UNIVERSIDADE NOVA DE LISBOA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE QUÍMICA

AUTOMATIC LEARNING FOR THE CLASSIFICATION OF CHEMICAL REACTIONS AND IN STATISTICAL THERMODYNAMICS

DIOGO ALEXANDRE ROSA SERRA LATINO

Lisboa
2008

n^{o} de arquivo
“copyright”

UNIVERSIDADE NOVA DE LISBOA

FACULDADE DE CIÊNCIAS E TECNOLOGIA

DEPARTAMENTO DE QUÍMICA

AUTOMATIC LEARNING FOR THE CLASSIFICATION OF CHEMICAL REACTIONS AND IN STATISTICAL THERMODYNAMICS

DIOGO ALEXANDRE ROSA SERRA LATINO

Tese orientada por:

Professor Doutor João Aires de Sousa

Professor Doutor Fernando M. S. S. Fernandes

Professora Doutora Filomena F. M. Freitas

Dissertação apresentada para obtenção do Grau de Doutor em Química
Especialidade de Química Orgânica,
pela Universidade Nova de Lisboa, Faculdade de Ciências e Tecnologia.

Lisboa
2008

Dedicada aos meus pais e avós

Declaration

The work presented in this Thesis is based on research carried out at CQFB (Centro de Química Fina e Biotecnologia), REQUIMTE, Departamento de Química, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa and at CCMM (Centro de Ciências Moleculares e Materiais), Departamento de Química e Bioquímica, Faculdade de Ciências, Universidade de Lisboa, Portugal.

The following Chapters or Sections are based on articles published or submitted during the PhD:

- Chapter 5 is based on the article:
  D. A. R. S. Latino, J. Aires-de-Sousa, “Linking Databases of Chemical Reactions to NMR Data: An Exploration of ^{1}H NMR - Based Reaction Classification”, *Anal. Chem.* **2007**, *79*, 854-862.

- Chapter 7 is based on the article:
  D. A. R. S. Latino, J. Aires-de-Sousa, “Genome-Scale Classification of Metabolic Reactions: a Chemoinformatics Approach”, *Angew. Chem. Int. Ed.* **2006**, *45*, 2066-2069.

- Chapter 8 is based on the article:
  D. A. R. S. Latino, Q.-Y. Zhang, J. Aires-de-Sousa, “Genome-Scale Classification of Metabolic Reactions and Assignment of EC Numbers with Self-Organizing Maps”, *Bioinformatics.* **2008**, *doi:10.1093/bioinformatics/btn405*

- Chapter 11 is based on the article:
  D. A. R. S. Latino, F. F. M. Freitas, J. Aires-de-Sousa, F. M. S. S. Fernandes, “Neural Networks to Approach Potential Energy Surfaces. Application to a Molecular Dynamics Simulation”, *Int. J. Quantum Chem.* **2007**, *107, (11)*, 2120-2132.

- Chapter 12 is based on the article:
  D. A. R. S. Latino, R. P. S. Fartaria, F. F. M. Freitas, J. Aires-de-Sousa, F. M. S. S. Fernandes, ”Mapping Potential Energy Surfaces by Neural Networks. The Ethanol / Au (111) interface”, *J. Electroanal. Chem.* **in press**

No part of this Thesis has been submitted elsewhere for any other degree or qualification and all experiments were performed by me unless in the following cases:

- In Chapter 8 Subsection 8.3.4 the experiments with chirality codes were done by Dr. Q.-Y. Zhang.
- In Chapter 11 Subsection 11.3.2 the Molecular Dynamics simulations were performed by Prof. F. F. M. Freitas.
- In Chapter 12 Subsection 12.2.1 the DFT calculations performed for the molecular orientations to be used as test set were performed in collaboration with Dr. R. P. S. Fartaria.

During the PhD the following articles were also published:

- D. A. R. S. Latino, L. M. V. Pinheiro, F. F. M. Freitas, F. M. S. Silva Fernandes, A. R. T. Calado, "Prediction of Lipophilicity by Feed-Forward Neural Networks using Topological Descriptors", *Revista Portuguesa de Farmácia*, ISSN 0484 - 811 X, Volume LII (n^{o}2), **2005.**
- F. M. S. S. Fernandes, R. P. S. Fartaria, D. A. R. S. Latino, F. F. M. Freitas "Computer Simulation of Solution/Electrode Interfaces", *Port. Electrochim. Acta*, ISSN 0872 - 1904, **2008,** *26, (1)*, 1-13**.**

Diogo Alexandre Rosa Serra Latino
(Licenciado em Química pela Faculdade de Ciências da Universidade de Lisboa)

Acknowledgements

Depois de um percurso de quatro anos, com bons e maus momentos, são muitas as pessoas a quem eu deveria agradecer. Como os Agradecimentos numa Tese devem ser dedicados realmente a quem teve maior influência no trabalho vou então dedicar-me apenas às pessoas, ou instituições, que contribuiram mais directamente para o desenrolar da minha investigação.

As minhas primeiras palavras de agradecimento vão para os meus orientadores, a Professora Doutora Filomena Freitas, o Professor Doutor Fernando Fernandes e o Professor Doutor João Aires de Sousa.

Tenho de referir que o meu percurso como investigador na área da Quimioinformática se iniciou em 2001, no Laboratório de Simulação Molecular da FCUL, ainda como aluno de Licenciatura pela acção da Professora Doutora Filomena Freitas. Foi ela a primeira pessoa que eu contactei demonstrando o meu interesse em realizar o estágio de final de curso na área da Química Computacional tendo-me encaminhado para o Professor Doutor Fernando Fernandes. Durante estes anos sempre me apoiou procurando motivar-me nos meus momentos de desânimo ou de menor motivação. Tenho que agradecer todo o tempo que despendeu comigo a discutir o trabalho assim como nas últimas semanas no processo moroso e difícil de revisão da Tese.

Ao Professor Doutor Fernando Fernandes tenho de agradecer o facto de me ter aceite como seu aluno ainda como estagiário, quando não tinha qualquer informação sobre mim, mas mais que isso, ter-me lançado o desafio da aplicação das redes neuronais artificiais em química. O Professor Doutor Fernando Fernandes deu-me a escolher dois temas, um em simulação molecular e outro em aplicação de redes neuronais em química, tendo frisado desde logo que relativamente ao último esta era uma área virgem no grupo e que por isso a sua ajuda não seria a mesma. Apesar disso, esta foi a área pela qual me senti atraído tendo aceite o seu desafio e continuando a trabalhar nela até hoje. Além disso, tenho de agradecer toda a motivação que sempre me procurou transmitir e a confiança que depositou em mim. Sempre procurou transmitir-me o sentido de individualidade, autonomia e independência procurando que eu trilhasse o meu caminho livre como o vento.

Em 2003 conheci o Professor Doutor João Aires de Sousa com o qual iniciamos uma colaboração tendo iniciado o doutoramento em 2004 também sob a sua supervisão. Tenho que lhe agradecer a sua orientação e tudo o que me ensinou sobre métodos de aprendizagem automática. Manteve-se quase que diariamente presente para acompanhar a evolução do trabalho mas, ao mesmo tempo, deixou-me seguir o caminho que eu considerava melhor dando-me liberdade e motivando-me para me dedicar a outros temas sem ser estritamente os do Doutoramento. Penso que uma das suas qualidades é conseguir analisar as virtudes e defeitos dos seus colaboradores e tirar rendimento disso. Todas as pessoas têm aptidões para realizar melhores tarefas que outras, formas diferentes de abordar e resolver os problemas e até horários de trabalho diferentes onde produzem mais. Ele consegue perceber e tirar partido disso.

Nestes anos tenho também de referir e agradecer todo o companheirismo dos meus colegas de grupo quer na FCUL quer na FCTUNL.

Na FCUL foram meus colegas o Doutor Pedro Celestino e o Doutor Rui Fartaria. O Doutor Celestino apesar do seu percurso conturbado esteve sempre disponível para discutir qualquer assunto ajudando-me inúmeras vezes, na realidade sempre que eu realmente precisei. Ao Doutor Fartaria um agradecimento especial. Foi com ele que eu passei mais tempo e a ele devo muitas das coisas que aprendi. Não tanto relativamente à minha investigação propriamente dita mas sobre tudo o resto que acaba por nos ajudar a aumentar o nosso rendimento e a atingir objectivos mais depressa.

Na FCTUNL foram meus colegas Gonçalo Carrera, Doutor Sunil Gupta, Doutor Q.-Y. Zhang, Doutor Yuri Binev, e Mestre Diana Almeida. Todos contribuíram para um ambiente de união e de camaradagem dentro do grupo mas por razões de afinidade de trabalho ou pessoais tive mais contacto com os três últimos. Devo agradecer nomeadamente ao Doutor Binev pela ajuda com o software SPINUS e ASNN e ao Doutor Zhang com quem colaborei mais directamente no desenvolvimento do método MOLMAP.

Em termos institucionais desejo agradecer às minhas instituições de acolhimento o REQUIMTE, CQFB, Departamento de Química da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa e o CECUL, numa primeira fase, e CCMM, no último ano, ambos do Departamento de Química e Bioquímica da Faculdade de Ciências da Universidade de Lisboa. A todos agradeço a disponibilização das instalações e recursos que me permitiram realizar o trabalho.

À Fundação para a Ciência e Tecnologia agradeço o apoio financeiro sob a forma de uma bolsa de Doutoramento (SFRH/BD/18347).

Na Tese encontram-se no final de cada Capítulo, se for caso disso, os Agradecimentos a instituições, entidades ou pessoas que contribuiram de alguma forma relevante mas não directa para o trabalho apresentado no Capítulo em questão. De uma forma mais geral desejo agradecer às seguintes entidades a disponibilização de dados e software usados nesta Tese:

- InfoChem GmbH (Munique, Alemanha) pela disponibilização do conjunto de dados relativos a reacções fotoquímicas extraído da base de dados SPRESI.

- Molecular Networks GmbH (Erlangen, Alemanha) pela disponibilização do programa PETRA e base de dados Biopath.
- Kyoto University Bioinformatics Center (Kyoto, Japão) pelo acesso à base de dados KEGG.
- ChemAxon Ltd (Budapeste, Hungria) pelo acesso aos programas JChem e Marvin.
- Doutor Igor Tetko (Institute for Bioinformatics and Systems Biology, Helmholtz Center Munich, Alemanha) pela disponibilização do programa ASNN para a implementação de ensembles de redes neuronais e redes neuronais associativas.

Um agradecimento especial para os meus pais. Se hoje me encontro aqui devo-o a eles por um lado pelo amor, carinho e todo o apoio dado ao longo dos anos, e por outro lado, pelo sentido de responsabilidade que me incutiram desde a mais tenra idade. Deram-me liberdade para eu escolher o meu caminho mas mostrando-me bem cedo que essa liberdade tinha de acarretar sentido de responsabilidade.

Um agradecimento a todos os meus amigos.

“Life is a long lesson in humility.”
James M. Barrie

”Opportunity is missed by most people because it is dressed in overalls and looks like work.”
Thomas A. Edison

“Being busy does not always mean real work. The object of all work is production or accomplishment and to either of these ends there must be forethought, system, planning, intelligence, and honest purpose, as well as perspiration. Seeming to do is not doing.”
Thomas A. Edison

“Para agir torna-se indispensável uma grande dose de defeitos. Um homem sem defeitos não serve para nada.”
Jacques Chardonne

Abstract

This Thesis describes the application of automatic learning methods for a) the classification of organic and metabolic reactions, and b) the mapping of Potential Energy Surfaces (PES). The classification of reactions was approached with two distinct methodologies: a representation of chemical reactions based on NMR data, and a representation of chemical reactions from the reaction equation based on the physico-chemical and topological features of chemical bonds.

NMR-based classification of photochemical and enzymatic reactions. Photochemical and metabolic reactions were classified by Kohonen Self-Organizing Maps (Kohonen SOMs) and Random Forests (RFs) taking as input the difference between the ^{1}H NMR spectra of the products and the reactants. The development of such a representation can be applied in automatic analysis of changes in the ^{1}H NMR spectrum of a mixture and their interpretation in terms of the chemical reactions taking place. Examples of possible applications are the monitoring of reaction processes, evaluation of the stability of chemicals, or even the interpretation of metabonomic data.

A Kohonen SOM trained with a data set of metabolic reactions catalysed by transferases was able to correctly classify 75% of an independent test set in terms of the EC number subclass. Random Forests improved the correct predictions to 79%. With photochemical reactions classified into 7 groups, an independent test set was classified with 86-93% accuracy. The data set of photochemical reactions was also used to simulate mixtures with two reactions occurring simultaneously. Kohonen SOMs and Feed-Forward Neural Networks (FFNNs) were trained to classify the reactions occurring in a mixture based on the ^{1}H NMR spectra of the products and reactants. Kohonen SOMs allowed the correct assignment of 53-63% of the mixtures (in a test set). Counter-Propagation Neural Networks (CPNNs) gave origin to similar results. The use of supervised learning techniques allowed an improvement in the results. They were improved to 77% of correct assignments when an ensemble of ten FFNNs were used and to 80% when Random Forests were used.

This study was performed with NMR data simulated from the molecular structure by the SPINUS program. In the design of one test set, simulated data was combined with experimental data. The results support the proposal of linking databases of chemical reactions to experimental or simulated NMR data for automatic classification of reactions

and mixtures of reactions.

Genome-scale classification of enzymatic reactions from their reaction equation. The MOLMAP descriptor relies on a Kohonen SOM that defines types of bonds on the basis of their physico-chemical and topological properties. The MOLMAP descriptor of a molecule represents the types of bonds available in that molecule. The MOLMAP descriptor of a reaction is defined as the difference between the MOLMAPs of the products and the reactants, and numerically encodes the pattern of bonds that are broken, changed, and made during a chemical reaction.

The automatic perception of chemical similarities between metabolic reactions is required for a variety of applications ranging from the computer validation of classification systems, genome-scale reconstruction (or comparison) of metabolic pathways, to the classification of enzymatic mechanisms. Catalytic functions of proteins are generally described by the EC numbers that are simultaneously employed as identifiers of reactions, enzymes, and enzyme genes, thus linking metabolic and genomic information. Different methods should be available to automatically compare metabolic reactions and for the automatic assignment of EC numbers to reactions still not officially classified.

In this study, the genome-scale data set of enzymatic reactions available in the KEGG database was encoded by the MOLMAP descriptors, and was submitted to Kohonen SOMs to compare the resulting map with the official EC number classification, to explore the possibility of predicting EC numbers from the reaction equation, and to assess the internal consistency of the EC classification at the class level.

A general agreement with the EC classification was observed, i.e. a relationship between the similarity of MOLMAPs and the similarity of EC numbers. At the same time, MOLMAPs were able to discriminate between EC sub-subclasses. EC numbers could be assigned at the class, subclass, and sub-subclass levels with accuracies up to 92%, 80%, and 70% for independent test sets. The correspondence between chemical similarity of metabolic reactions and their MOLMAP descriptors was applied to the identification of a number of reactions mapped into the same neuron but belonging to different EC classes, which demonstrated the ability of the MOLMAP/SOM approach to verify the internal consistency of classifications in databases of metabolic reactions.

RFs were also used to assign the four levels of the EC hierarchy from the reaction equation. EC numbers were correctly assigned in 95%, 90%, 85% and 86% of the cases (for independent test sets) at the class, subclass, sub-subclass and full EC number level, respectively. Experiments for the classification of reactions from the main reactants and products were performed with RFs - EC numbers were assigned at the class, subclass and sub-subclass level with accuracies of 78%, 74% and 63%, respectively.

In the course of the experiments with metabolic reactions we suggested that the MOLMAP / SOM concept could be extended to the representation of other levels of metabolic information such as metabolic pathways. Following the MOLMAP idea, the

pattern of neurons activated by the reactions of a metabolic pathway is a representation of the reactions involved in that pathway - a descriptor of the metabolic pathway. This reasoning enabled the comparison of different pathways, the automatic classification of pathways, and a classification of organisms based on their biochemical machinery. The three levels of classification (from bonds to metabolic pathways) allowed to map and perceive chemical similarities between metabolic pathways even for pathways of different types of metabolism and pathways that do not share similarities in terms of EC numbers.

Mapping of PES by neural networks (NNs). In a first series of experiments, ensembles of Feed-Forward NNs (EnsFFNNs) and Associative Neural Networks (ASNNs) were trained to reproduce PES represented by the Lennard-Jones (LJ) analytical potential function. The accuracy of the method was assessed by comparing the results of molecular dynamics simulations (thermal, structural, and dynamic properties) obtained from the NNs-PES and from the LJ function.

The results indicated that for LJ-type potentials, NNs can be trained to generate accurate PES to be used in molecular simulations. EnsFFNNs and ASNNs gave better results than single FFNNs. A remarkable ability of the NNs models to interpolate between distant curves and accurately reproduce potentials to be used in molecular simulations is shown.

The purpose of the first study was to systematically analyse the accuracy of different NNs. Our main motivation, however, is reflected in the next study: the mapping of multidimensional PES by NNs to simulate, by Molecular Dynamics or Monte Carlo, the adsorption and self-assembly of solvated organic molecules on noble-metal electrodes. Indeed, for such complex and heterogeneous systems the development of suitable analytical functions that fit quantum mechanical interaction energies is a non-trivial or even impossible task.

The data consisted of energy values, from Density Functional Theory (DFT) calculations, at different distances, for several molecular orientations and three electrode adsorption sites. The results indicate that NNs require a data set large enough to cover well the diversity of possible interaction sites, distances, and orientations. NNs trained with such data sets can perform equally well or even better than analytical functions. Therefore, they can be used in molecular simulations, particularly for the ethanol/Au (111) interface which is the case studied in the present Thesis. Once properly trained, the networks are able to produce, as output, any required number of energy points for accurate interpolations.

Keywords: Chemoinformatics, Bioinformatics, Automatic Learning Methods, Neural Networks, Classification of Reactions, Metabolism, EC number, Potential Energy Surfaces, Fitting, Molecular Simulations.

Resumo

Nesta Tese apresentam-se os resultados da aplicação de métodos de aprendizagem automática à resolução de vários problemas em Química, nomeadamente a classificação automática de reacções orgânicas e metabólicas e o mapeamento de superfícies de energia potencial (SEP). Relativamente à classificação automática de reacções químicas, esta Tese desenvolve duas metodologias distintas: a representação de reacções químicas baseada em dados de RMN dos reagentes e produtos da reacção e a representação de reacções químicas baseada em propriedades físico-químicas e topológicas das ligações químicas dos reagentes e produtos da reacção.

Classificação de reacções baseada em dados de ^{1}H RMN. Foi explorada a classificação de um conjunto de reacções fotoquímicas e de um conjunto de reacções metabólicas, por mapas auto-organizativos de Kohonen e Random Forests (RFs), utilizando como input a diferença entre os espectros de ^{1}H RMN dos reagentes e dos produtos. O desenvolvimento de tal representação de reacções químicas pode ter aplicação na análise automática das mudanças do espectro de ^{1}H RMN de uma mistura e a sua interpretação em termos das reacções químicas responsáveis por tais mudanças. Esta interpretação permitirá deduzir informação acerca de reacções químicas a ocorrer na mistura, mesmo sem elucidação estrutural dos produtos e/ou reagentes. Aplicações específicas poderão ser, por exemplo, a monitorização de processos químicos, a avaliação da degradação de químicos, ou até mesmo a interpretação de dados em metabonómica.

Dois aspectos relevantes da metodologia são a utilização de dados de RMN simulados e a representação adequada das diferenças entre os espectros dos produtos e reagentes. Neste último aspecto foi usada a diferença entre as intensidades dos espectros dos produtos e dos reagentes para cada desvio químico.

Os desvios químicos dos reagentes e dos produtos foram gerados pelo programa SPINUS tendo os sinais sido "fuzificados" por uma função triangular para obter uma representação do espectro. Todos os sinais (desvios químicos) dos reagentes de uma reacção são agrupados no mesmo "pseudo-espectro" (espectro simulado da mistura de reagentes) fazendo-se o mesmo para os produtos (no caso das reacções fotoquímicas incluiram-se apenas reacções com dois reagentes e um produto enquanto que no caso das reacções metabólicas o número de reagentes e produtos foi variável).

Para a experiência foram escolhidas reacções metabólicas catalisadas por transferases

(nº EC 2.x.x.x) tendo estas sido extraídas da base de dados de reacções enzimáticas KEGG LIGAND (de Janeiro de 2006). Escolheu-se, também, um conjunto de 189 reacções fotoquímicas classificadas manualmente em 7 tipos de classes de reacções: (A) fotocicloadição [3+2] de azirinas a C=C, (B) fotocicloadição [2+2] de C=C a C=O, (C) fotocicloadição [2+2] de C=N a C=C, (D) reacção de foto-Diels-Alder, (E) fotocicloadição [2+2] de C=C a C=C, (F) fotocicloadição [3+2] de piridazinas a C=C e (G) fotocicloadição [2+2] de C=C a C=S.

Utilizando o conjunto de 911 reacções metabólicas catalizadas por transferases, para treinar mapas auto-organizativos de Kohonen, 75% das reacções de um conjunto de teste independente foram correctamente classificadas de acordo com a subclasse (segundo dígito do número EC). Se forem utilizadas Random Forests em vez de mapas de Kohonen, para o mesmo conjunto de teste, 79% das reacções são classificadas correctamente. Relativamente às experiências com o conjunto de 189 reacções fotoquímicas, 86-93% das reacções de um conjunto de teste independente foram correctamente classificadas com base nas 7 classes definidas. Em ambos os casos foi possível observar um agrupamento assinalável das respectivas classes de reacções nos mapas de Kohonen.

A utilização de Random Forests em vez de mapas de Kohonen permitiu, também, associar uma medida de confiança a cada previsão, fazendo uso da probabilidade de cada previsão. Verificou-se que para o conjunto de teste de reacções fotoquímicas, 29 das 42 reacções (69%) foram previstas com probablidade superior a 0.5 e, destas, todas foram correctamente classificadas. Quanto à experiência com reacções metabólicas, 165 de 262 reacções (62%) do conjunto de teste foram previstas com uma probabilidade maior que 0.5. Destas apenas 7 foram classificadas incorrectamente (4.2%). Os resultados demonstram a utilidade deste parâmetro como medida de confiança na previsão obtida para novos objectos, ou como indicador da pertença do objecto ao domínio de aplicabilidade do modelo.

Para verificar se seria possível utilizar modelos treinados com dados de RMN simulados para aplicação a dados experimentais, foram realizadas experiências em que se testaram os modelos obtendo previsões para um conjunto de reacções representadas por uma combinação de dados experimentais e simulados. As classificações obtidas por RFs mostraram qualidade semelhante às obtidas para os conjuntos de teste em que as reacções eram descritas apenas por dados simulados. Os resultados obtidos com mapas de Kohonen foram ligeiramente inferiores.

É de salientar que este método baseado em dados de ^{1}H RMN é limitado pela existência de átomos de hidrogénio na vizinhança do centro da reacção e pela mudança dos desvios químicos resultantes da reacção.

Os bons resultados obtidos demonstram a possibilidade de estabelecer um método de classificação automática de reacções químicas a partir de espectros de ^{1}H RMN dos reagentes e dos produtos.

Numa extensão do trabalho, anteriormente descrito, foram simulados espectros de ^{1}H RMN de misturas em que ocorressem simultaneamente duas reacções. Para prever as classes das reacções que ocorrem a partir dos espectros de ^{1}H RMN dos reagentes e produtos treinaram-se redes de Kohonen e de Back-Propagation .

Deste conjunto de 181 reacções classificadas em seis classes simularam-se misturas em que duas reacções de diferentes classes ocorrem simultaneamente. Do conjunto de 189 reacções fotoquímicas, utilizado anteriormente, foram retiradas as reacções da classe C, (reacção de fotocicloadição [2+2] de C=N a C=C) visto que foi verificado no estudo anterior que este tipo de reacção não formava um bom agrupamento no mapa de Kohonen. A partir deste conjunto de 181 reacções de 6 classes foram geradas todas as combinações possíveis de duas reacções originando, assim, 12421 misturas de reacções aleatoriamente separadas em 8280 reacções para o conjunto de treino e 4141 para o conjunto de teste.

As redes de Kohonen não apresentam, neste caso, um agrupamento tão evidente como no estudo anterior, obtendo-se previsões correctas para 81-89% dos casos no conjunto de treino e 71-80% no conjunto de teste. A utilização de redes de Counter-Propagation não permitiu uma melhoria dos resultados. Foram, então, treinadas redes de Back-Propagation (ou Feed-Forward) com o intuito de obter uma capacidade de previsão superior. Estas redes tinham 120 neurónios de input (o espectro diferença foi dividido em 120 intervalos constituindo cada um deles um descritor da reacção) e 6 neurónios de output (um para cada tipo de reacção). Por exemplo, se tivermos uma mistura de reacções A e B, a rede é, então, treinada para activar o primeiro e o segundo neurónios da camada de output, mantendo os restantes neurónios inactivos. Ao variar o número de neurónios da camada escondida chegou-se à conclusão que são necessários apenas 5-10 neurónios nessa camada para a rede aprender a classificar misturas de reacções. As redes de Feed-Forward obtidas classificaram correctamente $\sim$ 98% dos casos, quer para o conjunto de treino quer para o de teste. A utilização de Random Forests permitiu obter, com este conjunto de treino e teste, 99% de misturas bem classificadas para o conjunto de teste. Foi, então, utilizada uma partição do conjunto de reacções diferente em que foi usada a partição do estudo anterior para fazer as misturas. A partir do conjunto de treino com 141 reacções fotoquímicas e conjunto de teste com 40 reacções foram feitas todas as misturas possíveis entre as reacções de classes diferentes dentro destes conjuntos. Trata-se de uma partição treino/teste mais exigente pois assim é assegurado que nenhuma reacção existe numa mistura do conjunto de treino e numa mistura do conjunto de teste simultaneamente, como acontecia na partição anterior. A percentagem de acertos para o conjunto de teste utilizando esta nova partição baixou para 63%, 58%, 77% e 80% utilizando, respectivamente, um ensemble dez redes de Kohonen, ensemble de dez redes de Counter Propagation, ensemble de dez redes de Back-Propagation e Random Forests (com 1000 árvores de classificação).

Este estudo mostrou que será possível obter modelos para classificar as reacções que ocorrem numa mistura tendo como base dados de ^{1}H RMN.

Classificação automática de reacções enzimáticas a partir da sua equação química. Estas experiências foram realizadas com as reacções codificadas pelo método MOLMAP. O descritor MOLMAP de um composto representa os tipos de ligações covalentes existentes na sua estrutura molecular. Os tipos de ligações são definidos automaticamente por mapas de Kohonen, com base em características topológicas e fisico-químicas das ligações e respectivos átomos. O descritor MOLMAP de uma reacção é a diferença dos MOLMAPs dos produtos e reagentes. Este método codifica numericamente as transformações estruturais resultantes da reacção química, representando o padrão de ligações que são quebradas, mudadas e criadas durante uma reacção química. A utilização dum MOLMAP-diferença permite representar numericamente reacções sem identificação explícita do centro da reacção.

Numa primeira fase, foi realizado um estudo preliminar com descritores MOLMAP de reacção, para classificar automaticamente um conjunto à escala genómica de reacções enzimáticas. A classificação teve como referência o sistema de classificação EC (de "Enzyme Commission"). Posteriormente fez-se um estudo completo, onde foi feita a análise e discussão das vantagens e limitações do método, além da exploração de algumas das ideias apresentadas no estudo preliminar.

Com a disponibilidade de bases de dados que incorporam informação de natureza química e genómica, a aplicação deste método a dados de reacções metabólicas permite toda uma série de novos estudos fazendo a ponte entre os domínios da quimioinformática e da bioinformática. A percepção automática de similaridades químicas entre reacções metabólicas é necessária para um conjunto variado de aplicações, desde a validação de sistemas de classificação, reconstrução ou comparação à escala genómica de vias metabólicas, ou a classificação de mecanismos enzimáticos, por exemplo.

A função catalítica das proteínas é, geralmente, descrita pelo número EC atribuído à reacção química catalisada. O número EC é, assim, empregue simultaneamente como um identificador de reacções, enzimas e genes de enzima, estabelecendo a ligação entre a informação genómica e metabólica. Apesar desta classificação estar bem estabelecida e ser muito difundida, algumas regras contêm ambiguidades e heterogeneidade sendo o seu uso na análise da diversidade de reacções metabólicas (reactoma) limitado. No contexto de reconstrução à escala genómica de vias metabólicas a definição de semelhança de reacções é fundamental. Para comparar reacções metabólicas independentemente dos números EC vários métodos devem estar disponíveis, mas simultaneamente, são necessários métodos para a atribuição automática de números EC a reacções ainda não classificadas oficialmente (dada a importância actual dos números EC).

Num estudo preliminar codificaram-se 3468 reacções enzimáticas da base de dados KEGG, por meio de MOLMAPs calculados a partir de 7 propriedades empíricas de ligações - diferença na carga total, diferença na carga π, diferença na electronegatividade σ, polaridade da ligação, estabilização por resonância das cargas geradas por heterólise, polarizabilidade da ligação, e energia de dissociação da ligação. Com estas reacções foi treinado um mapa de Kohonen tendo o mapa resultante mostrado um considerável agrupamento de reacções de acordo com o sistema de classificação EC, particularmente para oxidoreductases, transferases e hidrolases. Das classes menos representadas, as ligases também exibiram um bom agrupamento. Este estudo permitiu, ainda, identificar exemplos de reacções semelhantes mas com diferenças no número EC ao nível da classe.

Após o estudo preliminar, a investigação foi aprofundada, usando uma versão posterior da base de dados de reacções, novos descritores de ligações, verificando a influência de vários parâmetros nos resultados, considerando reacções em ambos os sentidos e estendendo as previsões de número EC à subclasse e sub-subclasse. Foram, também, exploradas outras técnicas de aprendizagem automática, como mapas de Kohonen supervisionados e Random Forests.

Os descritores de reacção MOLMAP foram aplicados ao mapeamento de um conjunto de dados, com cerca de 4000 reacções enzimáticas, por mapas de Kohonen tendo estes sido utilizados para classificar reacções e para uma análise sistemática de inconsistências em números EC ao nível da classe.

Tendo por base mapas de Kohonen foi possível atribuir correctamente a classe, subclasse e sub-subclasse a reacções de conjuntos de teste independentes (o conjunto de treino inclui uma reacção de cada número EC existente e o conjunto de teste é constituído pelas reacções restantes) em 92%, 80% e 70% dos casos, respectivamente. Estes resultados mostram uma compatibilidade entre o método de codificação proposto e o sistema de classificação EC, reflectindo ao mesmo tempo a similaridade entre reacções dentro dos três primeiros níveis da hierarquia EC.

Foram exploradas duas medidas de confiança associadas a cada previsão: a relação entre o número de votos obtidos por ensembles de mapas de Kohonen para uma reacção, e a distância Euclideana entre o descritor de reacção MOLMAP (um vector com descritores de reacção) e os pesos do neurónio vencedor no mapa de Kohonen. Foi utilizado um conjunto de teste com 1646 reacções. Este conjunto de teste foi escolhido a partir do conjunto total de reacções com base num mapa de Kohonen de dimensão 49×49, treinado com todas as reacções. Depois do treino, a divisão do conjunto inicial em conjunto de treino e teste é realizada escolhendo uma reacção, de forma aleatória, de cada um dos neurónios ocupados, para o conjunto de teste. As outras reacções constituem o conjunto de treino. Relativamente ao número de votos, verificou-se que, com um ensemble de 10 mapas, 97% das previsões obtidas, para o primeiro nível da hierarquia EC, com 10 votos, estavam correctas. Esta percentagem diminuiu para 82.3%, 75.7%, 68.8%, 64.6% e 47.1% para previsões obtidas com 9, 8, 7, 6 e 5 votos, respectivamente. A análise da distância Euclideana entre o descritor da reacção e os pesos do neurónio vencedor revelou que também esta poderá ser utilizada como uma medida de confiança na previsão. A percentagem de previsões correctas, de acordo com o primeiro dígito do número EC, diminuiu de 100% para reacções com uma distância Euclideana > 100 para 59.3% para reacções a uma distância Euclideana > 5000 (mais precisamente estes valores correspondem a dez vezes o quadrado das distâncias Euclideanas).

Para verificar a capacidade do método em classificar reacções novas, que não participaram no treino, foram realizadas várias experiências com diferentes níveis de semelhança entre os conjuntos de treino e teste. O teste realizado com maior grau de exigência utilizou um conjunto de teste que só incluía reacções de sub-subclasses que não participaram no treino, tendo sido obtidas 68% de previsões correctas para o primeiro dígito do número EC (nível da classe). Também foram realizadas experiências para a previsão do terceiro dígito do número EC (nível da sub-subclasse) tendo os resultados obtidos demonstrado a capacidade dos descritores de reacção MOLMAP em discriminar sub-subclasses.

A correspondência entre a semelhança química de reacções metabólicas e as semelhanças dos seus descritores MOLMAP foi confirmada com a detecção de várias reacções semelhantes mapeadas no mesmo neurónio do mapa de Kohonen mas que, oficialmente, pertencem a diferentes classes do número EC. Esta experiência permitiu demonstrar que a aproximação MOLMAP/SOM poderá ser utilizada para a verificação de consistência interna dos sistemas de classificação em bases de dados de reacções metabólicas.

Foi, também, realizado um estudo com o mesmo conjunto de reacções mas utilizando Random Forests (um método de aprendizagem supervisionada) em vez de mapas de Kohonen, para a atribuição dos quatro níveis de classificação dos números EC, assim como para a classificação automática das reacções com base apenas nos reagentes e produtos principais.

Relativamente à classificação de reacções baseada em todos os reagentes e produtos foi possível atribuir correctamente a classe, subclasse e sub-subclasse para reacções de conjuntos de teste independentes (o conjunto de treino inclui uma reacção de cada número EC que exista, sendo o conjunto de teste constituído pelas reacções restantes) em 95%, 90% e 85% dos casos, respectivamente. Para o estudo da atribuição do número EC completo foram apenas considerados números EC com pelo menos 4 exemplos de reacções, tendo o número EC completo sido previsto correctamente em 86% dos casos de um conjunto de teste independente com 220 reacções (correspondendo a 110 números EC diferentes).

Assim como no estudo em que se utilizaram mapas de Kohonen, foram feitas várias experiências com diferentes níveis de semelhança entre os conjuntos de treino e de teste. O conjunto de teste com menor semelhança com o de treino, e portanto com maior grau de dificuldade, era constituído apenas por reacções de sub-subclasses que não participaram no treino, tendo estas sido correctamente classificadas em 73% dos casos para o 1º nível do número EC. As Random Forests têm uma medida interna que permite estabelecer uma medida de confiança para a previsão obtida. A contagem dos votos em cada classe, obtidos para um objecto pelas árvores da floresta, permite determinar a probabilidade de esse objecto pertencer à classe vencedora (classe mais votada pelas árvores). Foi verificado que, para a atribuição do 1^{o} dígito do número EC, 97% das reacções com uma probabilidade maior ou igual a 0.5 foram correctamente classificadas.

Relativamente à tentativa de classificar reacções apenas com base nos reagentes e produtos principais, a metodologia para a construção do descritor é em tudo semelhante ao realizado anteriormente. A diferença está no último passo, em que não são utilizados todos os intervenientes da reacção mas apenas os que são considerados intervenientes principais (tal como estão identificados na base de dados KEGG) - os MOLMAPs que serviram de descritor para uma dada reacção são apenas os MOLMAPs dos reagentes e produtos principais. Avaliou-se, assim, a capacidade de classificar reacções sem incluir, por exemplo, co-factores. Random Forests treinadas com este tipo de dados foram capazes de prever correctamente a classe, subclasse e sub-subclasse em 78%, 74% e 63% das reacções, respectivamente, em conjuntos de teste independentes. Foi, também, mostrada a possibilidade de classificar reacções enzimáticas utilizando simultaneamente reacções completas e reacções incompletas (só com reagentes e produtos principais). Para este caso, foram bem classificadas 89%, 85% e 82% das reacções, em conjuntos de teste independentes.

Com as experiências sobre classificação de reacções metabólicas levadas a cabo, foi possível representar reacções enzimáticas de uma forma numérica, permitindo, assim, o mapeamento do conjunto de reacções enzimáticas (reactoma) num mapa de Kohonen. Tal possibilitou a análise da diversidade de reacções de vias metabólicas diferentes, quer dentro de um organismo, quer entre organismos diferentes. Quando se mapeiam todas as reacções metabólicas num mapa de Kohonen (independentemente dos organismos onde elas ocorrem) tem-se, assim, representada num único mapa a diversidade bioquímica reaccional conhecida na natureza.

Em suma, foi possível estender o fundamental deste conceito de codificação de reacções, a partir de propriedades locais de moléculas, a outros níveis de informação metabólica tais como a codificação de vias metabólicas ou a codificação de reactomas de organismos. Tal como na construção do MOLMAP de um composto é usado um mapa de Kohonen pré-treinado para mapear todas as ligações químicas desse composto, mapearam-se de uma forma análoga todas as reacções de uma via metabólica para obter um descritor dessa via metabólica. Assim, como o conjunto de neurónios activados por ligações de um composto origina o descritor (MOLMAP) do composto, agora o conjunto de neurónios (num novo mapa de Kohonen treinado com reacções) activados por reacções da via metabólica origina um descritor da via metabólica. Enquanto que para o primeiro caso o mapa de Kohonen era treinado com ligações, no segundo caso o mapa foi treinado com reacções. Se em vez de vias metabólicas (e suas reacções) considerarmos organismos (e suas reacções) obtemos um descritor da maquinaria bioquímica desse organismo. Podemos, assim, codificar o reactoma de um organismo de uma forma numérica e ter ao mesmo tempo uma representação de tamanho fixo. Ao fazê-lo obtemos uma impressão digital de um organismo em função do seu reactoma.

Depois de treinada uma rede com todas as reacções disponíveis, por mapeamento do reactoma individual (conjunto de reacções químicas que ocorrem num organismo), é efectuada a codificação do reactoma do organismo. Com a obtenção de um descritor de organismos, tornou-se possível treinar uma nova rede com os descritores dos organismos. Nesta fase os objectos que são dados para treinar a rede são (reactomas de) organismos.

O mapa treinado com organismos revelou agrupamentos correlacionados com a taxonomia.

Para a comparação e classificação de vias metabólicas treinou-se um mapa de Kohonen com todas as reacções metabólicas disponíveis e depois utilizou-se esse mapa para mapear as reacções de uma dada via metabólica. O padrão de neurónios activados pelas reacções de uma via metabólica é a representação das reacções que participam nessa via - uma impressão digital dessa via metabólica. Tem-se, desta forma, um descritor de vias metabólicas baseado nas reacções que as constituem, permitindo a comparação de diferentes vias metabólicas e de vias metabólicas de diferentes organismos.

Esta metodologia baseada em três níveis de classificação (classificação de ligações químicas, classificação de reacções e, por fim, classificação de vias metabólicas) permitiu mapear e encontrar, de uma forma automática, semelhanças entre vias metabólicas mesmo para vias de diferentes tipos de metabolismo e vias que não tenham necessariamente números EC em comum. Quando se treinou um mapa de Kohonen com várias vias metabólicas de diferentes tipos de metabolismo foi possível observar uma tendência para o agrupamento segundo o tipo de metabolismo. Identificaram-se, ao mesmo tempo, semelhanças químicas entre vias metabólicas classificadas, na base de dados KEGG, em tipos de metabolismo diferentes.

Este estudo demonstrou como a metodologia MOLMAP, associada a mapas de Kohonen ou Random Forests, permite classificar automaticamente reacções enzimáticas a partir da equação química, perceber semelhanças químicas entre reacções (ou vias metabólicas) e avaliar a consistência interna da classificação em bases de dados de reacções enzimáticas. O estudo permitiu percorrer todo um caminho desde a classificação de ligações químicas até à classificação de organismos, exclusivamente com base em propriedades topológicas e fisico-químicas locais das estruturas moleculares.

Mapeamento de superfícies de energia potencial (SEP). As primeiras experiências realizadas neste âmbito constituíram um estudo preliminar para investigar a possibilidade de redes neuronais gerarem modelos de confiança para mapear SEP de forma a serem utilizados em simulações de Monte-Carlo (MC) e Dinâmica Molecular (DM). A "qualidade" das SEP reveste-se de especial importância em simulação visto que a precisão dos resultados obtidos pelos métodos de Monte Carlo e de Dinâmica Molecular depende directamente daquelas. Idealmente, uma SEP deverá ter a precisão dos resultados obtidos por métodos *ab initio*/DFT. Como estes cálculos são normalmente muito demorados para sistemas complexos torna-se necessário o desenvolvimento de métodos que permitam uma interpolação rigorosa para estimar valores de energia não explicitamente calculados.

Recentemente, têm sido utilizadas redes neuronais artificiais para mapear superfícies de energia potencial a partir de conjuntos de dados de energia, na maior parte dos casos obtidos por cálculos *ab initio* ou DFT. Contudo, não tinha sido ainda efectuado um estudo sistemático sobre a qualidade dos resultados obtidos por simulação molecular tendo como base a utilização de potenciais tabelados gerados por vários tipos de redes neuronais.

O principal objectivo desta parte do trabalho foi treinar redes neuronais para reproduzir superfícies de energia potencial obtidas a partir de funções analíticas bem conhecidas (neste caso foi utilizado o potencial de Lennard-Jones) e avaliar a precisão dos resultados das simulações a partir da SEP gerada por redes neuronais por comparação com os obtidos usando a função analítica. Foram utilizados ensembles de redes de feed-forward, (EnsFFNNs) e redes associativas (ASNNs) de modo a reproduzir o potencial de Lennard-Jones (LJ) para um dado sistema, fornecendo como input os parâmetros do potencial bem como um conjunto de distâncias.

Recorrendo à função analítica do potencial LJ traçaram-se 15 curvas de potencial diferentemente parametrizadas (correspondentes a 15 substâncias diferentes). A seguir, foram treinadas EnsFFNNs e ASNNs com o objectivo de fazer o ajuste a essas curvas, tendo o modelo sido testado, posteriormente, com a curva relativa ao árgon (que não participou no treino).

Para testar a curva obtida através de redes neuronais, para além de se compararem os valores da energia potencial previstos pelas redes com os obtidos pela função analítica, realizaram-se, também, simulações por dinâmica molecular e calcularam-se propriedades térmicas, estruturais e dinâmicas do sistema, sendo estas comparadas com os valores obtidos nas simulações efectuadas com base na função analítica. Foi demonstrado que, para o potencial de Lennard-Jones, as redes neuronais podem ser treinadas e gerar superfícies de energia potencial com qualidade suficiente para serem usadas em simulações.

As ASNNs são uma extensão dos EnsFFNNs. Um ensemble de redes de feed-forward é um método de aprendizagem automática com supervisão, muitas vezes designado como método "sem memória" visto que durante o treino este não guarda qualquer informação de forma explícita sobre os dados utilizados na aprendizagem. Esta informação fica guardada de forma implícita nos pesos das redes. Por outro lado, as redes associativas são uma combinação de um "método sem memória" (ensembles de redes de feed-forward) e de um método "com memória" (os dados usados na aprendizagem são guardados numa "memória" sendo posteriormente utilizados para fazer correcções nas previsões com base em aproximações locais dos objectos guardados na memória). Esta combinação faz com que as ASNNs produzam, por vezes, melhores resultados do que as FFNNs individuais e do que os ensembles de redes de feed-forward, podendo vir a revelar-se um método útil para a obtenção de superfícies de energia potencial para sistemas mais complexos.

Foi demonstrada a possibilidade de treinar redes associativas com um conjunto de dados inicial e, posteriormente, melhorar a precisão do modelo usando novas “memórias” à medida que novos dados vão ficando disponíveis. Tal característica poderá ser importante em estudos em que a determinação da SEP é feita por métodos *ab initio*/DFT.

Demonstrou-se igualmente que a existência de curvas semelhantes no conjunto de treino contribui para a aprendizagem da rede e para a obtenção de previsões com menor erro, como seria de esperar. Para finalizar, mostrou-se a capacidade das redes neuronais para fazer interpolações entre curvas bastante distantes, gerando potenciais com precisão suficiente para serem utilizados em simulações moleculares.

De salientar que o objectivo principal deste estudo preliminar foi apenas o de analisar, de uma forma sistemática, a precisão de diferentes redes neuronais usando o potencial de Lennard-Jones como teste em simulações moleculares. Assim, sublinhe-se, isto não significa que se pretenda substituir o potencial de Lennard-Jones pelo potencial gerado pelas redes.

A grande motivação do estudo seguinte, no seguimento do trabalho anterior, consistiu na obtenção de SEP multidimensionais por redes neuronais para simular a adsorção e automontagem de moléculas orgânicas solvatadas em eléctrodos de metais nobres. Na verdade, para tais sistemas complexos e heterogéneos, o desenvolvimento de funções analíticas adequadas, que ajustem energias de interacção obtidas por métodos quânticos, para além de não trivial pode mesmo ser impossível.

Nas experiências realizadas, as redes foram treinadas para prever os valores de energia potencial obtidos por DFT, para o estudo da adsorção do etanol numa superfície de Au(111), calculados e utilizados anteriormente noutros trabalhos realizados no grupo. De referir que as energias de interacção foram avaliadas considerando sítios de adsorção relevantes, nomeadamente ”top”, ”hollow 1” e ”hollow 2”. O input das redes consistiu na distância da molécula de etanol e de dois ângulos que descrevem a sua orientação, relativamente à superfície, bem como de três descritores que codificam a posição da molécula relativamente aos sites escolhidos. Os valores utilizados no treino foram determinados colocando a molécula de etanol sobre um cluster de ouro e calculando a energia de interacção entre a molécula e o cluster, fazendo variar a sua distância e a sua orientação relativamente ao cluster (7 orientações diferentes) para os três sites, conduzindo a cerca de 400 valores de energia que foram seleccionados para a aprendizagem e validação interna da rede. Posteriormente, para testar as redes, calculou-se por DFT um conjunto adicional de energias de interacção para 6 orientações relativamente a cada sítio. A qualidade dos modelos foi verificada por: validação interna, procedimento “leave-one-out” do método e pelo conjunto de teste constituído por orientações que não participaram no treino. Nesta fase da investigação, as redes treinadas prevêem o potencial para cada uma das situações dadas com uma precisão pelo menos equivalente a outros métodos, por exemplo, os baseados no complexo e demorado desenvolvimento de uma função analítica.

Os resultados demonstram que se podem treinar redes neuronais para mapear SEP com

precisão suficiente para serem usadas em simulações moleculares se os dados disponíveis para o treino cobrirem, suficientemente, o espaço de orientações e de sítios de interacção possíveis da molécula, relativamente à superfície. Depois de treinadas, as redes podem facilmente produzir o número de pontos de energia que forem necessários de modo a efectuar interpolações rigorosas.

Palavras chave: Quimioinformática, Bioinformática, Métodos de Aprendizagem Automática, Redes Neuronais Artificiais, Classificação de Reacções, Metabolismo, Números EC, Superfícies de Energia Potencial, Ajustes, Simulação Molecular.

List of Symbols and Abbreviations

The neural networks (NNs) literature uses several notations and symbols, which could make the comparison between methods more difficult. The notation followed in this Thesis is the same used by Gasteiger and Zupan in their book "Neural Networks in Chemistry and Drug Design". [1]

- Scalar (single valued) values: small italic letters

$$a$$

the exception is the term *Net* that begins with a capital letter to not be confused with the terms "network" and "net"

- Vectors and matrices: bold italic letters

$$\boldsymbol{A}$$

- Individual values of an input vector ($\boldsymbol{Inp}$ or $\boldsymbol{X}$): small italic *inp* or x, indexed with a subscript i of dimension m

$$inp_i,\ x_i \quad (i = 1,\ 2,\ ...,\ m)$$

- Individual values of the output vector ($\boldsymbol{Out}$ or $\boldsymbol{Y}$): small italic *out* or y, indexed with a subscript j of dimension n

$$out_j,\ y_j \quad (j = 1,\ 2,\ ...,\ n)$$

- The weight matrix of a layer of neurons: bold italic $\boldsymbol{W}$
- The elements of a weight matrix: are represented by w_{ji} where the first index refers to the neuron being considered and the second index to the input unit (the preceding neuron that transmits the signal)

$$w_{ji}$$

- When weight matrices of different levels are compared the first weight matrix of level l, $\boldsymbol{W}^l$, has the indices i and j and the weight matrix of the next level, $\boldsymbol{W}^{l+1}$ the indices j and k with k from 1 to r:

$$w_{kj}$$

- If there are more than one input objects they are labeled with the subscript s having a maximum value of p. The input vector are labeled as $\boldsymbol{X}_s$ and the individual signals as:

$$x_{si}$$

- In a neural network with multiple layers, each layer is identified by a superscript l. The output vector of a layer l is $\boldsymbol{Out}^l$ and its individual values are:

$$out_j^l$$

- The iterations in the learning procedure of a neural network are labeled with a superscript t in parentheses, *(t)*. The initial value of a weight matrix is $\boldsymbol{W}^0$ that change in the next iteration to $\boldsymbol{W}^1$. The succession of iterations are indicated by the superscripts "*old*" and "*new*":

$$\mathbf{W}^{(old)}, \mathbf{W}^{(new)}$$

Symbols and Abbreviations

AI	Artificial Intelligence
ANN	Artificial Neural Network
ASNN	Associative Neural Network
aver	Average of the variable/descriptor
B3LYP	Becke, three-parameter, Lee-Yang-Parr
BDE	Bond Dissociation Energy
BP	Back- Propagation
BPE	Back-Propagation of Errors
BPNN	Back-Propagation Neural Network
BRG	Basic Reaction Graph
c	Van der Waals parameters
CART	Classification And Regression Trees
CICC	Conformation-independent chirality code
CPNN	Counter-Propagation Neural Network
CRG	Condensed Reaction Graph
Cv	Heat Capacity
CXC	Complete Reaction Concept
DC	Diffusion Coefficient
D	Self-diffusion coefficient
DFT	Density Functional Theory
DNA	Deoxyribonucleic acid
E	Total Energy
E_{ee}	Interaction energy between electrons
E_{Ne}	Interaction energy between electrons and the atomic nuclei
E_{XC}	Exchange correlation energy
EC	Enzyme Commission
EnsFFNN	Ensemble of Feed-Forward Neural Network
F	Total force
F_{HK}	Hohenberg-Kohn functional

$\hat{F}_{KS}$	Kohn-Sham operator
FFNN	Feed-Forward Neural Network
GA	Genetic Algorithm
GGA	Generalized Gradient Approximation
H	Enthalpy
H1	Hcp (hexagonal closed packed) site; the approach of the ethanol is made in the direction of the centre of a triangle formed between three gold atoms of the first layer with a gold atom of the second layer at the centre
H2	Fcc (face centred cubic) site; the ethanol approach is made in the direction of the centre of a triangle formed between three gold atoms of the first layer
HF	Hartree-Fock
hl	Hard-limiter
ITS	Imaginary Transition State
J	Classical electron interaction energy
JATOON	JAva TOOls for Neural networks
k	Kinetic energy
	constant force (in the context of molecular force field)
K	Kinetic energy (in the context of DFT theory and molecular simulation)
k_B	Boltzmann constant
K_S	Kinetic energy of non-interacting system
KEGG	Kyoto Encyclopedia of Genes and Genomes
KL	Kohonen Learning
KNN	K-Nearest Neighbor
l	Bond length
LDA	Local Density Approximation
LEPS	London-Eyring-Polanyi-Sato
LJ	Lennard-Jones
LOO	Leave-One-Out
LSDA	Local Spin Density Approximation
m_i	Mass of molecule i
MAE	Mean Absolute Error

Max	Maximum value that a variable x takes in the data set
MC	Monte Carlo
MD	Molecular Dynamics
Min	Minimum value that a variable x takes in the data set
min()	Minimum intrisic function
MALDI-TOF	Matrix-Assisted Laser Desorption/Ionization - Time Of Flight
MLRA	Multi Linear Regression Analysis
MOLMAP	MOLecular Map of Atom-level Properties
MS	Mass Spectrometry
MXC	Minimum Reaction Concept
n	New configuration (in the context of MC method)
	Periodicity of the rotation (in the context of the molecular force field)
	Number of objects (in the context of automatic learning)
N	Number of particles/molecules of the system
NMR	Nuclear Magnetic Resonance
NN	Neural Network
$norm_{0-1}(x)$	Normalization function between 0 and 1
$norm_{0.1-0.9}(x)$	Normalization function between 0.1 and 0.9
$norm_z(x)$	z-normalization
o	Old configuration (in the context of MC method)
OOB	Out-Of-Bag
OPLS	Optimized Potentials for Liquid Simulations
p	Pressure
PCA	Principal Component Analysis
PEOE	Partial Equalization of Orbital Electronegativity
PES	Potential Energy Surface
PETRA	Parameter Estimation for the Treatment of Reactivity Applications
PLS	Partial Least Squares
q	Partial charges
Q_σ	Bond polarity
QSAR	Quantitative Structure-Activity Relationships

QSPR	Quantitative Structure-Property Relationships
QSRR	Quantitative Structure-Retention Relationships
r	Distance between particles/molecules
r_c	Cutt-off distance
$\mathbf{r}_i$	Position vector of molecule i
$\mathrm{R}^{\pm}$	Resonance stabilization of charges generated by heterolysis
RC	Reaction Classification
RCG	Reaction Center Graph
RCP	Reaction Center Perception
rdf	Radial Distribution Function
RF	Random Forest
RG	Reaction Graph
RMS	Root Mean Square
RMSE	Root Mean Square of Errors
RNA	Ribonucleic acid
SAM	Self-Assembled Monolayers
sd	Standard deviation
SEQ	Symbolic EQuation
sf	Sigmoidal function
SOM	Self-Organizing Map
SPINUS	Structure - based Predictions In NUclear magnetic resonance Spectroscopy
SRGS	Superimposed Reaction Skeleton Graph
SVM	Support Vector Machine
t	Time
T	Temperature
tl	Threshold logic
Top	Site of the Au surface when the oxygen atom of ethanol approaches the surface directly over a gold atom of the first layer
u	Pair potential energy
u_{anal}	Potential energy generated by the LJ analytical function
u_{pred}	Potential energy generated by a NN

U	Potential energy
V	Volume
$\mathbf{v}_i$	Velocity vector of molecule/particle i
V_n	Rotational barrier height
V_{Ne}	Interaction potential nuclei-electron
vcf	Velocity Autocorrelation Function
vdW	Van de Waals
VOC	Volatile Organic Compound
XOR	exclusive-or logical operation
XC	Exchange Correlation
Y_{exp}	Predicted value by an automatic learning technique
Y_{calc}	Target value of an automatic learning technique
Z(t)	Velocity autocorrelation function
$\Delta n(r)$	Number of molecules/particles in the spherical shell of volume $4\pi r^2\Delta r$ at a distance r from the molecule/particle
Δq_π	Difference in π charge
Δq_{tot}	Difference in total charge
Δt	Integration time-step
ε	Potential well (in the context of the Lennard-Jones potential)
	Dielectric constant (in the context of molecular force field)
σ	Approximately the molecular diameter (in the context of the Lennard-Jones potential)
$\Delta\chi_\sigma$	Difference in σ electronegativity
α	Angle between the O-H bond and the normal to the surface
α_b	Effective bond polarizability
β	Angle between the plane H-O-C and the plane H-O-*normal to the surface*
ϕ	Angle between the projection of the r_{O-Au} vector on the surface plane and a reference surface vector beginning in a *Top* site and directed to a *H1* site
χ	Dihedral angle
φ_i	Kohn-Sham orbital i

ρ	Density
	Ensemble probability density function (in the context of MC method)
	Electron density (in the context of DFT theory)
θ	Bond angle (in the context of the molecular force field)
	Angle between the r_{O-Au} vector and the normal to the surface

Contents

List of Figures

Lista de Figuras

List of Tables

Lista de Tabelas

Preface

Most of the results presented in this Thesis are based on articles that have been published in international scientific journals with referees during the PhD work, or are in the submission or reviewing process. The Thesis consists of four Parts: a general introduction to the machine learning methods employed in this work, the research on classification of chemical reactions, the research on mapping of Potential Energy Surfaces (PES), and the conclusions.

Part I ("Introduction to Automatic Learning Methods in Chemistry") is an introduction to data analysis in the context of chemoinformatics and bioinformatics domains, focusing on the methods used for this Thesis. In Chapter 1 ("Introduction to Data Analysis") the domain of chemoinformatics and its main objectives are presented. The connections between chemoinformatics and bioinformatics are discussed followed by the presentation of some general concepts on data analysis and processing. Finally, the concept of automatic learning is presented and discussed. Chapter 2 ("Artificial Neural Networks") is dedicated to artificial neural networks models. Some historical notes are included followed by a description of the neuron model, how the neurons are linked in networks, and how a NN learns from data. The other Sections of this Chapter are dedicated to describe the concepts, architecture and learning algorithms of the different NN models applied here - Kohonen Self-Organizing Maps (Kohonen SOMs), Counter Propagation Neural Networks (CPNNs), Feed-Forward Neural Networks (FFNNs), and Associative Neural Networks (ASNNs). Chapter 3 ("Decision Trees and Random Forests") explains the Random Forest technique, the other automatic learning method employed.

Both Part II and Part III include their own introductions, describing the state of the art and specific methods related to the two topics of this work - the automatic classification of chemical reactions, and the mapping of potential energy surfaces. Both Introductions are followed by Chapters presenting the research performed for this Thesis. These Chapters are mostly based on articles published (or to be published) in scientific journals. Their contents were adapted to avoid multiple descriptions of common issues.

Chapter 4 ("State of the Art") of Part II presents an overview of reaction classification methods, an overview of other studies in genome-scale classification of metabolic reactions, and a full description of the MOLMAP reaction descriptors. Chapter 5 ("Linking Databases of Chemical Reactions to NMR data: ^{1}H NMR-Based Reaction Classification“)

presents the results concerning the development of a method for reaction classification based on ^{1}H NMR data. Chapter 6 (“Classification of Mixtures of Reactions from ^{1}H NMR data”) which consists in the preliminary results of application of the method presented in the last Chapter in classification of mixtures of reactions. Chapter 7 (“Genome-Scale Classification of Metabolic Reactions. A Preliminary Study”) presents a preliminary study of the application of the MOLMAP approach for classification of metabolic reactions that is extended and fully discussed in Chapter 8 (“Genome-Scale Classification of Metabolic Reactions and Assignment of EC Numbers”). Chapter 9 (“Genome-Scale Classification of Pathways and Organisms”) explored the extension of the MOLMAP approach for the encoding of other levels of metabolic information.

Chapter 10 (“Fundamental Concepts”) of Part III presents some of concepts of statistical mechanics, Monte Carlo and Molecular Dynamics methods. A short discussion on potential energy surfaces and Density Functional Theory is also presented. Finally the application of NNs to mapping PES is overviewed. Chapter 11 (“NNs to Approach Potential Energy Surfaces: Application to a MD Simulation”) reports the experiments to train NNs for reproducing PES, represented by well-known analytical potential functions, and then to assess the accuracy of the method by comparing the simulation results obtained from NNs and analytical PES. Chapter 12 (“Mapping Potential Energy Surfaces by NNs. The ethanol/Au (111) Interface”) is dedicated to the experiments to map multidimensional PES for the ethanol / Au (111) surface interaction regarding the simulation of the adsorption and self-assembly of alkylthiols solvated by ethanol.

Finally Part IV (“Conclusions and Future Work”) presents the main conclusions and ideas for improvements of the presented work, as well as suggestions for future applications based on the methods presented here.

Part I

Introduction to Automatic Learning Methods in Chemistry

Chapter 1

Introduction to Data Analysis

The information available today in databases, on all chemistry fields, far exceeds the knowledge of any individual chemist. Huge amounts of data are produced in chemical research that are collected in databases. Latent in the stored information there is a knowledge difficult to ferret out and to handle. Not only the explicit information is important, but also the relationships among the data could be relevant.

The acquisition/extraction of knowledge from databases and the correct representation of data to solve a specific problem must be done in an automatic way. Data analysis not only deals with the extraction of relevant information from databases or some specific data sets, but also with the generation of secondary information, such as the development of models to make predictions from the available data. The development of strategies to extract this latent knowledge making them useful to application in new situations is carried out by automatic learning methods. This Chapter presents an introduction to the methodology used in data analysis and automatic learning. Section 1.1 is an introduction to chemoinformatics and automatic learning. Section 1.2 mentions relationships between chemoinformatics and bioinformatics. Section 1.3 presents the concept of data, explaining the relationship between data, information and knowledge, and is followed by an overview on the preparation of data before their use by some automatic learning - data pre-processing, object selection and selection of training and test sets. Section 1.4 presents the concept of automatic learning, with a description of the learning process, and the differences between the two main learning strategies - supervised and unsupervised learning. The most general types of problems are presented.

1.1 Chemoinformatics and Automatic Learning

In Chemoinformatics, automatic learning methods are often employed to solve chemical problems on the basis of chemical information previously obtained. This research domain began when chemical information generated by chemists started to be stored in databases in electronic form. This field, for many years lacking a specific designation, deals with the storage, manipulation and processing of chemical information, and was designated almost 10 years ago as "Chemoinformatics". In Ref. [2] Gasteiger defined this field, domain or even discipline of chemistry in the following way:

"Chemoinformatics is the application of informatic methods to solve chemical problems"

other definitions are given by K. Brown [3]:

"The use of information technology and management has become a critical part of the drug discovery process. Chemoinformatics is the mixing of those information resources to transform data into information and information into knowledge for the intended purpose of making better decisions faster in the area of drug lead identification and organization"

and by G. Paris [4]:

"Chem(o)informatics is a generic term that encompasses the design, creation, organization, management, retrieval, analysis, dissemination, visualization and use of chemical information"

To clearly understand the scope of chemoinformatics, the broad definition of the chemistry domain itself must be considered.

"Chemistry deals with compounds, their properties and their transformations" from Ref. [2]

The structure, energy and many other properties of a molecule can be determined using *ab initio* methods. But for most complex structures like macromolecules the computation time needed to obtain an answer makes the first principles methods useless in many applications. The problem is even more complex for chemical reactions. Only simple reactions can be treated using *ab initio* methods, and even in simple cases it is difficult to take into account the influence of the reaction conditions such as the solvent or the temperature.

One way to avoid these limitations, and at the same time to use the information available from previous experiments, is to *learn from data.* A inductive learning method learns using the information available from a set of experiments. This information is used to develop models that can explain the experiments. These models are applied and confirmed, rejected or refined with new experiments, sometimes allowing the extraction of knowledge in the form of rules or even laws of the nature. The concept of automatic learning and inductive learning will be discussed again in Section 1.4.

There are several chemical problems where chemoinformatics techniques can be applied. One important task in chemistry is to obtain compounds with specific properties. The objective is to establish Structure-Property or Structure-Activity Relationships (SPR or SAR), or even to quantify these relationships in Quantitative Structure-Activity Relationships, or Quantitative Structure-Property Relationships (QSAR or QSPR) [5–8]. Such relationships enable the prediction of a specific property from the structural formula of the compound. Another example is *structure elucidation* - to derive the structural formula of an unknown compound using spectroscopic information (mainly infrared, NMR, and mass spectra) [9–11]. S*ynthesis design* and the *planning of chemical reactions* is another area where the use of computers has been researched for many years to solve chemical problems [12–23]. The classification of chemical reactions is another field that has recently attracted much attention due to the increasing relevance of metabolic reactions, and chemical reactivity in toxicology and medicinal chemistry [19, 23, 24]. In all these fields, the use of collected information to learn is an alternative to *ab initio* methods that cannot be applied in a straightforward manner to the resolution of many chemical problems.

1.2 Chemoinformatics and Bioinformatics

In the same way Chemoinformatics can be defined, in a general manner, as:

> *"the application of informatic methods to solve chemical problems"*

Bioinformatics can be defined as:

> *" the application of informatic methods to solve biological problems".*

Despite the focus on two different scientific areas, chemistry and biology, it is difficult to make a clear separation between the chemoinformatics and bioinformatics worlds. Bioinformatics mainly deals with proteins and genes, for example to study their sequence, structure, and properties. But at the same time proteins are chemical compounds to which chemoinformatics tools can also be applied. The function of enzymes is the catalysis of chemical reactions, which clearly belong to the domain of Chemistry. This shows that there are some specific areas of research that connect the chemoinformatics and bioinformatics domains. It is the case of structure elucidation of biomolecules such as proteins, DNA and RNA [25], and studies in structure/function relationships of enzymes [26–28], for example. In the last years several approaches have been presented for integrating chemical and biological data in "systems chemical biology" [29]. The chemical space, consisting of all possible organic molecules, also includes all organic molecules present in living organisms. The exploration of the chemical space with integration of biological knowledge can lead to a better understanding of the metabolism [30] and to the discovery of new drugs [31]. The exploration of these regions of the chemical space that participate in biological processes - "the biological activity space" can be made by the analysis of protein binding to small organic molecules and analysis of the phenotypic responses of the organisms to small organic molecules [32]. The virtual screening of chemical libraries to discover new ligands on the basis of biological structures is one application that clearly links the chemo and bioinformatics domains and can be used to better explore the biological activity space [33]. Also molecular engineering area, the manipulation of biomolecules to obtain compounds with specific structures and functions, is a field where chemoinformatics tools can play an important role [34]. Thornton and co-workers presented enzymatic structure/function studies and the comparison of the enzymatic sets of different species [27,35,36]. Kanehisa and co-workers are strongly contributing to integrate chemical and biological data with the development of the KEGG database [37–40]. The integration of chemical and genomic data in this database allows for several studies in classification of enzymatic reactions [41], regulation of metabolism and metabolic pathways [42–45].

This Thesis, presents a contribution in this field that consists in a method for the encoding and automatic classification of enzymatic reactions. A further incursion in the domain of Bioinformatics is presented in Part II where the representation of metabolic pathways and organisms is based on the methods developed for chemical reactions.

1.3 Data

Data in general, and chemical data in particular, is a mixture of information and noise. It is difficult to carry out experiments where the obtained information has no noise (the error of the measurement). The different level of noise depends on the type of the experiment but the magnitude of the information must be much greater than the magnitude of the noise. Only in this condition it is possible to extract knowledge from the information contained in some data [46].

In chemoinformatics, the features that describe an object (the measured data) are called independent variables, input variables, or descriptors, and the properties to be predicted are called dependent variables, output variables, targets, answers, responses.

In general the data are organized in vectors. The input and the output data are organized in separate vectors. Each component of the input or output vector is one input or output variable.

1.3.1 Data, Information and Knowledge

The huge amount of chemical and biological data presently available, and growing every day, do not mean that the required information to solve a problem is directly available . Differently, the extraction of knowledge is in most cases difficult to achieve.

The concept of information is related to the concept of uncertainty and can be exemplified with a chemical example: when a new molecule is obtained the information about it is null or almost null. There is no information about the physical or biological properties of the new compound nor even about its structure. By application of several analytical techniques the characterization of the compound is made. When the structure of the compound is found it can be said that there is no uncertainty about the structure but it remains for other properties that are still not determined. The information of a physical system is defined by Brillouin in his book about the subject as [47]:

"Information is a measure of decreasing uncertainty of the system by means of some physical activities."

Gasteiger presents other definition of information [2]:

"If data are put into context with other data, we call the result information. The measurement of the biological activity of a compound gains in value if we also know the molecular structure of that compound"

The concept of knowledge will be presented here in an intuitive way. It is intuitive that new knowledge can be obtained from the retrieval and analysis of data. Lekishvili defined it as [46]:

"Knowledge is the perception of the logical relations among the structure of the information."

Fischler and Firschein [48] present other generic definition of knowledge:

"Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world."

Gasteiger shows the actions needed to extract knowledge from information [2]:

"...obtaining knowledge need some level of abstraction. Many pieces of information are ordered in the framework of a model; rules are derived from a sequence of observations;

predictions can be made by analogy.”

The application of this concept to chemical data, chemical information and chemical knowledge is the major objective of chemoinformatics.

Two relevant aspects in the extraction of knowledge from data are the quality of the data itself, and the method used to learn and generalize the knowledge from the data. The automatic learning methods are dependent of the quality of data to extract reliable knowledge from them.

1.3.2 Data Pre-Processing

Some automatic learning techniques requires a pre-processing of the data. The type of pre-processing depends on the method and on the data. In general, data must be normalized in some specific interval. For Back-Propagation Neural Networks, for example, output values must be normalized between 0 and 1, 0.1 and 0.9 or even 0.2 and 0.8 - this is a consequence of the use of the sigmoidal function as transfer function that will be explained later.

In Kohonen Self-Organizing Maps the learning procedure and the internal corrections depend on the calculation of Euclidean distances between vectors composed of descriptors. In this case if the values of a descriptor are in a wider range than others, this descriptor will have more impact than others. Therefore, the data must be normalized for all descriptors to have the same impact.

For linear normalization of the data between 0 and 1 the following expression is applied:

$$norm\,(x) = \frac{x - Min}{Max - Min} \tag{1.1}$$

where x is the value of the descriptor to be normalized, Min is the minimum value that x takes in the data set and Max is the maximum value that x takes in the data set. If normalization is required between 0.1 and 0.9 the last equation is modified to:

$$norm\,(x) = 0.1 + 0.8 \times \frac{(x - Min)}{(Max - Min)} \tag{1.2}$$

In other cases, a different normalization is used based on the average value and standard deviation of the descriptor. This type of normalization is called z-normalization and is applied using the expression:

$$norm\,(x) = \frac{(x - aver)}{sd} \tag{1.3}$$

where $aver$ is the average of the descriptor and sd is the standard deviation. The normalized descriptor will have an average equal to 0 and a standard deviation equal to 1.

In some data, where the objects are represented by descriptors of different type, it may be necessary to normalize different descriptors in different ways.

1.3.3 Training and Test Sets

A data set consists in the pair of input and output data. The main objective of a model is to obtain accurate and reliable predictions for new data. Accurate predictions for the training set do not mean that the model will be reliable when new objects are presented for classification or prediction. In the learning process, overtraining or overfitting of the data can occur when the model "learns " the training set too well. In fact, the model may become too adapted to the training set, and despite the accurate predictions for the training data, it may perform poorly when applied to new objects. It is extremely important to test the obtained model by application to an independent data set with known targets. So, the initial data set must be partitioned into, at least, training and test sets.

It is common to monitor the training of a model by testing its application to a data set not used for its construction (validation set). The training of the model is optimized on the basis of the results obtained for this validation set. Although the validation set is not exactly used for training, it is employed to decide which of a series of possible models is to be preferred. It is, therefore, not an independent test set. In that case, a final validation of the model must be performed by application to a data set of known targets that had not been used for anything until that moment. The three data sets are usually called training set, validation (or control) set, and test (or prediction) set.

The resulting data sets should encompass the same regions of the universe as the initial data set. If the partition of the initial data set is wrongly made the validation and test sets may have less relevant information than the training set and, as a consequence, even a model that learns the problem and performs well in the tests will fail when asked to make predictions to sets whose contents do not correspond to the learned space.

Some machine learning techniques, consisting of ensembles of models, obviate the definition of a validation set because their training algorithm creates and uses a different training set for each model of the ensemble. The predictions obtained by each model for the objects left out of its specific training set are combined into a global measure of the accuracy of the ensemble - it is composed of predictions obtained for objects effectively not used for training the models that predicted them. An example of such technique is the Random Forest.

1.3.4 Selection of Objects

The initial data set can contain several similar objects. The partition of data containing some redundancy into training and test sets is an important problem. In the work presented in this Thesis different strategies for data set partition are followed depending on the data and problem. The most common and simple strategy is the random selection of objects. Another strategy consists of using a Kohonen Self-Organizing map (Kohonen SOM). This technique distributes objects over a 2D surface (a grid of neurons) in such a way that objects bearing similar features are mapped onto the same or adjacent neurons (see Section 2.6). In this strategy, the SOM is trained with all objects of the data set and then one reaction is randomly taken from each occupied neuron and moved to the test set. The training set is selected by this way with the aim of covering as much as possible the problem space. Other strategies can be followed if the objective is to test the limit of the prediction ability of the method and its reliability. In that case the experiments are performed with lower similarities between the training and test sets. Experimental design techniques are another possibility for building the training and test sets. These techniques allow to cover the entire space problem as much as possible with the available data, or only some particular region of the problem space if this region is more important.

Some software and algorithms have their own procedures to divide the training sets. For example, the software used in this work to implement ensembles of Feed-Forward Neural Networks (EnsFFNNs) and Associative Neural Networks (ASNNs) randomly partitioned, before the training, the whole training set into a learning set with 50% of the objects and a validation set with the other 50%. By this way each network in the ensemble is trained with a different training set. Another example, is the algorithm of Random Forests, which consist of an ensemble of decision trees, each tree built with a random subset of the training set (usually with one third of the objects).

1.4 Automatic Learning

Since the advent of the computer, scientists have tried to make them learn from data. The research on the methods used to make computers learn is called automatic learning or machine learning.

There are two different types of learning: deductive and inductive learning. Deductive learning is based on a theory that allows the calculation of some property of interest. This is the case of quantum mechanics in chemistry that make possible the calculation with high accuracy of the structure and other physical properties of compounds. However, in many common cases, properties of chemical systems result from complex unknown mechanisms, and predictive models cannot be put forward by first principle methods. This is the case of the prediction of the biological activity of a compound. The answer for these applications can be obtained from inductive learning. In inductive learning techniques, the knowledge of previous experiments is used to learn the problem, the information is incorporated in a model, and this is used to make predictions for new objects. In this Thesis inductive learning techniques are used to establish classification models (for photochemical reactions, enzymatic reactions or metabolic pathways) or quantitative models (prediction of potential energy).

Mitchell presents a broad definition of machine learning [49]:

"...any of computer program (algorithms) that improves its performance at some task through experience."

more precisely Mitchell define the automatic learning by a computer in the following way [49]:

*"A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."*

This area merges knowledge from several fields such as cognitive science, computer science, pattern recognition and statistics. All contribute in a different but complementary way to the development of automatic learning methods. For example, cognitive science studies the concepts of thinking and learning contributing with theories in these subjects. In a completely different way computer science contributes with the technological and methodological support.

1.4.1 Automatic Learning Process

The automatic learning process starts with the selection of the data set. The selection of the data set and how to divide it in training, validation and test sets was discussed in Section 1.3.

An automatic learning method use the training set to learn from the presented examples and then use the test set to evaluate the quality of the obtained model - if it is able to generalize. The main learning strategies are the unsupervised and supervised learning.

1.4.1.1 Supervised Learning

The main objective in any supervised learning technique is to build a system that, after the learning procedure, can associate the input data, $\boldsymbol{X}_s$, with the output or target data, $\boldsymbol{Y}_s$. Supervised learning needs a set of pairs ($\boldsymbol{X}_s$, $\boldsymbol{Y}_s$) as input. The input of the technique is the vector $\boldsymbol{X}_s$ and the target (correct answer) is the vector $\boldsymbol{Y}_s$. After the training, it is expected that the obtained model can give correct predictions for a new object $\boldsymbol{X}$. During the learning procedure, the output of the system for a given object is compared with the corresponding target, and an error is calculated. The supervised learning methods use this error to make corrections that try to minimize it. In this work Counter-Propagation

Neural Networks, Back-Propagation Neural Networks, Associative Neural Networks, and, Random Forests were employed as supervised learning techniques. Some of them will be described in Chapters 2 and 3.

1.4.1.2 Unsupervised Learning

In unsupervised learning there is no information about the classes or output of the training examples. Only the input data is given to the system to learn. It is said that the learning is performed without a "teacher".

The main objective is to make a system learn the relations between objects, building a representation of the data, in the most part of the cases in a lower dimensionality than the input data. Generally, these techniques have an application in clustering, data compression and outlier detection. They are used to solve classification problems. Kohonen Self-Organizing Maps is an example of an unsupervised learning technique that was much used in this Thesis.

1.4.2 Types of Problems

Automatic learning methods can solve several types of problems that, in a general level, can be classified into four main types: auto-association or hetero-association, classification, transformation or mapping, and modeling. Only the last three are discussed since they are the most frequent and related with the applications shown in this Thesis.

1. **Classification:** In classification an object described by a group of descriptors is assigned to an appropriate class. After the training the method is able to assign an unknown object to a class.

2. **Transformation or Mapping:** The objective is the transformation or mapping of a multivariate space into another of lower dimension. The 2D mapping is usually the most convenient, providing a clear visualization of multidimensional objects. The resolution of this type of problems is the most similar to the biological process of thinking, learning and reasoning that is considered as a process of mapping multivariate signals of our body into a 2-dimensional plane of neurons in the brain.

3. **Modeling:** The objective is to develop a model that is able to obtain a specific $n-$variable output for any $m-$variable input. In a neural network for example, it is not needed to know the type of the analytical function. The non-linearity of the neuron and a sufficient number of weights is enough for a NN to learn the relation between the input and the output signals. Differently, decision trees allow the extraction of rules that explain the output for a certain object.

Chapter 2

Artificial Neural Networks

This Chapter introduces the basic concepts of neural networks and presents an overview of the neural network models used for the experiments reported in this Thesis. The first four Sections present some historical notes on the development of NNs, explain the method in general, compare the biological neuron and the artificial neuron, and describe the processing of information by the artificial neuron. Then the formation of layers of neurons and the linking of layers are explained. The last Sections present the NNs models used for the experiments of this Thesis (Kohonen Self-Organizing Maps, Counter-Propagation Neural Networks, Feed-Forward Neural Networks, Ensembles of Feed-Forward Neural Networks and Associative Neural Networks).

2.1 Historical Notes

Despite the numerous current applications of neural networks in the most diverse areas, the history of this field started with a long and difficult process more than 60 years ago.

An early model of the biological neuron was presented by McCulloch and Pitts, in 1943. [50–52] This model described the neuron as a linear computation unit able to receive several signals and producing an answer (output). The answer may take two values, zero if the neuron remains inactive, or one if the neuron "fires". The neuron remains inactive if the resulting value of the signals processment does not reach some predefined value. This model did not answer the question of learning. This problem was discussed by Hebb [51–53] in 1949 with the publication of the book "The Organization of Behavior" where a learning rule for the modifications of the synaptic strengths was presented for the first time. By the Hebb rule the synaptic strength of a neuron has a proportional variation with the activity in the anterior and posterior part of the synapse. The first computers opened a possibility to solve several problems with new methods. By the same time appeared the field of "artificial intelligence, AI". Rochester et al, [51,52] in 1956 were responsible for the first attempt to test the neuronal theory based on the Hebb learning postulate. This is considered the first computational simulation of neural networks.

In 1959, Rosenblatt [51–53] developed a structure called perceptron. This structure basically consists of a neural network with one layer that produces an output with value 1 or -1. Despite the importance of this research, the perceptrons are only able to solve linearly separable problems (in a linearly separable problem every solution can be placed into one of two sets separated by a straight line). The solution to a linearly separable problem by a perceptron is simply the linearly-weighted sum of the inputs.

The exclusive-or logical operation, XOR condition, [53, 54] (XOR(x, y) is true if one and only one of the variables, x or y, is true) is an example of a simple problem with classes linearly inseparable. In 1969, Minsky and Papert [51, 52] published a book where they showed the limitations of the neural network models in solving even simple problems like the XOR problem. They also theorized that even multilayer perceptrons would not improve the results. As a consequence of these and other critics, and with the discouraging results obtained even in simple problems, the research in neural networks was almost dormant over the following years with few exceptions, e.g. some experiments carried out by Kohonen (1972) and Anderson (1972) .

In 1971 was published the first application of AI in chemistry. Jurs and Isenhour [55] implemented a decision tree with 26 binary decisions to predict the molecular formulas of compounds from their mass spectra. This application had the same limitation of the perceptron - the data had to be linearly separable.

Hopfield showed in 1982 [51, 52] that the neural network models of binary neurons can be treated as spin states establishing an isomorphism between such a recurrent network and the Ising model used in statistical physics. Beyond this important innovation, Hopfield was also responsible for the introduction of non-linear transfer functions giving a non-linear nature to the neuron. This non-linearity gave a greater flexibility to the networks when comparing with the old models. The new developed model was called the Hopfield network.

An important issue absent from this network model was a learning algorithm allowing the weight correction in a network with more than one layer. In fact there is no refinement of weights values in the Hopfield network because they are calculated directly from the object patterns.

The algorithm that solved this problem using a non-linear transfer function and the adaptation of the weights (learning) was developed independently by Parker (1985) and Rumelhart, Hinton and Williams (1986) but was, in fact, described earlier by Werbos in this PhD Thesis from the University of Harvard in August 1974, and is called Back-Propagation Algorithm. The name comes from the way the weights are adjusted, from the output layer until the first layer, layer by layer, and became the most popular learning algorithm in NNs. [51, 52]

It is to point out that even with the success in the development and application of these models, it is clear that the true understanding of the working mechanism of the brain is still far away. The artificial neural networks capabilities are still very rudimentary when compared with the biological networks that they want to simulate. However, they represent a very useful way for information processing and data mining. The applications in chemistry grew rapidly from 3 publications in 1988 to ∼1000 in 1997, and almost 5000 in the period 2001-2002, which gives an idea of the possibilities of application of NNs in all chemistry fields. [1]

2.2 Definition of an Artificial Neural Network

The availability of information is a defining feature of the contemporary world. However, the huge amount of available information can difficult the search of the correct data to solve a specific problem. Methods for data analysis that are able to extract the desired information are therefore essential.

Data analysis has been performed since long ago by statistical methods, but it is recognized that the human brain analyzes and processes information in a different way. The data acquisition by the brain is not performed by statistical methods, and the goal to simulate the biological acquisition of knowledge lead to the development of the mathematical models called artificial neural networks.

The neural networks learn by training with experiments, like the human beings, on the basis of pre-defined rules. Advances in neurophysiology and the development of experimental techniques like electroencephalography, magnetic resonance imaging, single-photon emission computerized tomography have helped in understanding the anatomy and function of the human brain. Based on the available information about the brain, mathematical models and algorithms started to be developed to mimic the way the brain acquires and processes information.

The Haykin [51] definition of NNs, viewed as an adaptive machine, is the following:

"A neural network is a massively parallel distributed processor made up of simple processing units, which has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:

1. *Knowledge is acquired by the network from its environment through a learning process.*

2. *Interneuron connection strengths, known as synaptic weights, are used to acquired knowledge."*

Following the approach of Gasteiger [1] this brief explanation of NNs will first consider a NN as a "black box" that can receive a group of input signals and transforms them into one or more output signals.

The input and output signals can be, for example, the spectra of a compound (the input) and his structure (the output) in a problem of structure elucidation, or vice-verse for a spectra prediction task. One advantage of NNs is the fact that the same learning algorithm can be used to solve several different problems.

A NN consists of basic operating units, artificial neurons, that are connected inside the “box”. The network receives the signals that pass along these connections and are distributed, transformed and eventually reunited to produce outputs. NNs are formed by neurons that are connected into networks.

A NN transforms a $m-$variable input into a $n-$variable output. The input and output variables can be:

- real numbers, preferably between the range from 0 to 1 or -1 to 1 (some methods can handle smaller or greater real values).
- binary numbers, 0 and 1
- bipolar numbers, -1 and +1

The number of outputs is, in general, smaller that the number of inputs.

2.3 Neuron Model

The artificial neuron is designed to simulate the function of biological neural cells of living organisms. This Section presents a description of the biological neuron and its analogy with the artificial neuron.

2.3.1 Analogy between Biological Neuron and Artificial Neuron

The human nerve system consists of $\sim 10^{10}$neural cells, or neurons. A typical neuron consists of a cell body (soma), with a nucleous, that has two types of extensions: the dendrites and the axon. Figure 2.1 shows a simplified model of a biological neuron.

The dendrites receive signals and send them to the soma. A neuron has many more dendrites than the representation shows and presents a large surface area to receive signals from other neurons. The axon is responsible for the transmission of signals to other neurons or to muscle cells and branches in several “collaterals”. The axon and collaterals of a neuron end in synapses, which are responsible for the contact with the dendrites of other neurons. A neuron motor has thousands of synapses (40% of the surface is covered by them).

The transfer of signal between the dendrites and the axon has an electrical nature (the transport of ions) while the transmission of the signal in the synapse has an chemical nature. The electrical signal in the axon releases a neurotransmitter that was stored in

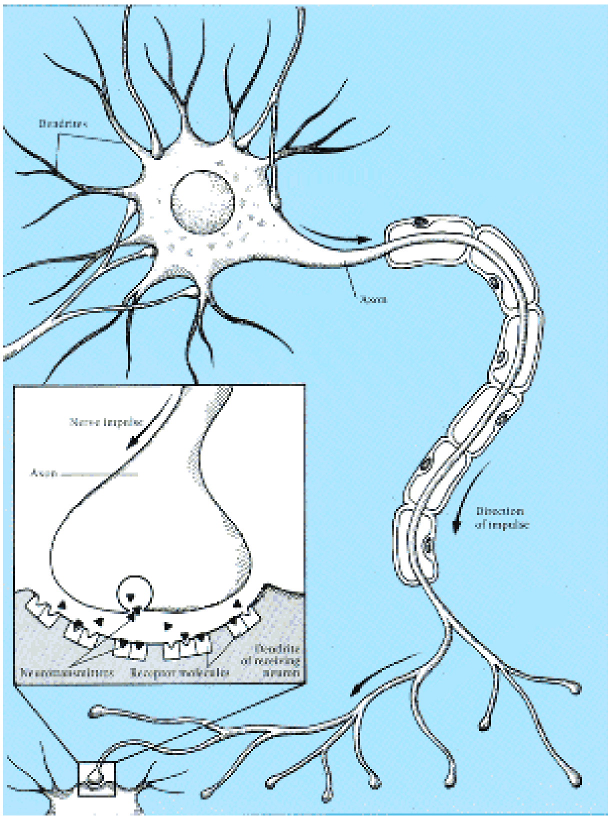


Figure 2.1: Biological neuron.

vesicles in the pre-synaptic membrane. This chemical substance is diffused across the synaptic gap and postsynaptic membrane into the dendrite of the other neuron as the zoom in Figure 2.1 shows. In the dendrite of this second neuron the neurotransmitter generates a new electrical signal that is passed through the neuron. The signal can be transmitted only in one direction. The postsynaptic membrane is not able to release the neurotransmitter, so synapses can only send the signal in one direction, functioning as gates.

The shape and appearance of signals generated by the neurons are very similar between species even when compared between the most primitive and the highly evoluted species. The intensity of the produced signals (the frequency of firing) by the neurons can differ depending on the intensity of the stimulus. This similarity suggests that brain functions are more dependent on the entire network of neurons and the way neurons are connected, than on the way that a single neuron works.

The signal passing between neurons is modulated by the synapses that work as same type of "barrier". The modulation of the signal depends on the *synaptic strengths* and are named in the artificial neurons as *weights, w.*

The synaptic strength determines the intensity of the signal that enters in the next neuron through the dendrites. **The adaptation of the synaptic strength to a particular problem is the essence of learning.**

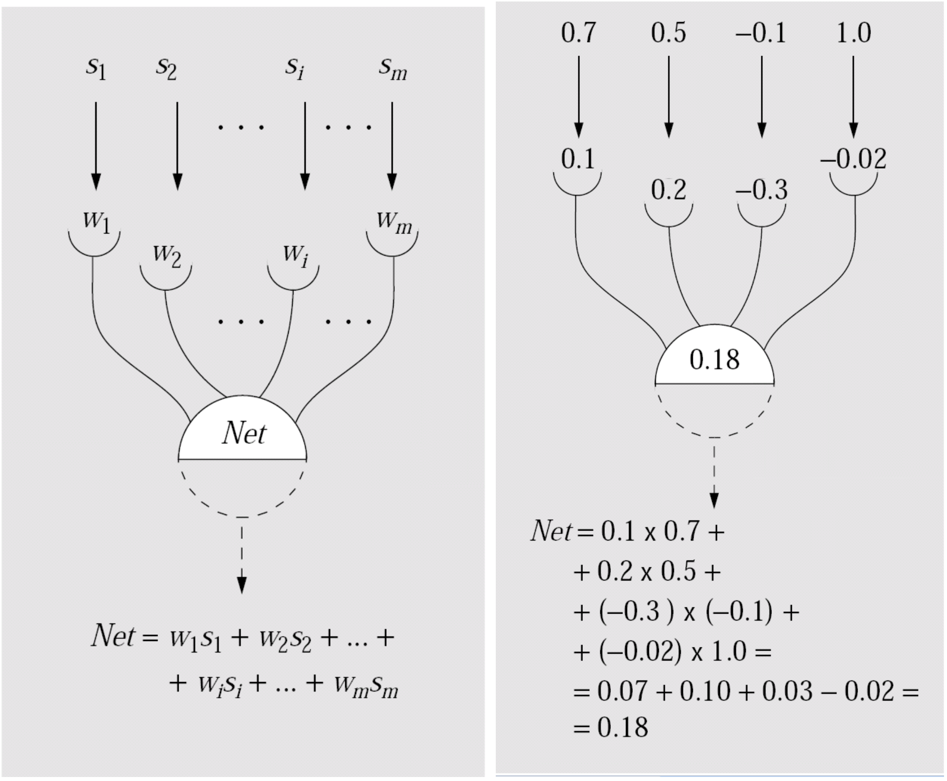


Figure 2.2: Calculation of the Net in a artificial neuron with m signals (the figures are adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

2.3.2 Synaptic strength of the Artificial Neuron (Weight)

The artificial neuron is able to receive a large number of signals simultaneously, like the biological neuron. The individual signals are labeled s_i and the corresponding synaptic strengths (weights) as w_i. At a given moment a neuron receives many signals that add together in a so-called *net input.* In the most used type of artificial neuron the net input (called *Net)* is a function of all signals s_i that reach the neuron within some gap of time, and of all synaptic strengths (weights, w_i). This function is the sum of products of the signals s_i and the corresponding weights w_i:

$$Net = w_1s_1 + w_2s_2 + ... + w_is_i + ... + w_ms_m \tag{2.1}$$

The *Net* value is not yet the output of the neuron. In this model of the artificial neuron, the output is calculated in two steps. The calculation of the first step is now illustrated with an example from ref [1]. Figure 2.2 shows how the signals are processed using Equation 2.1.

In this example the neuron has four synapses with weights 0.1, 0.2, -0.3 and -0.02. As this neuron has only four synapses, it can only receive 4 signals (with intensities 0.7, 0.5, -0.1 and 1.0) at the same time. The *Net* is calculated and has the value of 0.18. The second part of the neuron function will be described in the next Subsection. The group of signals $s_1, s_2, s_3, ..., s_i, ..., s_m$ received by the neuron from m neurons can be represented as a multidimensional vector $\boldsymbol{X}$, where the components are the individual signals x_i:

$$\{s_1,\, s_2,\, s_3,\, ...,\, s_i,\, ...,\, s_m\} = \boldsymbol{X}(x_1,\, x_2,\, ...,\, x_i,\, ...,\, x_m) \tag{2.2}$$

and the same was applied to all synaptic strengths that were represented in a multidimensional weight vector ***W:***

$$\boldsymbol{W} = (w1,\, w_2,\, w_3,\, ...,\, w_i,\, w_m) \tag{2.3}$$

Each neuron should have at least as many weights as connected neurons. With the vector notation the equation 2.1 was equivalent to:

$$Net = w_1x_1 + w_2x_2 + ... + w_ix_i + ... + w_mx_m = \boldsymbol{WX} \tag{2.4}$$

this is the dot product of two vectors: the weight vector $\boldsymbol{W}$ and the input vector $\boldsymbol{X}$. The input vector is a multivariate object described by several parameters. If more than one object exist they are distinguished by an index s, $\boldsymbol{X}_s$.

The univariate signals reaching the individual synapses are components of these multivariate objects and have two indices, x_{si}. The first index labels the multivariate object and the second the synapse to which this individual signal is linked. From this point the components x_{si} will be called *signals,* and the multivariate inputs, *objects,* or vector $\boldsymbol{X}_s$. So in equation 2.4 *Net* is the scalar product of a weight vector $\boldsymbol{W}$ and a multivariate vector $\boldsymbol{X}_s$ that represents an arbitrary object.

2.3.3 Transfer Functions in Artificial Neurons

The first part of the model of the neuron was presented in the last Subsection and was dedicated to the calculation of the signal *Net.* Now the second part will be explained. Basically the second function of the neuron consists in applying a non-linear transformation to the input signal *Net.* If only one step was considered, the weighted sum of the input signals, the signal *Net*, could be very large or negative. A very large value could be problematic in the learning procedures, and a negative value is unrealistic because one neuron only fires when certain value is reached. To make the output between certain intervals, a so-called transfer function is applied to the input signal *Net* of the neuron:

$$out = f\,(Net) \tag{2.5}$$

The transfer function has to produce the final output of the neuron with certain properties to make it more realistic. The final output signal should be non-negative, continuous and confined to a specific interval, such as between zero and one. There are several functions that can satisfy these conditions, for example the hard-limiter and the

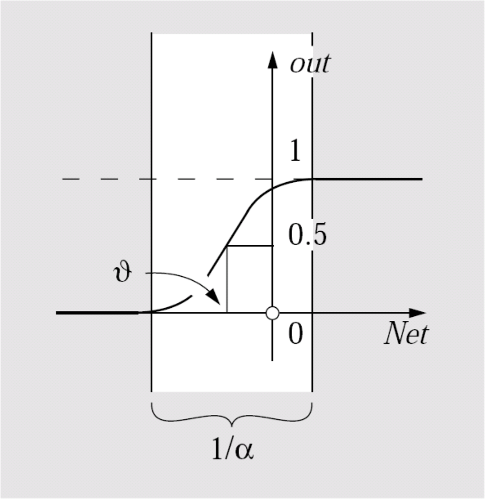


Figure 2.3: Sigmoidal transfer function (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

threshold logic functions, but only the most used in chemical applications - the sigmoidal function, sf - will be discussed here.

The sigmoidal function, shown in Figure 2.3, has the form

$$sf\,(Net,\,\alpha,\,\vartheta) = \frac{1}{1+\exp^{-\alpha(Net-\vartheta)}} \tag{2.6}$$

or

$$sf\,(Net,\,\alpha,\,\vartheta') = \frac{1}{1+\exp^{-(\alpha Net-\vartheta')}} \tag{2.7}$$

where α is the reciprocal width of the swap interval and ϑ is the threshold value, above which the neuron fires. These two parameters are illustrated in Figure 2.3.

The non-linearity is one of the advantages of the sigmoidal function. Although some neurons may show a linear relation between Net and the output, it is the non-linear nature of the transfer function that allows for the flexibility to learn different situations. The easy to obtain derivative is another advantage of the sigmoidal function. If equation 2.6 is simplified to the form

$$sf\,(x) = \frac{1}{1+\exp^{(-x)}} \tag{2.8}$$

the derivative is

$$\frac{d\,(sf\,(x))}{dx} = \frac{-1}{\left(1+\exp^{(-x)}\right)^2}\left[\frac{d\left(1+\exp^{(-x)}\right)}{dx}\right] = \frac{\exp^{(-x)}}{\left[1+\exp^{(-x)}\right]^2} =$$

$$= \frac{sf\,(x)\exp^{(-x)}}{1+\exp^{(-x)}} = sf\,(x)\left[\frac{-1+1+\exp^{(-x)}}{1+\exp^{(-x)}}\right] = sf\,(x)\,(-sf\,(x)+1) =$$

$$= sf\,(x)\,(1-sf\,(x)) \tag{2.9}$$

It is clear that where $sf\,(x) = 0$ or $sf\,(x) = 1$ the derivative is zero. This is an important property that will be explained later in the description of the learning procedure of some types of neural networks. The final advantage over other functions is the fact that by changing the parameters α and ϑ the swap interval of the decision can be influenced and the obtained predictions can be quantitatively evaluated according to the values of these parameters.

An extra parameter, called bias, is usually added to increase the adaptability of the network to the specif problem. This parameter is typically used in Back-Propagation Neural Network that will be described later.

An artificial neural network is described by the set of weights and by the parameters of the transfer function. The weights are, in general, initialized with small random numbers. In the transfer function, apart the swap interval, the other important parameter is the threshold ϑ, that is the point where the neuron starts to "react". The two equations that describe as the artificial neuron function are the Equations 2.4 and 2.6:

$$Net = w_1x_1 + w_2x + ... + w_ix_i + ... + w_mx_m = \boldsymbol{WX}$$

$$sf\,(Net,\,\alpha,\,\vartheta) \quad = \quad \frac{1}{1+\exp^{-\alpha(Net-\vartheta)}}$$

where:

- w_i - synaptic strength (weight) of the synapse i
- x_i - signal that are transmitted by a connected neuron to other neuron in the synapse i
- m - number of synapses in the neuron
- Net - net input of the neuron
- α is the reciprocal width of the swap interval
- ϑ is the threshold value of the sigmoidal transfer function

To better understand the introduction of the *bias* parameter, it is necessary to consider from the transfer function $\alpha Net - \vartheta$ (arg), so:

$$arg = \alpha w_1 x_1 + \alpha w_2 x_2 + ... + \alpha w_m x_m - \alpha \vartheta \tag{2.10}$$

by substitution of the products of the two values αw_i by the single value w_i', and $\alpha\vartheta$ by ϑ' the Equation 2.10 becomes:

$$arg = w_1' x_1 + w_2' x_2 + ... + w_m' x_m - \vartheta' \tag{2.11}$$

and if we regard ϑ' as a product of ϑ and a component x_{m+1}, that **is always equal to 1,** the Equation 2.11 was transformed in:

$$arg = w_1' x_1 + w_2' x_2 + ... + w_m' x_m - \vartheta' x_{m+1} \tag{2.12}$$

if $\vartheta'$is labeled as w_{m+1}' it is created a product of w_{m+1}' and a signal x_{m+1}(that is always equal to 1). As this is analogous in all products in the series the summation can be extended by one more element:

$$arg = w_1' x_1 + w_2' x_2 + ... + w_m' x_m + w_{m+1}' x_{m+1} = \sum_{i=1}^{m+1} w_i' x_i \tag{2.13}$$

inserting the obtained equation in the sigmoidal transfer function (Equation 2.6) it is obtained:

$$sf\,(Net,\, \alpha,\, \vartheta) = \frac{1}{1 + \exp^{-\sum_{i=1}^{m+1} w_i x_i}} \tag{2.14}$$

so the output produced by the neuron depends only on the weight vector $\boldsymbol{W}$ of dimension $(m+1)$ and on the input signal $\boldsymbol{X}$ of dimension $(m+1)$:

$$\boldsymbol{W} = (w_1,\, w_2,\, ...,\, w_m,\, w_{m+1}) \tag{2.15}$$

and

$$\boldsymbol{X} = (x_1,\, x_2,\, ...,\, x_m,\, 1) \tag{2.16}$$

the extra weight which always receive an input value of 1 is the bias.

The threshold value ϑ and the constant α can be regarded as an additional "synaptic strength" called bias to which a signal of value 1 is always transmitted.

2.4 From Neurons to Networks of Neurons

2.4.1 Generalities About Neural Networks

A biological neuron can fire at intervals of one millisecond, 10^{-3}s and the reaction time of most vertebrates is $\sim 10^{-1}$s. With these values we can conclude that all reactions of the brain to an external stimulus occur in less than 100 firing times. As 100 steps appear clearly insufficient to solve the complicated problems associated with rapid reactions of vertebrates, one must conclude that neurons are interconnected, forming a network - a massively parallel processor - that is able to use simultaneous parallel paths to process information.

With the model of the artificial neuron presented, the organization of neurons in a network to form an "artificial neural network" or simply neural network, will be described.

First, all neurons have to receive the same signal $\boldsymbol{X}$ for processing at the same time With this condition, the processing of the information giving rise to an output occurs simultaneously in all neurons. A group of neurons that produce a set of outputs simultaneously is called a layer of neurons. In a layer of neurons each neuron j receives a specific net input Net_j and gives origin to a specific out_j that is passed on to the next layer of neurons, transformed by the weights associated with the connections. Networks learn by modifying the weights.

In the next layer the individual output signals from each neuron of the previous layer are combined in the net input signal vector, *Net* and an output is produced in the same way. The output vector, ***Out***, of one layer is the input vector, ***Inp*** or $\boldsymbol{X}$, of the next layer of neurons.

Multilayer networks work sequentially, i.e. the neurons in a layer only receive signals from the layer before. Generally no more than two layers are used.

2.4.2 Networks with One Layer

A layer of neurons is usually represented by programmers and mathematicians as a matrix of weights. In a matrix of weights $\boldsymbol{W}$, the rows represent the neurons and each row j can be labeled as a vector $\boldsymbol{W}_j$ that represents a neuron j with m weights w_{ji}, $\boldsymbol{W}_j$=(w_{j1}, w_{j2}, ..., w_{jm}). All weights in the same column i, w_{ji} ($j = 1, 2, ..., n$) receive the same signal x_i simultaneously.

A one-layer network is showed in Figure 2.4. The network consists of a layer with three neurons, each one with 5 weights. Each neuron receives the same m signals (x_1, x_2, x_3, ..., x_{m+1}, 1), in the example of Figure 2.4 m=5. The weight w_{ji} is on the i-th position of the j-th neuron, the weight w_{23} represented in Figure 2.4 is the 3^{rd} weight of the 2^{nd} neuron. The circles in black correspond to the bias.

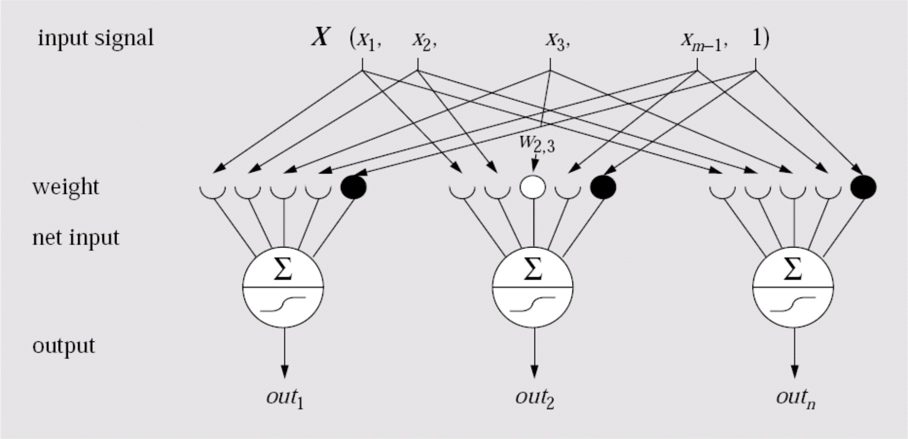


Figure 2.4: Network with one layer (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

2.4.3 The Input, Hidden and Output Layer in a Neural Network

The input signal x_i of the input vector $\boldsymbol{X}$ are “distributed” over as many weights as there are neurons in the layer. For example in Figure 2.4 each signal x_i is distributed over three weights corresponding to three neurons. For clarity of presentation, a non-active layer of “input neurons” are included that make no change in the input signals x_i. The “input neurons” have only the function of distributing the signals to the active layer and do not modify them. The input neurons are drawn as squares and the neurons in the active layers as circles. When the number of layers are counted to classify the architecture of the network the non-active layer is not included.

The layers below the input layer are generally called hidden layers because they are not connected to the “outside” as the input and output layers. The layer of neurons that produce the final signals, which are not transmitted to other layer, is called output layer.

In more complex neural networks models, which will not be treated in this Thesis, some of the signals may be the input of neurons at several layers.

The architecture of a network is defined by the following parameters:

- number of inputs and outputs
- number of layers
- number of neurons in each layer
- number of weights in each neuron
- how weights are linked together within or between the layers
- which neurons receive which signals

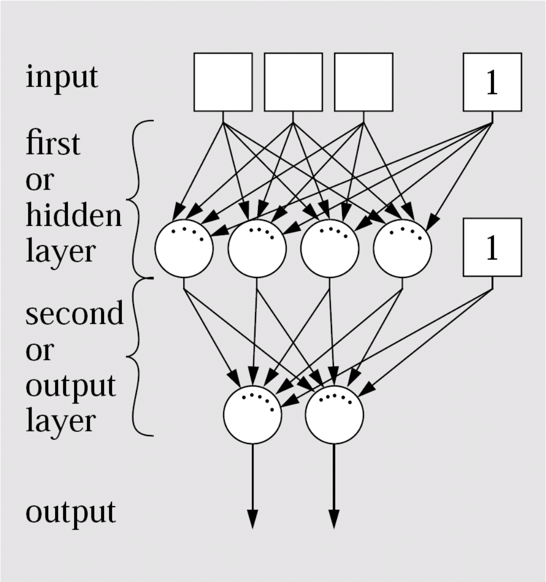


Figure 2.5: Example of a network with one input layer, one hidden layer and one output layer (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

The number of input and output neurons depends on the specific problem and on the specific procedure to encode input and output data. The number of hidden neurons is set empirically. They should be enough to enable learning the problem, but not too many to avoid overfitting.

Figure 2.5 shows an example of a network with architecture 3×4×2.

2.4.4 Matrix Representation of a Neural Network

In the graphical representation of the neural network presented in Figure 2.5 the neurons are represented as circles and the interconnection between layers by lines. The arrows represent the direction of the signal. This representation is based on the model of the biological neuron and is a simplification of how it works. However, for programming and description of learning procedures a matrix representation has some advantages. In a matrix representation a layer of n neurons, each one with m weights is considered as a weight matrix $\boldsymbol{W}$ of dimension $(n \times m)$. For a network with more than one layer it is introduced the superscript l to specify the index of the layer in each weight matrix:

$$w_{ji}^{l} \tag{2.17}$$

that refers to the i-th weight of the j-th neuron in the l-th layer. The weight matrix of the input layer, $\boldsymbol{W}^0$ that transmits m signals is a vector containing the value 1, n times:

$$\boldsymbol{W}^0 = (1,\, 1,\, 1,\, ...,\, 1) \tag{2.18}$$

The weight matrix of the first active layer will have the index 1, $\boldsymbol{W}^1$ and the output layer (the last layer) will have an index equal to the number of active layers in the network.

The matrix notation shows clearly that the input and output signals of the l-th layer, $\boldsymbol{X}^l$ and $\boldsymbol{Out}^l$ are vectors of dimension m and n. It is remembered that the input vector of a layer is the output vector of the above layer:

$$\boldsymbol{X}^l = \boldsymbol{Out^{l-1}} \tag{2.19}$$

and

$$\boldsymbol{Out^l} = \boldsymbol{X}^{l+1} \tag{2.20}$$

To submit an m-variate input signal to a one-layer network with n neurons (each one with m weights) the vector $\boldsymbol{X}(x_1,\, x_2,\, ...,\, x_{m-1},\, 1)$ is multiplied by the weight matrix $\boldsymbol{W}$ of dimension $(n \times m)$ resulting in a net input vector Net $(Net_1,\, Net_2,\, ...,\, Net_n)$ of dimension n.

$$Net = (Net_1, Net_2, ..., Net_j, ..., Net_n) =$$

$$= \begin{bmatrix} w_{11} & w_{12} & . & . & . & w_{1m} \\ w_{21} & w_{22} & . & . & . & w_{2m} \\ w_{31} & w_{32} & . & . & . & w_{3m} \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & w_{ji} & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ . & . & . & . & . & . \\ w_{n1} & w_{n2} & . & . & . & w_{nm} \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \\ . \\ . \\ . \\ x_i \\ . \\ . \\ . \\ x_{m-2} \\ x_{m-1} \\ 1 \end{bmatrix} \tag{2.21}$$

For layer l each component Net_j is calculated in the following way

$$Net_j^l = \sum_{i=1}^{m} w_{ji}^l x_i^l \qquad j = 1,\, 2,\, ...,\, n \tag{2.22}$$

The index j spans the n neurons and i the m weights in the j-th neuron.

The next matrix equation 2.23 incorporates the above equation for all net inputs in a one layer network:

$$Net^l = \boldsymbol{W}^l \boldsymbol{X}^l \tag{2.23}$$

The superscript l distinguishes the layers in a network with more than one layer. As the input of the l-th layer is the output of the $(l-1)$-th layer the Equations 2.22 and 2.23 can be written as:

$$Net^l = \boldsymbol{W}^l \boldsymbol{X}^l = \boldsymbol{W}^l \boldsymbol{Out^{l-1}}$$

or:

$$Net^l_j = \sum_{i=1}^{m} w^l_{ji} out^{l-1}_i \qquad j = 1,\, 2,\, ...,\, n \tag{2.24}$$

The output vector $\boldsymbol{Out}^l$ is obtained from the net vector Net^l by application of a transfer function. If the sigmoidal function is the transfer function applied:

$$\boldsymbol{Out^l} = sf(Net^l)$$

2.5 Learning from Information

One essential property in a NN is the ability to learn from an environment and improve its performance during the learning process. A NN learns by an iterative process of adjusting the synaptic strengths (weights). A network improves its internal representation of the problem with each iteration of the learning process. The definition of learning in the context of NNs given by Haykin [51] adapted from Mendel and Macclaren is the following:

"Learning is a process by which the free parameters of a neural network are adapted through a process of stimulation by the environment in which the network is embedded. The type of learning is determined by the manner in which the parameter changes take place."

This definition implies the following steps:

1. The NN is stimulated by an environment (a training set of objects each one described by m-variables)

2. The NN changes its free parameters as a result of the stimulation (the network corrects its weights)

3. The NN answers in a new way to the environment because of the changes occurred in its internal structure

The pre-defined rules for building a solution to the learning problem is called the learning algorithm. There are several learning algorithms each one differing in the way weights are adapted and depending on the architecture of the network.

The three fundamental features of a NN are the arithmetic operation in the neuron, the network architecture and the learning process. The following Sections describe the NN models and corresponding learning algorithms used to perform the work presented in this Thesis.

2.6 Kohonen Self-Organizing Maps

2.6.1 Introduction

This Section describes the learning procedure of the Kohonen Neural Networks or Kohonen Self-Organizing Maps (Kohonen SOMs). This is a neural network model with only one layer that learns in an unsupervised way by competitive learning.

In the learning phase the Kohonen SOM automatically adapts to the objects in such a way that similar objects are associated to topologically close neurons. This is an important feature when multidimensional data have to be analyzed in an automatic way. Topology of data is related to proximities and distances between data points, usually similarity relationships.

Compression of data allows handling large amounts of data with efficiency. When a file is compressed the information is still available and the file is reduced. The term compression can be interpreted here as a process of mapping a multidimensional input into an output space of smaller dimension. One of the problems that arise in compression of data is the loss of information, so a method has to be able to compress the data as much as possible with the minimum of information loss.

Teuvo Kohonen developed the concept of *self-organized topological feature maps.* These maps preserve the topology of multidimensional data in a two-dimensional plane - the topological relations among the data are preserved. This concept is the essential feature of the Kohonen approach.

2.6.2 Architecture

The Kohonen SOM is considered the most similar to biological neuron from all the neural networks types and architectures. A SOM consists of a single layer of neurons organized in a one-dimensional array or on a two-dimensional plane with a well defined topology (a well defined topology means that each neuron has a defined number of neurons as nearest

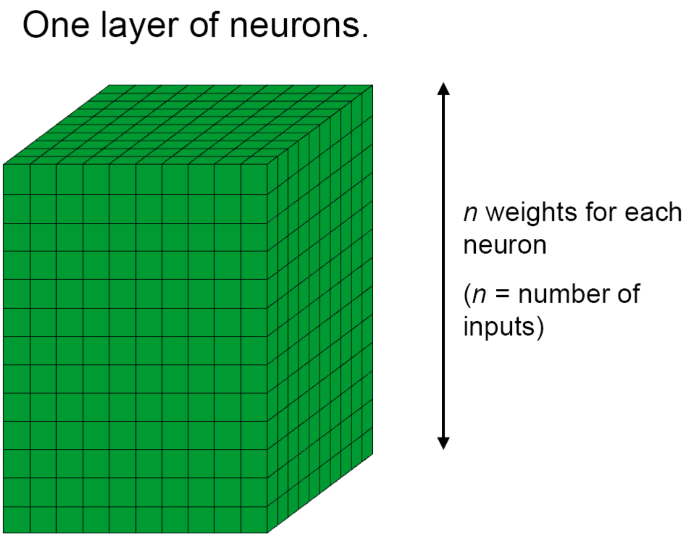


Figure 2.6: Kohonen SOM architecture.

neighbors, second nearest neighbors and so on). A neuron in a SOM can be represented in the form of a column. In Figure 2.6 it is easy to see that the weights that receive a specific variable of an object are in a single and well defined level of weights, i.e. each level of weights is affected separately by each input variable. The weights of each neuron are the weights of all levels that are aligned in a vertical column and there are as many weight levels as input variables describing the objects.

When the Kohonen map is defined as a grid of squares each neuron has eight nearest neighbors. In Kohonen SOMs the similarity between objects is related to topological relations between the neurons in the network. This concept allows for the mapping of objects in such way that similar objects excite neurons that are close in the plane.

The main objective of learning in a Kohonen SOM is to map similar objects in close positions (neurons) on the map. In order that each neuron on the plane (network) has the same number neighbors (Figure 2.7) the plane is transformed into a torus as Figure 2.8 shows. The upper row is considered to be adjacent to the bottom row, and then the left column to the right column. The indices of the edge neurons are converted to the toroidal topology , for the plane appear to wrap around.

In a Kohonen SOM only one layer of neurons is active. The active layer is arranged in a two-dimensional grid of neurons. All neurons in the active layer receive the same multidimensional input but, differently from other NN models, in a SOM the output of each neuron is not connected to another layer of neurons, or to all neurons on the plane. Only the topologically closest neurons know the output of the neuron but this information is only used to correct weights in such a way that close neurons have similar outputs when similar inputs are submitted.

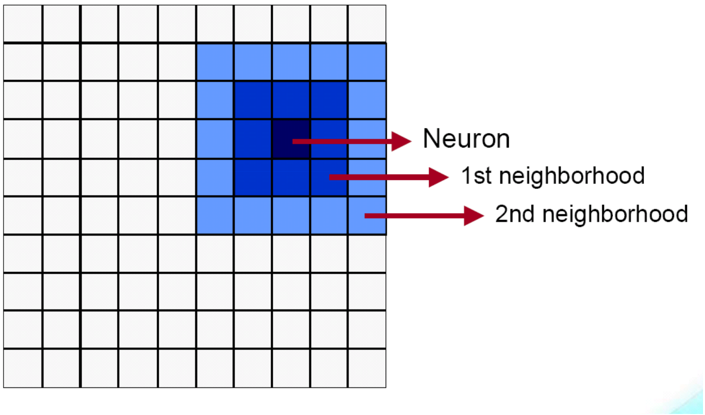


Figure 2.7: The square neighborhood have 8, 16, 24, ... neighbors in the 1^{st}, 2^{nd}, 3^{rd}, ... neighborhood.

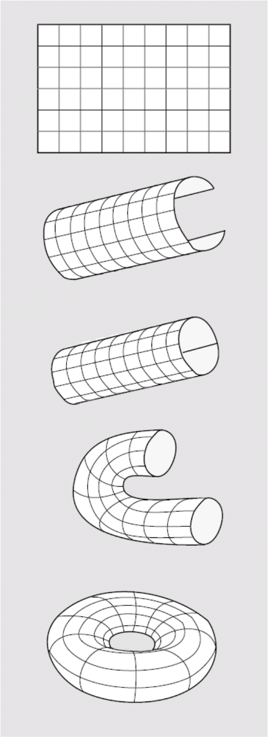


Figure 2.8: Wrapping a two-dimensional plane into a toroid (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

2.6.3 Unsupervised Competitive Learning

Competitive learning in a Kohonen SOM means that all neurons of the two-dimensional plane compete for activation when a stimulus is provided (an object is presented), but only one is selected. It is a "winner takes all" method. The winning neuron, $'c'$ (for central), is selected if it has either the largest output in the entire network:

$$out_c \leftarrow max\left(out_j\right) = max\left(\sum_{i=1}^{m} w_{ji} x_{si}\right) \quad j = 1,\, 2,\, ...,\, n \tag{2.25}$$

or the weight vector W$_j$ $(w_{j1},\, w_{j2},\, ...,\, w_{jm})$ most similar to the input signal $\boldsymbol{X}_s(x_{s1},\, x_{s2},\, ...,\, x_{sm})$:

$$out_c \leftarrow min\left\{\sum_{i=1}^{m} \left(x_{si} - w_{ji}\right)^2\right\} \quad j = 1,\, 2,\, ...,\, n \tag{2.26}$$

where the index j refers to a particular neuron, n is the number of neurons, m is the number of weights in each neuron, s is a particular input. The index j, that refers to a neuron in the Kohonen layer, depends on the layout of the network. The applications of Kohonen SOMs in this Thesis use the second rule, and a two-dimensional layout with the index j describing the position of a particular neuron in a two-dimensional plane. In general this position is described by two indices, corresponding to the two coordinates of the neuron in the plane.

Kohonen layers are often squares, but in some cases rectangular layers are also applied. After finding the winning neuron c, its weights w_{ci} are corrected to make them even more similar to the input object. It is said that the winning neuron was excited (or activated) by the object, and its weights are then adjusted to make them even more similar to the properties of the presented object. Not only the winning neuron has its weights, w_{ci}, adjusted but also the neurons in its neighborhood. The extent of adjustment depends, however, on the topological distance to the winning neuron c - the closer a neuron is to the winning neuron the larger is the adjustment of its weights, w_{ji}. The scaling function is a topology dependent function:

$$a\left(\cdot\right) = a\left(d_c - d_j\right) \tag{2.27}$$

where $d_c - d_j$ is the topological distance between the central neuron c and the current neuron j, while the level of the stimulation depends on the function $a\left(\cdot\right)$. Figure 2.9 shows the most used functions in Kohonen SOM. The correction decreases with the increasing of d_c and with each iteration cycle of the Kohonen learning process. An "epoch" corresponds to the presentation of all objects of the training set to the SOM, once, for learning, or to the presentation of as many objects as the size of the training set but randomly chosen.

In the Kohonen learning procedure it is assumed that the network "learns" - the weights

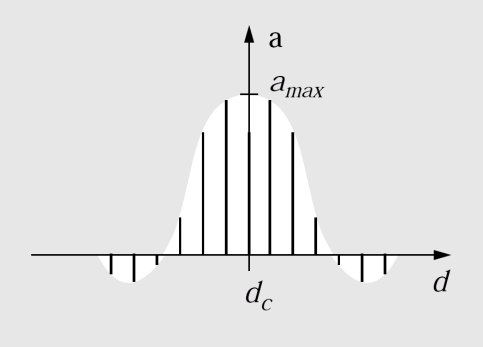


Figure 2.9: "Mexican Hat" function for scaling the corrections on neighbor weights (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

becomes more similar to the input objects so the Equation 2.27 is multiplied by another monotonically decreasing function $\eta(t)$:

$$f = \eta\,(t)\,a\,(d_c - d_j) \tag{2.28}$$

where t is the number of objects of the training set (if the number of objects is very large) or the number of epochs. The parameter t can be associated with time - the training time is proportional to the number of objects of the training set or to the number of epochs. The function $\eta\,(t)$ can be expressed as:

$$\eta\,(t) = (a_{max} - a_{min})\,\frac{t_{max} - t}{t_{max} - 1} + a_{min} \tag{2.29}$$

where t_{max} is the number of objects of the training set or the number of epochs predefined in the beginning of the training. The two constants a_{max} and a_{min} are the upper and lower limits between which the correction is decreasing from the beginning to the end of the training. If the function used is the "Mexican hat" it may develop empty spaces in the map in the borders between the categories due to the "destimulation" of the borders of the selected neighborhood. Also the span of the neighborhood for the scaling function changes during the training - as the training goes on fewer neurons have their weights corrected.

The corrections of the weights $w_{ji}$of the $j-$th neuron in the region defined by the function f is done in the following way:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \eta\,(t)\,a\,(d_c - d_j)\left(x_i - w_{ji}^{(old)}\right) \tag{2.30}$$

where x_i is a component of the input $\boldsymbol{X}_s$, c is the central or winning neuron and the only corrected is j, a particular weight of the neuron j is designated by i, t is the iteration cycle.

No matter if the difference between $x_i - w_{ji}^{(old)}$ is positive or negative, $w_{ji}^{(new)}$ will be more similar to x_i than $w_{ji}^{(old)}$.

The correction function for the maximum signal criterion, Equation 2.25, is similar:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \eta\left(t\right) a \left(d_c - d_j\right) \left(1 - x_i w_{ji}^{(old)}\right) \tag{2.31}$$

After the weight corrections, using Equation 2.31, the weights should be normalized to a constant value, usually 1:

$$\sqrt{\sum_{i=1}^{m} w_{ji}^2} = 1 \tag{2.32}$$

The output in Kohonen SOMs has only a qualitative meaning and not quantitative like in other types of neural networks, it only locate the neuron with the weights most similar to the object features.

The algorithm for one cycle of Kohonen learning can be summarized in the following steps (Figure 2.10 illustrates SOM training and architecture):

- Before the training starts, the weights take random values.

- An object $\boldsymbol{X}_s$ (vector input) of dimension m enters the network.

- The output of all neurons, each one with m weights, are calculated.

- The object $\boldsymbol{X}_s$ is mapped into the neuron with the most similar weights compared to its features. This is the central neuron, or winning neuron, c. It is said that the winning neuron was excited (or activated) by the object.

- The weights of neuron c are adjusted to make them even more similar to the properties of the presented object.

- The neurons in its neighborhood (arbitrarily defined) also have its weights adjusted. The extent of adjustment depends, however, on the topological distance to the winning neuron - the closer a neuron is to the winning neuron the larger is the adjustment of its weights.

- The next object $\boldsymbol{X}_s$ of the training set enter the network and the process is repeated.

- The objects of the training set are iteratively fed to the map, and the weights corrected, until a pre-defined number of cycles is attained.

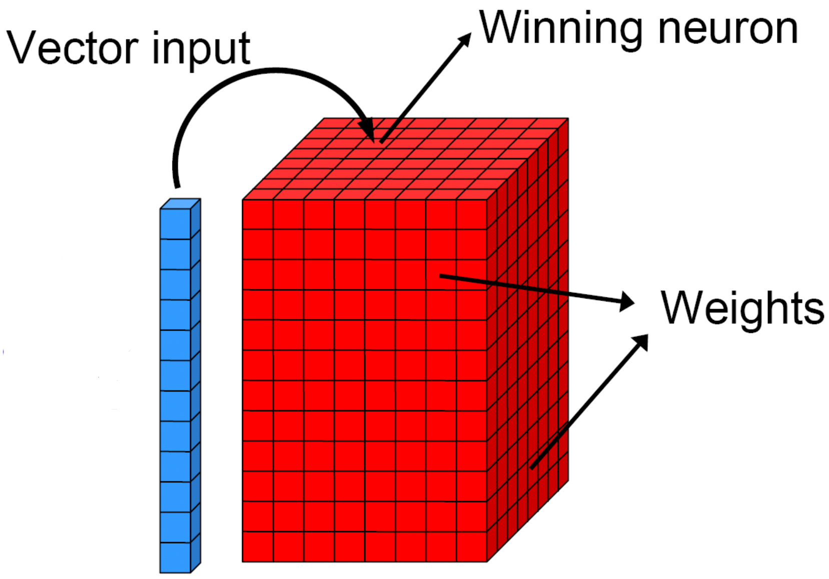


Figure 2.10: Illustration of Kohonen SOM training and structure.

2.7 Counter-Propagation Neural Networks

2.7.1 Introduction

A Counter-Propagation Neural Network (CPNN) consists of a Kohonen SOM linked to an output layer of neurons aligned with the Kohonen layer. A Kohonen self-organizing map (SOM) distributes objects over a 2D surface (a grid of neurons) in such a way that objects bearing similar features are mapped onto the same or adjacent neurons. They perform a non-linear projection of multidimensional objects onto a two-dimensional surface yielding maps of easy visual interpretation. The input data are stored in the two dimensional grid of neurons, each containing as many elements (weights) as there are input variables. In a CPNN the input layer is linked to an output layer. The output data (in this Thesis the classification of a reaction) are stored in the output layer that acts as a look-up table.

Before the training of a CPNN starts, random weights are generated. During the training, each individual object (a reaction) is mapped into that neuron of the input layer (central neuron or winning neuron) that contains the most similar weights compared to the input data (molecular descriptors). The weights of the winning neuron are then adjusted to make them even more similar to the presented data, and the weights of the corresponding output neuron are adjusted to become closer to the known output of the presented object. The neurons in the neighborhood of the winning neuron are also corrected, the extent of adjustment depending on the topological distance to the central neuron. The network is trained iteratively, i.e., all the objects of the training set are presented several times, and the weights are corrected, until the network stabilizes. Note that the output values are not used in determining the winning neuron, therefore the training of a CPNN can be described as semi-supervised.

After the training, the CPNN is able to make predictions for an object. The winning neuron is chosen and the corresponding weights in the output layer are used as the output.

2.7.2 Concept of Lookup Table in CPNNs

The output layer of a CPNN model can be treated as a lookup table, which is an area of the memory where the answers to some complex question are ordered in such way that is easy to find the "box" with the correct answer. The CPNN calculates the *address* of the correct answer (instead of a value) from the input data. CPNNs are useful, for example, when the output to be retrieved is only a weak function of the input, or when input data is "corrupted" or fuzzy .

It is to emphasize that the concept of lookup table is distinct from the concept of model, where the objective is to set up a procedure that gives a different answer for each different set of variables representing an object. In fact a lookup table and a model differ in the applications and in the data to which they can be applied. In a model, the number of experiments needs to be larger than the number of parameters that describe the model, while a lookup table usually requires at least one experiment (object) for each possible or expected answer (output). A model can give an infinite number of different answers while the number of the lookup table answers depends on the number of neurons (number of "answer boxes").

The best application of CPNN is not in modeling problems but in the generation of lookup tables where all the answers are known a priory.

2.7.3 Architecture

The CPNN has an architecture with two active layers: a Kohonen layer and an output layer. In this architecture the inputs are fully connected to the Kohonen network where competitive learning occurs.

The weights connecting the input unit i with the Kohonen neuron j are labeled as w_{ji} and each neuron in the Kohonen layer is described by the weight vector $\boldsymbol{W}_j$. Usually all neurons of the Kohonen layer are connected to the output layer but, in practice, although this connection is virtually made, after each input only a certain neighborhood of a given neuron is connected to the output neurons and only the weights of these neurons are corrected. Figure 2.11 illustrates this situation. The weights that connect the $j-$th neuron of the Kohonen layer to the $k-$th neuron of the output layer are represented as r_{kj} and the weight vector of a given output neuron as $\boldsymbol{R}_k$.

In this type of architecture the set of answers (output) is stored as the weights of the output neurons that receive signals from the Kohonen layer. The number of neurons in the Kohonen layer is thus equal to the number of answers stored, and the number of

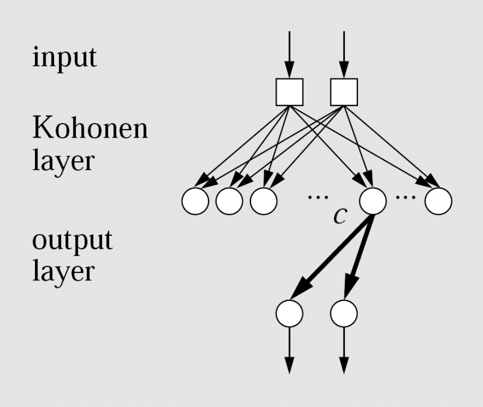


Figure 2.11: The answer to a given input is stored as a weight vector connecting the winning neuron in the Kohonen layer with all output neurons in the output layer. (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999)

weights in the output layer is equal to the number of variables that comprise the output answer. The input layer has the same number of weights as there are input variables.

2.7.4 Semi-supervised Competitive Learning

A supervised learning needs a set of pairs ($\boldsymbol{X}_s$, $\boldsymbol{Y}_s$) as input. The input of the network, in some cycle, is the vector $\boldsymbol{X}_s$ and the target (correct answer) is the vector $\boldsymbol{Y}_s$. The main objective in any supervised learning technique is to establish some system that after the learning procedure gives the correct answer $\boldsymbol{Y}_s$ for the corresponding input $\boldsymbol{X}_s$. After the training, it is expected that the network is able to make predictions for a new object $\boldsymbol{X}$.

A CPNN can give several types of predictions. It can be used for classification of multidimensional objects $\boldsymbol{X}$ in categories, content-dependent retrievals where incomplete or fuzzy data is used and the originals are recovered, and for modeling of complex multivariate non-linear functions (yielding a 1- or 2-D answer).

One disadvantage of this method is that all possible answers need to be covered (large quantities of data are required for learning complex problems). Another problem is that the size of the network is a limitation for the number of different answers that the network can produce. If there are too many answers a small network has not enough resolution.

As mentioned above the first active layer in a CPNN is a Kohonen layer. The selection of the winning or central neuron is done exactly in the same way as in Kohonen SOM. After the input of an object $\boldsymbol{X}$ of dimension m the selection of the winning neuron can be done by choosing the neuron with the largest output, out_c:

$$out_c \leftarrow max\,(out_j) = max\left(\sum_{i=1}^{m} w_{ji}x_{si}\right) \quad j = 1,\, 2,\, ...,\, n \tag{2.33}$$

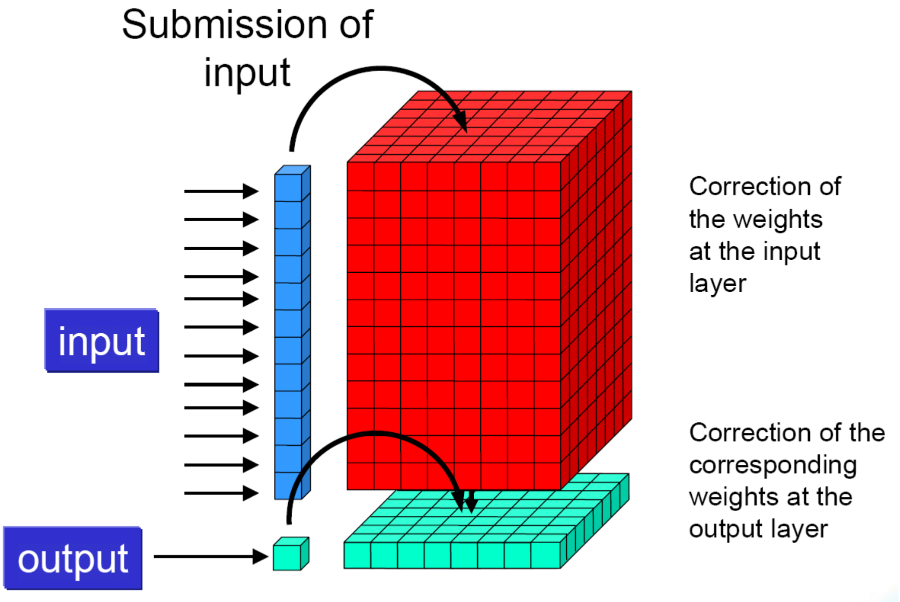

Submission of input
input
output
Correction of the weights at the input layer
Correction of the corresponding weights at the output layer

Figure 2.12: Illustration of CPNN training and architecture.

or by choosing the neuron j with the weight vector $\boldsymbol{W}_j\,(w_{j1},\, w_{j2},\, ...,\, w_{jm})$ most similar to the input signal $\boldsymbol{X}_s\,(x_{s1},\, x_{s2},\, ...,\, x_{sm})$:

$$out_c \leftarrow min\left\{\sum_{i=1}^{m}(x_{si}-w_{ji})^2\right\} \quad j=1,\,2,\,...,\,n \tag{2.34}$$

After the selection of the winning neuron two types of corrections on the weights are made:

- the correction of the weights w_{ji} of the neurons in the Kohonen layer
- the correction of the weights of the output layer

The correction of the weights in the Kohonen layer is made using the Equation 2.30:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \eta\,(t)\,a\,(d_c - d_j)\left(x_i - w_{ji}^{(old)}\right)$$

The neighborhood-dependent function $a\,(d_c - d_j)$ and the monotonically decreasing function $\eta\,(t)$ have been discussed in Section 2.6.

The second type of correction is made in the weights of the output layer. Figure 2.7.3 illustrates the organization of the layers.

The weights are corrected using the expression:

$$c_{ji}^{(new)} = c_{ji}^{(old)} + \eta\,(t)\,a\,(d_c - d_j)\left(y_i - c_{ji}^{(old)}\right) \tag{2.35}$$

where c is the index of the winning Kohonen neuron, j is the index of the neighboring neuron being corrected, i runs over all the weights connecting the Kohonen neuron j and the output neurons i. Each one of the n output neurons represents one component of the target vector $\boldsymbol{Y}= (y_1, y_2, ..., y_n)$.

The corrected weights do not need to be normalized because they are the output from the CPNN.

The CPNNs have three important properties:

- are able to deal with vectors (multidimensional objects) representing real values in the input and in the output (target) side
- a lookup table can be generated from a data set and can be used as a substitute of a mathematical model (explicit functional relationship)
- the correlation between the different variables can be obtained from the weights of the output layer.

The learning procedure can be summarized in the following steps:

- preparation of a data set of objects $(\boldsymbol{X}_s,\ \boldsymbol{Y}_s)$
- the weights are initialized with random values
- input the first object $\boldsymbol{X}_s$
- evaluate n sums in all neurons in the Kohonen layer:

$$out_j = \left\{ \sum_{i=1}^{m} (x_{si} - w_{ji})^2 \right\} \quad j = 1,\ 2,\ ...,\ n$$

- select the winning neuron c - the neuron with the minimum out_j:

$$out_c = min\left\{out_1,\ out_2,\ ...,\ out_n\right\} \quad j = 1,\ 2,\ ...,\ n$$

- correction of the weights of the winning neuron and of a given neighborhood around the winning neuron $c$in the Kohonen layer:

$$w_{ji}^{(new)} = w_{ji}^{(old)} + \eta\,(t)\,a\,(d_c - d_j)\left(x_i - w_{ji}^{(old)}\right)$$

- at the beginning of the training the product $\eta\,(t)\,a\,(d_c - d_j)$ is ~0.5 ($a_{max} = 0.5$ and $a_{min} = 0.01$) for the neurons in the neighborhood this values decreases as a function of the neighbor-ring and with the number of iteration training cycles performed.
- correction of the weights of the output layer:

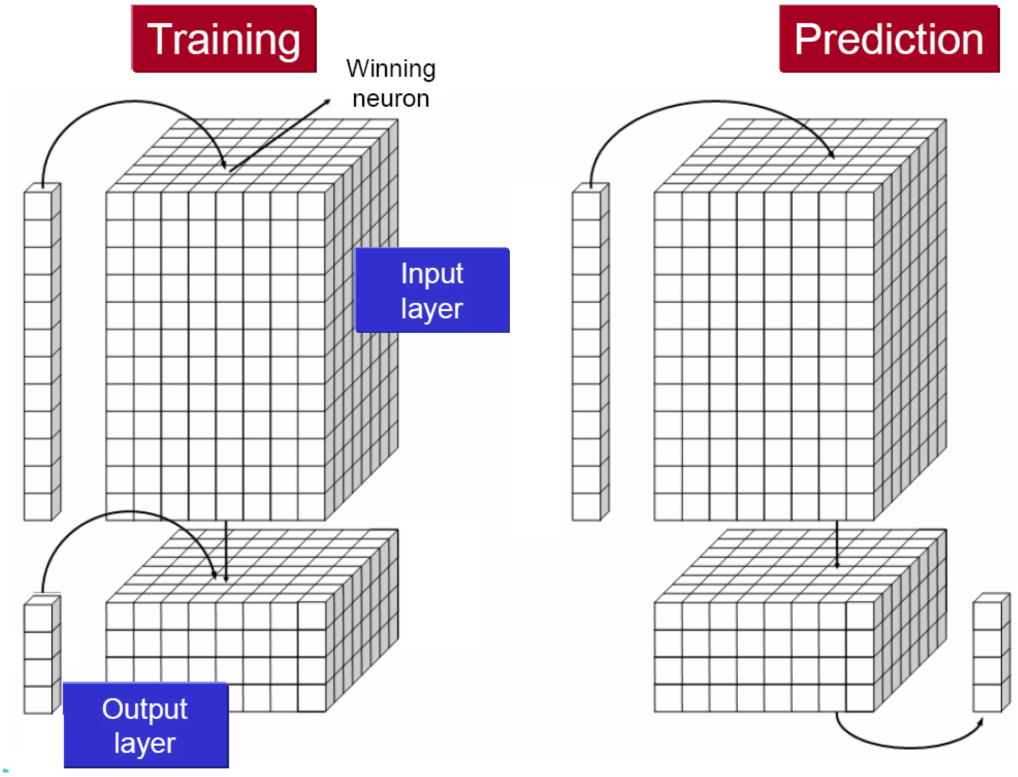


Figure 2.13: Illustration of the training procedure in a CPNN and the prediction for a new object.

$$c_{ji}^{(new)} = c_{ji}^{(old)} + \eta\left(t\right) a \left(d_c - d_j\right) \left(y_i - c_{ji}^{(old)}\right)$$

- repeat the procedure until all objects have been sent through the network in a predefined number of training cycles

During the training the results are evaluated using the root-mean-square (RMS) error between the targets and the output of the CPNN. The RMS error is calculated with the equation:

$$RMS = \sqrt{\frac{\sum_{s=1}^{n_i} \sum_{i=1}^{n} \left(y_{si} - out_{si}\right)^2}{n_i n}} \tag{2.36}$$

where $y_{si}$is the $i-$th component of the target $\boldsymbol{Y}_s$, out_{si} is the $i-$th component of the output for the $s-$th input vector, n_i is the number of inputs and n is the number of output variables. After the training is complete the CPNN is prepared to make predictions for new objects (Figure 2.13).

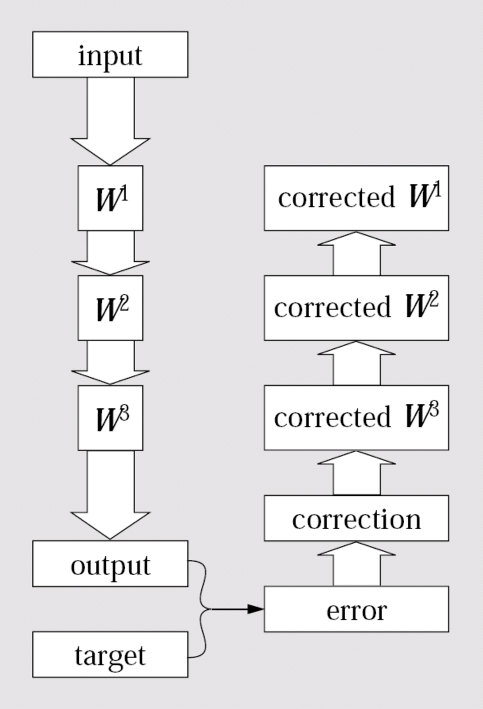


Figure 2.14: Illustration of weight correction in FFNNs (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

2.8 Feed-Forward Neural Networks and Ensembles of Feed-Forward Neural Networks

2.8.1 Introduction

This Section presents the Feed-Forward Neural Networks (FFNN) also known as Back-Propagation Neural Networks (BPNN). This is a multilayer network that learns in a supervised way. The FFNNs are the most frequently used NN model, namely in chemical applications.

This preference is a probably consequence of the well-defined, easy and explicitly set of equations for weight corrections used by this NN model, and its ability to address different types of problems, from classification to modeling. The correction of the weights starts in the last layer (output layer) and continues backwards until the first layer. (Figure 2.14)

The FFNN involves a supervised learning method, so training requires a set of pairs of objects, the inputs $\boldsymbol{X}_s$ and the targets $\boldsymbol{Y}_s$, $(\boldsymbol{X}_s,\ \boldsymbol{Y}_s)$. The trained network can be viewed as a model that was able to give a $m-$variate answer for each $m-$variable input (see Figure 2.15).

Three things have to be emphasized when FFNNs are compared with other supervised learning methods:

- all pre-processment of data before training used in the standard modeling techniques,

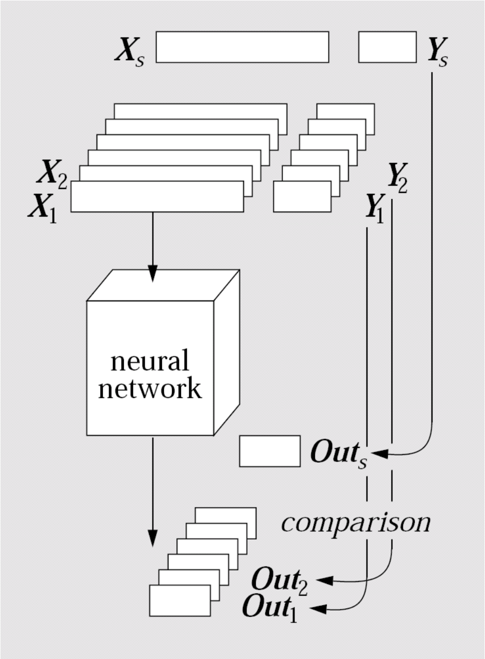


Figure 2.15: Supervised learning technique. (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999)

such as choice of descriptors, representation of objects, experimental design, etc, are important for the success of the FFNN learning procedure.

- differently of other modeling methods, in FFNNs there is no need to know *a priory* the form of the analytical function on which the model should be built - neither the functional type (polynomial, exponential, logarithmic, etc) nor the number and position of the parameters in the model function need to be given.
- it is difficult to obtain information from the weights after the training. In the most part of the cases a large number of weights are used to get the correct answers and so it is very hard to know what each weight is responsible for. A FFNN is often viewed as a "black-box" - it is difficult to get a physical interpretation and to establish a relation between its internal parameters and the answers.

2.8.2 Architecture

The architecture of the network influences the flexibility to learn a problem. The architecture is defined by the number of layers, number of neurons in each layer an by the way the neurons are connected.

The number of layers and neurons in each layer depends on the problem that the network is designed to solve and are determined by trial and error. Most frequently, and in the FFNNs used in this Thesis, the network has two fully connected active layers - one hidden layer and one output layer.

During the learning procedure a considerable number of inter layer calculations are performed. Each specific data involved in the training, input, output, weights, errors, corrections, will be indexed with a superscript that refers to the layer that each data belongs.

As it is said before, the output of one layer is the input of the layer below, and the input of one layer is the output of the layer above. In the notation followed in the learning procedure of FFNNs all signals will be labeled as outputs. The input signal that enters the network will be labeled as $\boldsymbol{Out}^0$, after be processed by the first layer it produces the $\boldsymbol{Out}^1$, then $\boldsymbol{Out}^2$, the output of the second layer, until the last layer where the the final output is obtained and labeled as $\boldsymbol{Out}^{last}$.

2.8.3 Learning Algorithm - Back-Propagation

The Back-propagation (BPG) algorithm is a supervised learning algorithm, which means that the weights are corrected for the network to produce the target values when the associated input values are submitted.

The correction of the weights is applied after each object of a training set is entered and processed by the network producing an output.

Each object $\boldsymbol{X}$ (input vector) of the training set is processed by the network and the output vector $\boldsymbol{Out}$ is compared with the target vector $\boldsymbol{Y}$. (The network is trained in order that $\boldsymbol{Out}$ values approximate $\boldsymbol{Y}$ values). The calculated error is used for the correction of the weights through all layers. The final equation for the correction of the weights is:

$$\Delta w_{ji}^{l} = \eta \delta_j^l out_i^{l-1} + \mu \Delta w_{ji}^{l(previous)} \tag{2.37}$$

where l is the index of the layer, j identifies the current neuron, i is the index of the neuron in the upper layer. δ_j^l is the error introduced by the corresponding neuron and is calculated in a different way depending on whether the neuron belongs to the last layer or to a hidden layer. The correction of the weights includes two terms. The first corrects the weights in a fast "steepest-descent" convergence and the second is a longer range function that prevents the solution from being trapped in a local minima. The constant η is the learning rate and μ is the momentum constant. The latter prevents sudden changes in the direction of the corrections, taking into account the correction of the previous cycle. The values of these two constants determine the influence of the two terms in the weight correction.

For the output layer ($l = last$) the error δ_j^l is calculated in the following way:

$$\delta_j^{last} = \left(y_j - out_j^{last}\right) out_j^{last} \left(1 - out_j^{last}\right) \tag{2.38}$$

for all other layers l ($l = last - 1\, to\, 1$) the error δ_j^l is calculated by:

$$\delta_j^l = \left(\sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1} \right) out_j^l \left(1 - out_j^l\right) \tag{2.39}$$

By substitution of Equation 2.39 in Equation 2.37 the full expression for weight correction in the hidden layer is obtained:

$$\Delta w_{ji}^l = \eta \left(\sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1} \right) out_j^l \left(1 - out_j^l\right) out_i^{l-1} + \mu \Delta w_{ji}^{l(previous)} \tag{2.40}$$

This equation shows that three layers of neurons influence the correction of a weight, the values from the current layer, l, the layer above $(l - 1)$ and the layer below $(l + 1)$. These equations assume the use of the sigmoidal function as the transfer function.

In FFNNs the data is processed in one direction and the correction of the weights is made in the opposite direction. The correction of the of the $i-$th weight on the $j-$th neuron in the $l-$th layer of neurons is:

$$\Delta w_{ji}^l = w_{ji}^{l(new)} - w_{ji}^{l(old)} \tag{2.41}$$

the weight w_{ji} links the $i-$th input with the $j-$th output signal (neuron j). These links between the two consecutive layers show the fact that the error is originated on the input and on the output side. The delta-rule is a good way to account for the two influences in the error and can be expressed as:

$$\Delta parameter = \eta g\left(output\, error\right) f\left(input\right) \tag{2.42}$$

The delta-rule states that the change of a parameter in an adapting process should be proportional to the input signal and to the error on the output side. The constant η, learning rate, determines the rate of the changes of the parameter during the iteration cycles. Equation 2.42 can be written in another form by substitution of the terms by the ones of the neural network approach:

$$\Delta w_{ji}^l = \eta \delta_j^l out_i^{l-1} \tag{2.43}$$

The two equations are identical. The parameter causing the error is the weight, w_{ji}, and the weight correction Δw_{ji} is proportional to the term δ_j^l that corresponds to the function g. The output out_i^{l-1} is the output of the $(l - 1)$ layer and is, at the same time, the input of the $l-$layer. The function f is just the input itself. The only parameter to determine is the function δ_j^l.

The value of the change δ_j^l is determined using the gradient descent method that enables the search of the value for a parameter that corresponds to the minimum of the

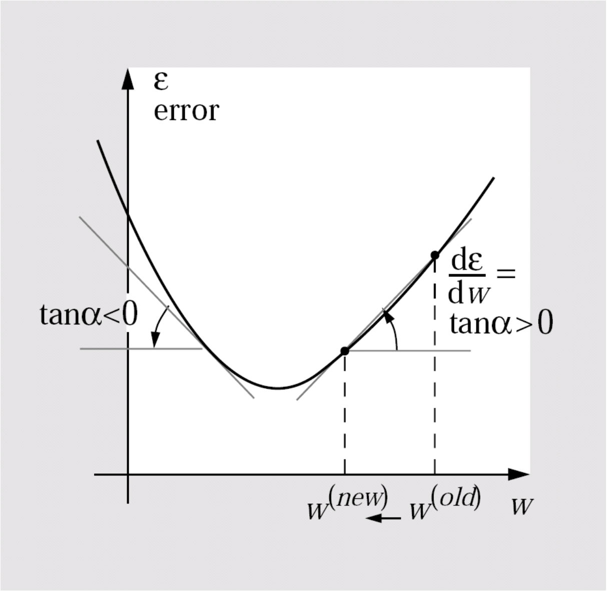


Figure 2.16: Error as a function of the weight (adapted from J. Zupan and J. Gasteiger, *Neural Networks in Chemistry and Drug Design*, Wiley-VCH, Weinheim, 1999).

error ε. In the plot of the error against the parameter, the value of the slope of the curve, at some point, indicates how to change the value of the parameter to make the error closer to the minimum. Figure 2.16 illustrates the correction in the weight to approach the minimum.

If the value of the weight is at the right of the minimum the derivative $d\varepsilon/dw$ is positive and so the new value of the parameter should be smaller than the old one and vice-verse:

$$\Delta w = w^{(new)} - w^{(old)} = -\frac{\kappa d\varepsilon}{dw} \tag{2.44}$$

where κ is a positive scaling factor. For a specific weight w^l_{ji} in the layer l the equation is transformed in:

$$\Delta w^l_{ji} = -\frac{\kappa d\varepsilon^l}{dw^l_{ji}} \tag{2.45}$$

As the error function is an indirect function of the parameters w^l_{ji}, it is possible to evaluate the derivative $\partial\varepsilon^l/\partial w^l_{ji}$ by application of the chain rule of derivatives:

$$\Delta w^l_{ji} = -\kappa\frac{\partial\varepsilon^l}{\partial w^l_{ji}} = -\kappa\left(\frac{\partial\varepsilon^l}{\partial out^l_j}\right)\left(\frac{\partial out^l_j}{\partial Net^l_j}\right)\left(\frac{\partial Net^l_j}{\partial w^l_{ji}}\right) \tag{2.46}$$

The derivatives of the error function ε^l are consecutively calculated with respect to out^l_j, Net^l_j and w^l_{ji}. All the derivatives are fully described in Ref. [1] or [56] for example. Here it will be only presented the final equations for the correction of the weights.

The three partial derivatives are:

$$\frac{\partial Net_j^l}{\partial w_{ji}^l} = out_i^{l-1} \tag{2.47}$$

$$\frac{\partial out_j^l}{\partial Net_j^l} = out_j^l \left(1 - out_j^l\right) \tag{2.48}$$

$$\frac{\partial \varepsilon^l}{\partial out_j^l} = \sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1} \tag{2.49}$$

by inserting these three equations in Equation 2.46 for the delta-rule:

$$\Delta w_{ji}^l = \eta \left(\sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1}\right) out_j^l \left(1 - out_j^l\right) out_i^{l-1} \tag{2.50}$$

Equation 2.50 clearly shows how the values of three different layers are involved in the calculation of the weight correction in the hidden layer l:

- the output out_i^{l-1} of the layer above - the input i of the $l-$th layer
- the out_j^l of the $j-$th neuron on the current layer l
- the correction on the weight w_{kj}^{l+1} of the $l+1$ layer

The learning algorithm in FFNNs can be summarized in the following steps:

- input an object $\boldsymbol{X}(x_1,\, x_2,\, ...,\, x_m)$
- the component x_i of the input object $\boldsymbol{X}$ is labeled as out_i^0 and a bias is added, so the input vector $\boldsymbol{X}$ becomes $\mathrm{Out}^0\left(out_1^0,\, out_2^0,\, ...,\, out_m^0,\, 1\right)$
- propagate Out^0 through the layers of the network with evaluation of the output vectors $\boldsymbol{Out}^l$. It is used the weights w_{ji}^l of the $l-$th layer and the output out_i^{l+1} from the previous layer (the input of layer l):

$$out_j^l = f\left(\sum_{i=1}^{m} w_{ji}^l out_i^{l-1}\right)$$

where f is the chosen transfer function, in this case the sigmoidal function.

- calculate the correction factor for all weights of the output layer δ_j^{last}, by using its output vector $\boldsymbol{Out}^{last}$ and the target vector $\boldsymbol{Y}$:

$$\delta_j^{last} = \left(y_j - out_j^{last}\right) out_j^{last} \left(1 - out_j^{last}\right)$$

- correct all weights w_{ji}^{last} on the last layer

$$\Delta w_{ji}^{last} = \eta \delta_j^{last} out_i^{last-1} + \mu \Delta w_{ji}^{last(previous)}$$

- calculate the correction factors δ_j^l layer by layer from $l = last - 1$ to $l = 1$:

$$\delta_j^l = \left(\sum_{k=1}^{r} \delta_k^{l+1} w_{kj}^{l+1} \right) out_j^l \left(1 - out_j^l\right)$$

- correct all weights w_{ji}^l on the layer l:

$$\Delta w_{ji}^l = \eta \delta_j^l out_i^{l-1} + \mu \Delta w_{ji}^{l(previous)}$$

- repeat the entire procedure with a new input-target object $(\boldsymbol{X},\ \boldsymbol{Y})$

Before the beginning of the training a few issues must be considered such as:

- choice of the initial neural network architecture
- initialization of the weights (randomly)
- selection of the learning rate η and momentum constant μ

The initial architecture of the network (number of layers, number of neurons) is usually only a starting point of the training procedure and is modified if it is needed after the evaluation of the results for training and test sets. The dimension of the input and output layer is predefined - the number of input units is equal to the number of input variables (number of descriptors of each object) and the number of output neurons in the output layer is equal to the number of outputs. The number of neurons in the hidden layer, or layers has to be defined by trial and error, and adjusted to each specific problem.

The weights are initialized as small random numbers. The value of learning rate constant is also an important parameter in the learning procedure. This parameter determines the speed of the weight changes and if this process occurs too fast a local minimum of the error can be reached instead of the global minimum. Another important parameter is the momentum constant which prevents sudden changes in the direction of the weight correction.

The optimum values of the two last parameters are usually found by trial and error, and are sometimes decreased during the learning process. Some authors say that the sum of them must be approximately one.

In FFNNs the processment of all objects by the network is called one iteration cycle or one epoch. In general, thousands of cycles are needed until convergence of the value of the error to a constant value for most complex problems.

After the training, the network must be able to recognize the training set and more importantly make accurate predictions for new data that did not participate in the training. A good selection of the training set is crucial for a successful training. The training set must be representative, covering as well as possible the space of the problem.

The output data must be normalized into values between 0.1 and 0.9 or even 0.2 and 0.8 as the sigmoidal function is the transfer function used, and this varies in the interval]0,1[.

After the training, a FFNN must be validated with an independent set of objects, and if predictions are not satisfactory, several approaches may be followed to improve the results:

- increase the number of training cycles
- change the network design
- get more data to the training set

The first issue deserves much attention. Too many training cycles can give rise to overtraining - the network is adapted too much to the training set and looses flexibility to make accurate predictions for new data outside the training set. The training must be stopped when the error for a monitoring test set reaches a minimum. Also networks with too many neurons can give origin to overfitting, in this case the number of parameters in the network is larger than needed to learn the problem. The number of weights in a network must be more or less the same or smaller than the number of objects in the training set. It is essential the monitoring of the RMSE of the training and test set to ensure that the network will not become overtrained. Obviously, the objects used in the test set for monitoring the training cannot be included in the final validation set.

2.8.4 The Levenberg-Marquardt Algorithm

2.8.4.1 Introduction

A second more efficient algorithm for the training of FFNNs is the Levenberg-Marquardt (LM) algorithm. For this Thesis it was used in the experiments with the ASNN software to implement ensembles of Feed-Forward Neural Networks and Associative Neural Networks.

The Levenberg-Marquardt (LM) algorithm is an iterative technique that can locate the minimum of a multivariate function. It can be thought of as a combination of steepest descent and the Gauss-Newton method. When the current solution is still far from the correct one, the algorithm behaves like a steepest descent method - with a slow progress but guaranteed to converge. When the current solution is close to the correct solution, it becomes a Gauss-Newton method.

The LM algorithm has become a standard technique for non-linear least-squares problems that has been widely used in a broad spectrum of disciplines, namely. to correct the weights of FFNNs in such way that the output of the network is as close as possible to the target.

A short description of the LM algorithm is presented. Detailed description of the LM algorithm is beyond the scope of this Thesis and the interested reader is referred to [57–59] for full description of the algorithm.

2.8.4.2 The Levenberg-Marquardt Algorithm in Neural Networks

The use of the Levenberg-Marquardt algorithm in FFNNs involves a training phase where the weights are adjusted like with the back-propagation of errors algorithm. The gradient-based algorithms (e.g. the back propagation of error algorithm described in the previous Subsection) are not totally efficient due to the fact that the gradient vanishes at the solution. Differently, the Hessian-based algorithms allow the NN to learn more subtle features of a complicated mapping. The training process has the advantage of quickly converging as the solution is approached, because the Hessian does not vanish at the solution.

To benefit from the advantages of Hessian based training of NNs the Levenberg-Marquardt Algorithm [60–62] was used. The LM algorithm is basically a Hessian-based algorithm for nonlinear least squares optimization. For neural network training the objective function is the error function of the type:

$$e(w) = \frac{1}{2} \sum_{k=0}^{p-1} \sum_{l=0}^{n_0-1} (y_{kl} - out_{kl})^2 \tag{2.51}$$

where out_{kl} is the actual output at the output neuron l for the input k and y_{kl} is the desired output at the output neuron l for the input k. The index p is the total number of training patterns and n_0 represents the total number of neurons in the output layer of the network. The letter w represents the weights and biases of the NN.

The steps involved in the training of a FFNN using the Levenberg-Marquardt algorithm are as follows:

1. Present all inputs to the FFNN. Calculate the corresponding outputs and errors and the mean square error over all inputs as in equation 2.51.

2. Determine the Jacobian matrix, $J(w)$ where w represents the weights and biases of the network.

3. Solve the Levenberg-Marquardt weight update equation to obtain Δw. (Equation 2.52)

4. Recalculate the error using $w+\Delta w$. If this new error is smaller than that computed in step 1, then reduce the *training parameter* μ by μ^-, let $w = w+\Delta w$, and go back to step 1. If the error is not reduced, then increase μ by μ^+ and go back to step 3. The parameters μ^+ and $\mu^-$are, usually, predefined values set by the user. Typically μ^+ is set to 10 and μ^- is set to 0.1.

5. The algorithm is assumed to have converged when the norm of the gradient is less than some predetermined value, or when the error has been reduced to some error goal.

In the above algorithm, the weight update vector Δw is calculated as:

$$\Delta w = \left[J^T(w)J(w) + \mu I\right]^{-1} J^T(w)R \tag{2.52}$$

where R is a vector of size pn_0 calculated as follows:

$$R = \begin{pmatrix} y_{11} - out_{11} \\ y_{12} - out_{12} \\ \dots \\ \dots \\ y_{21} - out_{21} \\ y_{22} - out_{22} \\ \dots \\ \dots \\ y_{pn_0} - out_{pn_0} \end{pmatrix} \tag{2.53}$$

$J^T(w)J(w)$ is referred to as the Hessian matrix. Let the dimension of the input space be n_i. Suppose there are n_0 classes that the input data is to be classified into. Also let the total number of patterns be p. If the popular Multi-Layer Perceptron (MLP) is used to do the classification it will necessarily comprise an n_i dimensional input layer and an n_0 dimensional output layer. If n_h is the number of hidden neurons, then the total number of weights and biases in the MLP will be

$$numw = n_i n_h + n_h n_0 + n_h + n_0 \tag{2.54}$$

with the above notation the dimension of the Jacobian will be $pn_0 \times numw$ while that of the Hessian will be $numw \times numw$ where $numw$ is the total number of weights and biases and is calculated as in equation 2.54.

2.8.5 Ensembles of Feed-Forward Neural Networks

An EnsFFNN is made of several independently trained FFNNs that contribute to a single prediction [63, 64]. Considering an ensemble of m FFNNs, EnsFFNNs:

$$enFFNNs = (FFNN_1,\ FFNN_2,\ ...,\ FFNN_m) \tag{2.55}$$

The prediction of an object $\boldsymbol{X}_i(x_1, x_2, ..., x_m)$ is represented by the vector of the output values $\boldsymbol{Out}_i = \left\{out_j^i\right\}^M$, this is the output vector of the object where $i = 1, 2, ..., M$ is the index of the network in the ensemble. The final prediction for an object is the average of the outputs from all FFNNs of the ensemble:

$$\overline{out_i} = \frac{1}{M}\sum_{i=1}^{M} out_j^i \tag{2.56}$$

This methodology smoothes random fluctuations in the individual predictions of individual FFNNs.

2.9 Associative Neural Networks

2.9.1 Introduction

The ASNNs is in fact a combination of FFNNs (an ensemble of FFNNs) with the K-Nearest Neighbor technique that often allows more accurate predictions for non-linear problems. An Associative Neural Network (ASNN) [65,66] is a combination of a memory-less (ensemble of Feed-Forward Neural Networks) and a memory-based method (K-Nearest Neighbor [67] technique). A memory-less method means that after training there is no explicit storage of the data in the system. The information about the data is stored in the neural networks weights. On the contrary in memory-based methods the data used to train the models are stored in a “memory” and used to make predictions for new data based on some local approximations of the stored examples. While neural networks are considered global methods the memory-based methods are considered local methods.

The EnsFFNNs is combined with a memory into a so-called Associative Neural Network (ASNN) [65,66]. The memory consists of a list of objects, represented by their input variables, and the corresponding targets. The ASNN scheme is employed for composing a prediction of the object from (a) the outputs from the EnsFFNNs and (b) the data in the memory. This approximation is important for the resolution of certain problems. For example, in non-linear problems a neural network can provide a good approximation of the global input data space but it may be difficult to reproduce the most subtle features of all regions of the space.

2.9.2 Learning Procedure

The learning procedure described in this Subsection only details the steps after the training of an EnsFFNNs.

When a query point is submitted to an ASNN, the following procedure takes place to obtain a final prediction of the object:

1. The descriptors of the object are presented to the ensemble, and a number of output values are obtained from the different FFNNs of the ensemble - the output profile of the query object.

2. The average of the values in the output profile is calculated. This is the uncorrected prediction for the query object.

3. Every target of the memory is presented to the ensemble to obtain an output profile.

4. The memory is searched to find the k-nearest neighbors of the query object. The search is performed in the output space, i.e. the nearest neighbors are the objects with the most similar output profiles (calculated in Step 3) to the query object (calculated in Step 1). Similarity is here defined as the Spearman correlation coefficient between output profiles.

5. For each of the KNN object, an (uncorrected) prediction is also obtained - the average of its output profile.

6. The uncorrected predictions for the KNN objects (calculated in Step 5) are compared with their target values in the memory. The mean error is computed.

7. The mean error computed in Step 6 is added to the uncorrected prediction of the query object (computed in Step 2) to yield the corrected prediction for the query point:

$$\overline{out'_i} = \overline{out_i} + \frac{1}{k} \sum_{j \in N_k(X)} \left(y_j - \overline{out_j} \right) \tag{2.57}$$

where $\overline{out_i}$ is the prediction of the ensemble, y_j is the target value. The summation is over the k-nearest neighbors determined by the Pearson's linear correlation coefficient. The difference $\left(y_i - \overline{out_i}\right)$ corresponds to the systematic error of the ensemble for the object i. The number of nearest neighbors, k is set empirically to provide the lowest possible leave-one-out (LOO) error.

The NNs were trained using the Early Stopping over Ensemble (ESE) method. The initial training set is randomly partitioned in equal parts in learning and validation sets for each FFNN in the ensemble giving to each one a different learning and validation set. The training is stopped when it is reached the minimum error for the validation set.

Chapter 3

Decision Trees and Random Forests

An important part of the work related to the classification of enzymatic reactions, in this Thesis, was performed with Random Forests (RFs). This Chapter presents a description of decision trees and RF methods. Similarly to EnFFNNs, a Random Forest is an ensemble of models - an ensemble of unpruned decision trees.

3.1 Decision Trees

A decision tree is a supervised automatic learning method that can be used for the purpose of classification or regression. It derives rules from a training set of objects represented by a number of descriptors. The derived rules are easy to interpret and can be visualized as a tree-like graph. Subsection 3.1.1 is an introduction to the method. Subsection 3.1.2 describes the learning procedure of a decision tree showing how the classification of an object is obtained based on classification rules.

3.1.1 Introduction

A decision tree is a supervised learning method. A decision tree is built based on a training set and, after the training, it can be applied to new objects for their prediction.

A decision tree consists of a root, nodes, branches descending from nodes, and leaves, in a similar way to a real tree. In a real tree the nutrients are taken from the soil to the root and carried to the branches and leaves. In the classification process of a decision tree an object is sorted down the tree from the root to some leaf node where the object is classified. The decision tree grows by partition of the objects from a parent node into two child nodes, or branches. Each node in the tree corresponds to a logical rule defined for some variable of the objects, and each branch descending from that node corresponds to one range of the possible values for the variable. Figure 3.1 presents a graphical representation of a decision tree for prediction of mutagenicity from the molecular structure [68]. An object to be classified by a decision tree is submitted to the root node of the tree, then the specific

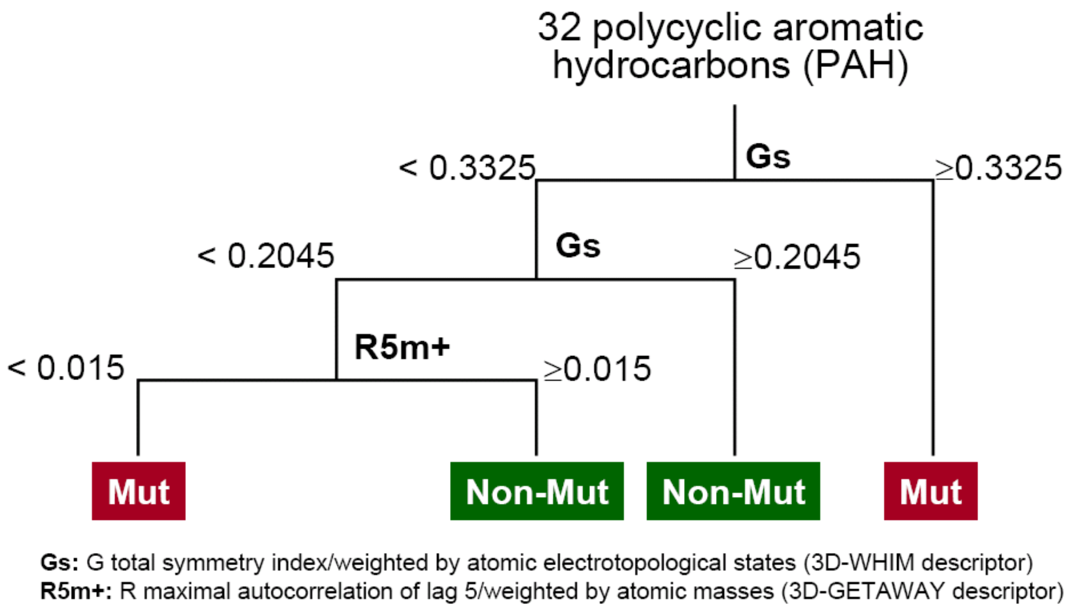

32 polycyclic aromatic hydrocarbons (PAH)
Gs
< 0.3325
≥0.3325
Gs
< 0.2045
≥0.2045
R5m+
< 0.015
≥0.015
Mut
Non-Mut
Non-Mut
Mut
Gs: G total symmetry index/weighted by atomic electrotopological states (3D-WHIM descriptor)
R5m+: R maximal autocorrelation of lag 5/weighted by atomic masses (3D-GETAWAY descriptor)

Figure 3.1: Example of a decision tree for prediction of mutagenicity. (adapted from P. Gramatica, E. Papa, A. Marrocchi, L. Minuti and A. Taticchi, *Ecotoxicol. Environ. Saf.* 2007, *66*, 353-361)

variable for this node is tested and the object is moved down to a branch according to the value of the variable. This process is repeated for the subtree rooted at the new node until the object reaches the final node, the leaf where it is classified. In the example of Figure 3.1 a tree was built for the classification of 32 polycyclic aromatic hydrocarbons as "mutagenic" or "non mutagenic" using only 2 descriptors from a total set of 381 molecular descriptors. The *Gs* descriptor is used twice with a different range of values.

Although in this work we are more interested in classification problems, decision trees can also be used for regression tasks. In that case, the predicted output for a given object is obtained as the average of the responses for the objects of the training set that are classified in the same terminal leave.

Differently from NNs, a decision tree is a rather transparent method - it is based on explicit classification rules that can be verbally expressed and graphically represented, and each rule can be associated with a measure of efficiency, calculated for the objects of the training set.

3.1.2 Decision Trees Structure and Learning Procedure

The main goal of a decision tree is to set up classification rules that allow the assignment of an object to a certain class based on a set of descriptors of the object. There is no limit to the number of descriptors presented to derive a tree, or for the dimension or structure of a tree. The learning procedure of classification and regression tree, CART, selects the optimum number of descriptors to build the most efficient and small tree. The criterion of selection of descriptors is based on *entropy*. The term *entropy* in communication theory is related to the expected information carried out by a message. The entropy of a message, $I(m)$, depends on the probability of the message, $p(m)$, and is expressed in bits as:

$$I(m) = -log_2 p(m) \tag{3.1}$$

The assignment of an object s_i from a training set S to a class c can be interpreted as a message m with the information "s_i is a member of class m_j". The classification can be viewed as the selection of a message m_j from a set of possible messages M, where the set of messages in M are the possible classes C to which the object belongs. The messages m_j have probability $p(m_j)$ so the average entropy of the classification of an object is given by:

$$I\left(S\right) = \sum_{j=1}^{|C|} -p\left(m\right) log_2 p\left(m_j\right) \tag{3.2}$$

the sum of the products of each possible message multiplied by its probability.

The average entropy for the subsets S_1, ..., S_n produced by partition of the data set S on the basis of the states of the descriptor k-th is given by the sum of the entropy for each subset, obtained from Equation 3.2, scaled by the subset's proportion to S.

$$I\left(S_1, \, ..., \, S_n\right) = \sum_{i=1}^{n} \frac{|S_i|}{|S|} I\left(S_i\right) \tag{3.3}$$

The building of the tree starts with a simple condition $X \leq d$, where X is a descriptor and d is a real number. All observations are placed at the root node in the beginning. The goal is to find the rule that will initially break up the data set and create groups that are more homogeneous than the root node. The procedure to build the tree from the root until the leaves can by summarized in the following steps:

1. Starting with the first variable (descriptor), CART splits the variable at all the values that the variable takes in the entire training set (sample). At each split point of the variable, the objects of the training set are partitioned in two binary or child nodes. The objects with a "yes" answer to the question posed are sent to the left node and the objects with "no" answers are sent to the right node.

2. Each split point is evaluated in terms of reduction in impurity, or heterogeneity [69]. The evaluation is based on the goodness of split criterion. The objective is to maximize the reduction in the degree of heterogeneity.

3. The best split on the variable - the split that most reduces the impurity - is selected.

4. Steps 1-3 are repeated for each of the remaining variables at the root node. After

this, CART ranks all of the “best” splits on each variable according to the level of impurity achieved by each one.

5. The variable and its split point that most reduces the impurity of the root or parent node is selected.

6. CART assigns classes to these nodes in a way that minimizes misclassification costs.

7. Steps 1-6 are repeated to each non-terminal child node at each of the successive stages.

8. The splitting process can be continued to build a large tree until every observation constitutes a terminal node. In this case, the tree will have a large number of terminal nodes, generally with a small number of objects and high purity.

With a large tree generated, the method creates a sequence of nested trees. This set of smaller, nested trees is obtained by obliteration (pruning) of certain nodes of the large tree previously obtained. The selection of the weakest branches is based on a cost-complexity [69] measure that decides which sub-tree, from a set of sub-trees with the same number of terminal nodes, has the lowest (within node) error. Finally, from the set of all nested sub-trees, the tree giving the lowest value of error in cross-validation (where the set of objects used to grow the tree is different from the prediction set) is selected as the optimal tree.

3.2 Random Forests

A Random Forest, RF, is an ensemble of unpruned classification trees that can be used for classification or regression of objects. Subsection 3.2.1 is an introduction to the method. Subsection 3.2.2 describes the learning procedure showing how the classification of an object is obtained from an ensemble of trees.

3.2.1 Introduction

A RF [70, 71] is an ensemble of unpruned classification trees created by using bootstrap samples of the training data, i.e. a random selection of objects, and random subsets of variables to define the best split at each node. It is a high-dimensional nonparametric method that works well on large numbers of variables. The predictions are made by majority voting of the individual trees. It has been shown that this method is very accurate in a variety of applications. [71] The RF method is currently receiving much attention, and has been recently applied with success in several areas such as genetics, molecular biology, chemistry, mathematics, or toxicology. In the domain of chemoinformatics several applications have been put forward, most of them related to QSAR / QSPR studies and classification of data.

Examples of QSAR applications include, for example, the prediction of anti-microbial activity of some 3-nitrocoumarins and related compounds [72]. The activity and the molecular properties that determine the anti-microbial effects of these compounds were explored using 64 descriptors extracted from semiempirical and density functional theory (DFT) calculations. Multivariate predictive models based on Random Forests (RF), and two hybrid classification approaches, genetic algorithms (GA) associated with either support vector machines (SVM) or k nearest neighbor (kNN), have been used for the establishment of QSARs.

Other example is the prediction of mutagenicity [73]. Zhang et al use empirical physico-chemical descriptors to predict mutagenicity (positive or negative Ames test) from the molecular structure. An error percentage of ∼15% were achieved for an external test set using Random Forests as automatic learning method.

RF have been also used by different groups to predict aqueous solubility [74, 75]. Palmer et al used RFs, Partial Least Squares (PLS), Support Vector Machines (SVM) and NNs in the development of QSPR models for the prediction of aqueous solubility. A RMSE of 0.69 logS units were obtained by RFs. A similar study was performed by Schroeter et al for the estimation of aqueous solubility of drug molecules using different automatic learning methods.

In chemometrics, RFs were used to predict chromatographic retention times [76, 77]. Put et al reviewed the different machine learning techniques used for Quantitative Structure-Retention Relationships (QSRR) and the molecular descriptors used to build the models.

Pardo et al. applied RFs to classify E-Nose data sets for food quality control [78]. The results were compared with the ones obtained from SVM and shruken centroids. The advantages of each method were explained.

Koike [79] compared different machine learning methods for the prediction of chemical compound-protein binding affinities. A similar work was reported by Bruce et al [80] that compared several state-of-the-art machine learning methods for drug data mining, showing some advantages of RF over other methods.

In a different study Plewczynski et al [81] assessed different classification methods for virtual screening, to select ligands of protein targets from large databases of compounds not used to train the models. Several machine learning methods were used, including RFs. The predicted activities as ligand or not are treated as binary, and significant differences in the performance of the methods were observed. The experiments showed that some methods can provide more information in some aspects of the prediction than others, and that the combination of the results of different methods, in certain cases, can improve the prediction accuracy over the most consistent method.

In bioinformatics, RF were applied, for example, for the identification of 16 medically relevant strains (from four species of Mycobacterium tuberculosis) from MALDI-TOF MS data [82]. The error rate for the classification of individual strains by RFs is less than 50% of that obtained by Linear Discriminant Analysis. Moreover, the use of RFs in the analysis of MALDI-TOF MS data allowed to correctly classify bacterial strains as either M. tuberculosis or non-tuberculous with 100% of accuracy in a test set. RFs were also applied in the diagnosis of lung cancer using as descriptor the exhaled breath analysed by a calorimetric sensor array [83]. Mazzone et al used the pattern of 36 volatile organic compounds (VOCs) in the exhaled breath of patients with lung cancer as descriptors . RFs then discriminated the pattern of VOCs of individuals with lung cancer, other lung diseases and healthy controls. Donald et al [84] employed RFs in the diagnosis of prostate cancer from SELDI-TOF mass spectral serum profiles. RFs were able to correctly classify 100% of the cancerous patients of an external test set.

3.2.2 Learning Procedure

A RF is a ensemble of trees, each tree grown as described in Section 3.1 without pruning. The RF is built according to the following procedure:

- each tree is grown on a bootstrap sample of the training set
- a number of variables (descriptors of the object), m, is specified, where m is smaller than the total number of variables, M of the object
- at each node, m variables are randomly selected from the total group of M variables; generally the default value is $m = \sqrt{M}$.
- the split is the best split on these m variables
- the trees are grown to its maximum depth

Differently from a single tree, where the consistency is probed forcing the number of cases in each terminal node to become large (a large tree is generated and then smaller nested trees are obtained by obliteration of certain nodes of the first tree), RFs are built to have a small number of objects in each terminal node.

The method can handle hundreds or thousands of input variables without loss of accuracy. The final prediction is made by majority voting of the individual trees. The performance of each tree is evaluated with the estimation of the prediction error for the objects left out (typically one random third of the data set) - out-of-bag (OOB) objects. All OOB estimations for an object n (obtained by the trees that left the object n out of the training) are taken together to yield a consensus prediction with a voting procedure. The accuracy of the predictions obtained for all objects of the data set in

OOB estimations is a measure of the model's predictive ability. Each OOB prediction is a consensus prediction like the predictions, obtained by the RF global model. Since each individual OOB estimation, is independent from the data used to grow that individual model, the global OOB result can be taken as an independent validation of the RF model.

Generally, the accuracy of RF predictions is best than CART. Furthermore, the RF method allows for the assessment of the relative importance of each input variable to the answer. RFs have three different ways to measure the importance of each variable. One way consists in randomly permuting all values of the m^{th} variable in the OOB objects for the k^{th} tree. The corrupted OOB objects in the m^{th} variable are then submitted to the corresponding tree, predictions are obtained and a new internal error rate is calculated. The importance of the m^{th} variable is the difference between the new error and the original OOB error. The second and third measures of variable importance are related to the number of obtained votes in each class. For the n^{th} object, its margin at the end of a run is the difference between the proportion of votes for its true class and the maximum of the proportion of votes for each of the other classes. The second measure of importance of the m^{th} variable is obtained by the average lowering of the margin across all cases when the m^{th} variable is randomly permuted. The third measure is simply the difference between the count of margins lowered and raised.

The voting system of a RF also allows the association of a ”probability” to each prediction that reflects the percentage of votes obtained by the winning class. The RFs also allow to obtain a measure of similarity between objects. Since the individual trees are unpruned, the terminal nodes will contain only a small number of objects. To obtain a measure of similarity between two objects, both are submitted to the trained model. Every time an individual tree classifies the two objects in the same terminal node the proximity between them is increased by one. At the end, the proximities are divided by the number of trees in the forest.

Part II

Automatic Classification of Organic and Metabolic Reactions

This Part of the Thesis concerns the application of automatic learning methods to the classification of organic and metabolic reactions. The classification of reactions was investigated with two distinct methodologies: a representation of chemical reactions based on NMR data (Chapter 5, “Linking Databases of Chemical Reactions to NMR data:^{1}H NMR-Based Reaction Classification“, and Chapter 6, “Classification of Mixtures of Reactions from ^{1}H NMR data”), and a representation of chemical reactions from the reaction equation based on the physico-chemical and topological features of chemical bonds (Chapter 7, “Genome-Scale Classification of Metabolic Reactions. A Preliminary Study”, and Chapter 8, “ Genome-Scale Classification of Metabolic Reactions and Assignment of EC Numbers”). Finally in Chapter 9 (“Genome-Scale Classification of Pathways and Organisms”) the extension of the MOLMAP approach for the encoding of other levels of metabolic information was explored.

Chapter 4

State of the Art

This Chapter describes the state of the art related to the automatic classification of chemical reactions. Section 4.1 reviews reaction classification methods proposed over the last 30 years, with an emphasis on recent developments. Section 4.2 is dedicated to the MOLMAP reaction descriptors (used in Chapter 7 and 8). Section 4.3 highlights the current interest in genome-scale classifications of metabolic reactions. Finally, Section 4.4 presents the description of the ^{1}H NMR-based approach to encode chemical reactions, and an overview of applications of NMR spectroscopy related to chemical reactions.

4.1 Reaction Classification Methods

The complete understanding of the course of chemical reactions, under the most variety of conditions, is still in the beginning and the prediction of the reaction products presents several problems. The calculation of the pathway of a reaction by first principles methods can only be handled for the most simple cases.

Chemists typically base their knowledge about chemical reactions and reactivity on the experience generated from the observation of individual reactions and accumulated over more than 200 years. Chemists are trained to learn inductively from experiments, to make predictions about the products, configuration of molecules, and reaction mechanism for new situations. With the technological advances in computers making them accessible to any chemist and not only to the "computational chemist", this latent knowledge about chemical reactions starts to be stored in reaction databases. A database of chemical reactions can contain several millions of chemical reactions. Chemoinformatics methodologies are required to assist the chemist in deriving knowledge from this information, e.g. for synthesis design, or reaction prediction. With these huge amount of data collected in databases, chemoinformatic methods are needed for the classification of chemical reactions. For example a relevant issue in database reaction search is the retrieval of similar reactions to a query, in addition to exact matches. Substructure searches can lead to a hit list of considerable size making the manual analysis a hard or impossible task. The developments of tools are needed for automatic analysis of the stored information from search results or for comparison purposes from reaction databases. Methods for classification of chemical reactions can assist the chemist in the retrieval of chemical reactions from databases which enables an alternative method for indexing databases, give access to generic types of reactions, a post-search method for management of large hit lists and simplification of a query generation. They are also needed and essential for linking chemical information of different sources and construction of knowledge bases for reaction prediction and synthesis design. In Bioinformatics the reconstruction of metabolic reactions from genomes requires the classification of enzymatic reactions.

A topic closely related to reaction classification is reaction representation. Particularly for machine learning techniques to process reaction data, reactions have to be represented by descriptors, just like compounds are encoded by molecular descriptors. One approach consists in representing the reaction center, i.e. the bonds broken and made, the correspondence between atoms in reactants and products, and how valence electrons are changed during the reaction. A bond belongs to a reaction centre if it is made or broken during the reaction, and an atom is defined as a reaction center if it changes the number of implicit hydrogens, number of valencies, number of π electrons, atomic charge, or belongs to a changing bond. Another strategy represents chemical reactions by the difference between descriptors of the products and reactants, which produces a representation of the differences between the structures of the products and the reactants - an implicit representation of the reaction center and their environments. The second approach avoids the assignment of the reaction center, but requires stoichiometrically balanced reactions.

The assignment of the reaction centre can be made manually, or automatically with the implementation of appropriate algorithms. This is not a trivial task in general, and although (few) programs are available, new methods are still being published [85].

In the last 30 years, several methods for the representation and classification of reactions have been put forward, [41,86–130] mostly based on the encoding of bonding changes at the reaction center, [41, 91, 96, 107, 127, 128] or on physico-chemical properties of the reaction center. [110, 111, 113] The reaction classification methods based on pre-defined models of the bonding changes at the reaction centre are usually called “model-driven methods”. When the classification is derived from a data set of reactions and their classification, the methods are called “data-driven methods”.

4.1.1 Model-Driven Methods

A first model-driven method for reaction classification was proposed by Theilheimer [86]. In their approach the classification was based on a predefined model of changes in the reaction center, and was implemented for four types of reactions (addition, elimination, rearrangement and exchange). In the Theilheimer approach the reaction centre is described in a simple way. The bond formed is indicated before the symbol ↑ and the bond broken after the symbol ↓. For example $CC \uparrow\downarrow CX$ describe the formation of a carbon-carbon bond from a halide. Balaban [87], in 1967, based his model-driven method on the cyclic shift of six electrons on six atoms to describe different reactions families. The method was based on the superimposing of the atoms unchanged connecting bonds in thirteen possible forms. These thirteen forms include reactions of addition or elimination, sigmatropic rearrangements, cycloadditions, etc. In all cases of this approach, when a double bond changes only the π electrons are moved, and the σ bond remain unchanged. The similar happens with triple bonds where the second π bond remains unchanged. Following this work, Hendrickson [88] recouped the Balaban approach for pericycles with four and six atoms and added the set of odd-atom pericycles. The thirteen forms of the six-atom cycle were characterized by the number of unchanging "shell" σ-bonds among the six atoms, and labeled by their shapes. The same was applied to cycles of four atoms and four electrons. Hendrickson described the interrelation among homovalent and ambivalent reactions on the 5-atom pericycles.

Arens [89] developed a more general classification system formulating all basic forms of reactions, linear or cyclic, as cyclic reactions. The basic patterns of bond shifts are represented by a sequence of + and - to indicate whether the number of bonds at each successive place in the cycle increased or decreased by one, respectively. For example the pattern for a six-cycle was simply written as $(+-)_3$ or $(-+-+-+)$, taken in a clockwise rotation around the ring.

Vladutz [90] considered the Theilheimer's reaction classification scheme too broad with the four reaction types too diverse and without subclassification for further discrimination. The Vladutz approach then considered all the bonds that make or break bonds in the reaction center. It uses the same hierarchical distinction of reaction centre atoms but, instead of reaction → product representation of reactions the approach condenses the two in a simpler format with a single drawing to represent the reaction. The bonds broken are represented with a stroke through the bond, and those made, with arrowheads ↔, on a single cycle of reaction atoms. The unchanging bonds were drawn as plain lines. The reaction is labeled as a "superimposed reaction skeleton graph" (SRSG). Beyond the proposal of the method Vladutz did not provide an overall system for reaction classification.

Fujita [91, 92] developed a single "unitary" representation - the "imaginary transition state" (ITS) to show all the features of a chemical reaction in a single "structure". The method was based on the same idea of the Vladutz approach. The formed bonds are marked with a circle on the bond ("in-bonds"), the bonds broken ("out-bonds") are marked with a double stroke ("), and the unchanged shell bonds are drawn as plain lines ("par-bonds"). The reactant is derived by deleting the in-bonds and the product by deleting the out-bonds. Comparing to the Vladutz approach, Fujita extended the method to more complex reactions on the same ITS. The simple unitary ITS format is used but simplified showing reactant bonds which break with plain lines, product bonds formed with dotted lines and unchanging shell bonds with boldface.

In the Fujita method the hierarchy of reactions is labeled with the basic reaction graph (BRG), the reaction graph (RG) with shell bonds, and the fully defined reaction center graph (RCG) with the atoms type specified.

In the method proposed by Zefirov [93] the homovalent and ambivalent reactions are distinguished by denoting the ambivalent atoms, with a valence change of ±2, as specific atoms, and the homovalent atoms as ordinary. Zefirov establish a hierarchy of generalization proceeding in the following way: identification of the reaction centre atoms in the complete reaction; identification of the reduced system or reaction equation by removing unchanging bonds to the substituted atoms on the reaction centre and showing the bond redistribution with arrows. The symbolic equation (SEQ) by generalization of the atoms next as ordinary atoms with no valence change (homovalent) or specific (ambivalent). Zefirov also partitioned the topology of reaction systems into linear and mono-cyclic, and stated that more complex cases, such as bicyclic, could be represented as sequences of linear and monocyclic primary reactions. The SYMBEQ [94, 95] program was developed by Zefirov, based on his approach for reaction classification. The program generates symbolic equations from topologies of bond changes at the reaction center.

Hendrickson developed a unified classification system that tried to merge those model-driven methods [96] and at the same time provide a general nomenclature for linear notation. The hierarchy is made by reaction centers, with subclasses of unchanging shell σ-bonds and these are further classified in terms of atom types of the reaction center atoms as in the Fujita approach.

Another reaction classification method with a different concept is the one developed by Ugi in the 1970s. This is a matrix-based reaction classification method [97, 98]. This method was applied in synthesis design and simulation of chemical reactions by the EROS system [99, 100], searching of reactions with the IGOR program [101], determination of minimum chemical distance between any reactant and product [102], and synthesis planning with the RAIN program [103]. The reaction representation is based on the addition of a connectivity matrix for the ensemble of reactant molecules ("beginning", B) to a reaction matrix (R) to yield the matrix of the ensemble of product molecules ("end", E), $B + R = E$. The R matrix can be applied to any ensemble of molecules (B) to show the products (E) characteristic of the reaction (R).

4.1.2 Data-Driven Methods

The Wilcox reaction classification method [104], differently from the previous methods is a data-driven method. No predefined rules are responsible for the classification of the presented reactions to the system. The Wilcox method is based on a bond-centered approach, with the changing bonds in the reaction represented in a unitary format. The unchanging bonds are drawn as plain, and the changing bonds dotted and annotated with numbers $m : n$ representing the bond multiplicity of the reactant and product respectively. A compound is defined with all bonds plain and a reaction with some bonds dotted. A line graph of these bonds is used to define the reaction in the computer. The computer calculates for each reaction the "minimum reaction concept" (MXC) from the dotted bonds and then the "complete reaction concept" (CXC) by adding to MXC all unchanging bonds adjacent to it.

Wilcox concluded that the approach retrieves reliable results only when the information concerning adjacent unchanging functionality is also given to the system. With this inclusion the system not only has information about the reaction centre but also on the influence of adjacent functional groups on the reaction.

The Gelernter reaction classification method [105] is based on a "conceptual clustering" technique. In this approach the "reaction context" is described as the attached functionality that is not transformed in the reaction. The reaction centers are first partitioned in primitive clusters and then subdivided in distinct groups of reactions based on the functional groups near the reaction center.

ClassCodes are implemented in InfoChem software for representing and classifying reactions based on hashcodes of the reaction centers. The developed algorithm (Classify) [106, 107] is a structure-topology approach based on the InfoChem's reaction center perception algorithm (RCP). A bond is considered as a reaction center if it is made or broken or changed, and an atom is considered as a reaction center if it changes the number of implicit hydrogens, the number of valences, the number of p-electrons, the atomic charge and/or in the case that the connecting bond is a reaction center.

The reaction center hashcodes are calculated for all reaction centers taking into account atom properties: atom type, valence state, total number of bonded hydrogens (implicit plus explicitly drawn), number of p-electrons, aromaticity, formal charges, bond reaction center information. The hashcode also depends on the contributions of the neighbor atoms of the reaction center. The sum of all reaction center hashcodes of all reactants and one product of a reaction provides the unique reaction classification code: the ClassCode. Multiple occurrences of identical transformations are handled as one. ClassCodes are defined at three levels of specificity depending on the spheres of atoms around the reaction center that are covered, as illustrated in Figure 4.1.

Kanehisa and co-workers developed the reaction classification numbers [41] for integrating chemical information on KEGG database (Kyoto Encyclopedia of Genes and Genomes). This approach allows the representation of chemical reactions that are catalysed by enzymes. In this method a reaction is decomposed into reactant pairs (a reactant and the corresponding product) and each pair is then structurally aligned to identify the reaction center (R), the matched region (M), and the difference region (D). The method

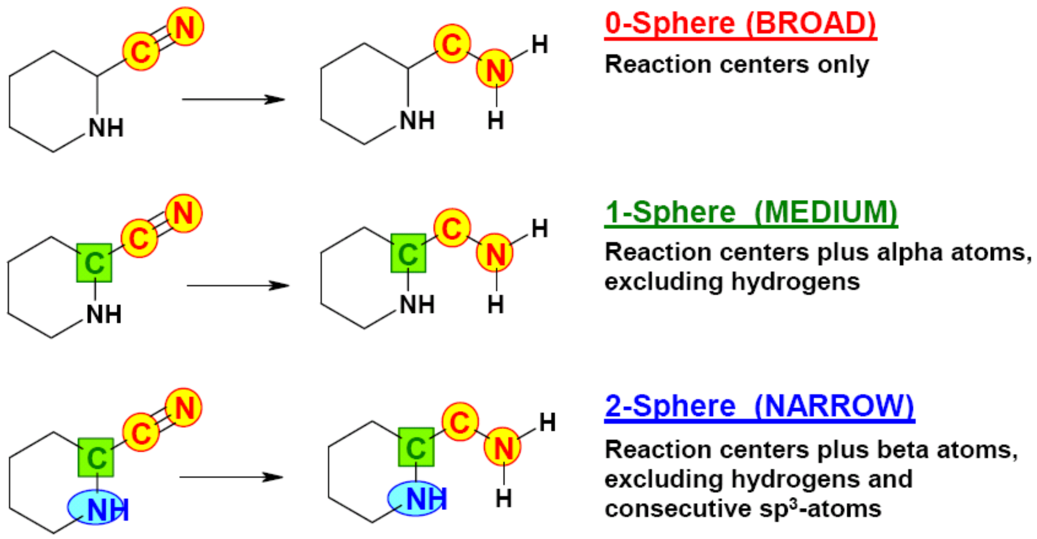

0-Sphere (BROAD)
Reaction centers only
1-Sphere (MEDIUM)
Reaction centers plus alpha atoms, excluding hydrogens
2-Sphere (NARROW)
Reaction centers plus beta atoms, excluding hydrogens and consecutive sp³-atoms
C
N
NH
H

Figure 4.1: The three levels of information of the Infochem method. Inclusion of atoms in the immediate environment (spheres) of the reaction center. From *Classify - The InfoChem Reaction Classification Program Version 2.9. to be found under http://www.infochem.de/en/ downloads/documentations.shtml.*

is based on a list of 68 predefined atom types (Figure 4.3 shows the definition of 23 atom types of carbon). From this list of atom types (68 different atom types) a list of numerical codes for conversion patterns of atom types, e.g. C1a -> C1a; C1a -> C1b C1b -> C1a;...; C1c -> C1c is defined. The conversion patterns of atom types in the three regions are encoded with numerical codes to obtain so-called RC numbers. Figure 4.2 shows an example of the encoding of an enzymatic reaction.

This representation of chemical reactions can be applied for the automatic assignment of EC numbers to enzymatic reactions. A query reaction is classified based on reactions sharing the same RC number in a database. Different restrictions are possible (e.g., RDM or only RD). This approach allowed the assignment of EC sub-subclasses with the accuracy of about 90% (coverage: 62% of a data set).

Mitchell and co-workers also developed a reaction representation scheme for enzymatic reactions [108]. Their approach provides a quantitative measure of the similarity of reactions based upon in their explicit mechanisms. Two approaches are presented to measure the similarity between individual steps of mechanisms: a fingerprint-based approach that incorporates relevant information on each mechanistic step; and an approach based only on bond formation, cleavage and changes in order.

In one case 58 features of a reaction are used as reaction descriptors, including the number of reactants, products - reactants, cycles in products - cycles in reactants, and the number of times a bond type is involved in the reaction (for 21 bond types). Descriptors also included the involvement of cofactors, the total number of each type of bond order change (bond formation, bond cleavage, changes in order from 1 to 2, 2 to 1, 3 to 2),

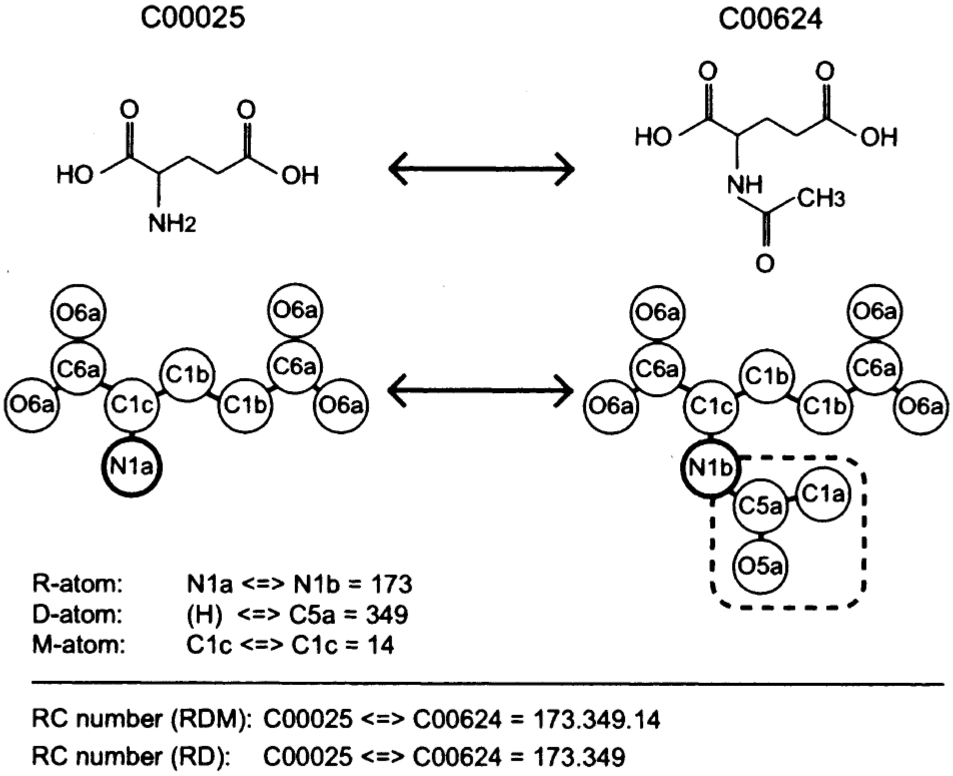
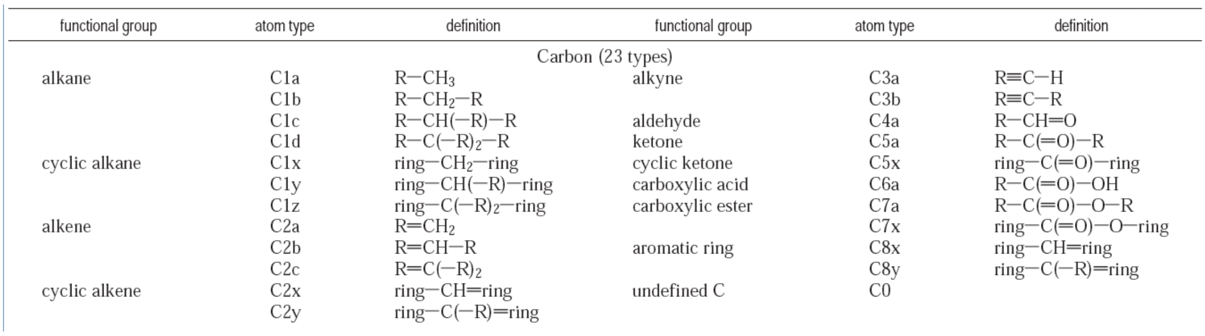

C00025
C00624
HO
OH
NH2
NH
CH3
O
O6a
C6a
C1c
C1b
N1a
N1b
C5a
C1a
O5a
R-atom: N1a <=> N1b = 173
D-atom: (H) <=> C5a = 349
M-atom: C1c <=> C1c = 14
RC number (RDM): C00025 <=> C00624 = 173.349.14
RC number (RD): C00025 <=> C00624 = 173.349

Figure 4.2: Assignment of the RC number, which describes the conversion patterns of the KEGG atom types for the reaction center atom (R-atom), the difference structure atom (D-atom), and the matched structure atom (M-atom). The definition of the R-, D-, and M-atoms are somewhat different for the cases of (A) a partial match with the difference structure (surrounded by dashed line). From M. Kotera, Y. Okuno, M. Hattori, S. Goto, M. Kanehisa, *J. Am. Chem. Soc.* 2004, *126*, 16487-16498.



functional group	atom type	definition	functional group	atom type	definition
		Carbon (23 types)			
alkane	C1a	R—CH$_3$	alkyne	C3a	R≡C—H
	C1b	R—CH$_2$—R		C3b	R≡C—R
	C1c	R—CH(—R)—R	aldehyde	C4a	R—CH=O
	C1d	R—C(—R)$_2$—R	ketone	C5a	R—C(=O)—R
cyclic alkane	C1x	ring—CH$_2$—ring	cyclic ketone	C5x	ring—C(=O)—ring
	C1y	ring—CH(—R)—ring	carboxylic acid	C6a	R—C(=O)—OH
	C1z	ring—C(—R)$_2$—ring	carboxylic ester	C7a	R—C(=O)—O—R
alkene	C2a	R=CH$_2$		C7x	ring—C(=O)—O—ring
	C2b	R=CH—R	aromatic ring	C8x	ring—CH=ring
	C2c	R=C(—R)$_2$		C8y	ring—C(—R)=ring
cyclic alkene	C2x	ring—CH=ring	undefined C	C0	
	C2y	ring—C(—R)=ring			

Figure 4.3: The 23 (carbon types) out of 68 atom typing for defining the 68 KEGG atoms. From M. Kotera, Y. Okuno, M. Hattori, S. Goto, M. Kanehisa, *J. Am. Chem. Soc.* 2004, *126*, 16487-16498.

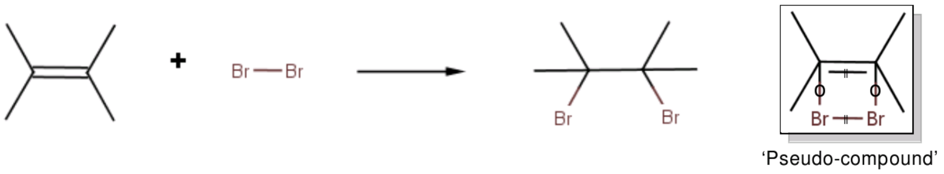


Figure 4.4: Reactants and products are merged in an imaginary transition state or pseudo-compound.

charge changes by atom type, and involvement of radicals. The method was applied to the assessment of similarity between individual steps of enzymatic reaction mechanisms and in the quantitatively measurement of the similarity between enzymatic reactions based upon their explicit mechanisms. The study used the MACiE database of enzyme reaction mechanisms, and identified some examples of convergent evolution of chemical mechanisms.

Recently, Varnek et al [109] developed a method for reaction classification based on the Condensed Reaction Graphs of Vladutz and Fujita [90–92] In this approach the reactants and products are merged in an imaginary transition state or pseudo-compound, as Figure 4.4 illustrates. Bonds types are defined according to their fate in the reaction ("no bond" to single, single to double, double to "no bond",...). Then fragments are derived around the reaction center, that are predefined-sized sequences of atoms of specific atom types connected by bonds of specific bond types. In this approach the descriptor of a reaction corresponds to the number of occurrences of a fragment. Descriptors can be used for the assessment of similarity between reactions, or for QSPR studies with reactions (just like molecular descriptors are used with molecules).

The methods presented until this point are only based on topological properties (atom elements and connectivity) of the reaction center and their neighborhood.

Diversely, Gasteiger and Funatsu groups proposed the representation of chemical reactions by physico-chemical features of the atoms and bonds at the reaction center [110–112] which are, in principle, related to the mechanism. Electronic factors such as charge distribution, inductive and resonance effects have a great influence in the reaction mechanism and their use was proposed as a possible way to indirectly assess mechanistic features of reactions in the absence of information about mechanisms in reaction databases. The HORACE reaction classification method developed by Rose et al [110, 112] use as descriptors of the reaction center empirical physico-chemical parameters implemented in the PETRA package [131] for the quantification of inductive and resonance effects at the atoms and bonds of the reaction center.

Chen and Gasteiger [113, 114] developed a reaction classification method based on physico-chemical properties of the reaction center, encoded in a numerical fixed-length code, which enabled to explore Kohonen Self-Organizing Maps (SOM) for reaction classification. In their first experiments [113] they explored the representation / classification

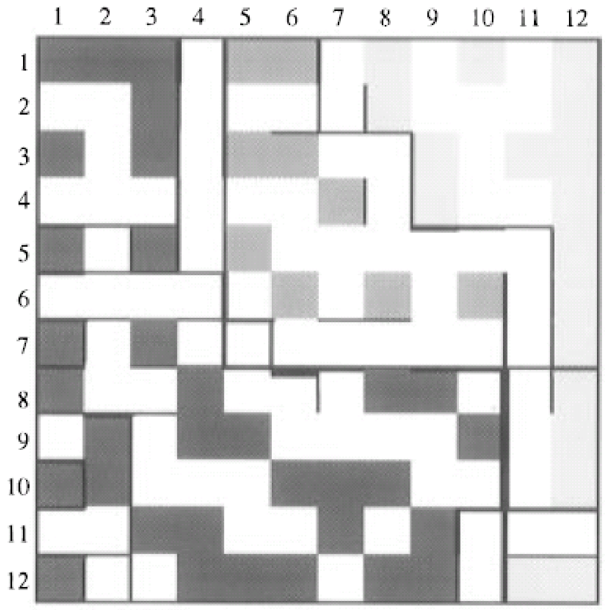


Figure 4.5: Kohonen SOM for the classification of chemical reactions. Indication of the weight distance between adjacent neurons by the thickness of lines separating them. The occupied neurons are marked in different gray levels: dark gray - nucleophilic aliphatic substitution of acyl chlorides, medium gray - acylation of C=C bonds, and light gray - acylation of arenes. (a) Only neurons with mapped reactions are marked. From L. Chen, J. Gasteiger, *J. Am. Chem. Soc.* 1997, *119*, 4033-4042.

of reactions with a common reaction center. In a first example each reaction was represented by five electronic parameters and these descriptors were used to map 74 reactions with a common reaction center onto a Kohonen Self-Organizing map. Figure 4.5 shows the obtained map. The method enables to discriminates similarities and relationships between reactions.

Chen and Gasteiger further explored the method by applying it to a set of 120 reactions used in the HORACE approach. In this experiment each reaction was represented by seven electronic parameters. The mapping of the 120 reactions in the trained map shows a good agreement between the classification performed by the method and the classification performed by chemists. The method was also able to detect errors in the database of reactions used.

Other experiments were performed with reactions also sharing features in the reaction center, but the feature being much less specific - the presence of an oxygen atom [111]. Satoh et al described the reactions by the changes in σ-charge, π-charge, π-electronegativity, σ-electronegativity, polarizability, and pKa values at the oxygen atoms of the reaction sites. Figure 4.6 illustrates the encoding of a reaction. This representation was used to map 152 O-atoms from 131 reactions on a Kohonen SOM .

More recently, the same method was applied to classify metabolic reactions of subclass EC 3.1.x.x (sharing features at the reaction center) on the basis of physico-chemical

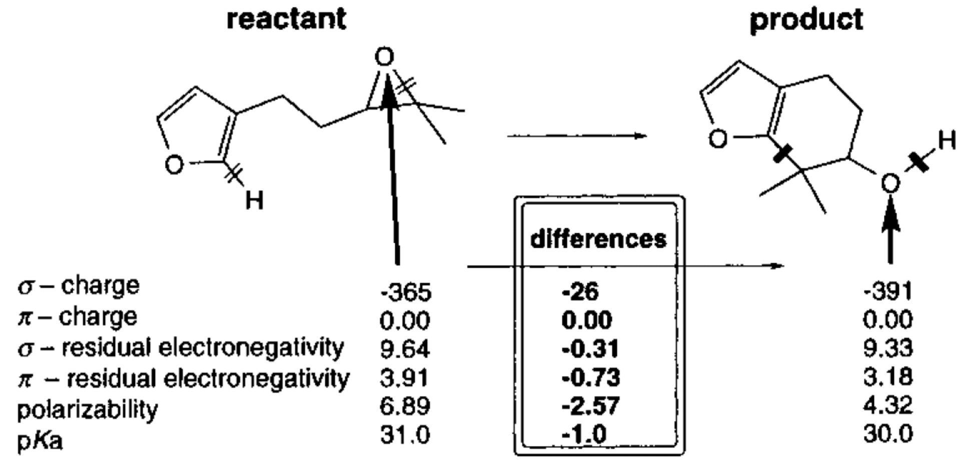


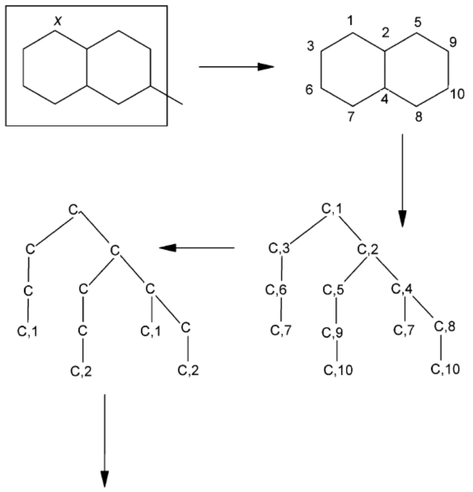
Figure 4.6: A data set for the analysis. The changes in σ-charge, π-charge, σ-residual electronegativity, π-residual electronegativity, polarizability, and pKa values at the 154 oxygen atoms of the reaction sites in going from the reactants to the products are taken as a characterization of the individual reactions and are used for their classifications. In this example, differences in these values between an oxygen atom in the epoxide of the reactant to an oxygen atom in the hydroxy group of the product are calculated. From H. Satoh, O. Sacher, T. Nakata, L. Chen, J. Gasteiger, K. Funatsu, *J. Chem. Inf. Comput. Sci.* 1998, *38*, 210-219.

properties of the reactants. [115]

Finally the concept of representing reactions by physico-chemical features of the reaction center was applied to data sets of reactions with diverse reaction centers. The Sacher-Gasteiger approach [116] encoded reactions with 6 physico-chemical properties for the bonds of products at the reaction center (bond order, difference in σ- and π-electronegativity between the two atoms of the bond, difference in total charge between the two atoms of the bond, stabilization of + and − charge by delocalization). The reactions were represented by vectors with room for up to 6 bonds, and for the assignment of positions of the bonds in the vector, bonds must be ranked. The method was used in the mapping of 33,613 reactions from the Teilheimer database on a 92×92 Kohonen SOM.

Differently from the methods described so far, the following methods do not explicitly represent the reaction center, but use instead differences between fingerprints (or signatures) of products and reactants to represent reactions.

A method implemented in Daylight software can represent reactions by "difference Daylight fingerprints". These are obtained from the difference in the Daylight fingerprints of reactant molecules and the fingerprints of product molecules. [117] The difference in the fingerprint of the reactant molecules and the fingerprint of the product molecules reflects the bond changes which occur during the reaction. This method avoids assignment of reaction centers and atom-to-atom mapping. Because fingerprints are binary, multiple occurrences of a path are not encoded, and a simple subtraction is not enough. It is required to keep track of the count of each path in the reactant and product and then



${}^0\sigma(x) = \text{C}$

${}^1\sigma(x) = \text{C (C C)}$

${}^2\sigma(x) = \text{C (C (C) C(C C))}$

${}^3\sigma(x) = \text{C (C (C (C,1)) C(C (C) C (C,1 C)))}$

${}^4\sigma(x) = \text{C (C (C (C,1)) C(C (C (C,2)) C (C,1 C(C,2))))}$

Figure 4.7: Atomic signature. The figure illustrates the five steps procedure for computing the atomic signature of atom x in methylnaphthene. (1) The subgraph containing all atoms at distance 4 from atom x is extracted. (2) The subgraph is canonized atom x having label 1. (3) A tree spanning all edges of the subgraph is constructed. (4) All labels appearing only once are removed, and the remaining labels are renumbered in the order they appear. (5) The signature is printed reading the tree in a depth-first order. Here we show atomic signatures from h = 0 to h = 4. From J.-L. Faulon, D. P. Visco, Jr., R. S. Pophale, *J. Chem. Inf. Comput. Sci.* 2003, *43*, 707-720.

subtract the counts of a given path. If the difference in count is non-zero, then the path is used to set a bit in the difference fingerprint. If the difference in count is zero, then no bit is set for that path in the difference fingerprint.

In a different way Faulon et al [118, 119] proposed “reaction signatures” which are defined as the difference between the molecular signatures of the products and the molecular signatures of the substrates. Molecular signatures are made of atomic signatures. An atomic signature is a canonical representation of the subgraph surrounding a particular atom (Figure 4.7). This subgraph includes all atoms and bonds up to a predefined distance from the given atom (the height, h). Each component of a molecular signature counts the number of occurrences of a particular atomic signature in the molecule. The Faulon approach was very recently applied to the classification of enzymatic reactions for the automatic assignment of EC numbers in a data set of 6,556 reactions from the KEGG database using Support Vector Machines (SVM). The method allowed for the correct assignment of EC class, subclass, and sub-subclass in up to 91%, 84% and 88% respectively (LOO cross-validation).

A similar approach for was implemented by Ridder et al [120]. Fingerprints are gener-

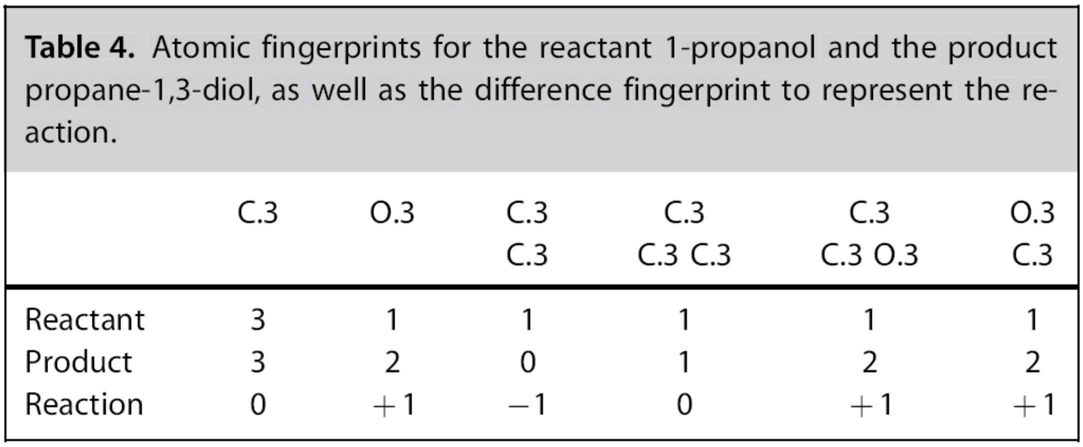


Table 4. Atomic fingerprints for the reactant 1-propanol and the product propane-1,3-diol, as well as the difference fingerprint to represent the reaction.

	C.3	O.3	C.3	C.3	C.3	O.3
			C.3	C.3 C.3	C.3 O.3	C.3
Reactant	3	1	1	1	1	1
Product	3	2	0	1	2	2
Reaction	0	+1	−1	0	+1	+1

Figure 4.8: Atomic fingerprints for the reactant 1-propanol and the product propane-1,3-diol, as well as the difference fingerprint to represent the reaction. From "L. Ridder, M. Wagener, *ChemMedChem* 2008, *3*, 821- 832".

ated for reactant and product molecules separately, based on Sybyl atom types and atom types augmented with a single layer around the central atom. The difference fingerprint is defined by the differences in occurrence of each atom type in the reactant and product fingerprints. Figure 4.8 shows and example of the encoding of a reaction. The method was applied in the classification of metabolic reactions to assist in the establishment of rules for reaction prediction. The reactions of a training set was projected on a 2D plane to optimally reflect reaction fingerprint distances calculated between all pairs of reactions. The method is based on stochastic proximity embedding, and optimizes the distances between points on a 2D plane to correspond as much as possible to the distances calculated in the fingerprint space between all pairs of metabolic reactions (Figure 4.9).

4.2 The MOLMAP Approach

The MOLMAP approach [122, 123] to encode molecular structures and reactions was developed in the research group of João Aires-de-Sousa in FCTUNL and was used by the author in most part of the experiments concerning classification of metabolic reaction.

The intrinsic chemical reactivity of a compound is related to its propensity for bond breaking and bond making, which mainly depends on the physico-chemical properties of the bonds. Gasteiger et al. [132–134] proposed a group of empirical physico-chemical properties particularly relevant in predicting the reactivity of a chemical bond toward heterolytic cleavage. The proposed properties are, for example, the difference in total charge, the difference in π charge, the difference in σ electronegativity, the bond polarity, the resonance stabilization of charges generated by heterolysis and the effective bond polarizability. These encode charge distribution effects, [135, 136] inductive effects, [137] resonance effects, [135] and polarizability effects [138]. In the methods proposed by Gasteiger et al

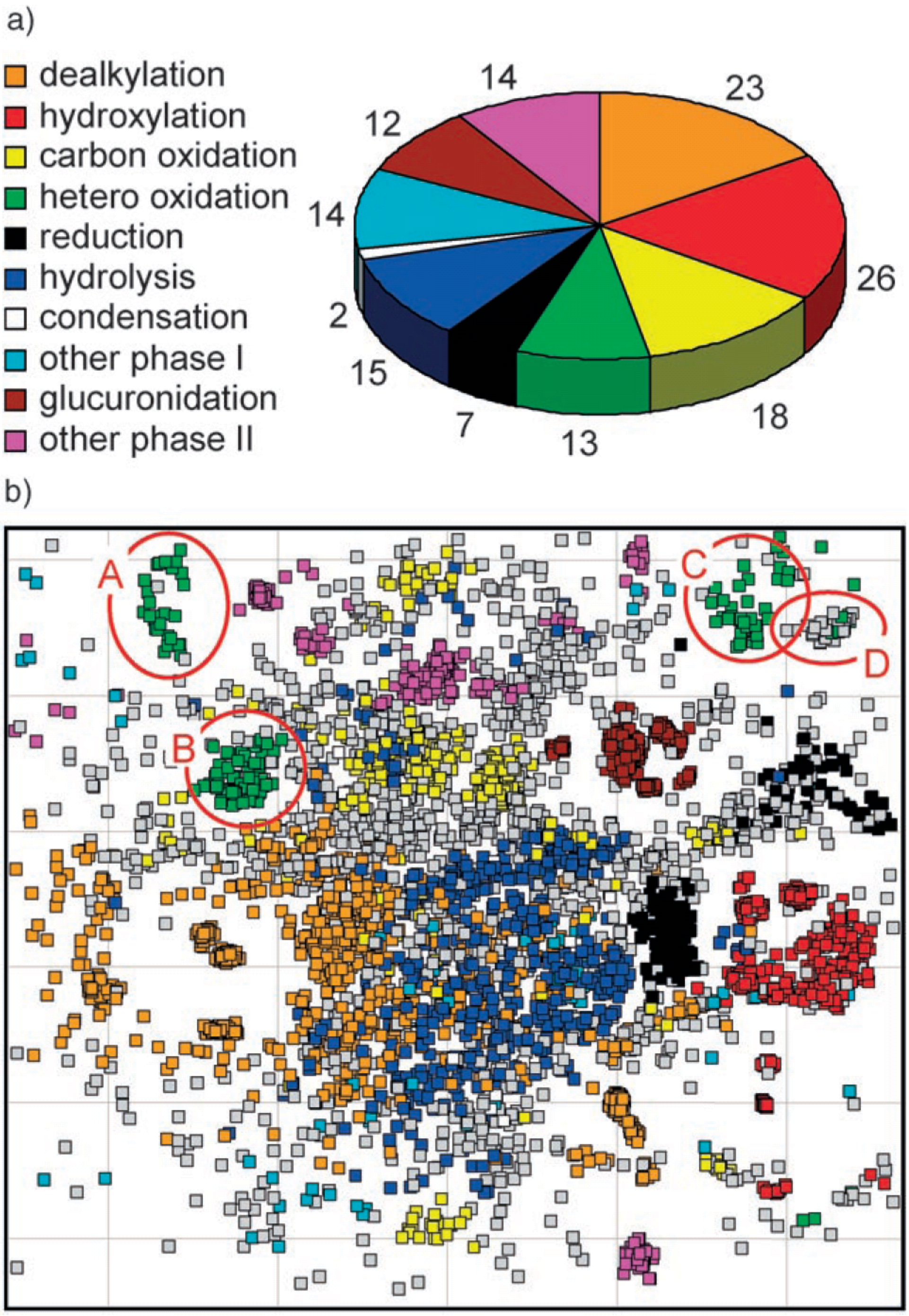


Figure 4.9: a) Distribution of the rules for the various types of metabolic reactions. b) Projection of all reactions in the training set on a 2D plane to optimally reflect reaction fingerprint distances calculated between all pairs of reactions. For each reaction, it was verified whether SyGMa could reproduce the metabolite in up to three subsequent reaction steps at a certain point during rule development. Reactions reproduced by SyGMa in two or three steps were excluded from the analysis. Reactions reproduced by SyGMa in one step are colored according to the rule they matched, within one of the indicated categories. Reactions not matched by SyGMa in up to three subsequent reaction steps are represented by gray dots. Four clusters of hetero-atom oxidation reactions (A-D) are circled. From "L. Ridder, M. Wagener, *ChemMedChem* 2008, *3*, 821- 832".

this information is only used for the encoding of the reaction center. To use all this information for an entire molecule, and at the same time have a fixed-length representation, all the bonds of a molecule are mapped into a Kohonen SOM - a MOLMAP (MOLecular Map of Atom-level Properties). The method is based on the classification of bonds by a Kohonen SOM from their physico-chemical and topological features. A trained Kohonen SOM can automatically assign bonds types and by this way a MOLMAP encodes the bond types that exist in a molecule. The generation of a reaction MOLMAP involves the following three main steps:

- The training of a Kohonen SOM with bonds
- The generation of molecular MOLMAPs from the trained Kohonen SOM
- The generation of reaction MOLMAPs from the molecular MOLMAPs

The MOLMAP descriptors are used in this Thesis to represent reactions and to classify genome-scale data sets of metabolic reactions [123–125] . It can also be applied directly (the molecular MOLMAPs) in QSAR studies related to chemical reactivity [73, 139, 140].

4.2.1 Training a Kohonen SOM with Bonds

A Kohonen SOM must be trained with a diversity of bonds selected randomly or by other methods to cover the chemical space of the problem under study. Each bond is described by a set of physico-chemical and/or topological properties (in the preliminary study it is used only seven physico-chemical properties and in the main study different sets of physico-chemical and topological properties are used to better evaluate the impact in the classification accuracy). The Kohonen SOM learn by unsupervised training distributing objects over a 2D surface (a grid of neurons) in such a way that bonds bearing similar features are mapped onto the same or adjacent neurons. [1,141] The input to the Kohonen SOM is the set of bond properties (Figure 4.10)

The learning process is carried out as presented in Section 2.6. Figure 4.11 shows how different regions of the surface are activated by different types of bonds.

4.2.2 Generation of MOLMAP Molecular Descriptors

A representation of the set of bonds existing in a molecule can be obtained by mapping the bonds of that molecule on the SOM previously trained with a diversity of bonds. Each bond activates one neuron. The pattern of activated neurons (the MOLMAP) is a map of the bonds available in the structure (it may also be interpreted as a fingerprint of reactivity features of the molecule). [122] For easier computational processing, the pattern of activated neurons is encoded numerically. Each neuron is given a value equal to the

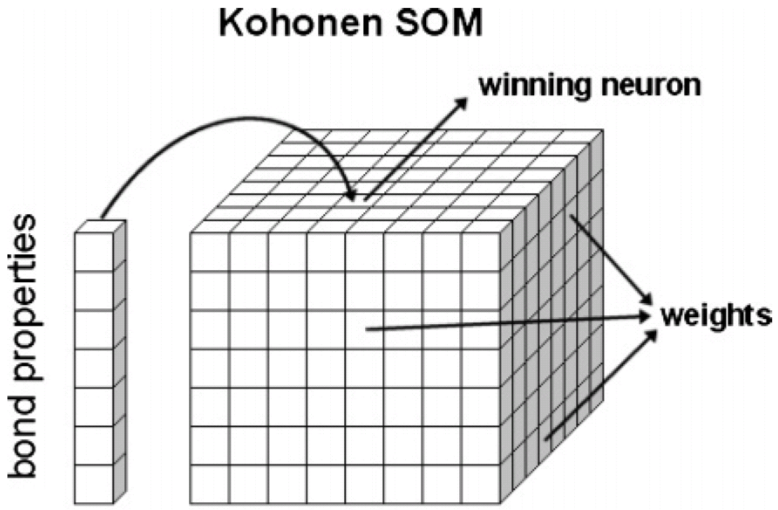


Figure 4.10: Representation of a Kohonen SOM for processing chemical bonds. Every small box of the block represents a weight. Each column of small boxes in the block represents a neuron. The Kohonen SOM is trained by the iterative presentation of objects (bonds described by physico-chemical and/or topological parameters)

number of times it was activated by bonds of the molecule. The map (a matrix) is then transformed into a vector by concatenation of columns. To account for the relationship between the similarity of bonds and proximity in the map, a value of 0.3 was added to each neuron multiplied by the number of times a neighbor was activated by a bond. If an empty neuron is a direct neighbor of more than one activated neuron, its value is the sum of the contributions from each activated neuron. Figure 4.12 illustrates the generation of the MOLMAP for a molecule.

4.2.3 Generation of MOLMAP Reaction Descriptors

A chemical reaction is defined by the changes operated in the reactants, leading to the formation of the products. The MOLMAP of the reaction is obtained by subtracting the MOLMAP of the reactants from the MOLMAP of products. Such a procedure eliminates MOLMAP components corresponding to unchanged bonds, and yields non-null values for components corresponding to bonds that changed, disappeared from the reactants, or were made anew in the products. The resulting MOLMAP is thus a representation of the changes occurring with the reaction, which implicitly represents features of the reaction center and its neighborhood. If the reaction involves more than one reactant, the MOLMAPs of all the reactants must be overlapped and numerically summed, and the same has to be done for the products. Figure 7.1 shows a step-by-step illustration of the procedure to encode a reaction (a-c).

This method provides a fixed-length numerical representation of chemical reactions, which is based on topological and physico-chemical molecular features, does not require

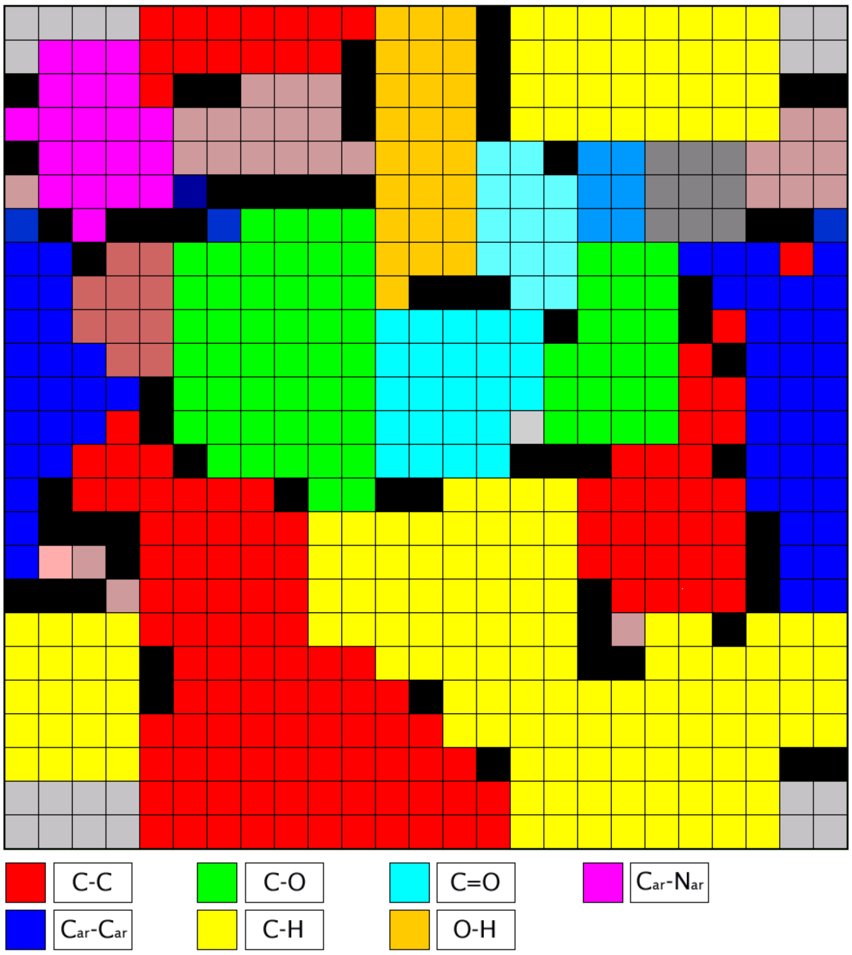
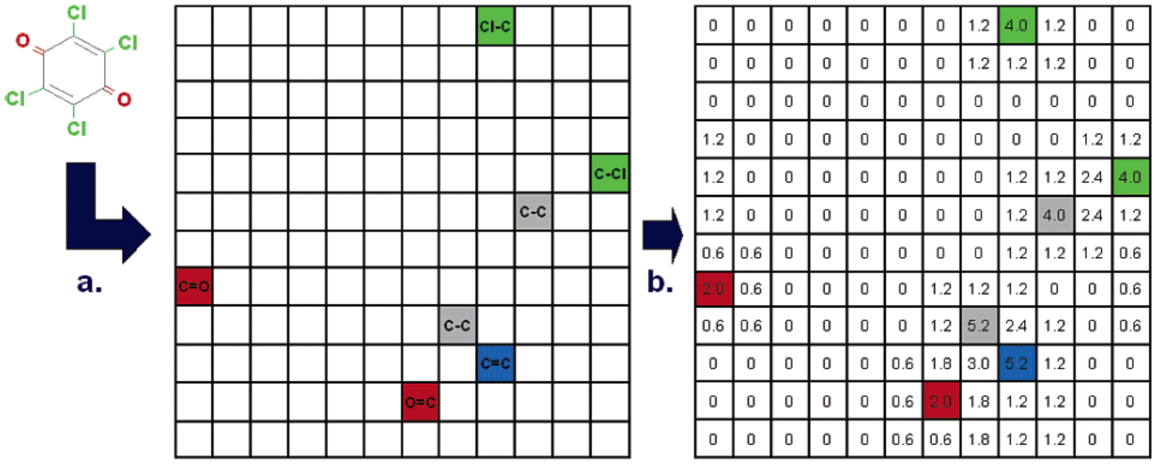


Figure 4.11: A Kohonen SOM trained with bonds represented by physico-chemical and topological descriptors. Each neuron was colored after the training, according to the types of chemical bonds that were mapped onto it. The figure shows how different regions of the surface are activated by different types of bonds.



0	0	0	0	0	0	0	1.2	4.0	1.2	0	0
0	0	0	0	0	0	0	1.2	1.2	1.2	0	0
0	0	0	0	0	0	0	0	0	0	0	0
1.2	0	0	0	0	0	0	0	0	0	1.2	1.2
1.2	0	0	0	0	0	0	0	1.2	1.2	2.4	4.0
1.2	0	0	0	0	0	0	0	1.2	4.0	2.4	1.2
0.6	0.6	0	0	0	0	0	0	1.2	1.2	1.2	0.6
2.0	0.6	0	0	0	0	1.2	1.2	1.2	0	0	0.6
0.6	0.6	0	0	0	0	1.2	5.2	2.4	1.2	0	0.6
0	0	0	0	0	0.6	1.8	3.0	5.2	1.2	0	0
0	0	0	0	0	0.6	2.0	1.8	1.2	1.2	0	0
0	0	0	0	0	0.6	0.6	1.8	1.2	1.2	0	0

Figure 4.12: Generation of a molecular MOLMAP descriptor for a molecule. (a) Submission of bonds to a previously trained SOM. (b) Numerical processing of the pattern of activated neurons. Adapted from Q.-Y. Zhang and J. Aires-de-Sousa, *J. Chem. Inf. Model.* 2005, *45*, 1775-1783.

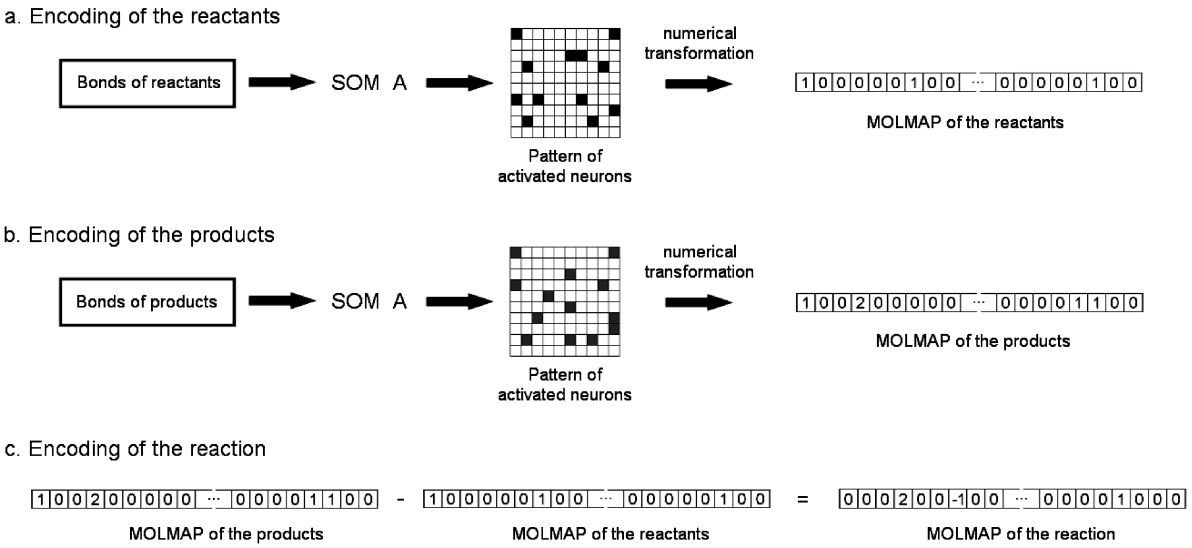


Figure 4.13: Simplified illustration of the procedure used to encode a reaction (a-c).

the assignment of reaction centers, i.e. the explicit identification of the bonds that change in the reactants or are formed in the products, and avoids atom-to-atom mapping, i.e. the correspondence between atoms in the reactants and products.

4.3 Genome-Scale Classification of Metabolic Reactions

The understanding of the multiple ways small molecules affect biological systems is demanding an integration of chemical and biological data in 'systems biology' approaches. [29] In this context, 'enzymatic function' emerges as a key gateway between the universes of biology and chemistry. Employed for the annotation of genes and proteins, or encoded as an edge in graph representations of metabolic networks, 'enzymatic function' has an intrinsic chemical nature - it is the catalysis of a chemical reaction. The automatic comparison and classification of enzymatic reactions, a current issue in bioinformatics, requires specific chemical knowledge and chemoinformatics methodologies. Bioinformatics applications of reaction classification and comparison include a) computer-aided validation of classification systems, e.g. the assignment of EC (Enzyme Commission [142]) numbers in the context of an ever increasing number of enzymatic reactions; b) genome-scale reconstruction of metabolic pathways, where similarity searches of metabolic reactions are helpful for the proposal of enzyme sequences from their functions, and for the annotation of unknown genes; [41, 143, 144] c) classification of enzymatic mechanisms; [108] d) visualization of reactomes; e) enzymatic structure-function studies; [26,27,35,145] f) alignment of metabolic pathways. [146] This Thesis reports the classification of a genome-scale data set of metabolic reactions by self-organizing maps (SOM) and Random Forests using physico-chemical and topological features of reactants/products to represent reactions. The data set encompasses all possible reactions in the KEGG database, which includes exhaustive lists of known metabolic reactions from different organisms.

Protein catalytic functions are officially classified by the EC numbers assigned to the catalysed chemical reactions. [147, 148]

The Enzyme Commission number (EC number) is a numerical classification scheme for enzymes, based on the chemical reactions they catalyse. EC numbers do not specify enzymes, but enzyme-catalysed reactions. If different enzymes (for instance from different organisms) catalyse the same reaction, then they receive the same EC number. Every enzyme code consists of the letters ”EC” followed by four numbers separated by dots. Those numbers represent a progressively finer classification of the enzyme.

The EC system is a hierarchical classification system that proceeds from a first level of the basic form of the reaction change (six classes - oxidoredutases, transferases, hidrolases, lyases, isomerases and ligases) and developing each one down to successive levels of generalization with more detail , finally arriving at particular defined reactions. The third level is a specific reaction and the fourth digit is in principle a serial number for the substrate, or for the enzyme. There is no general rule, because the meaning of these digits is defined separately for each class. The assignment of a reaction with the subclass EC 1.1.x.x has a different rule than the assignment of a reaction with the subclass EC 2.1.x.x.

EC 1.x.x.x, oxidoreductases, catalyse oxidation/reduction reactions (transfer of hydrogen and oxygen atoms or electrons from one molecule to another). The typical reaction is:

$$AH + B \rightleftharpoons A + BH \quad (reduced)$$

$$A + O \rightleftharpoons AO \quad (oxidized)$$

EC 2.x.x.x, transferases, catalyse reactions with transfer of an atom or functional group between two molecules (the group may be methyl-, acyl-, amino- or phosphate group), excluding reactions in other classes (specifically exclude oxidoreductases and hydrolase reactions). The typical reaction is:

$$AB + C \rightleftharpoons A + BC$$

EC 3.x.x.x, hydrolases, catalyse hydrolysis reactions (formation of two products from one substrate and water). The typical reaction is:

$$AB + H_2O \rightleftharpoons AOH + BH$$

EC 4.x.x.x, lysases, catalyse non-hydrolytic addition or removal of groups from substrates (C-C, C-N, C-O, C-S bonds can be cleaved), often leaving double bonds. The typical reaction is:

$$AB \rightleftharpoons A + B$$

EC 5.x.x.x, isomerases, catalyse intramolecular rearrangement (isomerization changes within a single molecule). The typical reaction is:

$$A \rightleftharpoons A'$$

EC 6.x.x.x, ligases, catalyse the synthesis of new bonds (C-O, C-S, C-N or C-C bonds) coupled with the breakdown of a pyrophosphate bond in ATP or other nucleoside triphosphate. The typical reaction is:

$$X + Y - ATP \rightleftharpoons XY + ADP + Pi$$

Every EC number is associated with a recommended name for the respective enzyme.

The EC number is often simultaneously employed as an identifier of reactions, enzymes and enzyme genes, linking metabolic and genomic information. The assignment of EC numbers to new enzymes is performed based on published experimental data that include the full characterization of the enzymes and their catalytic functions. Although chemically meaningful, and widespread, the EC rules are not always straightforward to apply, and often establish similarities and differences between chemical reactions that are biased towards their biological significance. [149] EC numbers are often problematic in practice when an enzyme catalyses more than one reaction, or when the same reaction is catalysed by different enzymes. [41] Different methods should be available to automatically compare metabolic reactions from their reaction formulas, independently of EC numbers. Such methods are mandatory for the comparison of reactions with incomplete official EC numbers, with EC numbers still not assigned, or with no EC number (if they are not catalysed by enzymes). Advantageously, classification of reactions should take into account physico-chemical features of reactants and products, as these affect reactivity and reaction mechanisms.

The MOLMAP method for numerically encoding the structural transformations resulting from a chemical reaction will be used in Chapter 7 and 8 to classify genome scale data sets of metabolic reactions.

4.4 ^{1}H NMR Data and Automatic Learning

4.4.1 Processing ^{1}H NMR Data by Automatic Learning Methods

Processing ^{1}H NMR data by automatic learning techniques is expanding the application of this spectroscopy to domains far beyond the classical use in structure elucidation. [150] Most notably, ^{1}H NMR spectroscopy is playing a central role in the emerging area of metabonomics, [151] in which complex multivariate data from biological samples are analyzed by machine learning and statistical methods, in order to detect certain metabolites and potential biomarkers or to assess disease pathology, drug efficacy, toxicological profiles, gene expression, and mode of action of bioactive compounds. [152–158] Lindon [159] reviewed applications of pattern recognition methods in biomedical magnetic resonance. These include unsupervised learning methods, such as principal component analysis, nonlinear mapping, and hierarchical cluster analysis, for reduction of data complexity, and supervised learning methods for sample classification, such as SIMCA, partial least-squares, linear discriminant analysis, and artificial neural networks.

Kohonen self-organizing maps (SOMs) [1,141] have been used to process ^{1}H NMR data in varied applications. Beckonert et al. [160] used ^{1}H NMR spectroscopy and SOMs to study metabolic changes in breast cancer tissue. Van et al. [161] distinguished individuals affected by interstitial cystitis and by bacterial cystitis from non affected individuals on the basis of mass spectrometry and ^{1}H NMR spectral patterns of urine. Bathen et al. [162] investigated SOMs for the classification of ^{1}H NMR spectra of the human blood plasma lipoprotein fractions from healthy volunteers and patients with cancer or coronary heart disease.

The usefulness of ^{1}H NMR spectroscopy in reaction and process monitoring has also been demonstrated; for example, in liquid-phase combinatorial synthesis, [163] estimation of velocities of enzyme-catalysed reactions, [164] or using on-line NMR spectroscopy. [165] Kalelkar et al. described a SOM analysis of ^{1}H NMR spectra from combinatorial parallel synthesis with the aim of identifying outliers in the libraries. [166] However, to the best of our knowledge, SOMs have not been investigated for reaction classification on the basis of NMR data.

4.4.2 ^{1}H NMR-Based Reaction Classification

The SPINUS program [167–169] for the estimation of ^{1}H NMR chemical shifts from the molecular structure allows linking a database of chemical reactions to the corresponding ^{1}H NMR data. The following Chapter explores the possibility of classifying chemical reactions by Kohonen SOMs and Random Forests (RFs), [70] taking as input the difference between the ^{1}H NMR spectra of the products and the reactants. The rationale behind this proposal is that the substructures of the reactants that are far from the reaction center mostly will have their chemical shifts unchanged, whereas the chemical shifts of the atoms near the reaction center should change with the reaction. The pattern of changes can be interpreted as a descriptor of the reaction. Such a representation additionally has the potential to encode 3D effects, even if related to substructures topologically distant from the reaction center, and can, in principle, be applied when more than one reaction occurs simultaneously. Clearly, ^{1}H NMR spectroscopy has its own limitations, particularly for reactions with a small number of hydrogen atoms in the neighborhood of the reaction center. At the same time, the use of ^{1}H NMR has considerable advantages in comparison to NMR of other nuclei, such as the speed and the amount of sample required.

Automatic analysis of changes in the ^{1}H NMR spectrum of a mixture and their interpretation in terms of chemical reactions taking place have a diversity of possible applications. The changes in the ^{1}H NMR spectrum of a stored chemical can be interpreted in terms of the chemical reactions responsible for degradation. A database of metabolic reactions linked to the corresponding NMR data can assist in the monitoring of a biotechnological process by ^{1}H NMR spectroscopy and yield information about the enzymatic reactions taking place, or the alterations in the spectrum of a biofluid can be related to changes in metabolic reactions.

Chapter 5

Linking Databases of Chemical Reactions to NMR Data

5.1 Introduction

The automatic classification of chemical reactions from the molecular structures of the reactants and products is currently a topic of high interest in chemoinformatics. Several methods for the representation and classification of reactions have been put forward, as it was showed in the previous Chapter. Most of these methods were based on codes of the reaction center or on physico-chemical properties of atoms and bonds at the reaction center and require atom-to-atom mapping and assignment of the atoms or bonds involved in the reaction.

In this Chapter, the classification of chemical reactions is explored taking as input the difference between the ^{1}H NMR spectra of the products and the reactants. The chemical shifts of the reactants and products were fuzzified to obtain a crude representation of the spectra. The possibility of inferring the type of reaction exclusively from ^{1}H NMR spectra was analysed. To address this problem, it is simulated a situation in which the structures of the reactants and products are unknown, but their ^{1}H NMR spectra are available. The SPINUS program [167–169] for the estimation of ^{1}H NMR chemical shifts from the molecular structure allows linking a database of chemical reactions to the corresponding ^{1}H NMR data.

Automatic analysis of changes in the ^{1}H NMR spectrum of a mixture and their interpretation in terms of chemical reactions taking place have a diversity of possible applications, from the monitoring of reaction processes or degradation of chemicals to metabonomics.

Two machine learning techniques, Kohonen SOMs and Random Forests, were investigated to classify photochemical and metabolic reactions from the ^{1}H NMR chemical shifts of the products and the reactants. These machine learning methods differ in the type of learning. Whereas Kohonen SOMs are trained with unsupervised learning (competitive learning), Random Forests are trained with supervised learning.

We first explored a data set of photochemical cycloadditions. This was partitioned into a training set of 147 reactions and a test set of 42 reactions, all manually classified into seven classes. The ^{1}H NMR spectra were simulated from the molecular structures by SPINUS, [167–169] and a reaction was represented by the difference between the spectrum of the product and the spectra of the reactants. After the predictive models for the classification of chemical reactions were established on the basis of simulated NMR data, their applicability to reaction data from mixed sources (experimental and simulated) was evaluated. The experiments with this data set yielded 86-93% of correct classifications for an independent test set of 42 reactions. The models were further validated with a test set combining experimental and simulated chemical shifts.

A second data set was also explored, consisting of 911 metabolic reactions catalysed by transferases (EC number 2.x.x.x) classified into eight subclasses according to the Enzyme Commission (E.C.) system. [147] Differently from the first data set, not all the reactions have the same number of reactants and products. Apart from that, automatic classification was investigated in the same way. With this data set, classification according to subclass (second digit of the EC number) could be achieved with up to 84% of accuracy.

The results support our proposal of linking databases of chemical reactions to NMR data for automatic reaction classification and show the usefulness of the predictions obtained by the SPINUS program for the estimation of missing NMR experimental data.

The next Section 5.2 contains the methodology and computational details. Section 5.3 discusses the results. Finally Section 5.4 presents the concluding remarks.

5.2 Methodology and Computational Details

The experiments here described involve two steps: the generation of a reaction descriptor from the simulated ^{1}H NMR spectra of the products and reactants and the development of predictive models for reaction classification using Kohonen self-organizing maps or Random Forests.

5.2.1 Data Sets of Reactions

A data set of 189 photochemical reactions, involving two reactants and one product (bearing at least one hydrogen atom covalently bonded to a carbon atom) was extracted from the SPRESI database (InfoChem GmbH, Munich, Germany). The reactions were manually assigned into seven classes: [3 + 2] photocycloaddition of azirines to C=C (class A, 20 reactions), [2 + 2] photocycloaddition of C=C to C=O (class B, 31 reactions), [2 + 2] photocycloaddition of C=N to C=C (class C, 8 reactions), [4 + 2] and [4 + 4]

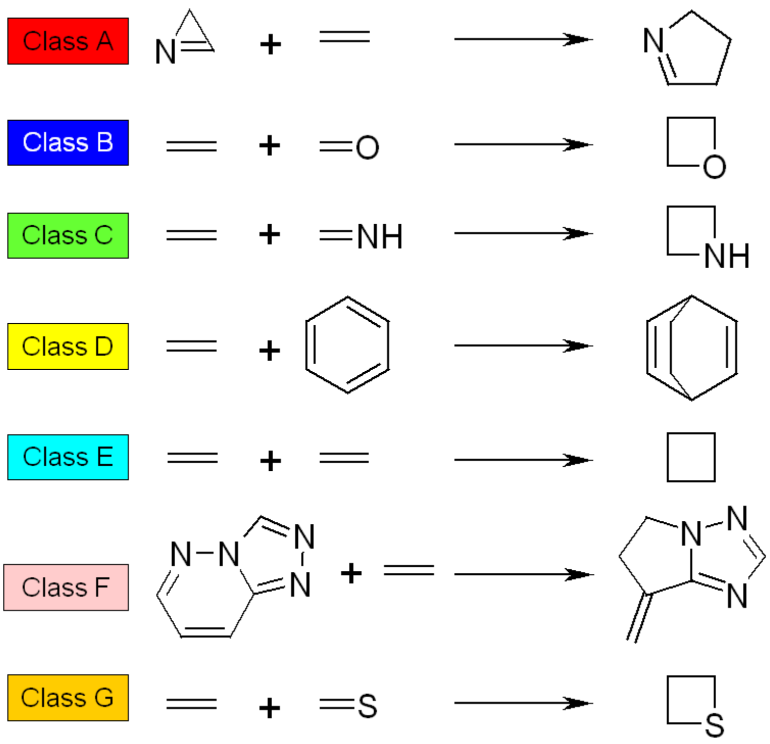


Figure 5.1: Classes of photochemical reactions.

photocycloaddition of olefins to carbon-only aromatic rings (class D, 20 reactions), [2 + 2] photocycloaddition of C=C to C=C (class E, 73 reactions), [3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (class F, 10 reactions), and [2 + 2] photocycloaddition of C=C to C=S (class G, 27 reactions). Scheme 1 summarizes the different types of reactions. The data set of 189 reactions were randomly partitioned into a training set of 147 reactions (16 of type A, 23 of type B, 7 of type C, 16 of type D, 56 of type E, 8 of type F, and 21 of type G) and a test set of 42 reactions (4 of type A, 8 of type B, 1 of type C, 4 of type D, 17 of type E, 2 of type F, and 6 of type G), assuring that both sets cover the whole range of reactions.

For validating the models with experimental ^{1}H NMR data, a subset of 26 reactions was assembled from the original set, with the simulated chemical shifts replaced by experimental values for some reactants or products. The chemical shifts were obtained from the SDBS database [170] or from the references associated with the reactions in the SPRESI database. In three reactions, experimental data was included for both reactants and the product; in 14 reactions, for only one reactant; in six reactions for only the product; in one reaction, for the product and one reactant; and in two reactions, for both reactants. The subset of 26 reactions (9 were from the test set and 17 from the training set) includes 3 reactions of class A, 3 of class B, 1 of class C, 4 of class D, 8 of class E, 2 of class F, and 5 of class G.

For the second application, a data set was assembled with metabolic reactions catalysed by transferases (EC number 2.x.x.x) extracted from the KEGG LIGAND database [39] of enzymatic reactions (release of January 2006). The selected data set excluded reactions listed with more than one EC number or an incomplete EC number. Reactions involving a compound not accepted by SPINUS, as well as unbalanced reactions, were also excluded. Reactions differing only in stereochemical features were considered as duplicates and were included only once. The cleaning procedure was based on chemical hashed fingerprints generated by JChem package, version 3.1.7.1 (ChemAxon, Budapest, Hungary, www.chemaxon.com) with a length of 64 bytes, a maximum number of five bonds in patterns, and two bits switched on for each pattern in the structure. The data set finally consisted of 911 reactions catalysed by transferases (EC number 2.x.x.x) belonging to eight different subclasses (second digit of the EC number): 133 reactions transferring one-carbon groups (EC 2.1.x.x, here labeled with “class A”), 9 reactions transferring aldehyde or ketonic groups (EC 2.2.x.x, class B), 171 reactions catalysed by acyltransferases (EC 2.3.x.x, class C), 201 reactions catalysed by glycosyltransferases (EC 2.4.x.x, class D), 41 reactions transferring aryl or alkyl groups other than methyl (EC 2.5.x.x, class E), 75 reactions transferring nitrogenous groups (EC 2.6.x.x, class F), 259 reactions transferring phosphorus-containing groups (EC 2.7.x.x, class G), and 22 reactions transferring sulfur-containing groups (EC 2.8.x.x, class H). The data set of 911 reactions was partitioned into training and test sets using a 18×18 Kohonen SOM. The SOM was trained with all reactions, and after training, one reaction was randomly selected from each occupied neuron and moved to the test set. With this procedure, a test set resulted with 262 reactions (40 of subclass 2.1.x.x, 3 of subclass 2.2.x.x, 60 of subclass 2.3.x.x, 62 of subclass 2.4.x.x, 15 of subclass 2.5.x.x, 24 of subclass 2.6.x.x, 52 of subclass 2.7.x.x, and 6 of subclass 2.8.x.x). The training set consisted of the remaining 649 reactions (93 of subclass 2.1.x.x, 6 of subclass 2.2.x.x, 111 of subclass 2.3.x.x, 139 of subclass 2.4.x.x, 26 of subclass 2.5.x.x, 51 of subclass 2.6.x.x, 207 of subclass 2.7.x.x, and 16 of subclass 2.8.x.x).

5.2.2 Simulation of ^{1}H NMR Spectra with SPINUS

The SPINUS program [167–169] was used for the estimation of ^{1}H NMR chemical shifts from the molecular structures. In these simulations, only hydrogen atoms bonded to carbon atoms were predicted. A crude representation of the spectrum was obtained by fuzzifying the predicted chemical shifts with a triangular function. The triangular function was used with widths 0.2, 0.3, and 0.4 ppm at each side of the chemical shift, which approximate the observed mean absolute error of SPINUS predictions (0.2-0.3 ppm). [167] No information concerning coupling constants was used.

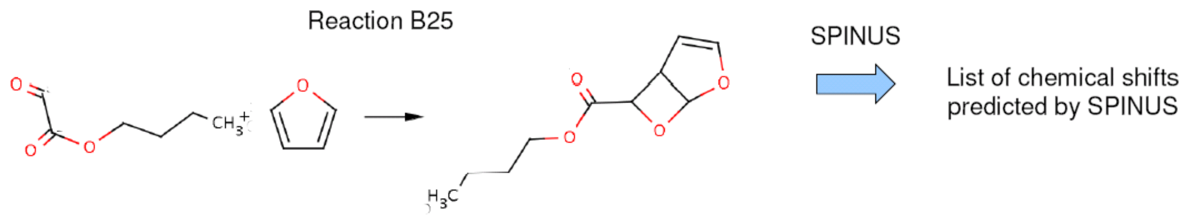


Figure 5.2: Example of generation of a ^{1}H NMR reaction descriptor (step 1).

5.2.3 ^{1}H NMR-Based Descriptor of Chemical Reactions

The generation of the ^{1}H NMR-based reaction descriptor can be summarized in the following steps:

1. Prediction of the ^{1}H NMR chemical shifts from the molecular structures of the reactants and products of the reaction using the SPINUS program (Figure 5.2).

2. Fuzzification of the predicted chemical shifts with a triangular function to obtain a crude representation of the spectrum. The width used by the triangular function depends on the experiment and on the observed mean absolute error of SPINUS predictions (0.2-0.3 ppm). [167] (first part of Figure 5.3)

3. All the signals, integrating proportionally to the number of protons, arising from all reactants of one reaction were taken together (a simulated spectrum of the pseudo-mixture of reactants). The same was done for products if necessary. (second part of Figure 5.3)

4. The simulated spectrum of the pseudo-mixture of reactants were subtracted from the simulated spectrum of the pseudo-mixture of products. (Figure 5.4)

5. The difference spectrum is used as the representation of the chemical reaction. The difference-spectrum, covering the range of 0-12 ppm, was converted into a 120-positions code, each position integrating the intensities within an interval of 0.1 ppm. (Figure 5.4)

6. The data set of chemical reactions represented with the ^{1}H NMR-based reaction descriptor is submitted to the automatic learning technique to perform the training of the model. Figure 5.4 illustrates the situation where the machine learning method used is a Kohonen Self-Organizing Map.

In the generation of the ^{1}H NMR-based reaction descriptor no information concerning coupling constants was used.

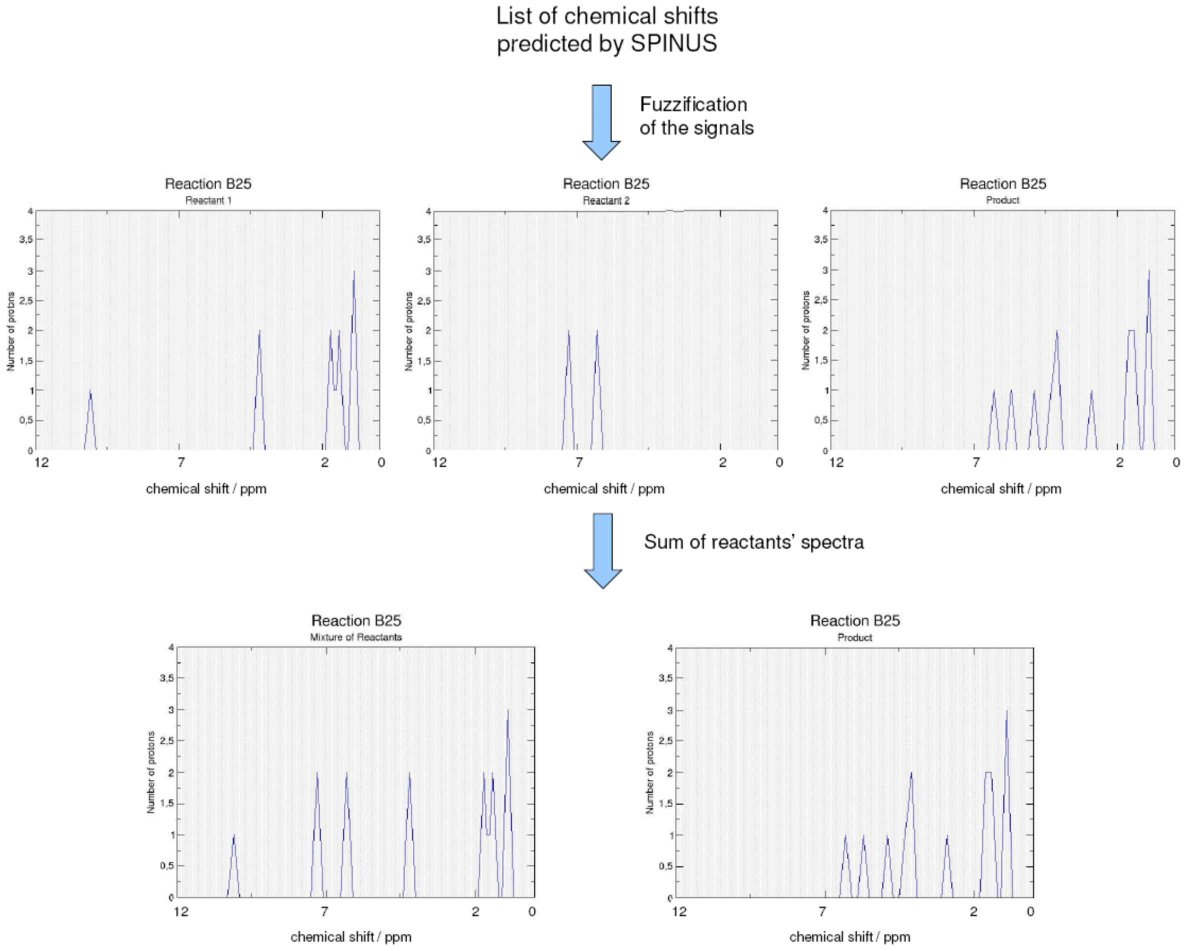


Figure 5.3: Example of generation of a ^{1}H NMR reaction descriptor (step 2 and 3).

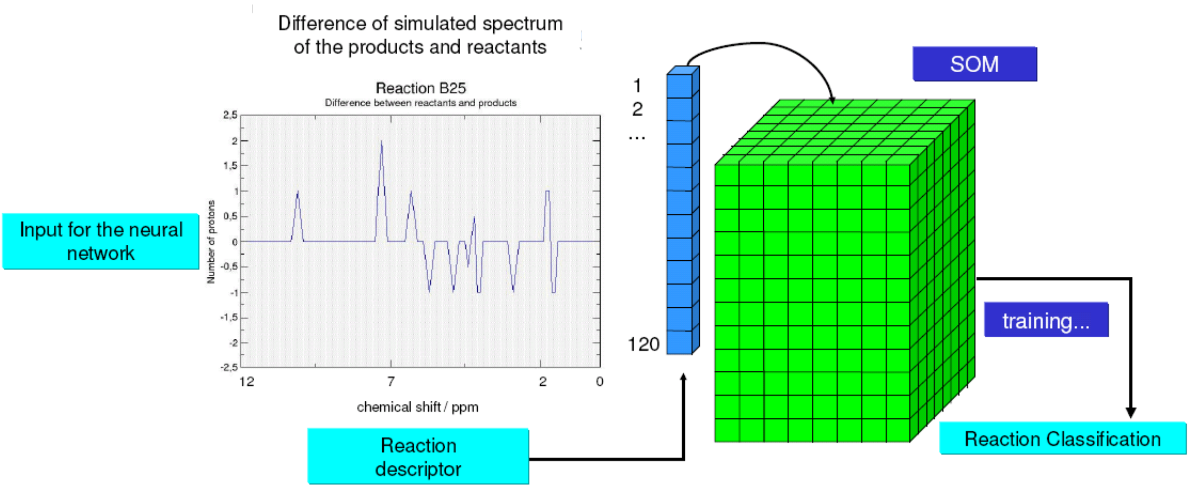


Figure 5.4: Example of generation of a ^{1}H NMR reaction descriptor (step 4-6).

In the case of photochemical reactions, all reactions have two reactants and one product, but in the data set of metabolic reactions, the number of reactants and products varies among the reactions.

5.2.4 Kohonen Self-Organizing Maps

A Kohonen self-organizing map [1, 141] distributes objects over a 2D surface (a grid of neurons) in such a way that objects bearing similar features are mapped onto the same or adjacent neurons. SOMs perform a nonlinear projection of multidimensional objects onto a two-dimensional surface, yielding maps of easy visual interpretation. The full description of this method including details about the learning procedure could be found in Section 2.6. The input data are stored in the two-dimensional grid of neurons, each neuron containing as many elements (weights) as there are input variables. In the investigations described in this Chapter, the input variables are the above-mentioned 120 reaction descriptors derived from ^{1}H NMR spectra. We trained SOMs with a diversity of reactions to investigate the ability of the ^{1}H NMR descriptors to cluster and classify chemical reactions. SOMs with toroidal topology and sizes varying between 13×13 and 15×15 for photochemical reactions and 25×25 or 29×29 for metabolic reactions were trained with the training set and tested with the test set. The SOMs learn in an unsupervised way, which means that the distribution of objects on the map relies on only the NMR data, not on the preassigned classes. After the training, each neuron (a position of the map) was assigned to a class, depending on the reactions that were mapped into it, or into its neighbors if it was empty. If a winning class could not be identified for a neuron, the neuron was classified as undecided.

Training of the SOMs was performed by using a linear decreasing triangular scaling function with an initial learning rate of 0.1. The weights were initialized with random numbers that are calculated using the mean and the standard deviation of each variable in the input data set as parameters. For the selection of the winning neuron, the minimum Euclidean distance between the input vector and neuron weights was used. The training was performed over 50-100 cycles, with the learning span and the learning rate linearly decreasing until 0. The Kohonen SOM was implemented with in-house-developed software based on JATOON Java applets. [171] To overcome fluctuations induced by the random factors influencing the training, five independent SOMs were trained with the same objects, generating an ensemble of SOMs, and the final classifications were obtained by majority voting of the individual maps.

5.2.5 Random Forests

A random forest [70, 71] is an ensemble of unpruned classification trees created by using bootstrap samples of the training data and random subsets of variables to define the best split at each node. It is a high-dimensional nonparametric method that works well on large numbers of variables. The predictions are made by majority voting of the individual trees. The details about Random Forests could be found in Chapter 3. The performance is internally assessed with the prediction error for the objects left out in the bootstrap procedure. In this work, RFs were grown with the R program version 2.0.1, [172] using the randomForest library, [173] and were used to classify the reactions from the NMR reaction descriptor, the difference between the ^{1}H NMR spectra of the product and reactants. The number of trees in the forest was set to 1000, and the number of variables tested for each split was set to default (square root of the number of variables).

5.3 Results and Discussion

5.3.1 Validation of the Chemical Shifts Predicted by SPINUS

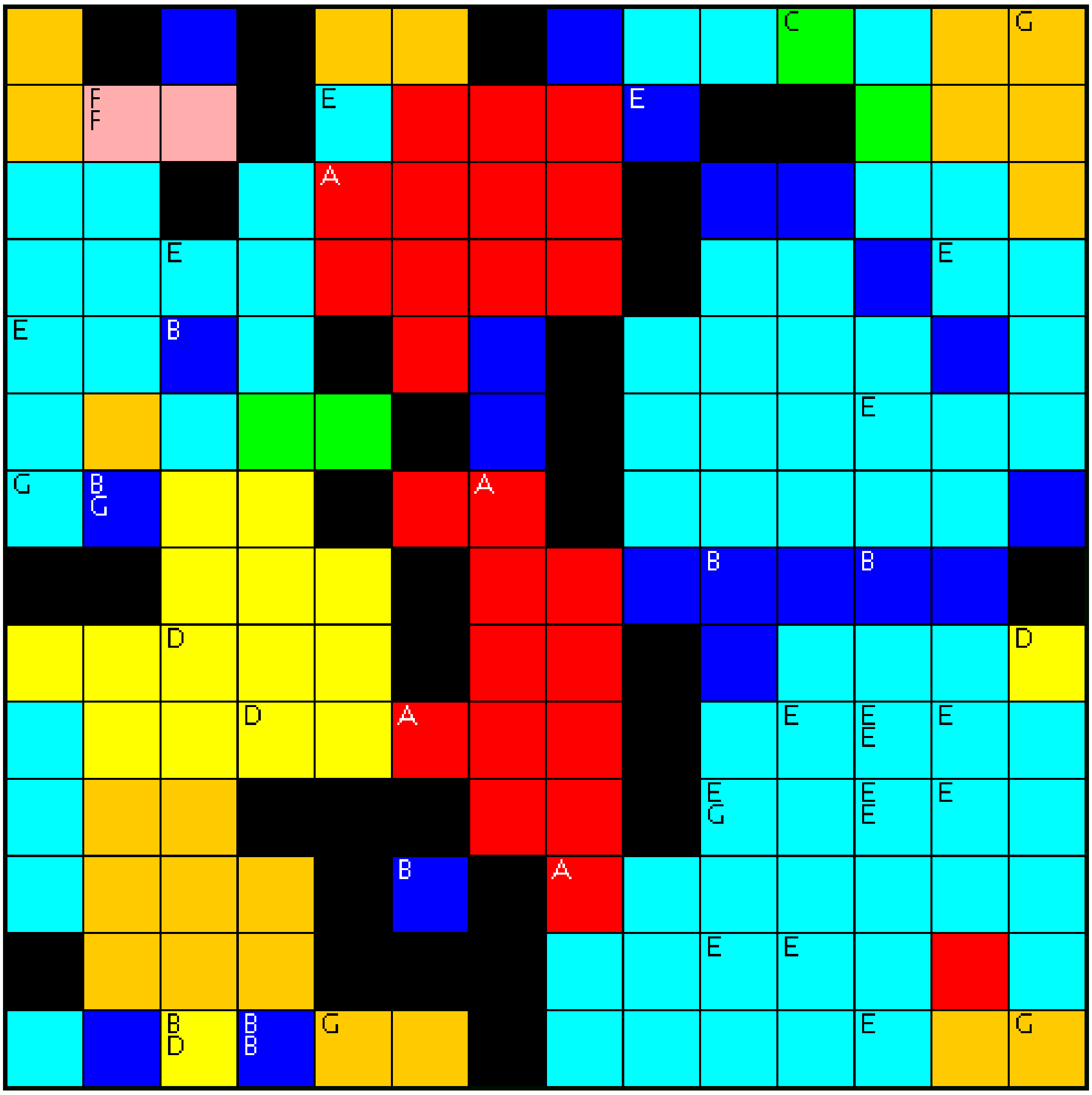
Although SPINUS has been tested with large, diverse data sets, [168, 174] we validated the predicted chemical shifts for some specific types of structures involved in this study by comparing predictions with experimental chemical shifts for some reactants and products in our data set of photochemical reactions. The experimental chemical shifts were obtained from the SDBS database [170] or from the references associated with the reactions in the SPRESI database.

Comparisons were performed for 349 chemical shifts from 36 molecular structures covering reactants and products of all classes. Assignment of the experimental chemical shifts to individual protons was done on the basis of similarities between the predictions and experimental values. This is acceptable for this study, since the global accordance between simulated and experimental spectra is here more relevant than the accuracy of predictions for individual protons. A mean absolute error (MAE) of 0.24 ppm was obtained for the 349 chemical shifts, which is similar to the results of previous tests. [168, 174] Table 5.1 details the results by class of reaction.

The predictions were particularly accurate for compounds of reaction classes A, D, and G (MAE of 0.16, 0.18, and 0.21 respectively). A MAE lower than 0.2 ppm was achieved for 60% of the cases.

5.3.2 Classification of Photochemical Reactions from ^{1}H NMR Data

SOMs of different sizes were trained with the reactions of the training set using different fuzzification parameters. The (toroidal) surface of a SOM is illustrated in Figure 5.5, where a trend of some classes to form well-defined clusters is clear. It is to point out that the learning method does not use the information about the classes of the reactions during



🟥 [3 + 2] photocycloaddition of azirines to C=C

🟦 [2 + 2] photocycloaddition of C=C to C=O

🟩 [2 + 2] photocycloaddition of C=N to C=C

🟨 [4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings

▢ [2 + 2] photocycloaddition of C=C to C=C

🟪 [3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C

🟧 [2 + 2] photocycloaddition of C=C to C=S

Figure 5.5: Toroidal surface of a 14×14 Kohonen self-organizing map trained with the ^{1}H NMR-based descriptor for photochemical reactions belonging to classes A-G. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors. The reactions of the test set were mapped onto the trained SOM and are represented by the label of their true classes. A fuzzification parameter of 0.2 ppm was used in this experiment for the simulation of NMR spectra.

Table 5.1: Accuracy of the estimated ^{1}H NMR chemical shifts in the validation set for photochemical reactions.

Class	Compounds		Chemical	MAE/	AE / ppm				
	React.	Prod.	Shifts	ppm	0-0.1	0.1-0.2	0.2-0.3	0.3-0.4	>0.5
A	5	1	59	0.16	24 (41%)	12 (20%)	13 (22%)	7 (12%)	3 (3%)
B	2	2	44	0.31	17 (39%)	4 (9%)	3 (7%)	5 (11%)	15 (34%)
C	-	1	10	0.29	3 (30%)	2 (20%)	3 (30%)	0 (0%)	2 (20%)
D	5	1	40	0.18	12 (30%)	20 (50%)	2 (5%)	1 (2.5%)	5 (12.5%)
E	10	2	91	0.31	37 (41%)	14 (15%)	11 (12%)	9 (10%)	20 (22%)
F	-	2	20	0.22	3 (15%)	9 (45%)	4 (20%)	1 (5%)	3 (15%)
G	2	3	85	0.21	44 (52%)	14 (16%)	11 (13%)	3 (4%)	13 (15%)
Total	24	12	349	0.24	140 (40%)	75 (21%)	47 (13%)	26 (7%)	61 (17%)

MAE - Mean Absolute Error; AE - Absolute Error

the training (unsupervised learning). The reactions of the test set were mapped onto the same surface to illustrate the ability of the map to classify unseen reactions. Class A as well as classes C and D clusters into two regions, and inspection of the reactions activating each region showed that they mainly correspond to different types of substrates. Class B is scattered through several regions of the map, generally in the vicinity of class E. Class G clusters relatively well with the training set, although the predictions for the test set were correct for only one-half of the cases, even though two of these wrongly classified reactions activated neurons in the neighborhood of a G neuron. The only wrongly classified reaction of class E activated a neuron assigned to class B that was empty with the training set and that is a neighbor of an E neuron. The wrongly classified reaction of class B (B1 in Scheme 2) was classified as D. Inspection of the reactions in the training set that hit the same neuron revealed reactions D2 and D3, which have reactants that are very similar to B1 (Figure 5.6). In the test set, another reaction with the same reactants as B1 but giving a different product (D1) also activated this neuron. This shows how the method is sensitive not only to the reaction center but also to the structural environment and, thus, to the structure of the reactants and products.

The percentage of correct classifications obtained for the training and test sets by SOMs are presented in Table 5.2. Correct predictions could be achieved for 94-99% of the training set and for 81-88% of the test set. The size of the network and the fuzzification parameter have not exhibited a significant influence on the prediction ability.

Consensus predictions involving an ensemble of five independent SOMs considerably improved the predictions both for the training and the test sets (see Table 5.2). For the training set, only one reaction was always wrongly classified: this is reaction D4 of Figure that has the same reactants as reaction E1 and was mapped into the same neuron. Into

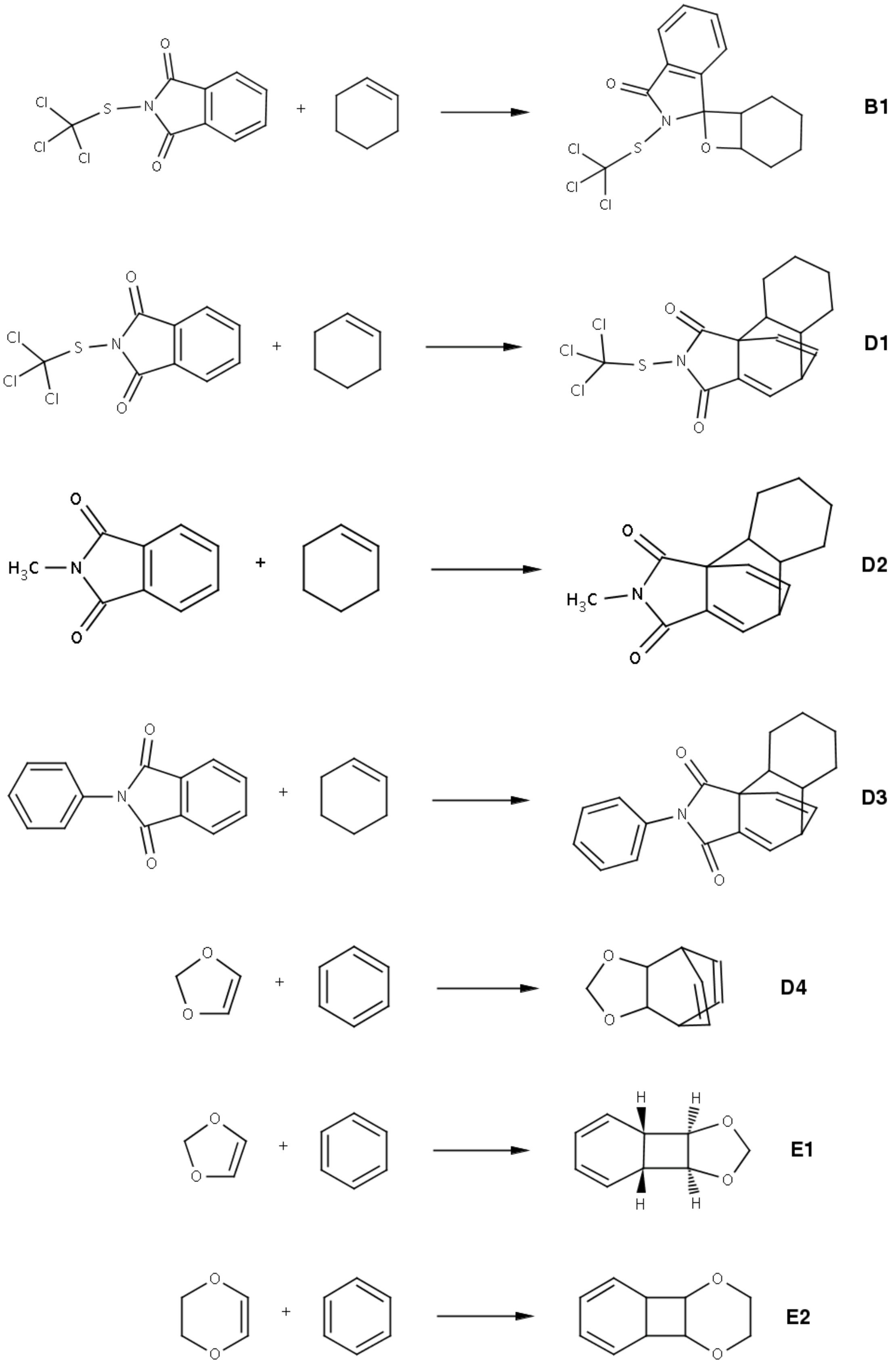


Figure 5.6: Examples of pairs of reactions with the same or similar reactants and yielding different products.

Table 5.2: Classification of photochemical reactions by Kohonen SOMs and Random Forests from NMR data using different fuzzification parameters.

Machine Learning	Fuzzification width	% Correct Predictions[a]	
Method	/ ppm	Training set	Test set
SOM 13×13	0.2	99 (97, 96, 98, 97, 96)	88 (76, 79, 81, 76, 81)
	0.3	99 (96, 91, 93, 97, 97)	86 (76, 83, 81, 76, 86)
	0.4	99 (95, 96, 97, 95, 97)	86 (76, 83, 81, 86, 83)
SOM 14×14	0.2	99 (98, 95, 96, 97, 95)	93 (79, 79, 81, 83, 88)
	0.3	99 (96, 92, 94, 98, 96)	86 (76, 81, 86, 76, 76)
	0.4	99 (97, 98, 97, 97, 95)	86 (86, 81, 81, 81, 81)
SOM 15×15	0.2	99 (96, 97, 98, 99, 95)	93 (81, 81, 81, 81, 86)
	0.3	99 (95, 99, 99, 97, 98)	88 (79, 81, 79, 81, 74)
	0.4	99 (99, 95, 97, 97, 97)	93 (83, 81, 86, 86, 86)
RF	0.2	89[b]	81
	0.3	85[b]	81
	0.4	85[b]	83
RF (classes A-D, F, G, repeated)	0.2	-	86
	0.3	-	83
	0.4	-	83

[a] In the SOMs rows, the results were obtained by consensus prediction from an ensemble of five SOMs (within the parentheses are the results for the five individual SOMs). [b] Predictions obtained by out-of-bag estimation (internal cross-validation).

this neuron was also mapped reaction E2 (Figure 5.6). Predictions for the test set reached up to 93% accuracy. With the ensemble of SOMs of dimension 14×14 and fuzzification parameter 0.2 ppm, only one reaction of the test set was wrongly classified (reaction B1 of Figure 5.6), and two reactions were undecided (one of class A and one of class G). This ensemble could, thus, overcome most of the problems associated with class G in individual SOMs.

After experimenting with unsupervised learning, we investigated the performance of Random Forests, which learn in a supervised way. The obtained results are also shown in Table 5.2. For the training set, the displayed results were obtained in internal cross-validation tests, which rely on the bootstrap procedure employed by the Random Forests: out-of-bag estimation. Consistency was observed between these results and those obtained for the test set. Most of the wrong predictions were false E. Because class E is the most populated class in the training set, with 38% of the reactions, it was decided to balance the training set by including the reactions of the classes B and G twice, reactions of classes A and D three times, reactions of class F six times and reactions of class C seven times. A slight improvement of the prediction ability for the test set could be achieved, particularly for the experiment with fuzzification parameter 0.2 ppm. In this case, six reactions of the test set were wrongly classified: one of class B, one of class E, and four of class G. Most of the misclassifications were again reactions wrongly classified as class E. Significantly, the problematic reactions of class G (consistently misclassified by SOMs and RFs) have no hydrogen atoms bonded to the atoms of the reaction center.

The voting system of a RF enables the association of a probability to each prediction, corresponding to the proportion of votes obtained by the winning class and by the other classes. For the test set, 29 out of 42 reactions (69%) were predicted with a probability higher than 0.5, and all were correctly predicted.

5.3.3 Validation of the Classification Models with Experimental Chemical Shifts

To assess if the models trained with simulated data could be applied to experimental data, predictions were obtained for a data set of 26 photochemical reactions in which the simulated chemical shifts for some reactants or products were replaced by experimental values (see Methodology). Both the ensemble of SOMs and the RF were applied (see Table 5.3). The RF exhibits a higher robustness than the ensemble of SOMs. With the SOMs, the accuracy of the classifications consistently degrades with the inclusion of experimental data in comparison with the experiments based exclusively on simulated data for the same subset of reactions. On the contrary, the accuracy of the RF predictions is quite stable; it is globally the same for simulated data and for the combination of simulated and experimental data.

Table 5.3: Classification of 26 reactions on the basis of experimental and simulated NMR data combined or on the basis of simulated data alone by the ensemble of five Kohonen SOMs and by the Random Forest trained with simulated data.

Machine Learning	Fuzzification width	% Correct predictions	
Method	/ ppm	Experimental + simulated data	Simulated data
SOM 13×13	0.2	81	100
	0.3	77	96
	0.4	81	92
SOM 14×14	0.2	77	100
	0.3	73	96
	0.4	85	96
SOM 15×15	0.2	77	100
	0.3	77	96
	0.4	77	96
RF	0.2	81	85[a]
	0.3	81	77[a]
	0.4	92	85[a]

[a] For the reactions included in the training set, the predictions are from internal cross-validation obtained by out-of-bag estimation.

5.3.4 Classification of Metabolic Reactions from ^{1}H NMR Data

The concept of ^{1}H NMR-based classification of reactions was then explored with a larger, more complex data set consisting of metabolic reactions catalysed by transferases. SOMs were trained with a data set of metabolic reactions on the basis of their ^{1}H NMR descriptor, and neurons were assigned to classes at the end of the training. Figure 5.7 shows the resulting surface for such a SOM, each neuron colored according to the Enzyme Commission subclass of the reactions activating it, that is, the second digit of the EC number. On the same surface are displayed the second and third digits of the EC numbers corresponding to the reactions hitting each neuron.

The map reveals a remarkable clustering of the reactions according to EC numbers, exclusively from the NMR chemical shift-derived descriptors. In a few cases, subregions of the same color (same first three digits of the EC number) could be interpreted in terms of structural features of the reactants or products. For example, reactions with EC numbers 2.6.1 (transaminases, in pink) clustered into two main regions, one around neuron C7, and the other around neuron N1. We observed that the first cluster corresponds to reactions involving transformation of oxoglutaric acid into glutamic acid, whereas the second is associated with transformations of pyruvic acid into alanine.

Another example is the region spanned by subclass 2.4 (glycosyltransferases, in dark yellow), where separation is observed for EC numbers 2.4.1 (hexosyltransferases), 2.4.2 (pentosyltransferases, around neuron AA29), and 2.4.99 (transferring other glycosyl groups, around neurons C13 and AB12). In addition, subclass 2.5 (light blue) is clustered into three regions. Reactions activating neurons around position S1 transfer alkyl groups from (S)-adenosyl-L-methionine, the cluster around neuron A21 corresponds to reactions transferring alkyl groups from phosphorylated compounds, and the region centered on neuron AB9 was typically activated by O-acetyl- or O-succinyl-L-homoserine lyases.

In some cases, odd mapping of reactions can be interpreted from their molecular structures. For example, two reactions of subclass 2.7 were mapped on neuron X15, in the middle of the region assigned to subclass 2.4. These two reactions (EC 2.7.7.8) involve the cleavage of a phosphoester bond in RNA and release of a nucleotide. Significantly, cleavage of a sugar-phosphate bond is not typical of subclass 2.7, but is common in the reactions of subclass 2.4 mapped on this region.

Inspection of the map also reveals limitations of the method. Neuron P21 was activated by reactions producing no changes in the ^{1}H NMR spectra. Reactions from different subclasses were mapped on this neuron and its neighborhood, illustrating an expected limitation: when there are no hydrogen atoms in the neighborhood of the reaction center or when the spectra of the products are no different from those of the reactants, the reactions cannot be distinguished or even described. Furthermore, in this study, only chemical shifts of hydrogen atoms bonded to carbon atoms were considered. Examples of

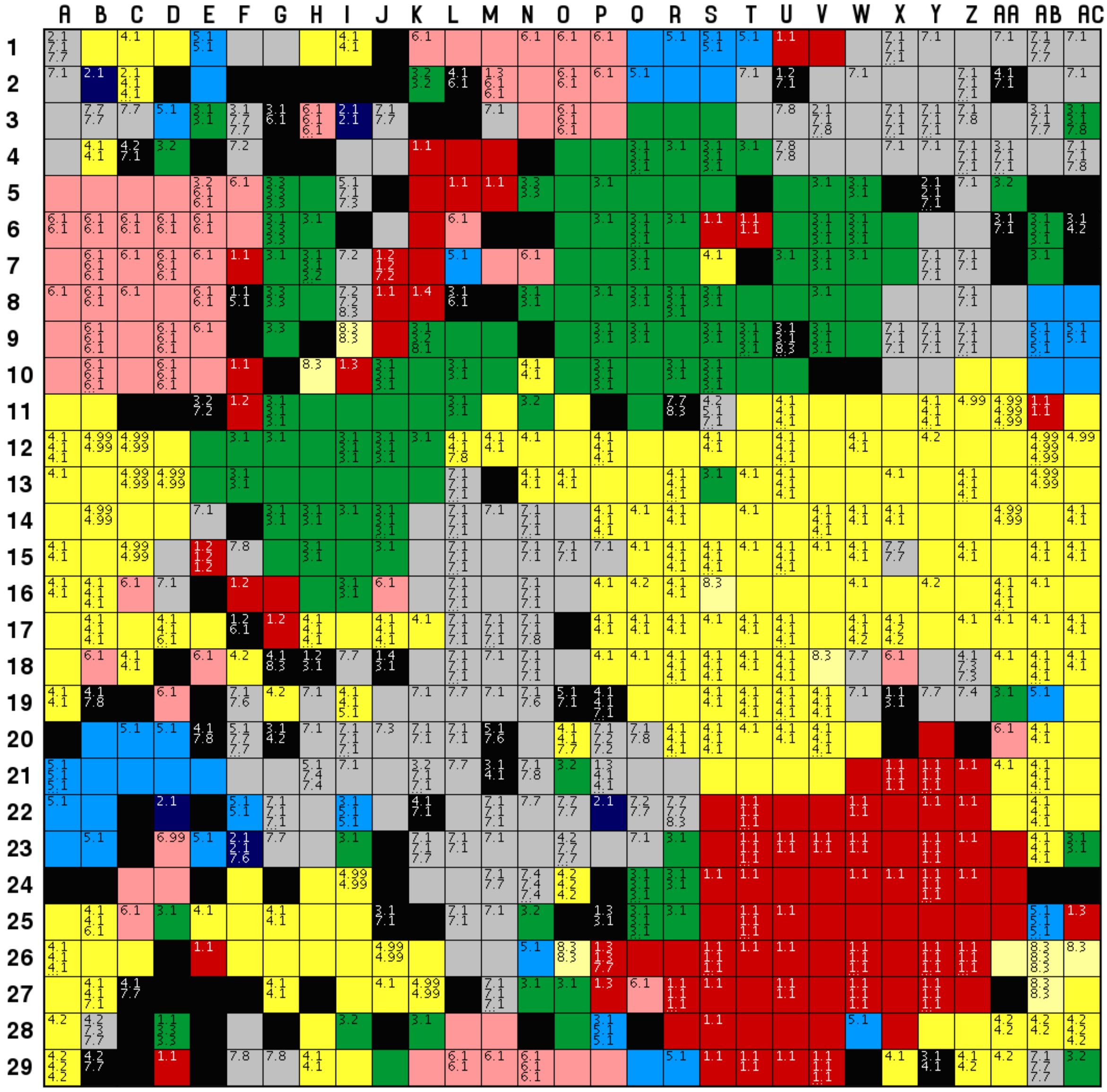


Figure 5.7: Toroidal surface of a 29×29 Kohonen self-organizing map trained with metabolic reactions belonging to EC subclasses 2.1-2.8 on the basis of the ^{1}H NMR descriptor. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors. The second and third digits of the EC numbers corresponding to the reactions of the training set are displayed on the neurons they activated. A fuzzification parameter of 0.2 ppm was used for the simulation of NMR spectra.

reactions hitting neuron P21 or its neighbors are

$$alanine + 2 - oxo\,acid \rightarrow pyruvate + amino\,acid\,(EC\,2.6.1.12)$$

$$ATP + formate \rightarrow ADP + formyl\,phosphate\,(EC\,2.7.2.6)$$

$$carbamoyl\,phosphate + oxamate \rightarrow ortophosphate + oxalureate\,(EC\,2.1.3.5)$$

In a series of reactions of the same type, particular features of the spectra can sometimes emphasize differences between substrates over common aspects. This is illustrated by the subclass 2.8. In this study, all reactions of subclass 2.8 belong to sub-subclass 2.8.3 (CoA-transferases) and can be formally written as

$$CoA - S - C(=O) - R^1 + R^2CO_2H \rightarrow CoA - S - C(=O) - R^2 + R^1CO_2H$$

Only slight changes in the ^{1}H NMR spectra result from these reactions, and we observed that the structure of R^1 determined the mapping; reactions with R^1=CH$_3$ clustered around neuron AB26 with the others scattered through different regions.

Several Kohonen SOMs, as well as Random Forests, were trained to classify reactions according to the second digit of the EC number, and their performances were quantitatively evaluated. Table 5.5 shows the percentage of correct classifications obtained for training and test sets by Kohonen SOMs of sizes 25×25 and 29×29, and by RFs. A fuzzification parameter of 0.2 ppm was used. In some experiments, the models were trained with the whole data set; in others, the data set was first partitioned into a training set and a test set.

With individual SOMs of size 25×25 and 29×29, correct predictions were achieved for 86% and 89-90%, respectively, of the entire data set of metabolic reactions. Ensembles of five networks improved the results up to 94 and 96% of correct predictions for SOMs of size 25×25 and 29×29, respectively. The test set was predicted with 66-67% of accuracy by individual SOMs, and ensembles of five SOMs achieved up to 73% of correct predictions. SOMs of size 29×29 yielded slightly better results than SOMs of size 25×25. Tables 5.5 and 5.6 show the confusion matrices obtained for the test set using the best individual SOM and the ensemble of five SOMs of size 29×29.

The classification performance was worse for subclasses 2.2 and 2.5. These subclasses are subrepresented in the data set, and most of the misclassifications were as false subclass 2.7.

Employing a random forest as the machine learning method, the predictions for the

Table 5.4: Classification of metabolic reactions by Kohonen SOMs and Random Forests.

Machine Learning	N. Reactions,	% Correct Predictions[a]	
Method	Training Set	Training Set	Test Set (262 reactions)
SOM	911	94 (86, 86, 86, 86, 86)	-
25×25	649	96 (89, 89, 91, 89, 88)	73 (67, 66, 66, 66, 66)
SOM	911	96 (89, 89, 89, 90, 89)	-
29×29	649	97 (92, 93, 91, 92, 89)	75 (68, 65, 66, 68, 68)
Random	911	84[b]	-
Forests	649	84[b]	79

[a]In the SOMs rows, the results were obtained by consensus prediction from an ensemble of five SOMs (within the parentheses are the results for the five individual SOMs). [b] Internal cross-validation obtained by out-of-bag estimation.

Table 5.5: Confusion matrix for the classification of metabolic reactions in the test set by the best 29 × 29 Kohonen SOM.

	2.1.	2.2.	2.3.	2.4.	2.5.	2.6.	2.7.	2.8.	X[a]	% Correct Predictions
2.1.	22	-	1	1	1	1	3	-	4	67
2.2.	-	1	-	-	-	-	2	-	1	25
2.3.	-	-	35	-	-	-	9	1	5	70
2.4.	1	-	-	80	-	-	4	-	6	88
2.5.	-	-	1	-	4	-	5	-	3	31
2.6.	-	-	-	1	1	20	1	-	1	83
2.7.	-	-	-	2	-	-	46	-	6	85
2.8.	-	-	-	-	-	-	-	3	-	100

Table 5.6: Confusion matrix for the classification of metabolic reactions in the test set by an ensemble of five 29 × 29 Kohonen SOMs.

	2.1.	2.2.	2.3.	2.4.	2.5.	2.6.	2.7.	2.8.	X[a]	% Correct Predictions
2.1.	24	-	2	-	1	2	1	-	3	73
2.2.	-	-	-	-	-	-	3	-	1	0
2.3.	-	-	36	-	1	-	7	1	5	72
2.4.	-	-	-	85	-	-	5	-	1	93
2.5.	-	-	-	2	6	-	4	-	1	46
2.6.	-	-	-	-	-	20	2	1	1	83
2.7.	1	-	-	3	-	-	50	-	-	93
2.8.	-	-	-	-	-	-	-	3	-	100

Table 5.7: Confusion matrix for the classification of metabolic reactions in the test set by a Random Forest.

	2.1.	2.2.	2.3.	2.4.	2.5.	2.6.	2.7.	2.8.	% Correct Predictions
2.1.	23	-	2	1	-	-	6	1	70
2.2.	-	2	-	-	-	-	2	-	50
2.3.	-	-	43	1	-	-	6	-	86
2.4.	-	-	-	88	-	-	3	-	97
2.5.	-	-	1	2	4	1	5	-	31
2.6.	-	-	2	1	-	20	1	-	83
2.7.	1	-	-	3	-	-	50	-	93
2.8.	-	-	-	-	-	-	-	3	100

test set achieved 79% of correct classifications (see Table 5.4). This result is consistent with the out-of-bag estimation for the entire data set and for the training set (84% of correct classifications). Again, the subclasses with fewer reactions (2.2 and 2.5) were more difficult to classify, and most of the misclassifications were reactions wrongly classified as subclass 2.7. The confusion matrix for the random forest prediction is available as Table 5.7.

The probability associated with each prediction by the RF was again meaningful. In the test set, 165 out of 262 reactions (62%) were predicted with a probability higher than 0.5, and only 7 of these were wrongly classified (4.2%).

The consistent large number of false subclass 2.7 classifications can be explained in part by the very small differences in the ^{1}H NMR spectra of reactants/products predicted by SPINUS for several reactions of subclass 2.7. This was already observed in the SOM of Figure 5.7, where a number of reactions from different subclasses hit neuron P21 due to their exhibiting null or almost null differences between the spectra of the reactants and products, even though neuron P21 was assigned to subclass 2.7, because most of the reactions by which it was activated belong to that subclass.

5.4 Conclusions

Automatic classification of chemical reactions from differences between ^{1}H NMR spectra of reactants and products was demonstrated with a high level of accuracy for a data set of photochemical reactions and for a data set of transferase enzymatic reactions. Ensembles of Kohonen SOMs with consensus predictions allowed for improvement of the results in comparison to individual SOMs. Random forests exceeded SOMs for the metabolic reactions, but not for the photochemical reactions. RFs also allowed associating a meaningful probability to each classification. With the data set of photochemical reactions, the SOM and RF models were tested with a subset of reactions for which experimental and simulated chemical shifts were combined. Classifications of the same overall quality as those obtained with simulated values alone were obtained by the RF, but SOMs were less robust against the inclusion of experimental data.

The approach is limited by the availability of hydrogen atoms in the neighborhood of the reaction center and by the sensitivity of their chemical shifts to the changes resulting from the reaction. The results support our proposal of linking reaction and NMR data for automatic reaction classification. They also show the usefulness of SPINUS predictions of NMR data in that context for the generation of training sets and for the estimation of missing experimental data.

Inference of reaction type from NMR experiments can have an application in the assessment of enzymatic function and in the development of biotechnological processes. Classification of the changes in the NMR spectrum of a mixture containing an enzyme and a pool of potential substrates can assist in the exploration of the catalytic ability of the enzyme as well as its versatility.

In most practical situations, a reaction is accompanied by side reactions, which would require that the NMR interpretation system is able to identify a reaction even in the presence of a mixture of reactions. Preliminary results using mixtures of reactions to train automatic learning methods indicate that it is possible to infer the reaction type when more than one reaction occur simultaneously. This results will be presented in the next Chapter.

Acknowledgements

The authors of the work presented in this Chapter thank InfoChem GmbH (Munich, Germany) for sharing the data set of photochemical reactions from the SPRESI database. The assistance of Dr. Yuri Binev with the simulation of NMR data is gratefully acknowledged.

Chapter 6

Classification of Mixtures of Reactions from ^{1}H NMR Data

6.1 Introduction

Following the work developed in the last Chapter, the ^{1}H NMR-based reaction descriptor was applied for the classification of mixtures of reactions. Kohonen Self-Organizing maps, Counter Propagation Neural Networks, Feed-Forward Neural Networks and Random Forests were trained for the automatic classification of mixtures of reactions from ^{1}H NMR spectra of the products and reactants.

As in the last Chapter, the classification of mixtures of chemical reactions is explored taking as input the difference between the ^{1}H NMR spectra of the products and the reactants, but now the products of two reactions of different classes are taken together, as well as the reactants. The SPINUS program [167–169] for the estimation of ^{1}H NMR chemical shifts from the molecular structure was used to simulate the ^{1}H NMR data. Four different automatic learning methods were used, which differ in the type of learning. Whereas Kohonen SOMs and CPNNs are trained with unsupervised learning (competitive learning), FFNNs and RFs are trained with supervised learning.

A subset of the photochemical cycloadditions used in the last Chapter was used to simulate the data set of mixtures.

The main objective is to predict the classes of the occurring reactions, from the simulated ^{1}H NMR spectra of the reactants and products in a mixture with two reactions taking place simultaneously.

6.2 Methodology and Computational Details

The experiments here described involve three main steps: the generation of a reaction descriptor from the simulated ^{1}H NMR spectra of the products and reactants, the generation of the simulated mixtures of two reactions from the ^{1}H NMR-based reaction descriptor, and the development of classification models for mixtures of reactions.

6.2.1 Data Sets of Reactions

The data set of reactions was a subset of the 189 photochemical reactions used in the last Chapter, from which the reactions belonging to class C, [2 + 2] photocycloaddition of C=N to C=C, were excluded.

The final data set consisted of six classes: [3 + 2] photocycloaddition of azirines to C=C (class A, 20 reactions), [2 + 2] photocycloaddition of C=C to C=O (class B, 31 reactions), [4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (class D, 20 reactions), [2 + 2] photocycloaddition of C=C to C=C (class E, 73 reactions), [3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (class F, 10 reactions), and [2 + 2] photocycloaddition of C=C to C=S (class G, 27 reactions). For an easier comparison the labels of the reaction classes are the same as in the last Chapter.

From the data set of 181 reactions, classified in six classes, mixtures of reactions, where two reactions of different classes simultaneously occur, are simulated. From these 181 reactions of six classes, all possible combinations of two were generated yielding a data set of 12421 mixtures of reactions. From this data set, 8280 mixtures were randomly selected to the training set and the remaining 4141 to the test set. Another partition of the data set was also used. From the partition used in the last Chapter (140 reactions in the training set and 41 in the test set without the reactions of class C) all the possible combinations within each data set (training and test sets) was performed yielding a partition of mixtures with 7578 mixtures in the training set and 593 mixtures in the test set. Table 6.1 shows the constitution of each data set and the colors of the labels that were used in the experiments with Kohonen SOMs.

The ^{1}H NMR spectra of the products and reactants were simulated as described in the last Chapter.

6.2.2 Kohonen Self-Organizing Maps

The description of this method including details about the learning procedure could be found in Section 2.6. In the investigations described in this Chapter, the input variables are the 120 reaction descriptors derived from ^{1}H NMR spectra of the reactants and products of two reactions of different classes. The spectra of all products of the two reactions were summed, and the spectra of all reactants of the two reactions were summed. The total spectrum of the reactants was then subtracted from the total spectrum of the products to yield the descriptor of the reactions mixture. SOMs were trained with a diversity of mixtures to study the ability of the ^{1}H NMR descriptors to cluster and classify mixtures of chemical reactions. SOMs with toroidal topology and dimension 49×49 were trained

Table 6.1: Number of reaction mixtures in each mixture class (mixture of two reactions of different classes) for the two partitions of the data set.

Label	Reaction 1	Reaction 2	Partition 1	Partition 2
🟥 (A)	[3 + 2] photocycloaddition of azirines to C=C (A’)	[2 + 2] photocycloaddition of C=C to C=O (B’)	413/207	368/32
🟦 (B)	[3 + 2] photocycloaddition of azirines to C=C (A’)	[4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (D’)	266/133	256/16
🟩 (C)	[3 + 2] photocycloaddition of azirines to C=C (A’)	[2 + 2] photocycloaddition of C=C to C=C (E’)	975/487	896/68
🟨 (D)	[3 + 2] photocycloaddition of azirines to C=C (A’)	[3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (F’)	132/67	128/8
🟦 (E)	[3 + 2] photocycloaddition of azirines to C=C (A’)	[2 + 2] photocycloaddition of C=C to C=S (class G	360/180	352/20
🟥 (F)	[2 + 2] photocycloaddition of C=C to C=O (B’)	[4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (D’)	413/206	368/32
🟧 (G)	[2 + 2] photocycloaddition of C=C to C=O (B’)	[2 + 2] photocycloaddition of C=C to C=C (E’)	1510/754	1288/136
🟪 (H)	[2 + 2] photocycloaddition of C=C to C=O (B’)	[3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (F’)	206/104	184/16
⬜ (I)	[2 + 2] photocycloaddition of C=C to C=O (B’)	[2 + 2] photocycloaddition of C=C to C=S (G’)	558/279	506/40
⬜ (J)	[4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (D’)	[2 + 2] photocycloaddition of C=C to C=C (E’)	974/486	896/68
🟥 (K)	[4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (D’)	[3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (F’)	133/67	127/8
🟫 (L)	[4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings (D’)	[2 + 2] photocycloaddition of C=C to C=S (G’)	360/180	353/20
⬜ (M)	[2 + 2] photocycloaddition of C=C to C=C (E’)	[3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (F’)	498/250	448/34
🟦 (N)	[2 + 2] photocycloaddition of C=C to C=C (E’)	[2 + 2] photocycloaddition of C=C to C=S (G’)	1302/651	1232/85
🟦 (O)	[3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C (F’)	[2 + 2] photocycloaddition of C=C to C=S (G’)	180/90	176/10

and tested using the two different partitions of the data set. For visualization and prediction, each label of the neurons (color) corresponds to a combination of reactions from two specific classes (Table 6.1)

Training of the SOMs was performed by using a linear decreasing triangular scaling function with an initial learning rate of 0.1. The weights were initialized with random numbers that are calculated using the mean and the standard deviation of each variable in the input data set as parameters. For the selection of the winning neuron, the minimum Euclidean distance between the input vector and neuron weights was used. The training was performed over 50-100 cycles, with the learning span and the learning rate linearly decreasing until zero. The Kohonen SOM was implemented with in-house-developed software based on JATOON Java applets. [171] To overcome fluctuations induced by the random factors influencing the training, five or ten independent SOMs were trained with the same objects, generating an ensemble of SOMs. Ensemble predictions were obtained by majority vote of the individual maps.

6.2.3 Counter Propagation Neural Networks

Counter Propagation Neural Networks learn and map objects in a very similar manner to Kohonen SOMs, but have a second layer (output layer) that acts as a look-up table and stores output data (the classification of the mixture of reactions). Details about this method can be found in Section 2.7.

To use the information about the class of the mixture, the classification was encoded with a vector with dimension six, corresponding to the number of classes of reactions. This vector is the output for each object (mixture). The two components of the vector corresponding to the classes of the reactions present in the mixture take the value one, the others take the value zero. During the training, the winning neuron is determined exclusively on the basis of the input layer (^{1}H NMR spectra descriptors), but the weights of the corresponding output neuron are adjusted to become closer to the vector representing the classification of the presented mixture. After the training, the CPNN is able to classify the reactions occurring in the mixture on input of its ^{1}H NMR-based descriptors - the winning neuron is chosen and the corresponding weights in the output layer are used for prediction. Prediction of the two reactions in the mixture was based on the two weights with the highest value at the output layer of the winning neuron.

Training settings were the same as in the experiments with Kohonen SOMs. Ensembles of five or ten independent CPNNs were trained, and predictions were obtained by majority vote of the individual maps.

6.2.4 Feed-Forward Neural Networks

FFNNs [1] were defined with 120 input neurons, one hidden layer of neurons, and six output neurons. In the input layer and in the hidden layer, an extra neuron (called bias) with the constant value of one was also added. Section 2.8 presents the full description of FFNNs.

The networks were trained with the JATOON software [171, 175] to predict the reactions occurring in a mixture, taking as input the difference spectrum of the reactants and products of the mixture of reactions, and producing a six-values output encoding the classes of the reactions in the mixture, just like for the CPNNs.

Corrections were performed on the weights during the training (learning) using the Back-Propagation of Errors algorithm (more details in Section 2.8 or in Ref. [1]). The number of neurons in the hidden layer was optimized for each case, generally in the range 5-10. The activation function was the logistic function and each input and output variable was linearly normalized between 0.1 and 0.9 on the basis of the training set. The maximum number of iterations used in the training was set between 300 and 2,000. The training was stopped when there was no further improvement in the root mean squared error (RMSE) for the test set [56].

Similarly to the experiments performed with Kohonen SOMs and CPNNs, five or ten independent FFNNs were trained, generating an ensemble of FFNNs. Ensemble predictions were obtained by majority vote of the individual maps.

6.2.5 Random Forests

A random forest [70, 71] is an ensemble of unpruned classification trees created by using bootstrap samples of the training data and random subsets of variables to define the best split at each node. It is a high-dimensional nonparametric method that works well on large numbers of variables. The description of RFs method can be found in Chapter 3. The performance is internally assessed with the prediction error for the objects left out in the bootstrap procedure. In this Chapter, RFs were grown with the R program version 2.0.1, [172] using the randomForest library, [173] and were used to classify the reactions present in a mixture of reactions from the same NMR reaction descriptors as for the NNs. The number of trees in the forest was set to 1000, and the number of variables tested for each split was set to default (square root of the number of variables).

6.3 Results and Discussion

6.3.1 Mapping of Mixtures of Reactions on a SOM

Kohonen SOMs of size 49×49 were trained with the two different training sets as described before. In the learning procedure the network made no use of the information related to mixture class. After the training, each neuron of the surface was assigned to a class (one of the 15 different mixtures of reactions), according to the majority class of the mixtures in that neuron or in its neighbors. Figure 6.1 shows a Kohonen SOMs of size 49×49 trained with 8280 mixtures corresponding to the training set of partition 1.

The results show a trend for some types of mixtures to cluster, namely type B (mixture of [3 + 2] photocycloaddition of azirines to C=C and [4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings), type L (mixtures of [4 + 2] and [4 + 4] photocycloaddition of olefins to carbon-only aromatic rings and [2 + 2] photocycloaddition of C=C to C=S), type M (mixtures of [2 + 2] photocycloaddition of C=C to C=C and [3 + 2] photocycloaddition of s-triazolo[4,3-b]pyridazine to C=C), and type N (mixtures of [2 + 2] photocycloaddition of C=C to C=C and [2 + 2] photocycloaddition of C=C to C=S).

In this experiment with partition 1 an individual SOM was able to correctly classify 80,6% and 71.1% of the mixtures of reactions in the training and test sets respectively (Table 6.2). An improvement in the accuracy of the predictions was observed if predictions from ensembles of five and ten Kohonen SOMs were considered. Correct predictions of the mixtures were achieved for 86.7% and 89% of the training set and 77.4% and 79.6% of the test set using ensembles of five and ten SOMs respectively.

Then experiments were performed using a partition of the data with lower similarities between training and test set. In partition 2, no reaction in mixtures of the test set was present in any mixture of the training set. All mixtures of partition 1 are different, but the same reaction can be present in a mixture of the training and in a mixture of the test set. Not surprisingly, the prediction accuracy decreased considerably for the test set. An ensemble of ten Kohonen SOMs was able to correctly classify 62.3% of the mixtures in the test set.

6.3.2 Mapping of Mixtures of Reactions on a CPNN

CPNNs differently of Kohonen SOMs have a second layer (output layer) that acts as a look-up table and stores output data (the classification of the mixture of reactions). Figure 6.2 shows the six output layers (corresponding to the six possible classes of reactions in the mixtures) of a 49×49 CPNN trained with 7578 mixtures (partition 2). A fuzzification parameter of 0.1 ppm was used for the simulation of NMR spectra. High values of the weights in each output layer are represented by blue, and low values by red (a reaction

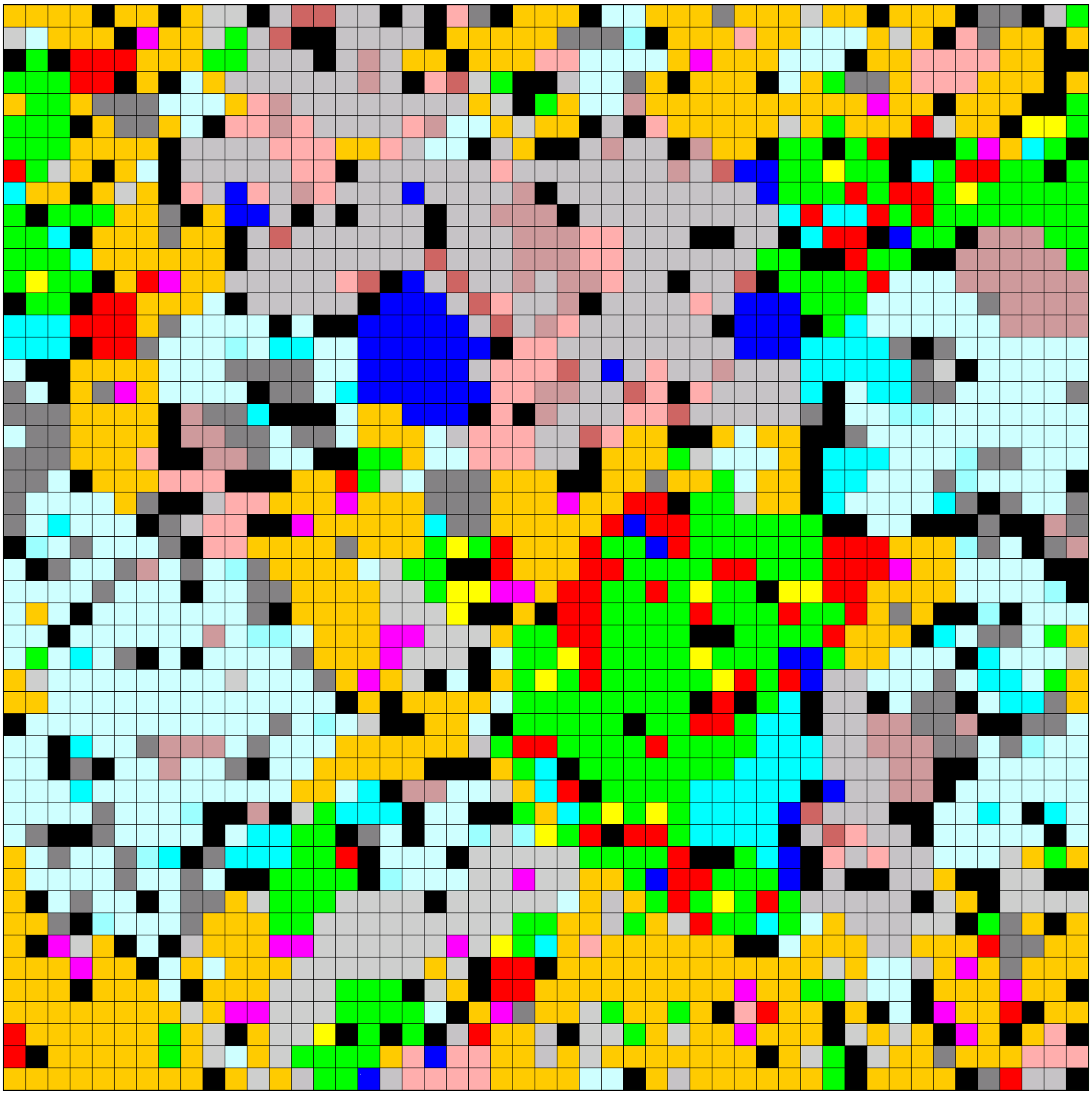


Figure 6.1: Toroidal surface of a 49×49 Kohonen SOM trained with 8280 mixtures of two photochemical reactions encoded by the ^{1}H NMR descriptor of size 120. After the training, each neuron was colored according to the mixture of reactions in the training set that were mapped onto it or onto its neighbors. The colors correspond to the classes in Table 6.1. The black neurons correspond to conflicts.

Table 6.2: Classification of mixtures of reactions (mixtures of two reactions) by Kohonen SOMs of dimension 49×49.

data sets		Nr of	% Correct predictions		
		reactions	Best ind. SOM	Ens. of five SOMs	Ens. of ten SOMs
Partition	Training	8280	80.6	86.7	89.0
1	Test	4141	71.1	77.4	79.6
Partition	Training	7578	82.9	89.4	91.4
2	Test	593	52.6	59.4	62.6

Table 6.3: Classification of mixtures of reactions (mixtures of two reactions) by Counter-Propagation Neural Networks of dimension 49×49.

Data sets		Nr of	% Correct predictions		
		reactions	Best ind. CPNN	Ens. of five CPNNs	Ens. of ten CPNNs
Partition	Training	8280	61.3	73.0	75.6
1	Test	4141	57.7	69.1	71.8
Partition	Training	7578	68.4	77.4	78.6
2	Test	593	47.2	57.2	57.5

class was identified in the mixture of the reactions if the mixture activated a neuron with a high value of the weight corresponding to that reaction class). A CPNN assigns a class to a mixture when two and only two of the output layers have a value higher than 0.5. Otherwise the mixture is classified as undecided.

CPNN did not yield superior predictions to Kohonen SOMs (Table 6.3). For partition 1 an ensemble of ten CPNNs were only able to correctly classify 75.6% and 71.8% of the mixtures of the training and test set respectively. The prediction accuracy for the test set decreased in partition 2 to 57.5%.

Kohonen SOMs and CPNNs are unsupervised learning methods presenting the advantage of an easy visualization of the objects in a map, and revealing relationships between similarities of objects descriptors and objects classes. However, they are based on global comparisons of the descriptor profile, and are not expected to learn associations between classes and reduced numbers of specific descriptors. Such associations may well occur in the studied data set. Therefore experiments were performed with supervised learning techniques that are described next.

6.3.3 Assignment of Reaction Class in Mixtures of Reactions by FFNNs

FFNNs yielded improved prediction accuracies when compared to Kohonen SOMs and CPNNs (Table 6.4). For the more challenging test set of partition 2, correct classifications were achieved for 77% of the mixtures.

6.3.4 Assignment of Reaction Class in Mixtures of Reactions by RFs

The results obtained with Random Forests are displayed in Table 6.5 for the two partitions. The results presented for for training sets were from the internal cross validation obtained by out-of-bag (OOB) estimation. Similarly with the experiments with FFNNs the accuracies of the predictions for partition 1 reached 99% both for OOB estimation of

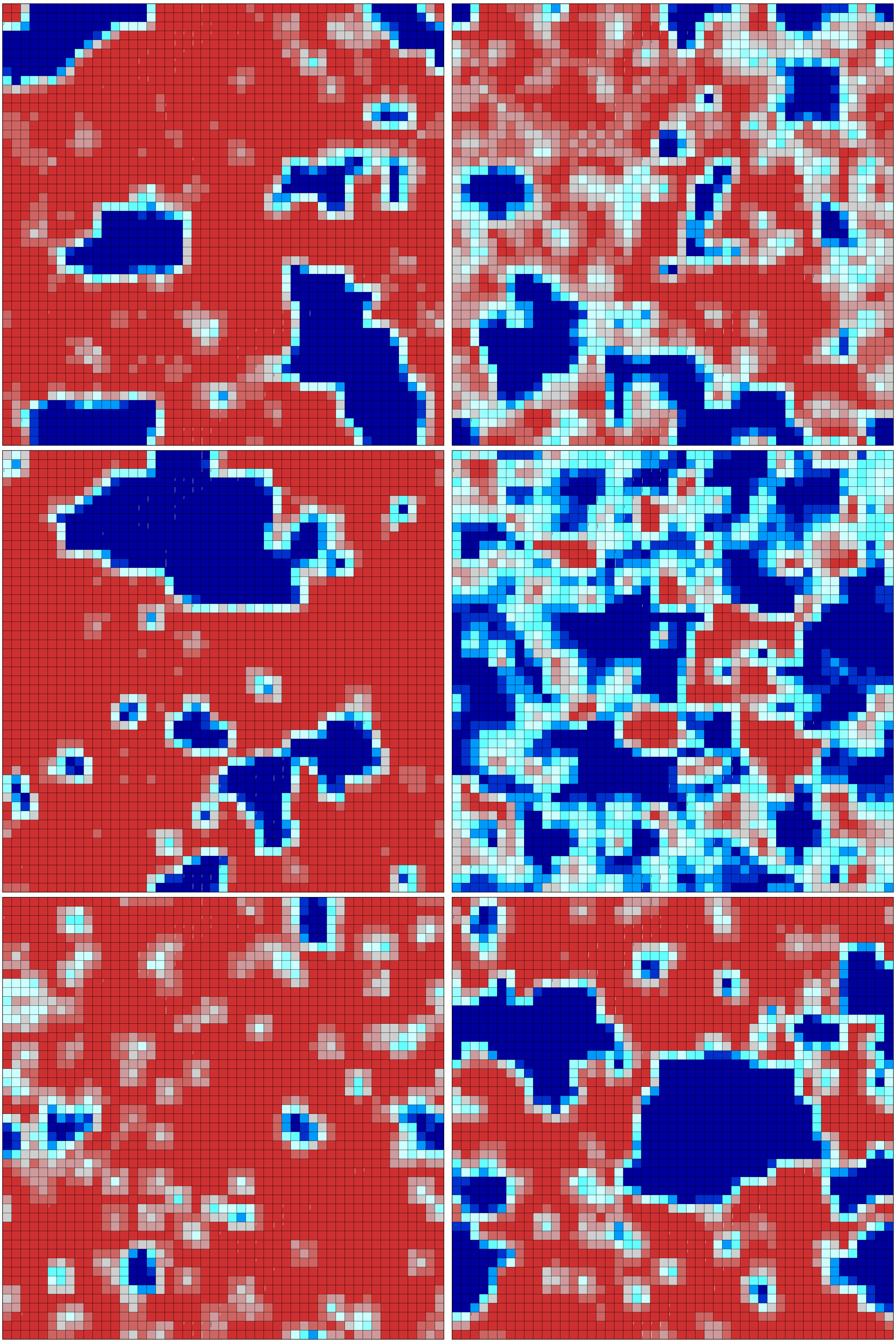


Figure 6.2: Representation of the six output layers of a 49×49 CPNN trained with 7578 mixtures of two reactions encoded by ^{1}H NMR descriptors. High values of the weights in each output layer are represented by blue, and low values by red (a reaction class was identified in the mixture of the reactions if the mixture activated a neuron with a high value of the weight corresponding to that reaction class).

Table 6.4: Classification of mixtures of reactions (mixtures of two reactions) by Feed-Forward Neural Networks.

Data sets		Nr of	% Correct predictions		
		reactions	Best ind. FFNN	Ens. of five FFNNs	Ens. of ten FFNNs
Partition	Training	8280	98.1	98.0	97.9
1	Test	4141	97.7	98.0	97.9
Partition	Training	7578	96.6	94.7	95.0
2	Test	593	75.6	75.7	76.9

Table 6.5: Classification of mixtures of reactions (mixtures of two reactions) by Random Forests.

Data sets		Nr of reactions	% Correct predictions
Partition 1	Training 1	8280	99.2
	Test 1	4141	99.1
	Training 2	4141	97.6
	Test 2	8280	98.4
Partition 2	Training 1	7578	99.6
	Test 1	593	80.3
	Training 2	593	91.4
	Test 2	7578	67.5

the training set and for the test set. With a totally independent test set (partition 2) the accuracy of the predictions were 80%. From the four automatic learning methods tried, RFs performed best.

6.3.5 Comparison Between the Different Methods

Table 6.6 compares the ability of the four automatic learning methods to identify each of the 15 types of reactions mixtures. The types of reactions more difficult to predict typically involved mixtures with reactions of class G' (types E, I, L, N, and O). In the study reported in the previous Chapter, class G yielded some wrong predictions mainly due to a few reaction centers lacking hydrogen atoms.

Table 6.6: Classification of the different types of mixtures of reactions by the different methods used for the test set of partition 2.

Class of	Method			
Mixture	Ens. of ten SOMs	Ens. of ten CPNNs	Ens. of ten FFNNs	Random Forests
A	18.8	21.9	84.4	78.1
B	68.8	50.0	100.0	93.8
C	64.7	66.2	88.2	88.2
D	25.0	0.0	87.5	100.0
E	40.0	35.0	45.0	50.0
F	40.6	40.6	59.4	87.5
G	75.0	72.1	91.2	99.3
H	31.3	12.5	81.3	100.0
I	25.0	15.0	60.0	52.5
J	91.2	88.2	85.3	86.8
K	87.5	75.0	75.0	100.0
L	70.0	45.0	70.0	55.0
M	73.5	73.5	73.5	100.0
N	70.6	63.5	56.5	48.2
O	20.0	0.0	60.0	50.0
Total	62.6	57.5	76.9	80.3

6.4 Conclusions

The automatic classification of mixtures of chemical reactions from differences between the ^{1}H NMR spectra of reactants and products using unsupervised and supervised learning methods was demonstrated with an acceptable level of accuracy for a data set of mixtures of photochemical reactions. The fact that supervised learning methods yielded significantly better predictions suggest that specific changes in the ^{1}H NMR spectra are valuable as markers of reaction types.

These results demonstrate that it is possible to train models to identify the types of reactions occurring in a mixture, from the changes in its ^{1}H NMR spectrum, even if more than one reaction are occurring simultaneously.

The method relies on NMR spectra, which means that it can work without structural information on the reactions participants. Clearly, the approach is limited by the availability of hydrogen atoms in the neighborhood of the reaction center and by the sensitivity of their chemical shifts to the changes resulting from the reaction. The results support our proposal of linking reaction and NMR data for automatic reaction classification of mixtures of reactions.

Acknowledgements

The authors of the work presented in this Chapter thank InfoChem GmbH (Munich, Germany) for sharing the data set of photochemical reactions from the SPRESI database.

Chapter 7

Genome-Scale Classification of Metabolic Reactions. A Preliminary Study

7.1 Introduction

The chemical reactivity of a compound is related to its propensity for bond breaking and bond making, which mainly depends on the physico-chemical properties of the bonds. Gasteiger et al. [132–134] proposed that seven empirical physico-chemical properties are particularly relevant in predicting the reactivity of a chemical bond toward heterolytic cleavage: difference in total charge, difference in π charge, difference in σ electronegativity, bond polarity, resonance stabilization of charges generated by heterolysis, effective bond polarizability, and bond dissociation energy. These encode charge distribution effects, [135, 136] inductive effects, [137] resonance effects, [135] polarizability effects, [138] and bond dissociation energies, [176] and are calculated by empirical procedures implemented in the program PETRA. [131] To use all this information for an entire molecule, and at the same time have a fixed-length representation, we first map all the bonds of a molecule into a Kohonen SOM - a MOLMAP (molecular map of atom-level properties). Each reaction is represented numerically using the MOLMAP approach without assignment of the reaction center, by the difference between molecular descriptors of the products and the substrates. [122]

This Chapter reports the mapping of the reactome into a self-organizing map (SOM) [1, 141] to classify metabolic reactions, and to assign EC numbers from the molecular structures of the substrates and products.

The next Section 7.2 contains the methodology and computational details of the experiments. Section 7.3 discusses the results and Section 7.4 presents the concluding remarks. The experiments presented here are a preliminary study about the application of the

MOLMAP approach to genome-scale data sets of reactions that is fully discussed in the next Chapter.

7.2 Methodology and Computational Details

7.2.1 Data Set of Enzymatic Reactions

A data set of 3468 reactions was extracted from the KEGG LIGAND database of enzymatic reactions (release of July 2005), [37, 38, 40, 177] which consists of the reactions catalysed by 1105 oxidoreductases, 1098 transferases, 622 hydrolases, 335 lyases, 172 isomerases, and 136 ligases. The selected data set excluded reactions listed with more than one EC number, an incomplete EC number, or no assigned EC number. Reactions involving a compound not accepted by PETRA, as well as unbalanced reactions, were also excluded.

7.2.2 Generation of Reaction MOLMAP descriptors

A Kohonen SOM distributes objects over a 2D surface (a grid of neurons) in such a way that objects bearing similar features are mapped onto the same or adjacent neurons. [1, 141] We trained a 15×15 SOM with a diversity of bonds randomly selected from covering the chemical space of the metabolic compounds, each bond described by the above-mentioned seven physico-chemical properties. Then the bonds of one molecule were submitted to the trained SOM (SOM A in Figure 7.1), each bond activating one neuron. The pattern of activated neurons is a map of reactivity features of that molecule (MOLMAP) - a fingerprint of the bonds available in the structure. [122] For easier computational processing, the pattern of activated neurons is encoded numerically. Each neuron is given a value equal to the number of times it was activated by bonds of the molecule. The map (a matrix) is then transformed into a vector by concatenation of columns (Figure 7.1 - a,b).

To focus on functional groups, only bonds that involve a heteroatom, or an atom of a π system, were considered.

The difference between the MOLMAP of the products and the MOLMAP of the reactants is a descriptor of the reaction and represents the structural changes operated by the reaction. It can be interpreted as a fingerprint of the reaction [122] and was used in this work as a numerical code of the reaction (Figure 7.1 - c). The procedure for the generation of the MOLMAP reaction descriptors, and for reaction classification, is fully described in reference [122].

The reaction descriptors (MOLMAPs of the reactions) were calculated for all the reactions, and the relationship between the descriptors and the EC numbers was investigated with a new SOM (SOM B in Figure 7.1 - d). It must be emphasized that this SOM is

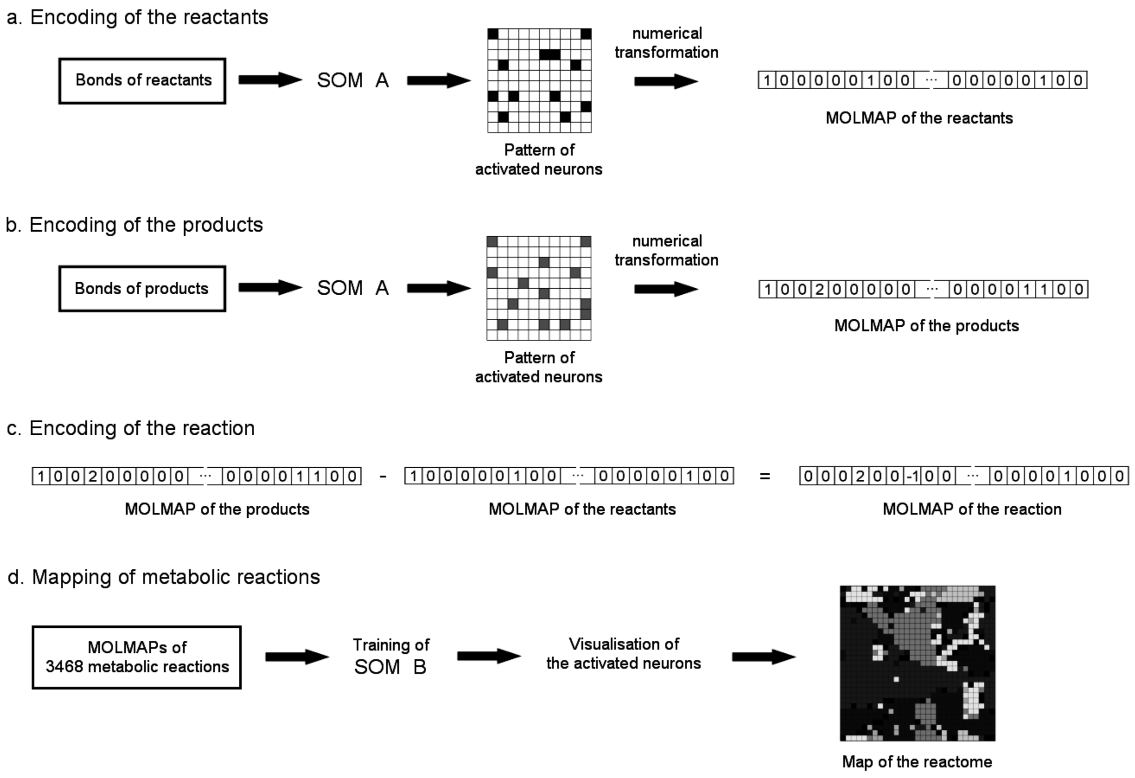


Figure 7.1: Simplified illustration of the procedure used to encode a reaction (a-c), and to classify metabolic reactions (d).

independent of the SOM defined for the generation of MOLMAPs (SOM A in Figure 7.1 - 1). The objects presented to SOM A are bonds, whereas the objects presented to the new SOM B are reactions. After training SOM B with the 3468 reactions, the neurons were colored according to the classes of the reactions that activated them.

7.3 Results and Discussion

7.3.1 Mapping of Enzymatic Reactions in a SOM

A Kohonen SOM was trained with the data set of 3468 reactions encoded with MOLMAPs of size 15×15 using as bond descriptors seven empirical physico-chemical properties.

The resulting map (Figure 7.2) shows a remarkable clustering of the reactions according to the EC classification, with the three most populated classes showing the best clustering: oxidoreductases, transferases, and hydrolases. Ligases also exhibited good clustering.

The zooms in Figure 7.2 illustrate how the reactions were clustered in terms of the first three digits. For most of the map, the majority of the reactions activating the same neuron shared the first three EC digits. The method exhibited robustness in relation to the selected physico-chemical bond properties. For example, an experiment using the above-mentioned properties, except bond dissociation energy and resonance stabilization

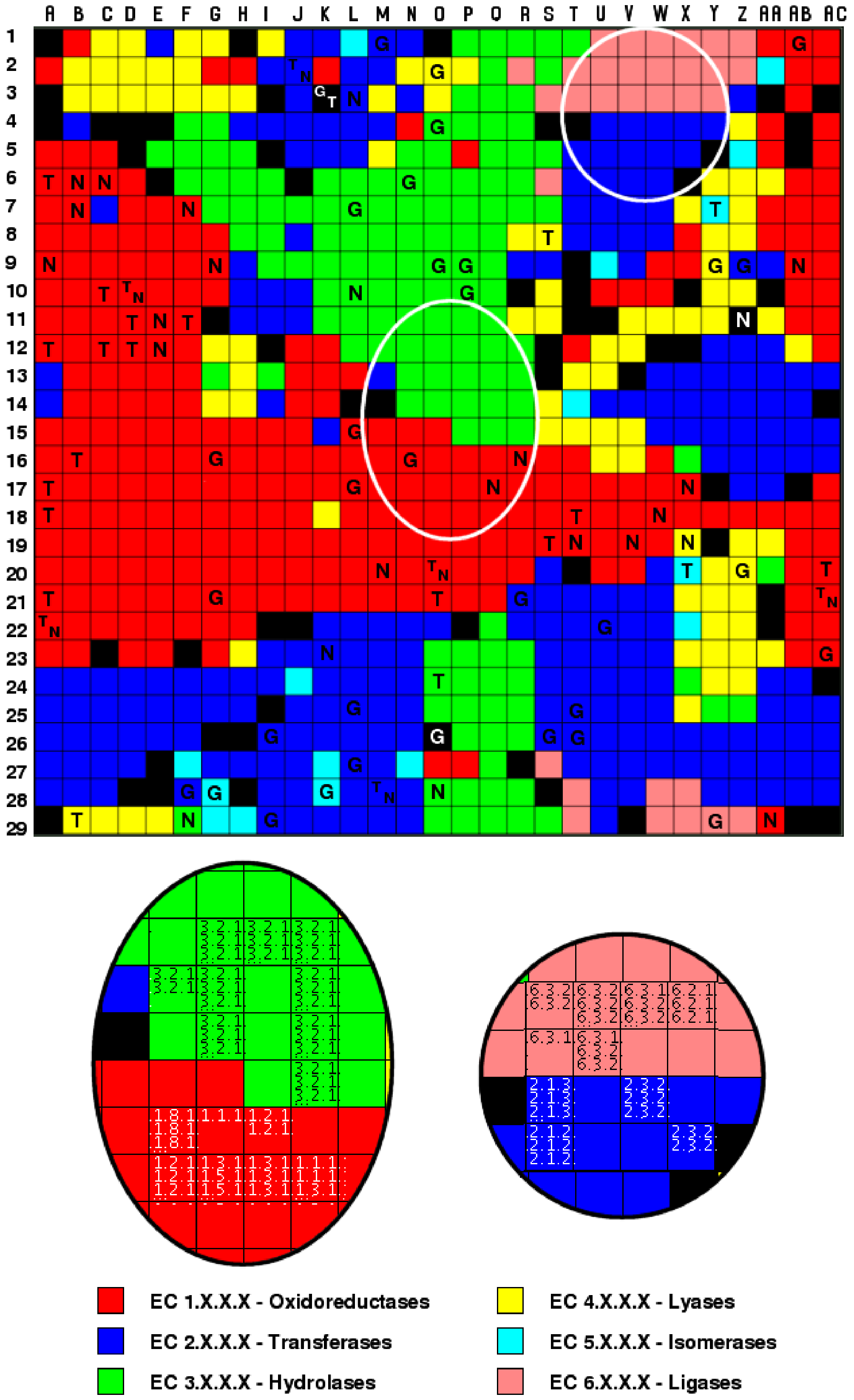


Figure 7.2: SOM of the reactome and its relationship with EC numbers. The ambiguous neurons were colored with black. The neurons activated by reactions of three different metabolic pathways are highlighted (G glycolysis/glyconeogenesis, N nitrobenzene degradation, T toluene and xylene degradation).

of charges generated by heterolysis, yielded essentially the same results. Neuron H,23 (Figure 7.2) was activated by a reaction catalysed by uracilylalanine synthase, which was listed in the database as EC 4.2.99.16 (a lyase). However, the neighbors of that neuron were assigned either to the transferase or the oxidoreductase classes. Interestingly, the EC number of this enzyme was transferred in 2002 to EC 2.5.1.53 (a transferase), [178, 179] and only the September 2005 release of KEGG incorporated this change in the reaction file. A different case is the reaction catalysed by 17αhydroxyprogesterone aldolase, listed as EC 4.1.2.30 but exciting neuron K,18 in the middle of an oxidoreductase region. Significantly, this reaction involves the oxidation of nicotinamide adenine dinucleotide phosphate (NADPH) to NADP$^+$. All the reactions mapped into adjacent neurons are catalysed by oxidoreductases and involve NADP$^+$ and NADPH. In 405 reactions involving oxidation/reduction of NADPH/NADP$^+$, only 11 are not officially classified as oxidoreductases.

The examples illustrate a new way to analyze relationships between reactions in genome-scale databases of metabolic transformations, to detect similarities not revealed by EC numbers, and to identify problematic classifications in terms of EC number.

7.3.2 Mapping of Metabolic Pathways in a SOM

With the wide range of biochemical reactions defined by the SOM of Figure 7.2, the analysis and comparison of metabolic pathways are also possible. This is demonstrated by mapping all the reactions from three different pathways - glycolysis/gluconeogenesis, nitrobenzene degradation, and toluene and xylene degradation - onto the same map (Figure 7.2).

The pattern of neurons activated by glycolysis / gluconeogenesis reactions is significantly distinct from the other two, not only because different regions are involved, but also because different classes are preferred. For example, transferases and hydrolases are dominant in glycolysis / gluconeogenesis, whereas oxidoreductases are preferred by the two metabolic pathways of xenobiotics (which overlap considerably). This finding is in agreement with the wellknown biological strategy of metabolizing aromatic compounds through oxidation. Neuron D,10 is an example of a neuron activated by reactions from the two pathways of xenobiotics: oxidation of 3-hydroxytoluene to 2,3-dihydroxytoluene and oxidation of phenol to catechol. Both are catalysed by phenol 2-monooxygenase (EC 1.14.13.7).

7.4 Conclusions

The results demonstrate the ability of MOLMAPs to classify metabolic reactions, and show a general compatibility between the new scheme and the EC classification. They open the way to a host of new investigations on the diversity analysis of metabolic reactions and comparison of metabolic pathways that will fully discussed in the next Chapter.

Acknowledgements

The authors of the work presented in this Chapter acknowledge Molecular Networks GmbH (Erlangen, Germany) for access to the PETRA program and Kyoto University Bioinformatics Center (Kyoto, Japan) for access to the KEGG database.

Chapter 8

Genome-Scale Classification of Metabolic Reactions and Assignment of EC Numbers

8.1 Introduction

The automatic perception of chemical similarities between metabolic reactions is required for a variety of applications ranging from the computer-aided validation of classification systems, to genome-scale reconstruction (or comparison) of metabolic pathways, to the classification of enzymatic mechanisms. Comparison of enzymatic reactions has been mostly based on EC numbers, which are extremely useful and widespread but not always straightforward to apply, and often problematic when an enzyme catalyses more than one reaction, when the same reaction is catalysed by different enzymes, when official EC numbers are incomplete (or still not assigned), or when reactions are not catalysed by enzymes. Different methods should be available to automatically compare metabolic reactions from their reaction formulas, independently of EC numbers. Simultaneously, methods are required for the automatic assignment of EC numbers to reactions still not officially classified.

We have proposed the MOLMAP reaction descriptors to numerically encode the structural transformations resulting from a chemical reaction. MOLMAPs do not require the previous assignment of reaction centers, and represent the pattern of bonds that are broken, changed, and made during a chemical reaction, on the basis of physico-chemical and topological properties of covalent bonds.

The last Chapter presented the preliminary results concerning the application of MOLMAPs to the classification of a genome-scale data set of enzymatic reactions. [123] (More recently, Faulon et al. reported a very similar approach but using molecular signatures of topological atom neighborhoods, and support vector machines as the learning algorithm to classify metabolic reactions). [119]

Here, such descriptors are applied to the mapping of a genome-scale database of almost 4,000 metabolic reactions by Kohonen self-organizing maps (SOM), and its screening for inconsistencies in EC numbers (reactions revealed as similar but belonging to different EC classes). We report the training of SOM and Random Forests with the enzymatic reactions in the KEGG database in order to their classification, to study the agreement between EC numbers and the MOLMAP-based classification, and to assign the EC numbers from the reaction formula. Different levels of similarity between training and test sets were explored. Reactions were included in both possible directions. They were processed both with all the reactants and products, or only with the main reactants and products.

8.2 Methodology and Computational Details

8.2.1 Data Set of Chemical Reactions

Enzymatic reactions were extracted from the KEGG LIGAND database (release of November 2006) [37,38,40,177] in MDL .mol format, and were pre-processed in the following way. Unbalanced reactions were removed (or manually balanced in simple cases). In some cases, general fragment symbols were replaced, such as 'X' by a chlorine atom, or 'R' by methyl, adenine, cytosine or other type of fragment depending on the reaction. Reactions were removed that involved a compound producing no output by the software employed for the calculation of descriptors (STANDARDIZER and CXCALC tools from JChem package - ChemAxon, Budapest, Hungary, www.chemaxon.com or PETRA - Molecular Networks GmbH, Erlangen, Germany). Duplication of reactions was eliminated. Reactions differing only in stereochemical features were considered as duplicates and were included only once. The detection of reaction duplicates was based on chemical hashed fingerprints generated by the GenerFP tool from JChem package, version 3.1.7.1 (ChemAxon, Budapest, Hungary, www.chemaxon.com) with a length of 128 bytes, a maximum number of 10 bonds in patterns, and 3 bits switched on for each pattern in the structure. Because most reactions are reversible, and a classification system should be able to account for reversibility, all reactions were represented in both directions (The MOLMAP descriptors have symmetrical values for the two directions). After the representation in both directions, the procedure to exclude duplicated reactions was applied again. Reactions were also removed that were listed with more than one EC number, or no assigned EC number. Reactions with incomplete EC numbers were not used for training. Finally, reactions with null reaction MOLMAP (eg. reactions where only stereochemistry changes occur) were not used.

In the experiments to predict the subclass (or sub-subclass), only subclasses (or sub-subclasses) with four or more reactions were considered.

The number of reactions used in the experiments to evaluate the impact of the MOLMAP descriptor size and type of bond descriptors in RFs learning were displayed in Table 8.1.

Table 8.1: Composition of the data set at each level of the EC system.

	Nr. of classification groups	total	train	test
1^{st} level	6 classes	3784	3156	628
2^{nd} level	49 subclasses	3764	3140	624
3^{rd} level	110 sub-subclasses	3671	3066	605
4^{th} level	111 EC numbers	682	571	111

Reactions represented only in one direction. The data set of 3784 reactions was partitioned into training and test sets using a 29×29 Kohonen SOM. The SOM was trained with all reactions and after the training one reaction was randomly selected from each occupied neuron and moved to the test set.

Table 8.2: Number of reactions in each subclass (for the prediction of subclass).

EC 1.x		EC 2.x		EC 3.x		EC 4.x		EC 5.x		EC 6.x	
EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.
1.1	369	2.1	177	3.1	234	4.1	144	5.3	42	6.1	20
1.2	126	2.2	15	3.2	99	4.2	129	5.4	35	6.2	44
1.3	113	2.3	222	3.3	21	4.3	26	5.5	22	6.3	73
1.4	61	2.4	269	3.4	8	4.4	19	EC 5	99	6.4	7
1.5	64	2.5	84	3.5	190	4.5	5			6.5	4
1.6	13	2.6	82	3.6	49	4.6	4			EC 6	148
1.7	20	2.7	305	3.7	18	4.99	6				
1.8	37	2.8	64	3.8	12	EC 4	333				
1.9	3	EC 2	1218	EC 3	631						
1.10	13										
1.11	19										
1.12	5										
1.13	89										
1.14	313										
1.16	8										
1.17	26										
1.18	4										
1.21	9										
EC 1	1292										

The final data set ("whole data set") includes each reaction in both directions and consists of 7482 reactions - 2594 of class EC 1 (oxidoreductases), 2438 of class EC 2 (transferases), 1278 of class EC 3 (hydrolases), 666 of class EC 4 (lyases), 206 of class EC

5 (isomerases), and 300 of class EC 6 (ligases). The number of reactions of each subclass and sub-subclass used in the experiments to assign the second and third digit of the EC number could be found in Tables 8.2 and 8.3. Different criteria were used for building training and test sets, in some cases aiming at covering the maximum possible diversity of reactions, in other cases aiming at tuning the level of similarity between training and test sets on the basis of EC numbers (e.g. by not allowing the same full EC number, or the same sub-subclass, to appear both in the training set and in the test set).

8.2.2 Kohonen Self-Organizing Maps (SOM)

SOMs with toroidal topology were used in this study for two independent tasks, the classification of chemical bonds for the generation of a molecular descriptor, and the classification of enzymatic reactions. SOMs learn by unsupervised training, distributing objects through a grid of so-called neurons, on the basis of the objects features. This is an unsupervised method that projects multidimensional objects into a 2D surface (a map). SOMs can reveal similarities between objects, mapped into the same or neighbor neurons. Each neuron of the map contains as many elements (weights) as there are input variables (objects features). Before the training starts, the weights take random values. During the training, each individual object is mapped into the neuron with the most similar weights compared to its features. This is the central neuron, or winning neuron. It is said that the winning neuron was excited (or activated) by the bond, and its weights are then adjusted to make them even more similar to the properties of the presented bond. Not only does the winning neuron have its weights adjusted, but also the neurons in its neighborhood. The extent of adjustment depends, however, on the topological distance to the winning neuron - the closer a neuron is to the winning neuron the larger is the adjustment of its weights. The objects of the training set are iteratively fed to the map, and the weights corrected, until a pre-defined number of cycles is attained. A trained Kohonen SOM reveals similarities between objects of a data set in the sense that similar objects are mapped into the same or closely adjacent neurons.

Training was performed by using a linear decreasing triangular scaling function used with an initial learning rate of 0.1 and an initial learning span of half the size of the map (except for maps of size 49×49, for which an initial learning span of 7 was used). The weights were initialized with random numbers that were calculated using as parameters the mean and standard deviation of the corresponding variables in the input data set. For the selection of the winning neuron, the minimum Euclidean distance between the input vector and neuron weights was used. The training was typically performed over 50, 75 or 100 cycles, with the learning span and the learning rate linearly decreasing until zero. SOMs were implemented throughout this study with an in-house developed Java application derived from the JATOON Java applets. [171]

Table 8.3: Number of reactions in each sub-subclass (for the prediction of sub-subclass).

EC 1.x.x		EC 1.x.x		EC 2.x.x		EC 3.x.x		EC 4.x.x		EC 5.x.x		EC 6.x.x	
EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.	EC	Nr.
1.1.1	304	1.17.1	11	2.1.1	151	3.1.1	88	4.1.1	85	5.3.1	15	6.1.1	20
1.1.3	36	1.17.4	11	2.1.2	14	3.1.2	18	4.1.2	33	5.3.3	15	6.2.1	44
1.1.4	4	1.18.6	4	2.1.3	9	3.1.3	77	4.1.3	21	5.3.99	9	6.3.1	13
1.1.99	21	1.21.3	6	2.2.1	15	3.1.4	33	4.1.99	5	5.4.2	6	6.3.2	33
1.2.1	99	EC 1	1237	2.3.1	192	3.1.6	10	4.2.1	96	5.4.3	7	6.3.3	4
1.2.3	12			2.3.2	18	3.1.8	4	4.2.2	4	5.4.99	16	6.3.4	13
1.2.4	8			2.3.3	12	3.2.1	84	4.2.3	28	5.5.1	22	6.3.5	10
1.2.99	4			2.4.1	213	3.2.2	15	4.3.1	18	EC 5	90	6.4.1	7
1.3.1	94			2.4.2	37	3.3.2	18	4.3.2	5			6.5.1	4
1.3.3	11			2.4.99	19	3.4.13	4	4.4.1	19			EC 6	148
1.3.99	6			2.5.1	84	3.5.1	95	4.5.1	5				
1.4.1	22			2.6.1	80	3.5.2	19	4.6.1	4				
1.4.3	37			2.7.1	167	3.5.3	26	4.99.1	6				
1.5.1	49			2.7.2	12	3.5.4	32	EC 4	329				
1.5.3	10			2.7.3	8	3.5.5	10						
1.5.99	4			2.7.4	28	3.5.99	8						
1.6.5	8			2.7.6	5	3.6.1	48						
1.7.1	10			2.7.7	51	3.7.1	18						
1.7.3	5			2.7.8	28	3.8.1	12						
1.8.1	20			2.8.1	8	EC 3	619						
1.8.3	4			2.8.2	34								
1.8.4	8			2.8.3	21								
1.10.3	8			EC 2	1206								
1.11.1	19												
1.13.11	76												
1.13.12	11												
1.14.11	47												
1.14.12	33												
1.14.13	151												
1.14.14	35												
1.14.15	10												
1.14.16	9												
1.14.18	5												
1.14.21	8												
1.14.99	11												
1.16.1	6												

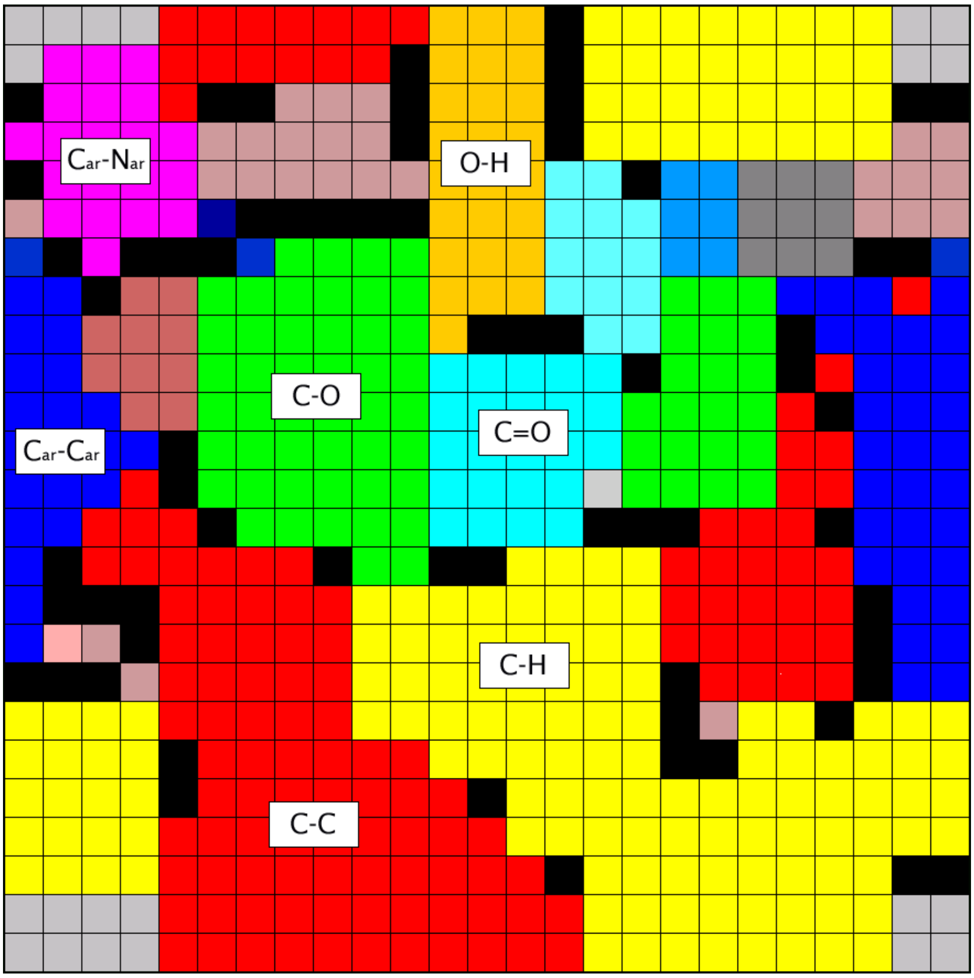


Figure 8.1: A 25×25 SOM trained with 1568 bonds represented by the subset of physico-chemical and topological descriptors. Each neuron was red after the training, according to the types of chemical bonds that were mapped onto it. The figure shows how different regions of the surface are activated by different types of bonds.

8.2.3 Generation of MOLMAP Reaction Descriptors

The generation of MOLMAP molecular descriptors is based on a SOM that distributes chemical bonds through the grid of neurons. The chemical bonds are represented by topological and physico-chemical features (Tables 8.4 and 8.5).

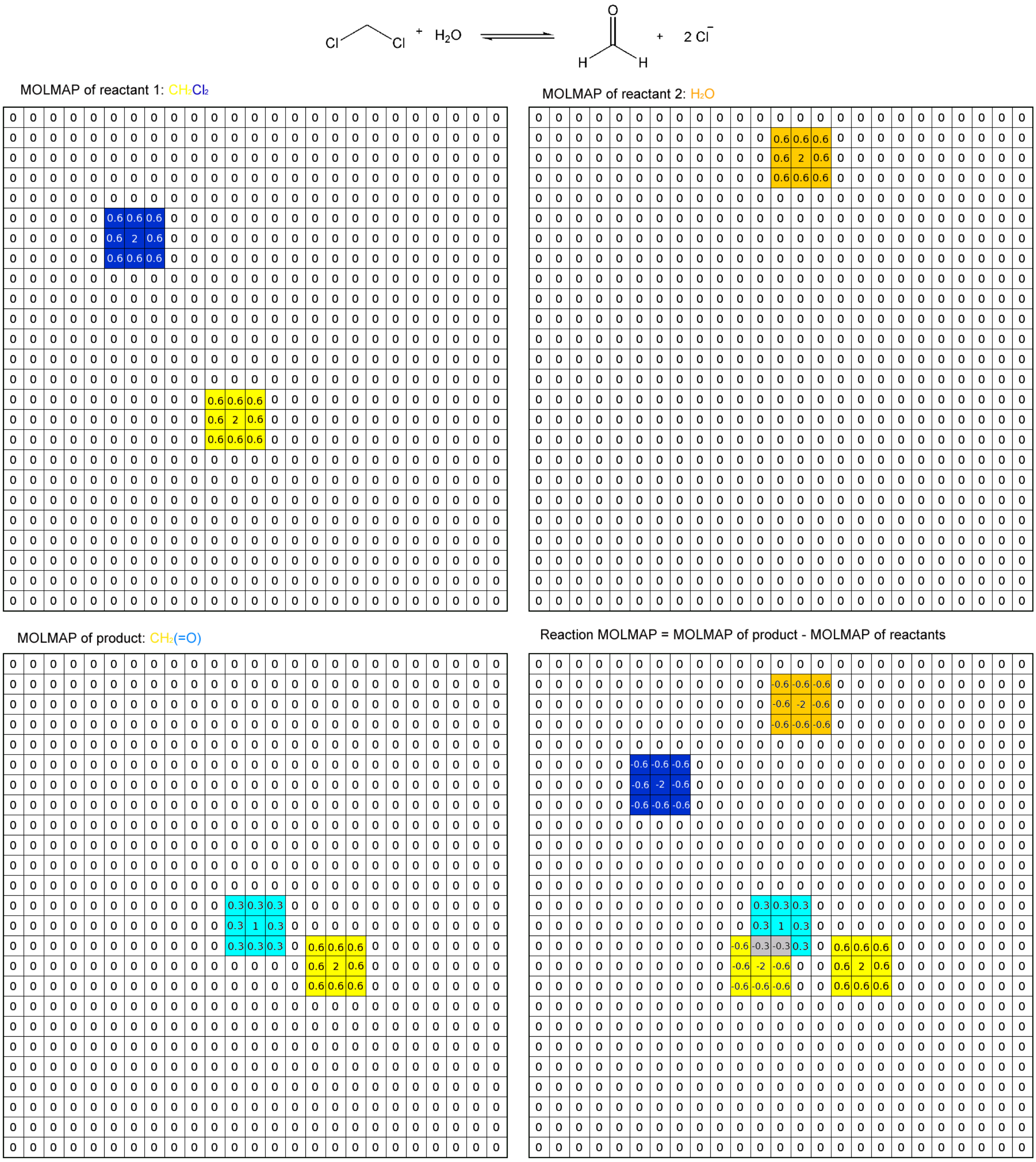
The SOM is trained with a diversity of bonds, taken from a diverse data set of molecules. After the training, the SOM can reveal similarities between chemical bonds, mapped into the same region of the surface (Figure 8.1), and can describe the types of bonds present in a molecule. The bonds existing in a molecule can be represented as a whole by mapping all the bonds of that molecule onto the SOM previously trained with a diversity of bonds. This is illustrated in the scheme of Figure 8.2 (the calculation of a reaction MOLMAP for Reaction R00603 - Entry 1 of Table 8.7) for the two reactants and one product separately - in each map the neurons activated by the bonds of the compound as well as the neighbor neurons are displayed with colors.

Table 8.4: List of topological bond descriptors. A and B are the two atoms of the chemical bond.

Topological descriptors
1. A is a hydrogen, AH
2. A is carbon, AC
3. A is nitrogen, AN
4. A is oxygen, AO
5. A is phosphorus, AP
6. A is sulphur, AS
7. A is halogen, AX
8. B is hydrogen, BH
9. B is carbon, BC
10. B is nitrogen, BN
11. B is oxygen, BO
12. B is phosphorus, BP
13. B is sulphur, BS
14. B is halogen, BX
15. number of H-atoms bonded to A, HnA
16. number of C-atoms bonded to A, CnA
17. number of N-atoms bonded to A, NnA
18. number of O-atoms bonded to A, OnA
19. number of P-atoms bonded to A, PnA
20. number of S-atoms bonded to A, SnA
21. number of halogen atoms bonded to A, XnA
22. A is C sp1, Csp1A
23. A is C sp2, Csp2A
24. A is C sp3, Csp3A
25. number of H-atoms bonded to B, HnB
26. number of C-atoms bonded to B, CnB
27. number of N-atoms bonded to B, NnB
28. number of O-atoms bonded to B, OnB
29. number of P-atoms bonded to B, PnB
30. number of S-atoms bonded to B, SnB
31. number of halogen atoms bonded to B, XnB
32. B is C sp1, Csp1B
33. B is C sp2, Csp2B
34. B is C sp3, Csp3B
35. bond order, boord
36. number of C sp1 neighbors of A
37. number of C sp2 neighbors of A
38. number of C sp3 neighbors of A
39. number of C sp1 neighbors of B
40. number of C sp2 neighbors of B
41. number of C sp3 neighbors of B

Table 8.5: List of physico-chemical bond descriptors. A and B are the two atoms of the chemical bond.

Physico-chemical descriptors
42. π charge of A, q_{π_A}
43. π charge of B, q_{π_B}
44. Difference of π charges, Δq_{π}
45. σ charge of A, q_{σ_A}
46. σ charge of B, q_{σ_B}
47. Difference of σ charges, Δq_{σ}
48. Total charge of A, q_{tot_A}
49. Total charge of B, q_{tot_B}
50. Difference of total charges, Δq_{tot}
51. Maximum charge of A neighbors, q_{max_A}
52. Maximum charge of B neighbors, q_{max_B}
53. Minimum charge of A neighbors, q_{min_A}
54. Minimum charge of B neighbors, q_{min_B}
55. π electronegativity of A, χ_{π_A}
56. π electronegativity of B, χ_{π_B}
57. Difference of π electronegativity, $\Delta\chi_{\pi}$
58. σ electronegativity of A, χ_{σ_A}
59. σ electronegativity of B, χ_{σ_A}
60. Difference of σ electronegativity, $\Delta\chi_{\sigma}$
61. Maximum polarizability of A neighbors , α_{max_A}
62. Maximum polarizability of B neighbors , α_{max_B}
63. Minimum polarizability of A neighbors, α_{min_A}
64. Minimum polarizability of B neighbors , α_{min_B}
65. π charge density of A, qdens_{π_A}
66. π charge density of B, qdens_{π_B}
67. Total charge density of A, qdens_{tot_A}
68. Total charge density of B, qdens_{tot_B}



MOLMAP of reactant 1: CH_2Cl_2

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.6	2	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.6	2	0.6	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MOLMAP of reactant 2: H_2O

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.6	2	0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MOLMAP of product: $CH_2(=O)$

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0.3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.3	1	0.3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0.3	0	0.6	0.6	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	2	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Reaction MOLMAP = MOLMAP of product - MOLMAP of reactants

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-0.6	-0.6	-0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-0.6	-2	-0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	-0.6	-0.6	-0.6	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.6	-0.6	-0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.6	-2	-0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	-0.6	-0.6	-0.6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.3	0.3	0.3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0.3	1	0.3	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.6	-0.3	-0.3	0.3	0	0.6	0.6	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.6	-2	-0.6	0	0	0.6	2	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	-0.6	-0.6	-0.6	0	0	0.6	0.6	0.6	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 8.2: Example of the calculation of a reaction MOLMAP (Entry 1 of Table 8.7 - Reaction R00603.

The pattern of activated neurons is interpreted as a fingerprint of the available bonds in the molecule, and it was used as a molecular descriptor (MOLMAP). For numerical processing, each neuron gets a value equal to the number of times it was activated by bonds of the molecule. For example, in the map of the product illustrated in the scheme the neuron activated by the C=O bond got a value of 1, while the neuron activated by the two (equivalent) C-H bonds received a value of 2. The map is then transformed into a vector by concatenation of columns. In order to account for the relationship between similarity of bonds and proximity in the map, a value of 0.3 is added to each neuron multiplied by the number of times a neighbor was activated by a bond.

A similar idea was proposed by Atalay to encode the primary structure of proteins with a SOM on the basis of aminoacid features. [180]

The MOLMAP descriptor *of a reaction* is calculated as the difference between the MOLMAP of the products and the MOLMAP of the reactants. If there are more than one reactant in a reaction, the MOLMAPs of all reactants are summed, and the same for the products. The reaction MOLMAP is a representation of structural changes operated by the reaction, and was used in this work as a numerical code of the reaction.

A full description of the methodology for the generation of the MOLMAP reaction descriptors can be found in Ref. [122]

In this study, the SOM was trained with the chemical bonds from a diverse representative set of molecules involved in enzymatic reactions. The molecules were selected from the database of reactions by the Ward's minimum variance method [181–183] (a hierarchical clustering method designed to optimize the minimum variance within clusters) implemented in the WARD tool of the JChem package, and was used here with chemical hashed fingerprints as variables. The algorithm begins with one large cluster encompassing all objects to be clustered. In this case, the error sum of squares is 0. The program searches objects that can be grouped together while minimizing the increase in error sum of squares. The method is based on molecular chemical hashed fingerprints with a length of 64 bytes, a maximum number of 5 bonds in patterns, and 2 bits switched on for each pattern in the structure. The Kelley method [184] was used to decide the number of clusters - 45. The central compounds of the clusters were chosen, and all their bonds were extracted to the training set of the SOM.

The original molfiles of the compounds were treated with the JChems's STANDARDIZER tool to add hydrogens, clean stereochemistry, and aromatize. The physico-chemical and topological descriptors listed in Tables 8.4 and 8.5 were computed with in-house developed software from properties calculated with the JChem CXCALC tool. As the descriptors for a bond depend on the orientation of the bond, each bond was always oriented from the atom with higher charge to the atom with lower charge. To make all the descriptors equally relevant in the training of the SOM, z-normalization was applied to each descriptor 42-68 based on the whole data set of chemical bonds. Descriptors 1-14 and 35-41 were multiplied by 3.

SOMs of sizes 7×7, 10×10, 15×15, 20×20, 25×25 and 29×29, yielding MOLMAPs of dimension 49, 100, 225, 400, 625, and 841, respectively, were trained with a data set of 1568 bonds extracted from the selected compounds using the Ward method.

To focus on substructures around functional groups, only bonds were considered that include (or are at a one bond distance from) a heteroatom or an atom belonging to a π system. Experiments were performed using exclusively topological descriptors of bonds, only physico-chemical descriptors, the whole set of descriptors, or the subset of descriptors 1-43, 45, 46, 48, 49, 55, 56, 58, 59, 65, 66, 67, 68 from Tables 8.4 and 8.5.

8.2.4 Classification of Enzymatic Reactions

MOLMAP reaction descriptors were calculated for all the reactions in the data set using a SOM, as described above. Then, classification of the reactions was performed by new SOMs, unrelated and independent from the first. While the first SOM was trained with chemical bonds to generated molecular descriptors, the new SOMs were trained with enzymatic reactions to predict EC numbers. Chemical bonds were the objects of the first SOM, reactions were the objects of the new SOMs. The data set of reactions was partitioned into a training and a test set, using different criteria depending on the experiment (see Section 8.3 Results and Discussion). In the experiments aimed at predicting the EC class, a SOM was trained with the MOLMAPs of the reactions as the input. After the training, the whole training set was mapped on the surface, and each neuron was assigned to a reaction class depending on the majority of reaction classes that activated the neuron (or its neighbors, if the neuron was empty). When a majority could not be obtained, the neuron was classified as undecided. The test set was then submitted to the SOM, and each reaction was classified based on the class of the neuron it activated.

In the experiments for predicting EC subclass, or sub-subclass, reactions belonging to each EC class were treated separately, i.e. six experiments were performed, one for each EC class. Due to the large number of sub-subclasses of classes EC 1 and EC 2, counter propagation neural networks (CPNN) were used in those cases, instead of Kohonen NNs. CPNNs learn and map objects in a very similar manner to Kohonen SOMs, but have a second layer (output layer) that acts as a look-up table and stores output data (the classification of the reaction). For this task, the classification of the reaction was encoded with a vector with a dimension corresponding to the number of sub-subclasses. The component of the vector corresponding to the sub-subclass of the reaction takes the value 1, the others take the value zero. During the training, the winning neuron is determined exclusively on the basis of the input layer, but the weights of the corresponding output neuron are adjusted to become closer to the vector representing the classification of the presented reaction. After the training, the CPNN is able to classify a reaction on input of its MOLMAP descriptor - the winning neuron is chosen and the corresponding weights in the output layer are used for prediction. Prediction of the sub-subclass was based on the weight with the highest value at the output layer of the winning neuron.

To overcome fluctuations induced by the training random factors, five or ten independent SOMs were trained, generating an ensemble of SOMs. Ensemble predictions were obtained by majority vote of the individual maps. The relationship between the number of votes for the winning class, and the reliability of the prediction, was investigated. Another measure of reliability was investigated, the Euclidean distance between a reaction MOLMAP (a vector with the reaction descriptors) and the weights of the corresponding winning neuron.

To evaluate the impact of the MOLMAP size, experiments were performed with MOLMAPs of sizes 15×15, 20×20, 25×25, and 29×29. Four different sets of bond descriptors were also evaluated.

8.2.5 Classification of Racemase and Epimerase Reactions

An experiment was performed with 18 reactions of subclass 5.1 extracted from the BioPath database version Nov. 2005 (Molecular Networks GmbH, Erlangen, Germany) with stereochemistry explicitly assigned in molecules, and chirality codes [185] were used as molecular descriptors instead of MOLMAPs. Chirality codes are descriptors of molecular chirality that can distinguish stereoisomers, namely enantiomers. Here, conformation-independent chirality codes (CICC) with a dimension of 50 were calculated with the following parameters: (a) all the atoms in each molecule were considered, including hydrogen atoms, (b) partial atomic charge (calculated by PETRA 3.2) was used as the atomic property, (c) the chirality code function was sampled in the interval [-r,+r] with r equal to 0.070 e^2Å^{-1}, (d) only combinations of four atoms with maximum interatomic path distances of six bonds were considered, (e) the smoothing parameter was set to (code length/range of u)^2. The 50 dimensional vectors were normalized by their vector sum.

8.2.6 Random Forests Assignment of EC numbers

MOLMAP reaction descriptors were calculated for the different data sets using a SOM, as described in the last Subsection. Then, automatic assignment of EC numbers was performed by Random Forests (RF) for the four levels of the EC classification system. A random forest [70, 71] is an ensemble of unpruned classification trees created by using bootstrap samples of the training data and random subsets of variables to define the best split at each node. It is a high-dimensional nonparametric method that works well on large numbers of variables. The predictions are made by majority voting of the individual trees. It has been shown that this method is very accurate in a variety of applications. [71] Additionally, the performance is internally assessed with the prediction error for the objects left out in the bootstrap procedure (out-of-bag estimation, OOB). In this work, RFs were grown with the R program, [172] using the randomForest library, [173]. The number of trees in the forest was set to 1000, and the number of variables tested for each split was set to default (square root of the number of variables). The data set of reactions was partitioned into a training and a test set, using different criteria depending on the experiments (see Section Results and Discussion). Also a different measure of reliability was investigated, the "probability" assigned to each prediction by the RF. The voting system of a RF allows the association of a "probability" to each prediction that reflects the percentage of votes obtained by the winning class.

To evaluate the influence of MOLMAP size and type of bond descriptors on the accuracy of the EC assignments at the four levels, experiments were performed with MOLMAPs of size 7×7, 10×10, 15×15, 20×20, 25×25, and 29×29, each one using the four different sets of bond descriptors.

8.3 Results and Discussion

8.3.1 Mapping the Whole Data Set of Enzymatic Reactions on a Self-Organizing Map (SOM)

The MOLMAP representation of chemical reactions yielded numerical descriptors of enzymatic reactions that could be further processed by a SOM. Training a SOM in an unsupervised manner distributes objects (in this case reactions) over a surface in order that objects represented by similar descriptors generally activate a common region of the map. Therefore, SOM were here applied to evaluate to which extent similarities between reaction MOLMAPs correspond to similarities in EC classification, and whether MOLMAPs could be used for the classification of enzymatic reactions. Note that MOLMAPs represent overall reactions, and do not explicitly consider the mechanisms (although the physico-chemical and topological bond properties used for the calculation of MOLMAPs are in principle related to mechanisms). EC numbers are also assigned based on overall reactions. Because many reactions are reversible, and a general classification system should be able to represent reactions in both directions, each reaction in the database was included in both directions (the MOLMAP descriptors have symmetrical values for the two directions). A 49×49 Kohonen SOM was trained with the whole data set of 7482 enzymatic reactions encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. During this stage, the network made no use of the information related to EC numbers. After the training, each neuron of the surface was assigned to a class (first digit of the EC number), according to the majority class of the reactions activating that neuron. The resulting map is displayed in Figure 8.3. It clearly shows a trend for reactions to cluster according to the EC class, particularly those catalysed by oxidoreductases (EC

1), tranferases (EC 2), hydrolases (EC 3), and ligases (EC 6) (to better understand the concept of toroidal surface and visualize the clustering according EC classes in the limits of the map Figure 8.4 shows the tiling of six Kohonen SOMs equivalent to Figure 8.3). Consistent mapping was observed for 91.4% of the data set - the percentage of correctly classified reactions in terms of EC class when the whole data set is submitted again to the SOM trained with the whole data set. (In five experiments with randomized class labels of the reactions, consistency was only 41-42%). The more scattered lyase class has not a so typical pattern of overall changes in chemical bonds, which suggests similarities with reactions of other classes. Isomerase reactions (EC 5) are particularly difficult to classify by MOLMAP descriptors, as they often consist of subtle changes in the molecular structures of reactants and products. These problems will be discussed later with a more detailed analysis of the map, and the assessment of internal consistency of the EC system.

The influence of MOLMAP size, and type of MOLMAP bond descriptors, on the consistency of EC class mapping was verified with a smaller version of the data set (encompassing reactions represented only in the direction of the KEGG reaction file - 3784 reactions) on a 29×29 SOM (Table 8.6). Variation of MOLMAP size between 225 (15×15) and 841 (29×29) affected the consistency of mapping in only 0.3 - 4.1%, depending on the type of bond descriptors used for the MOLMAPs. Topological and physico-chemical bond descriptors combined yielded the most robust MOLMAPs in such experiments. MOLMAPs were generated with four sets of bond descriptors (see Section Methods): topological, physico-chemical, topological + physico-chemical, and topological + subset of physico-chemical descriptors. For the same size of MOLMAPs, the type of descriptors never affected the consistency of mapping in more than 3%. Among the 16 tried combinations of MOLMAP size and type of bond descriptors the highest and lowest consistency of mapping differed in only 4.1%. Experiments with bond features derived from physico-chemical descriptors calculated with PETRA software (Molecular Networks GmbH, Erlangen, Germany) did not yield superior results.

A map such as the one shown in Figure 8.3 has a number of possible applications. The trend for reactions to cluster according to EC numbers allows to explore the SOM for the automatic assignment of EC numbers from the reaction equation (see below). Simultaneously, the inspection of reactions belonging to different EC classes but activating the same neuron may indicate possible inconsistencies in EC numbers, problematic classification of enzymes, similarity between reactions hidden by specific EC rules, mistaked in database entries, as well as limitations of the MOLMAP approach. A systematic analysis of such cases uncovered reactions similarities hidden by differences in EC numbers at the class level, several interesting cases, ten of which are displayed in Table 8.7 (examples 1-5) and Table 8.8 (examples 6-10). While some cases may deserve a revision of EC numbers, others illustrate problematic aspects of the application of EC rules related to reversibility of reactions, or enzymes catalysing more than one type of reactions.

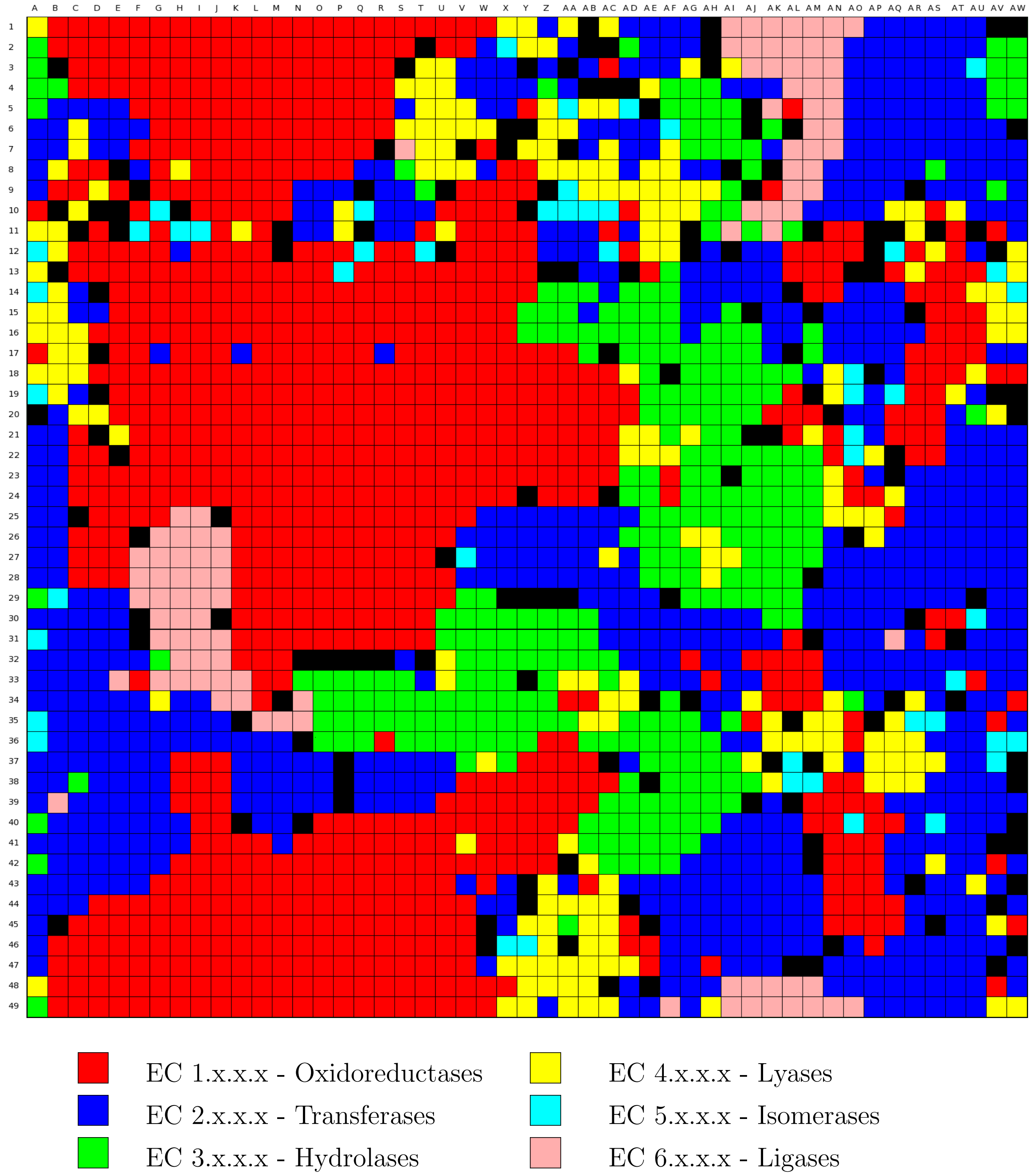


Figure 8.3: Toroidal surface of a 49×49 Kohonen SOM trained with 7482 enzymatic reactions encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. After the training, each neuron was red according to the reactions in the training set that were mapped onto it or onto its neighbors. The black neurons corresponds to ambiguous ones.

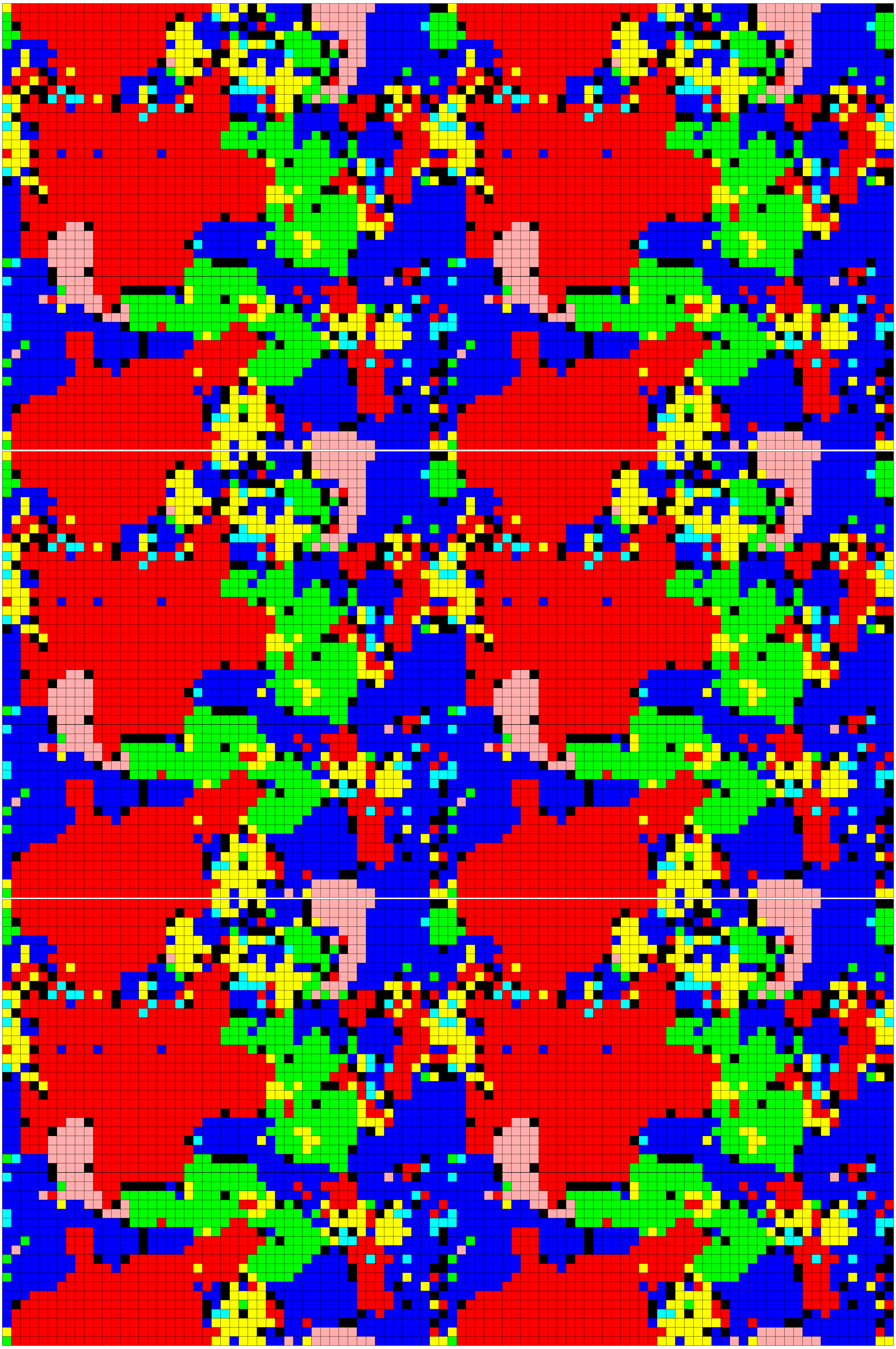


Figure 8.4: Tiling six identical toroidal surfaces of a 49×49 Kohonen SOMs equivalent to Figure 8.3.

Table 8.6: Classification of enzymatic reactions according the first digit of EC number with different sets of bond descriptors and different sizes of MOLMAPs.

Size of	% Correct predictions			
MOLMAP	Top.	P.C.	P.C.+Top.	subset P.C. + Top.
225	86.6	88.5	89.4	89.3
400	89.1	87.9	89.4	89.7
625	90.7	88.5	89.7	90.0
841	89.5	88.1	89.6	89.0

Top. - Topological bond descriptors (41 descriptors); P.C. - Physico-chemical bond descriptors (27 descriptors); P.C. + Top. - Physico-chemical bond descriptors and topological bond descriptors (27+41 descriptors); subset P.C. + Top. - Subset of physico-chemical bond descriptors and topological bond descriptors (14+41 descriptors).

The first entry of Table 8.7 corresponds to reactions R00603 and R03523 of the KEGG database, the first listed as a lyase (dichloromethane dehalogenase, EC 4.5.1.3) and the second listed as a hydrolase (alkylhalidase, EC 3.8.1.1). Surprisingly, the only difference between them is that one eliminates two chloride ions and the other eliminates one chloride and one bromide ion.

In entry 2, the two reactions are the same, only the substrates are different. They get different classes probably because the enzyme responsible for one of them (R05086, EC 4.2.3.1) also catalyses a different type of reaction.

The two overall reactions of entry 4 are essentially the same, although they are catalysed by different enzymes, and listed as belonging to different EC classes. The first is catalysed by the broad group of glutathione S-transferases (EC 2.5.1.18), and the second is catalysed by leukotriene-C4 synthase (EC 4.4.1.20). Significantly, the latter was named as 2.5.1.37 until 2004.

The two reactions of entry 5 are hydrolysis of epoxides. While the first is catalysed by hepoxilin-epoxide hydrolase (EC 3.3.2.7), the second is catalysed by a P-450 unspecific monooxygenase and the EC number 1.14.14.1 is more an identifier of the enzyme than of the catalysed reaction.

The main difference between the two overall reactions of entry 6 (Table 8.8) is that one involves a ketone and an aldehyde, while the other involves two aldehydes. Apart from that, they are formally the same. However, the first is associated with a transferase (2-hydroxy-3-oxoadipate synthase, EC 2.2.1.5) and the second with a lyase (tartronate-semialdehyde synthase, EC 4.1.1.47).

Globally, the first reaction of entry 7 is an intramolecular version of the second reaction, but the first is catalysed by a lyase (ornithine cyclodeaminase, EC 4.3.1.12) while the second is catalysed by a transferase (methylamine-glutamate N-methyltransferase, EC 2.1.1.21).

Table 8.7: Examples of reactions activating the same neuron, but labeled with different EC classes (examples 1-5).

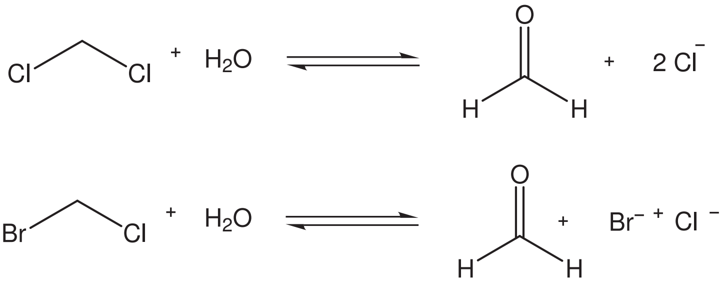
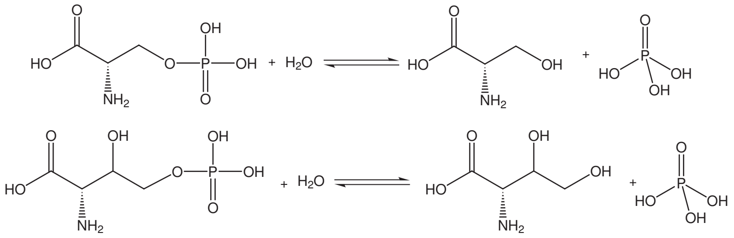
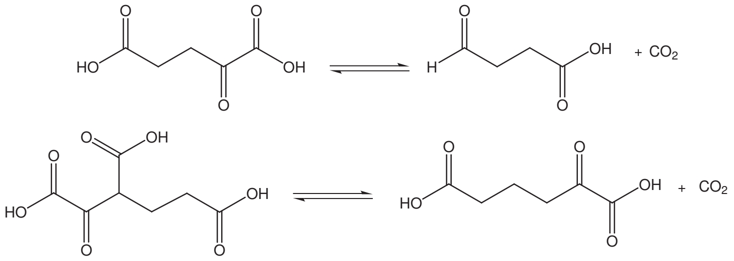
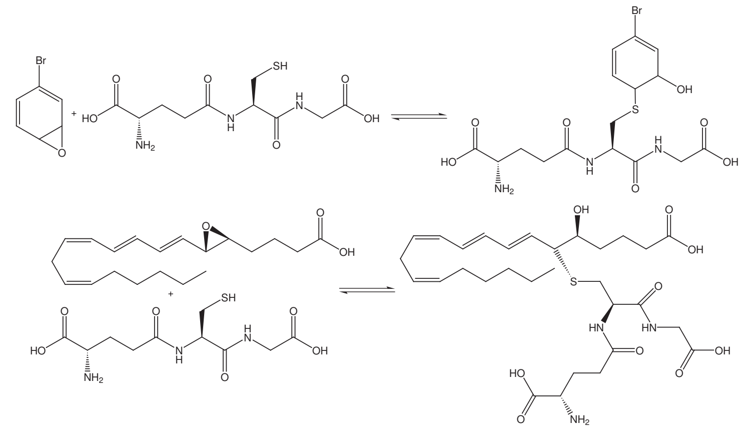
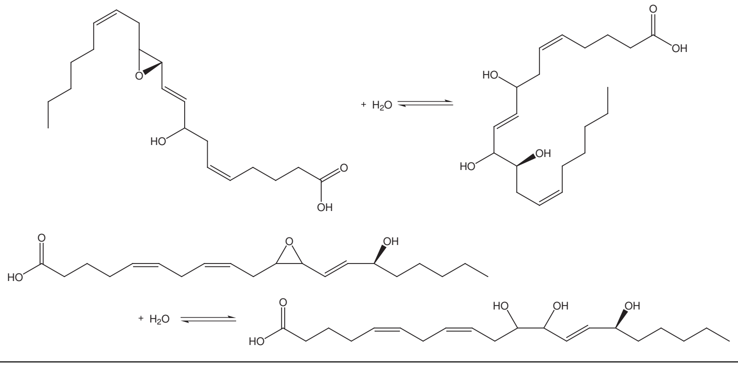
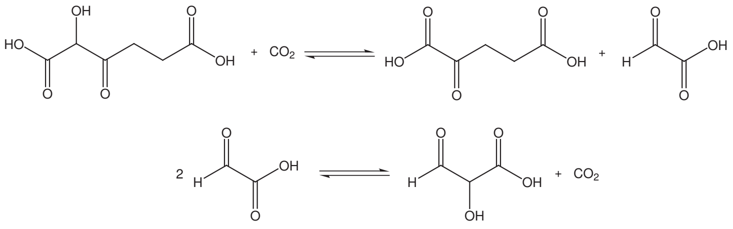
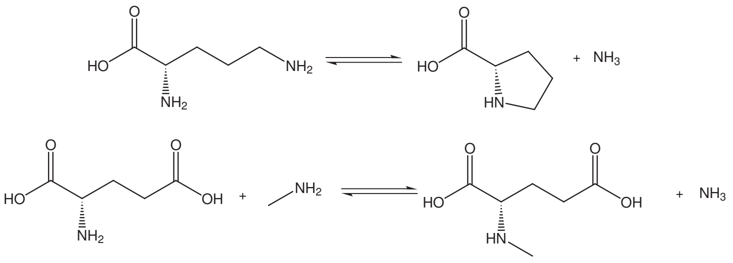
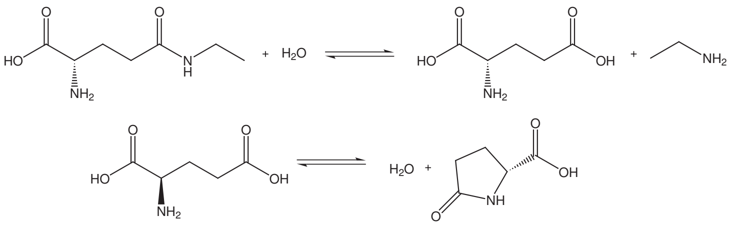
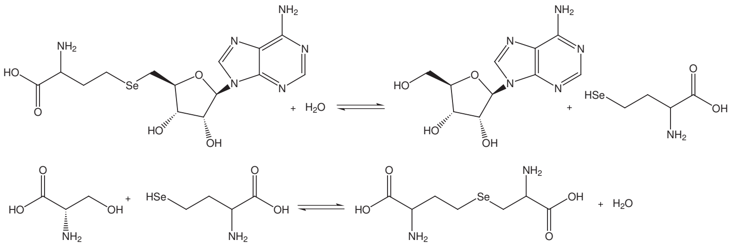
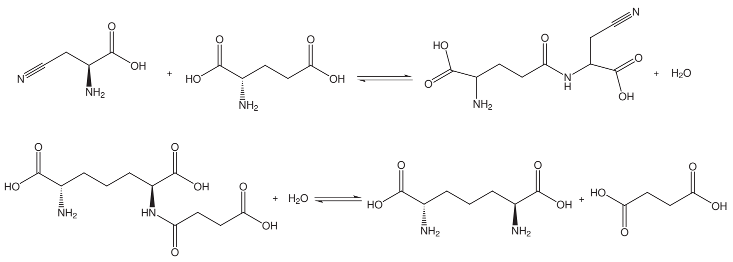
Entry	Reactions diagrams	EC Nr	KEGG identifier	Neuron
1		4.5.1.3 3.8.1.1	R00603 R03523	W:37
2		3.1.3.3 4.2.3.1	R00582 R05086	A:3
3		4.1.1.71 1.1.1.87	R00272 R01936	AF:10
4		2.5.1.18 4.4.1.20	R07069 R03059	AR:32
5		3.3.2.7 1.14.14.1	R04609 R07044	X:31

Table 8.8: Examples of reactions activating the same neuron, but labeled with different EC classes (examples 6-10).

Entry	Reactions diagrams	EC Nr	KEGG identifier	Neuron
6		2.2.1.5 4.1.1.47	R00474 R00013	AV:17
7		4.3.1.12 2.1.1.21	R00671 R01586	AV:33
8		3.5.1.65 4.2.1.48	R02930 R01583	AI:37
9		3.3.1.1 4.2.1.22	R04936 R04942	A:49
10		2.3.2.2 3.5.1.18	R03970 R02734	AK:16

[a](multistep, first oxidation of OH to ketone)

Although the two reactions of entry 8 are officially classified into different EC classes, both are formally hydrolysis of amides (the second occurring intramolecularly, and represented in the opposite direction). Such a similarity is completely hidden by their EC numbers.

The reactions of entry 9 are represented in the directions corresponding to their metabolic pathways (both are currently listed in KEGG as irreversible). The first reaction is a hydrolysis (and its EC number corresponds to a hydrolase, EC 3.3.1.1), and the second reaction involves the cleavage of a C-O bond with an EC number corresponding to a carbon-oxygen lyase (EC 4.2.1.22). However, if we consider both reactions in both directions, they are the same overall reaction. A similar situation happens in entry 10. The two reactions are currently listed in the KEGG database as irreversible, and are represented in the same directions as in the metabolic pathways to which they belong. However, if the first reaction is considered in the opposite direction, then both reactions are hydrolysis of amides. The second reaction has in fact an EC number corresponding to a hydrolase (EC 3.5.1.18), but the first is associated with a transferase (EC 2.3.2.2).

Other neurons activated by reactions of conflicting EC classifications highlighted limitations of the MOLMAP/SOM approach that perceived truly non-similar reactions as similar (Table 8.9). In many of those cases, the two reactions result in the formation of similar bonds, although the bonds broken in the reactants are different. Globally the reaction MOLMAPs bear some similarities and may activate the same neuron even if the reactants and the reactions are not similar. There are also cases in which reactants very similar to the products result in MOLMAPs with only few non-null components - two such reactions may yield globally similar (almost null) MOLMAPs, although the non-null values are different because they correspond to different types of bonds being broken, changed or formed (different types of reactions). In other situations, the overall reaction involves more than one transformation making the MOLMAP somewhat similar to that of another reaction where only one of the transformations occur. Still in other cases, two reactions activate the same neuron but one (or both) are at a high Euclidean distance to the neuron (probably due to a lack of similar reactions in the database) - the fact that the winning neuron is the most similar neuron to the MOLMAP does not imply they are very similar. Wrongly perceived similarities between reactions may also derive from wrongly perceived similarities and differences between bonds in the SOM employed for the mapping of bonds.

8.3.2 SOM-Based Assignment of EC First Digit from the Reaction Equation

The MOLMAP/SOM approach was explored to automatically assign EC numbers of reactions from the structures of reactants and products. The whole database of enzymatic

Table 8.9: Examples of conflicts detected by the SOM illustrating limitations of the MOLMAP/SOM.

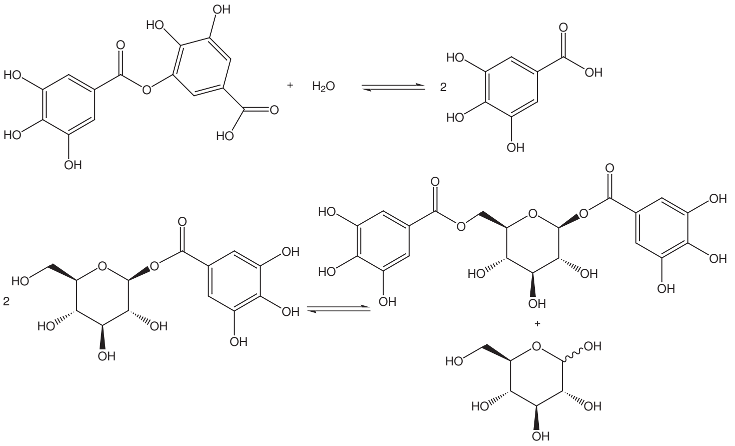
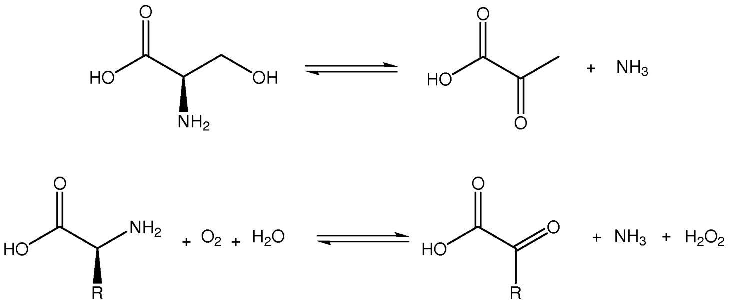
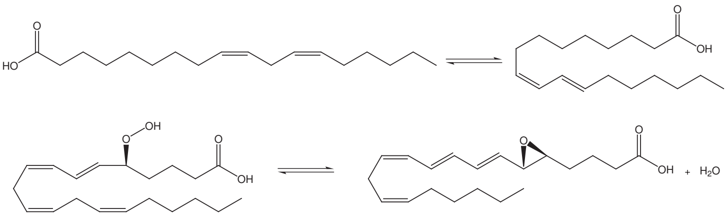
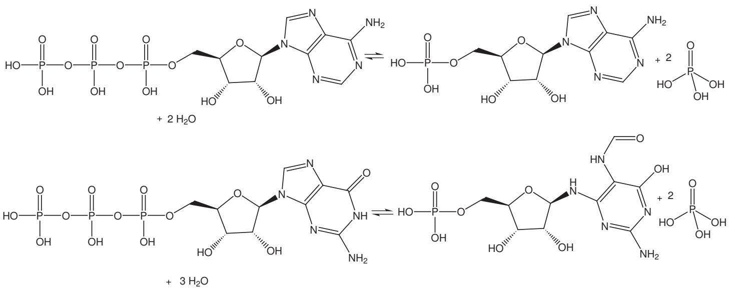
Entry	Reactions diagram	EC	KEGG	Neuron
		Nr	identifier	
1		3.1.1.20 2.1.1.90	R00053 R00049	A:45
2		5.2.3.1 1.13.11.34	R03627 R03058	H:10
3		4.3.1.18 1.4.3.2	R00221 R01259	U:3
4		3.6.1.5 3.5.4.29	R00085 R07306	N:33

Table 8.10: SOM assignment of the first digit of EC numbers.

Data sets		Nr of	% Correct predictions		
		reactions	Best ind. SOM	Ens. of five SOMs	Ens. of ten SOMs
	Training 1	5855	92.1	94.6	95.1
Partition	Test 1	1646	79.6	83.4	84.3
1	Training 2	5855	95.4	96.3	96.4
	Test 2	1646	85.4	86.3	85.9
Partition	Training	5246	88.5	93.0	93.9
2	Test	2236	88.3	91.4	91.7
Partition	Training	350	86.0	94.3	96.0
	Test 1	4896	66.7	72.7	74.4
3	Test 2	7132	68.5	74.4	75.7
Partition	Training	310	74.8	90.0	92.6
4	Test	40	67.5	65.0	67.5

reactions was partitioned into several training and test sets. Test sets were not presented to the networks until the training was finished. MOLMAPs of size 625 generated with topological and physico-chemical bond descriptors were employed.

In a first experiment (Partition 1), training and test sets were selected with a 49×49 Kohonen SOM. The SOM was trained with all reactions, then one reaction was randomly taken from each occupied neuron and moved to the test set, resulting in a training set with 5855 and a test set with 1646 reactions. Correct predictions of the EC class were achieved for up to 79.6% of the test set (Table 8.17). In order to overcome random fluctuations, and to improve predictive ability, consensus predictions were obtained with ensembles of five or ten independent SOMs trained with the same data set. An increased accuracy of predictions was in fact observed for the training set, and then also for the test set (84.3%), particularly for the lyase class. The number of wrong classifications obtained with an individual SOM and an ensemble of ten SOMs is similar - the improvement in the number of correct predictions using an ensemble mainly derives from reactions that were classified as undecided with a single SOM and became correctly classified by the ensemble. The confusion matrix for the test set (Table 8.11) show a higher prediction accuracy for classes EC 2 and EC 3, and the worse predictions for classes EC 4 and EC 5. Reactions catalysed by isomerases (EC 5) are generally more problematic as very often they involve no substantial structural changes, and they are also less represented in the data set. The results (confirmed by the map of Figure 8.3) reveal that MOLMAP patterns for reactions catalysed by lyases (class EC 4) are not so well defined, and most frequently confused with those corresponding to hydrolases and transferases. Significantly, all but two of the examples highlighted in Tables 8.7 and 8.8 involve lyases.

Table 8.11: Confusion matrix for the classification of enzymatic reactions according the first digit of the EC number (test set of partition 1; ensemble of ten 49×49 Kohonen SOMs).

	EC 1	EC 2	EC 3	EC 4	EC 5	EC 6	X	% Correct predictions
EC 1	573	31	11	21	3	4	8	88.0
EC 2	13	414	13	9	3	1	6	90.2
EC 3	1	10	214	6	0	1	5	90.3
EC 4	11	20	19	115	2	1	13	63.5
EC 5	5	13	2	11	19	0	4	35.2
EC 6	0	5	2	2	0	52	3	81.2

Partition 2 also involved a strategy for the training set to cover as much the reaction space as possible - one reaction of each available EC number (full EC number) was selected into the training set (5246 reactions), and the remaining reactions were moved to the test set (2236 reactions). This partition guaranteed that all the full EC numbers in the test set were represented in the training set, which generally means a high similarity between reactions in both sets. In fact, assignment of the EC class was obtained with an accuracy of 91.7% for the test set. For this and the next partitions, the two entries of each reaction (in the two opposite directions) were included in the same set (training or test).

Then experiments were performed with lower similarities between training and test sets. With Partition 3, the model was trained only with one reaction from each sub-subclass (first three digits of the EC number, 350 reactions), and tested with one reaction from each of the remaining full EC numbers (4896 reactions). In this way, all reactions in the training set belong to a different sub-subclass, and all sub-subclasses were represented. The test set includes only reactions with full EC numbers non-available in the training set. Despite the test set being 14 times larger than the training set, and the exclusion of similarities at the level of the full EC number, 74.4% of the test set could still be correctly predicted in terms of EC class by a 25×25 SOM. An even more stringent test was performed with Partition 4, where all the reactions of the test set belonged to different sub-subclasses of those in the training set. The map for the training set is displayed in Figure 8.5. From the set of 350 reactions with no duplicated sub-subclass, a test set of 40 reactions was randomly selected that could be predicted with 67.5% accuracy by a 20×20 SOM.

Not surprisingly, the results indicate that the EC class can be better predicted for unseen reactions if similar reactions exist in the training set. On that basis, we explored the possibility of obtaining some measure of reliability associated to each prediction. Tables 8.12 and 8.13 shows the performance of two such measures, the Euclidean distance to the winning neuron, and the number of votes in the consensus prediction by the ensemble of SOMs, for the test set of Partition 1.

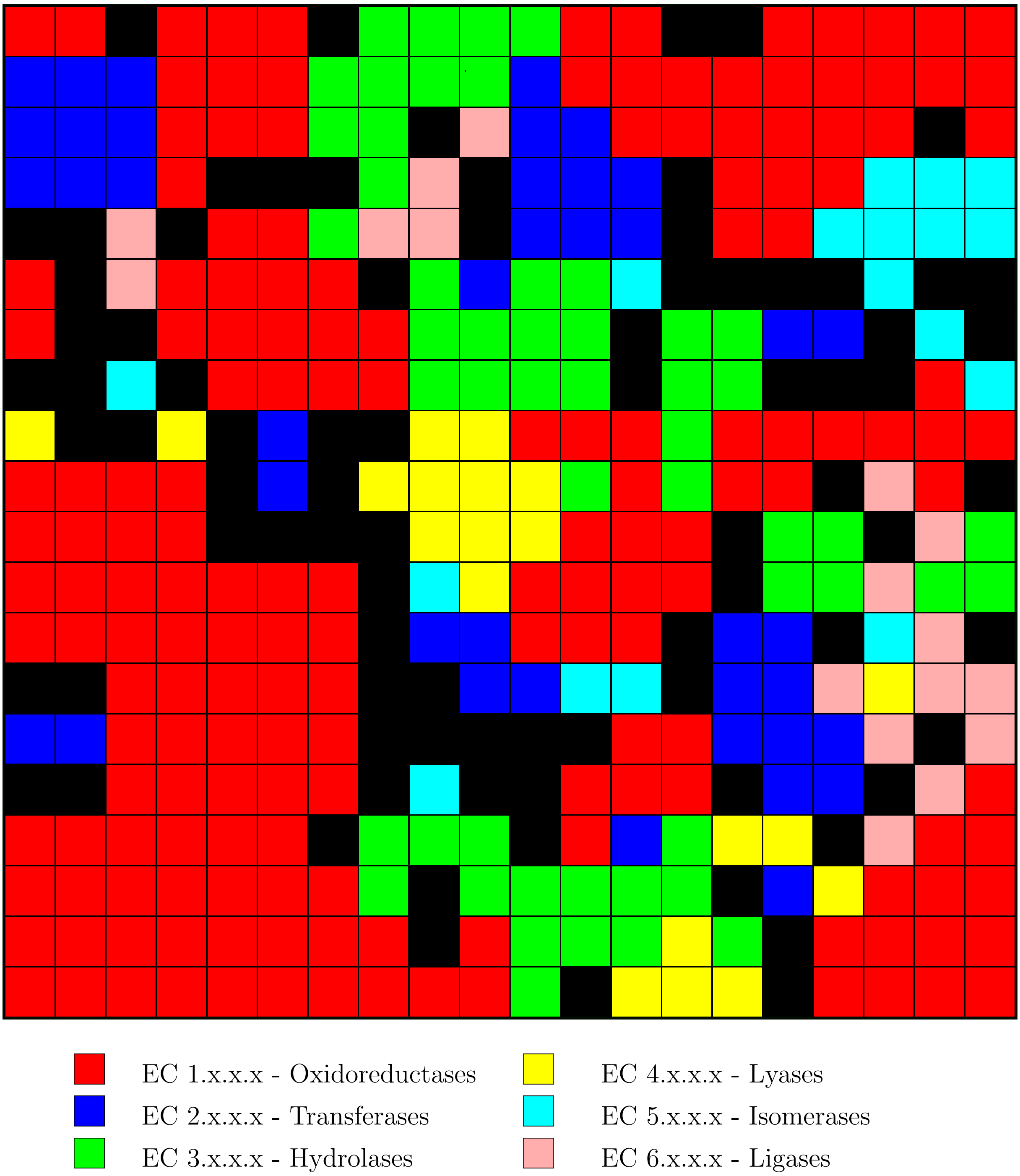


Figure 8.5: Toroidal surface of a 20×20 Kohonen SOM trained with 310 enzymatic reactions (all from different sub-subclasses) encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors. The black neurons are ambiguous ones.

Table 8.12: Relationship between prediction accuracy and the Euclidean distance of the reaction MOLMAP to the winning neuron (test set of partition 1; classification according to the first digit of the EC number).

Euclidean Distance[a]	N. of reactions	% of total reactions	% Correct predictions
< 50	14	0.9	100
< 100	26	1.6	100
< 500	186	11.3	94.1
< 1000	422	25.6	94.8
$\geq$1000 and < 2000	418	25.4	84.2
$\geq$2000 and < 3000	305	18.5	73.1
$\geq$3000 and < 5000	280	17.0	72.9
$\geq$5000	221	13.4	59.3

[a] - Euclidean distance (in fact the numbers displayed in the Table correspond to 10 times the squared of the Euclidean Distance) between the reaction MOLMAP (vector of reaction descriptors) and the winning neuron (neuron where the reaction was mapped).

Table 8.13: Relationship between prediction accuracy and the number of votes for the predicted class in an ensemble of ten Kohonen SOMs (test set of partition 1; classification according to the first digit of the EC number).

	Ten votes		Nine votes		Eight votes	
	Nr. reactions	Nr. correct	Nr. reactions	Nr. correct	Nr. reactions	Nr. correct
EC 1	482	475 (98.6)	47	44 (93.6)	20	18 (90.0)
EC 2	298	288 (96.4)	60	47 (78.3)	31	24 (77.4)
EC 3	148	143 (96.6)	43	33 (76.7)	16	10 (62.5)
EC 4	35	29 (82.9)	18	14 (77.8)	26	17 (65.4)
EC 5	5	5 (100)	1	1 (100)	4	4 (100)
EC 6	39	37 (94.9)	6	5 (83.3)	6	5 (83.3)
Total	1007	977 (97.0)	175	144 (82.3)	103	78 (75.7)

	Seven votes		Six votes		Five votes	
	Nr. reactions	Nr. correct	Nr. reactions	Nr. correct	Nr. reactions	Nr. correct
EC 1	11	9 (81.8)	16	12 (75.0)	12	8 (66.7)
EC 2	36	21 (58.3)	23	14 (60.9)	22	10 (45.5)
EC 3	7	4 (57.1)	18	9 (50.0)	22	11 (50.0)
EC 4	22	17 (77.3)	23	16 (69.6)	22	9 (40.9)
EC 5	2	2 (100)	1	1 (100)	4	1 (25.0)
EC 6	2	2 (100)	1	1 (100)	3	1 (33.3)
Total	80	55 (68.8)	82	53 (64.6)	85	40 (47.1)

With the ensemble of 10 SOMs, and for the test set, 97% of the predictions obtained with 10 votes are correct, the percentage decreasing to 82.3%, 75.7%, 68.8%, 64.6%, and 47.1% for predictions obtained with 9, 8, 7, 6, and 5 votes, respectively. The Euclidean distance between the winning neuron and the MOLMAP of the query reaction also performed well. The percentage of correct predictions gradually decreased from 100% for reactions at an Euclidean distance <100, to 59.3% for reactions at a distance > 5000.

Another independent test was performed using 930 reactions in the KEGG database with incomplete EC numbers. These are often problematic cases, and are therefore expected to present an increased level of difficulty. The ensemble of 10 SOMs trained with all the 7482 reactions with full EC numbers was able to correctly predicted the EC class for 73.8% of such cases.

Although SOMs are trained in an unsupervised manner, it is possible to render the training supervised by adding new descriptors encoding the classes of the objects. In our case, six new descriptors were appended to the MOLMAP descriptors, each one corresponding to an EC class. For one reaction, five of the descriptors were zero, and the descriptor corresponding to the EC class took a value of 80. This increases the overall similarity between reaction descriptors of the same class, forcing the SOM to cluster according to class (Figure 8.6). After the training, for the SOM to map a new reaction (possibly of unknown class), the six layers of the network corresponding to the class codes are not used to determine the winning neuron. For Partition 1, such a technique allowed a single SOM to improve the accuracy of predictions for the test set up to 85.4% (similar to the results obtained with the ensemble of ten unsupervised SOMs), but ensembles of supervised SOMs could only marginally improve this number.

8.3.3 SOM-Based Assignment of EC Second and Third Digits from the Reaction Equation

SOMs were also used to classify reactions according to the second and third digits of the EC number (subclass and sub-subclass). Independent networks were trained for different classes. For the sub-subclass experiments concerning classes EC 1 and EC 2, CPNNs were used instead of SOMs due to the larger number of sub-subclasses to classify - see Section Methods. Each output neuron of the CPNN had as many weights as the number of the sub-subclasses in the data set.

The data sets were partitioned into training and test sets in two different ways. In Partition 1 the test set included one (random) reaction of each subclass (or sub-subclass), and the training set the remaining reactions. Accurate predictions of subclass were achieved for 62% of the test set (42% for EC 1 reactions, 88% for EC 2, 63% for EC 3, 57% for EC 4, 100% for EC 5, and 80% for EC 6). Correct prediction of the sub-subclass was obtained for 52% of the test set. In Partition 2 the test set was selected with a SOM as

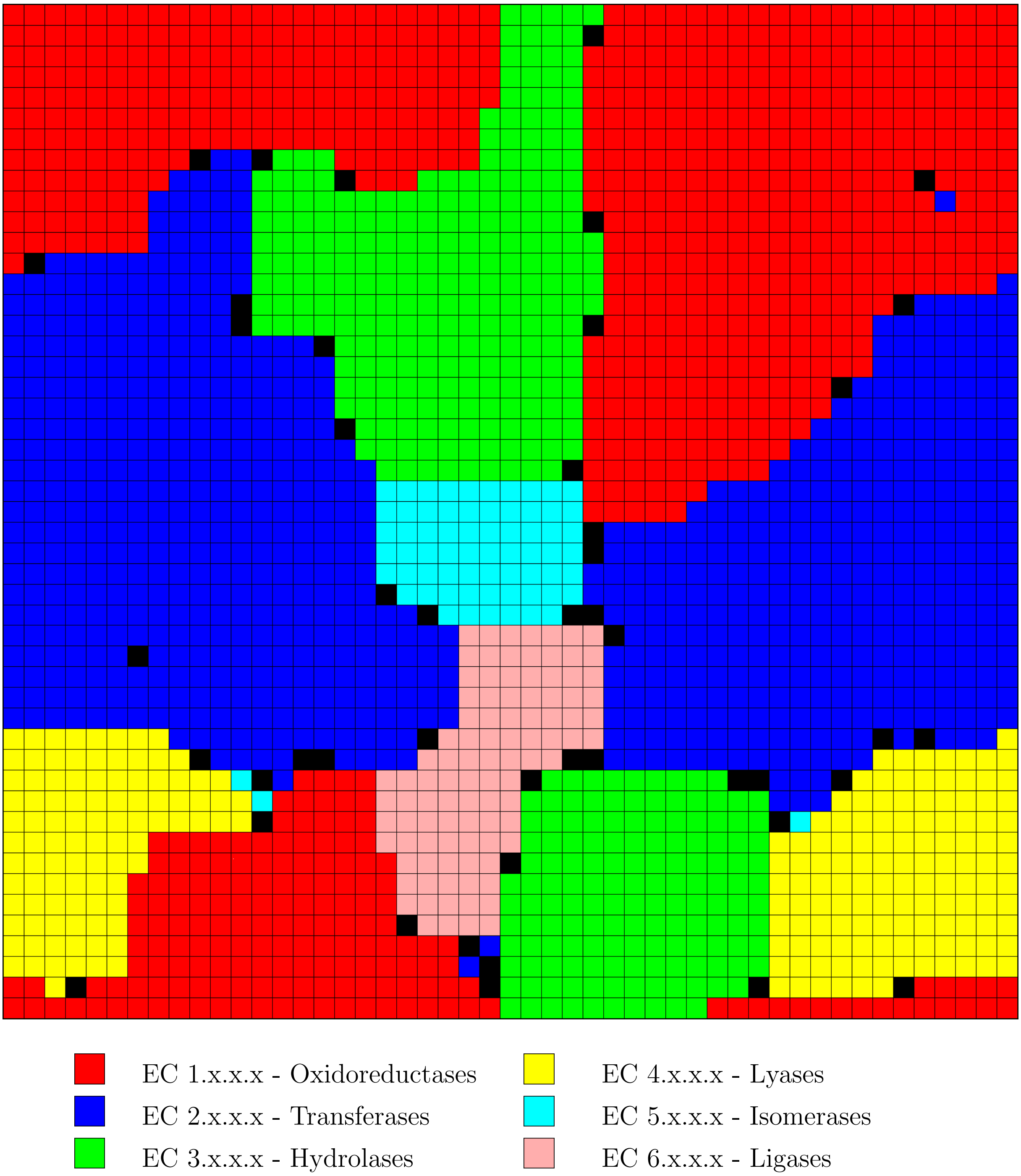


Figure 8.6: Toroidal surface of a 49×49 Kohonen SOM trained with 5855 enzymatic reactions encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors and six additional descriptors with the value of 80 or 0 according the class of reaction. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors. The black neurons are ambiguous ones.

Table 8.14: SOM assignment of the second digit of EC numbers.

	% Correct predictions (number of reactions)		
	All reactions	Partition 1 Training / Test	Partition 2 Training / Test
EC 1	90.7 (2584)	91.6/41.7 (2548/36)	93.5/74.3 (1805/779)
EC 2	95.5 (2436)	94.9/87.5 (2420/16)	96.1/86.1 (1759/677)
EC 3	96.7 (1262)	96.0/62.5 (1246/16)	96.4/87.4 (912/350)
EC 4	89.3 (666)	94.8/57.1 (652/14)	90.1/69.5 (453/213)
EC 5	93.9 (198)	91.2/100 (192/6)	94.6/70.6 (130/68)
EC 6	97.6 (296)	97.9/80.0 (286/10)	99.5/92.1 (208/88)

All reactions - Whole data set of reactions for each class; Partition 1 - test set with one reaction of each subclass randomly chosen and training set with the remaining reactions; Partition 2 - The data set was partitioned into training and test sets using a Kohonen SOM. The SOM was trained with all reactions and, after the training, one reaction was randomly selected from each occupied neuron, and moved to the test set.

for the experiments at the class level. While the second test set has a similar distribution of subclasses (or sub-subclasses) to the whole data set, in the test set of Partition 1 the subclasses (or sub-subclasses) with smaller number of reactions have proportionally more impact on the calculated prediction accuracy. Accurate predictions of subclass were achieved for 80% of the test set of Partition 2 (74% for EC 1 reactions, 86% for EC 2, 87% for EC 3, 70% for EC 4, 71% for EC 5, and 92% for EC 6). Correct prediction of the sub-subclass was obtained for 70% of the test set. Tables 8.14 and 8.15 show the number of reactions in each training and test sets, and detail the predictions for the second and third digits of the EC number. The size of the maps was chosen such that the number of reactions is approximately twice the number of neurons. Maps of size 35×35, 35×35, 25×25, 18×18, 10×10 and 12×12 were used for classes EC 1, EC 2, EC 3, EC 4, EC 5, and EC 6. CPNNs of size 35×35 were used to assign sub-subclasses of the EC 1 and EC 2 classes.

The general trend for Partitions 2 to yield better predictions than Partitions 1 suggests that predictions are generally easier for subclasses (and sub-subclasses) with more examples available (in the test sets of Partitions 2 these are proportionally more abundant). It is thus expected that EC numbers can be more easily predicted when similar reactions are known catalysed by the same enzyme with other substrates. Eventhough, 62% and 52% of the reactions in the test set (respectively at the subclass and sub-subclass levels) could be correctly predicted in the more severe situations of Partitions 1. Inspection of the results for individual subclasses (or sub-subclasses) reveals in general a lower percentage of correct predictions for subclasses (or sub-subclasses) with a smaller number of reactions in the data set. While the experiments at the class level with Partition 4 assessed

Table 8.15: SOM assignment of the third digit of EC numbers.

	% Correct predictions (number of reactions)		
	All reactions	Partition 1 Training / Test	Partition 2 Training / Test
EC 1	75.1 (2474)	76.7/30.0 (2394/80)	77.7/65.4 (1864/610)
EC 2	84.8 (2412)	82.8/72.7 (2368/44)	85.0/73.9 (1960/452)
EC 3	92.7 (1238)	91.3/73.7 (1200/38)	93.7/75.7 (859/379)
EC 4	86.9 (658)	87.2/26.9 (632/26)	90.4/65.3 (439/219)
EC 5	88.3 (180)	87.4/85.7 (166/14)	91.4/57.8 (116/64)
EC 6	93.6 (296)	92.8/66.7 (278/18)	95.6/75.8 (205/91)

All reactions - Whole data set of reactions for each class; Partition 1 - test set with one reaction of each sub-subclass randomly chosen and training set with the remaining reactions; Partition 2 - The data set was partitioned into training and test sets using a Kohonen SOM. The SOM was trained with all reactions and, after the training, one reaction was randomly selected from each occupied neuron, and moved to the test set.

the ability of the MOLMAPs to identify similarities between sub-subclasses of the same class, these experiments with classifications at the sub-subclass level assess the ability to discriminate between different sub-subclasses. The surfaces of the best SOMs obtained for the classification into subclasses of the six EC classes were illustrated in the Figure 8.7, as well as the maps of the reactions of the EC 3, EC 4, EC 5, EC 6 classes colored according the sub-subclasses in Figure 8.8.

8.3.4 Classification of Racemases and Epimerases Reactions.

The connectivity of reactants is not changed in isomerase reactions of subclass 5.1 (racemases and epimerases) - only stereochemical changes occur. MOLMAPs can not perceive stereochemical features, and therefore cannot represent such reactions. We retrieved 18 reactions of subclass 5.1 from the BioPath database, where stereochemistry was assigned to chiral structures, and tried to represent reactions using chiral descriptors. Chirality codes [185] were used as molecular descriptors instead of MOLMAPs. Reactions were represented by the difference of the chirality code of the products and the chirality code of the reactants. Then a SOM was trained to assess the ability to distinguish between sub-subclasses of racemases and isomerases (Figure 8.9).

Each reaction was included twice, corresponding to the two directions. Although the data set is rather small, the map shows some separation between the two main sub-subclasses (EC 5.1.1 acting on aminoacids and derivatives, and EC 5.1.3 acting on carbohydrates and derivatives). Within each of these sub-subclasses two regions are differenti-

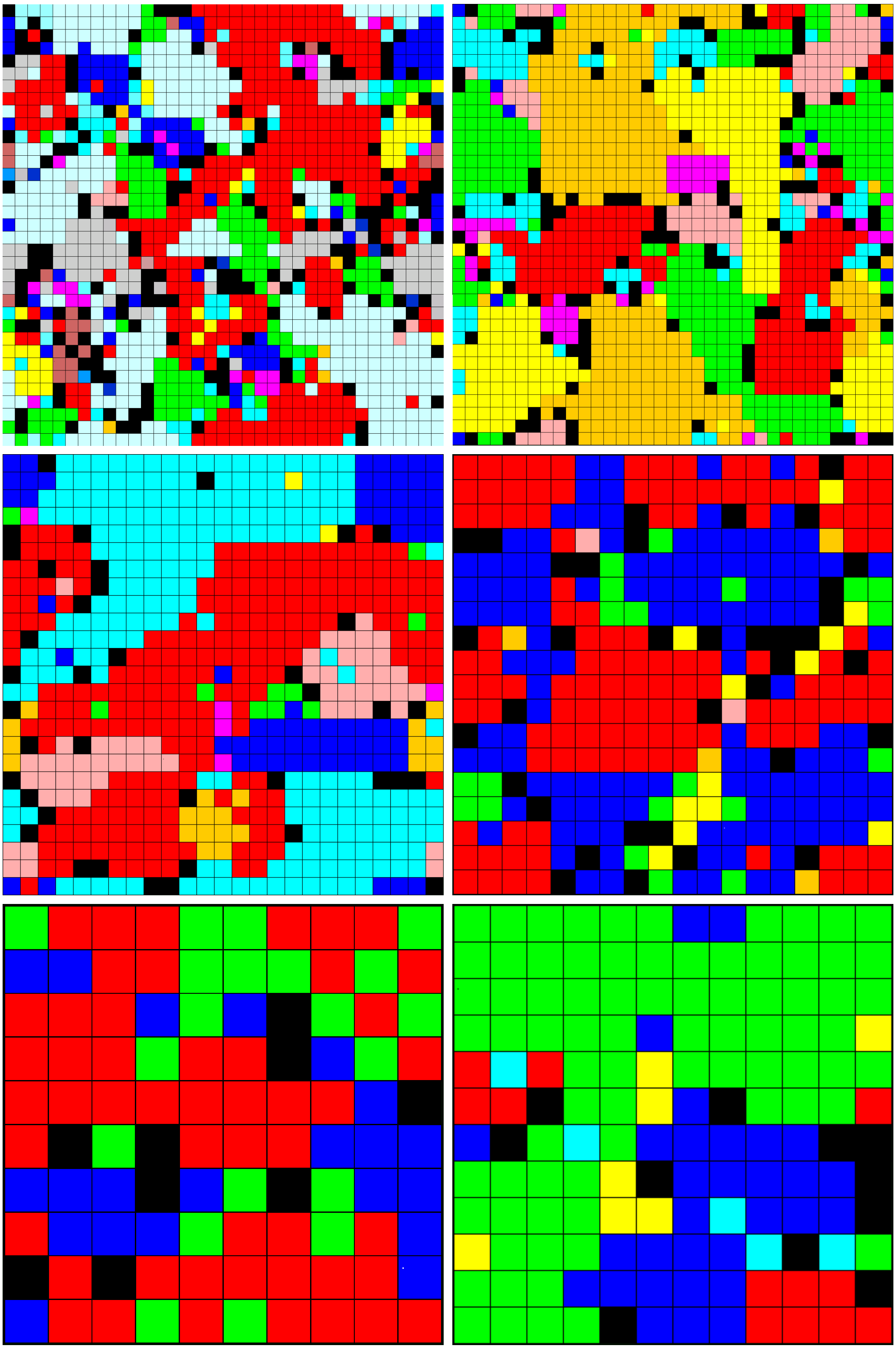


Figure 8.7: First row: Toroidal surfaces of a 35×35 Kohonen SOMs. Left) Kohonen SOM trained with 2584 oxidoreductases (EC 1); Right) Kohonen SOM trained with 2436 transferases (EC 2). Second row: Left) Toroidal surface of a 25×25 Kohonen SOM trained with 1262 hydrolases (EC 3); Right) Toroidal surface of a 18×18 Kohonen SOM trained with 666 lyases (EC 4); Third row: Left) Toroidal surface of a 10×10 Kohonen SOM trained with 198 isomerases (EC 5); Right) Toroidal surface of a 12×12 Kohonen SOM trained with 296 ligases (EC 6); In all cases the enzymatic reactions were encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors.

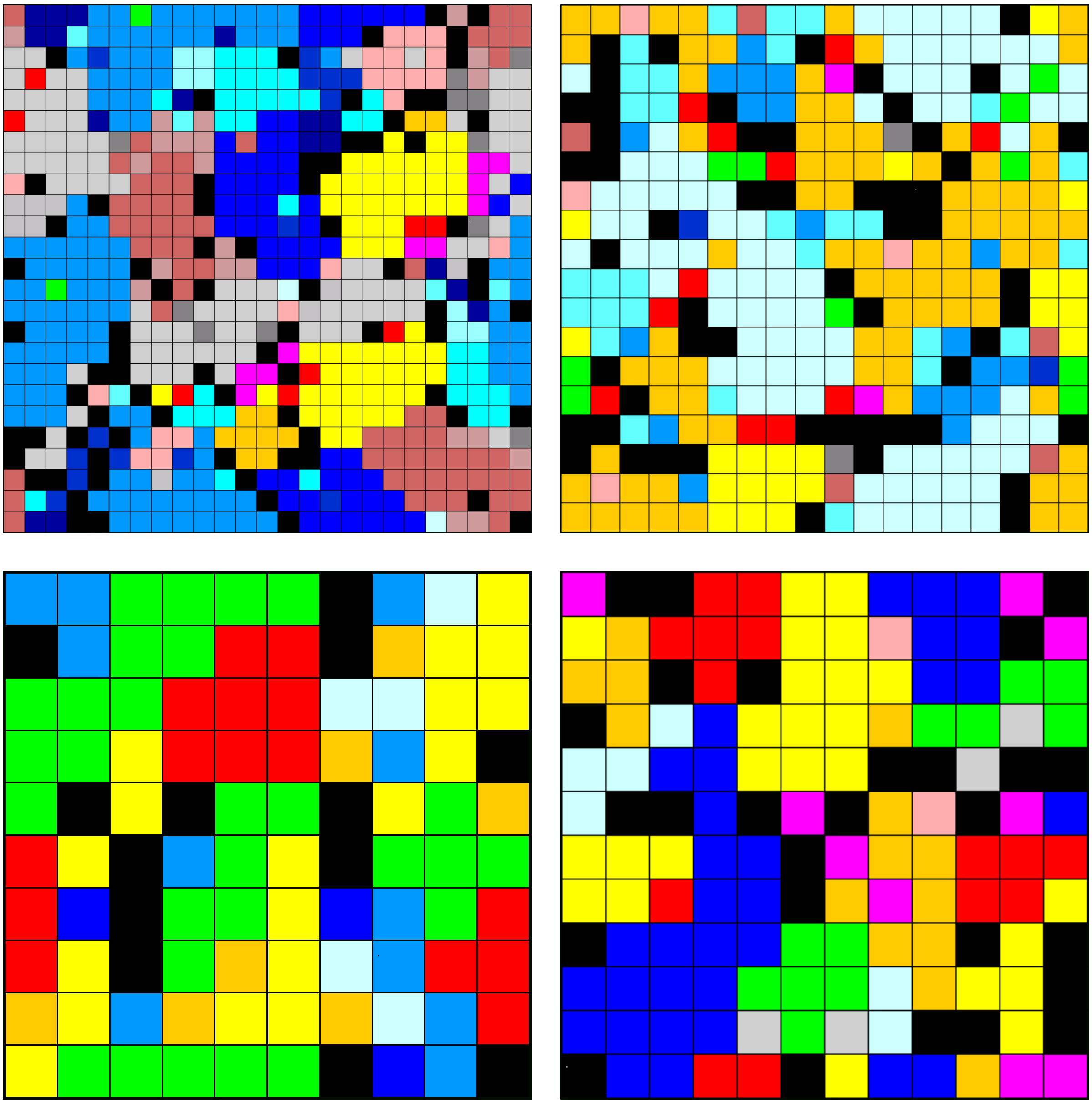
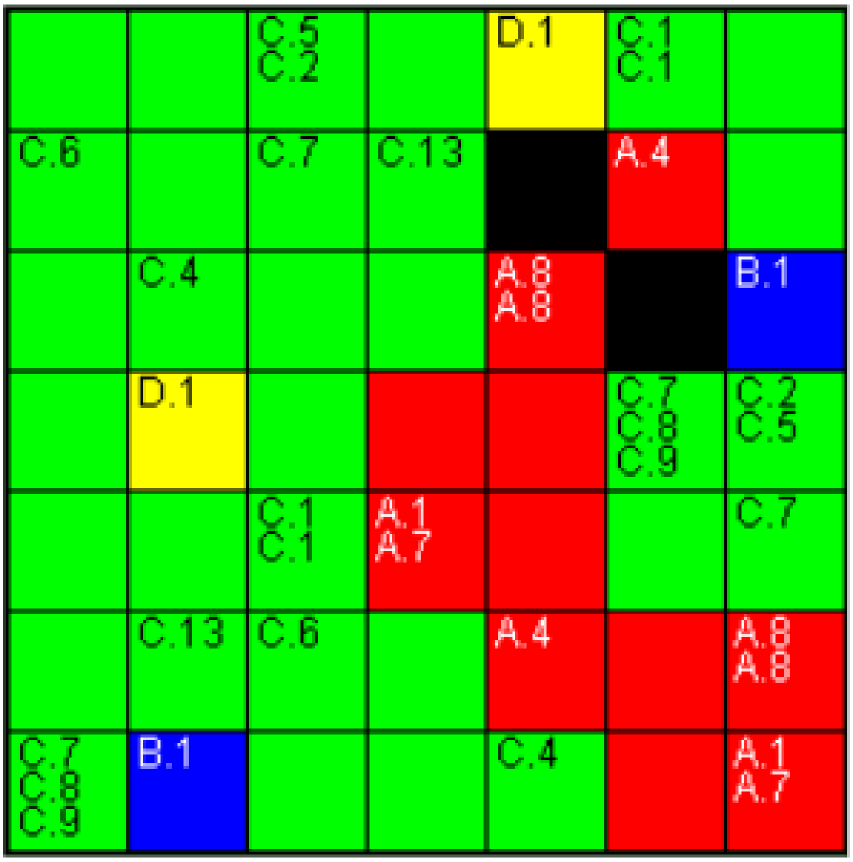


Figure 8.8: First row: Left) Toroidal surface of a 25×25 Kohonen SOM trained with 1238 hydrolases (EC 3); Right) Toroidal surface of a 18×18 Kohonen SOM trained with 658 lyases (EC 4); Second row: Left) Toroidal surface of a 10×10 Kohonen SOM trained with 180 isomerases (EC 5); Right) Toroidal surface of a 12×12 Kohonen SOM trained with 296 ligases (EC 6); In all cases the enzymatic reactions were encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. After the training, each neuron was colored according to the reactions in the training set that were mapped onto it or onto its neighbors.



🟥 EC 5.1.1.x - Acting on amino acids and derivatives

🟦 EC 5.1.2.x - Acting on hydroxy acids and derivatives

🟩 EC 5.1.3.x - Acting on carbohydrates and derivatives

🟨 EC 5.1.99.x - Acting on other compounds

Figure 8.9: Mapping of isomerase and racemase reactions (subclass EC 5.1) encoded by chirality codes. A (red) - sub-subclass 5.1.1 - acting on amino acids and derivatives; B (blue) - sub-subclass 5.1.2 - acting on hydroxy acids and derivatives; C (green) - sub-subclass 5.1.3 - acting on carbohydrates and derivatives; D (yellow): sub-subclass 5.1.99 - acting on other compounds. The activated neurons are identified with the labels corresponding to the sub-subclass and the last digit of the EC number.

ated corresponding to the two directions of the reactions.

8.3.5 Evaluation of the Impact of MOLMAP Parameters (Size and Bond Descriptors) on RFs Learning.

The influence of MOLMAP size and type of MOLMAP bond descriptors on the accuracy of the predictions for the four levels of EC numbers were evaluated with the data set of enzymatic reactions represented only in the direction of the KEGG reaction file - 3784 reactions.

Random Forests were trained with MOLMAPs of dimension 49, 100, 225, 400, 625 and 841 each one generated using the four sets of bond descriptors (topological, physico-chemical, topological + physico-chemical, topological + subset of physico-chemical). Experiments were performed for the assignment of the four levels of the EC number, and the results were analysed using the internal cross validation of the RF obtained by out-of-bag (OOB) estimation for the training set, or the predictions for a test set. The results are displayed in Table 8.16.

In general the results for the test set were ∼5% worse than the OOB results for the total data sets or for the training set. The following discussion is based on the results of the internal cross validation (OOB estimation) for the training sets. MOLMAPs of size 49 (7×7) yielded the worst results. This size does not provide sufficient resolution to distinguish between the different types of chemical bonds, and so to encode the different types of chemical changes operated in the reactants by a reaction. The variation of the MOLMAP size between 100 (10×10) and 841 (29×29) affected the accuracy of the predictions (the best result compared to the worst) in 4.1, 8.0, 10.7 and 17.3% for the four levels of the EC hierarchy depending on the type of bond descriptors. In experiments performed with the same MOLMAP size (100), the type of descriptors affected the accuracy at most in 2.8, 3.2, 4.3, and 12.4% for the four levels of the EC number, respectively. With MOLMAPs of size 400, the accuracy of the predictions was only affected in 0.5, 2.0, 1.1 and 5.3%.

With MOLMAPs of size larger than 100, topological descriptors generally perform poorer than the others. The best results were obtained using the combination of topological descriptors and subsets of physico-chemical descriptors for MOLMAPs of sizes 625 and 841. No significant benefit was observed from increasing the MOLMAP size from 625 to 841. MOLMAPs of size 625 using topological descriptors and the subset physico-chemical bond descriptors were thus chosen for the subsequent experiments. This combination yielded an error of 6.3, 13.8, 14.7 and 23.5% for the assignment of the class, subclass, sub-subclass and full EC number, respectively. When applied to the test sets, this system yielded predictions with errors ∼5% higher than OOB estimations for the training sets, except for the full EC number (for which the error was less than for the training set).

Table 8.16: Impact of MOLMAP parameters on the accuracy of EC number assignment by RF.

		% Uncorrect Predictions											
		1^{st}level			2^{nd}level			3^{rd}level			4^{th}level		
		all	tr	te	all	tr	te	all	tr	te	all	tr	te
49	T	19.32	19.42	22.61	33.40	33.89	35.42	37.10	37.77	41.98	48.83	51.31	49.55
	PC1	11.44	10.96	17.20	19.95	19.78	25.32	23.26	23.91	30.41	27.71	31.87	28.83
	PC1T	18.23	18.00	21.82	28.21	28.63	30.77	33.97	33.20	38.02	45.16	46.23	49.55
	PC2T	20.35	20.09	23.73	33.79	34.68	37.34	37.95	38.01	43.14	47.80	49.21	51.35
100	T	10.17	9.66	14.65	18.25	18.41	23.72	23.78	24.08	30.41	35.63	38.35	33.33
	PC1	9.12	8.75	15.45	16.50	17.52	20.99	19.56	20.91	25.12	22.58	26.97	24.32
	PC1T	7.53	7.38	12.10	18.20	19.08	20.51	24.30	24.72	30.41	33.58	36.08	39.64
	PC2T	10.60	10.17	15.29	20.43	20.67	24.68	25.01	25.24	31.40	36.51	39.40	35.14
225	T	8.27	8.24	12.42	15.14	15.48	19.23	18.71	19.50	23.64	27.71	30.30	27.03
	PC1	7.43	7.16	12.58	14.35	15.16	18.43	15.83	17.78	21.65	20.38	26.27	18.02
	PC1T	6.50	6.27	10.83	13.26	13.54	17.63	16.04	16.99	21.49	23.02	27.50	23.42
	PC2T	6.45	6.18	10.67	14.16	14.94	18.91	17.62	18.59	23.97	23.75	31.70	23.42
400	T	6.37	6.46	10.03	12.59	13.82	16.51	16.32	16.77	20.83	24.93	28.37	24.32
	PC1	6.92	6.81	11.15	13.71	14.81	18.11	15.77	16.99	21.98	19.35	23.82	18.02
	PC1T	6.21	6.31	9.55	12.11	12.80	16.19	14.41	15.92	20.33	19.65	23.12	20.72
	PC2T	6.53	6.43	9.87	12.54	13.22	17.63	15.45	15.99	20.17	20.82	25.39	18.92
625	T	6.34	6.34	9.55	12.49	13.25	16.67	16.07	16.84	20.99	24.19	29.42	25.23
	PC1	7.16	7.54	11.46	13.42	14.87	18.39	15.36	16.37	21.82	19.21	24.17	15.32
	PC1T	6.05	6.21	9.55	12.33	13.06	17.31	14.14	14.87	20.50	19.21	24.17	15.32
	PC2T	6.32	6.34	10.99	12.01	13.76	16.83	13.70	14.68	20.99	19.06	23.47	16.22
841	T	6.63	6.40	10.67	13.04	13.57	17.31	16.18	17.09	21.65	23.46	27.50	25.23
	PC1	6.42	6.65	10.99	13.15	14.43	18.11	14.85	16.11	21.16	19.35	24.84	17.12
	PC1T	6.40	6.18	10.19	12.06	12.71	16.99	13.67	14.51	20.00	18.04	22.07	18.02
	PC2T	6.21	6.12	10.51	11.90	12.74	17.15	13.81	14.88	19.67	17.74	23.12	16.22

T - Topological bond descriptors (41 descriptors); PC1 - Physico-chemical bond descriptors (27 descriptors); PC1T - Physico-chemical bond descriptors and topological bond descriptors (27+41 descriptors); PC1T - Subset of physico-chemical bond descriptors and topological bond descriptors (14+41 descriptors).

Table 8.17: Classification of enzymatic reactions by RFs.

Data sets		Uncorrect Predictions / %			
		1^{st}level	2^{nd}level	3^{rd}level	4^{th}level
All	data	6.19 (7482)	11.07 (7442)	13.25 (7258)	16.69 (1342)
Partition	Training 1	5.84 (5855)	11.62 (5794)	14.03 (5659)	-
1	Test 1	9.17 (1646)	15.95 (1624)	18.60 (1575)	-
Partition	Training	7.68 (5246)	15.35 (5206)	17.34 (5046)	-
2	Test	5.05 (2236)	10.24 (2236)	15.05 (2212)	-
Partition	Training	28.57 (350)	- (320)	-	-
3	Test 1	20.08 (4896)	49.51 (4886)	-	-
(subsub)	Test 2	18.28 (7132)	48.16 (7122)	-	-
Partition	Training	29.68 (310)	-	-	-
4	Test	27.50 (40)	-	-	-
Partition	Training 1	-	-	-	19.43 (1122)
5	Test 1	-	-	-	14.09 (220)

Partition 1 - The data set of 7501 reactions was partitioned into training and test sets using a 49×49 Kohonen SOM. The SOM was trained with all reactions and after the training one reaction was randomly selected from each occupied neuron and moved to the test set resulting in a training set with 5855 and a test set with 1646 reactions; Partition 2 - Training set with one reaction of each EC number of the data set and test set with the remaining reactions, 5246 and 2236 reactions respectively; Partition 3 - Training set with one reaction of each sub-subclass, 350 reactions. Two test sets, the first with one reaction of each EC number of the remaining reactions, 4896 reactions and the second set with all reactions excluding the 350 using in training, 7132 reactions; Partition 4 - Training set with one reaction of each sub-subclass excluding 40 reactions for test set. Training set with 310 reactions and test set with 40.

8.3.6 RFs Assignment of EC Numbers from the Reaction Equation.

MOLMAPs of size 625 (25×25) generated using topological and physico-chemical bond descriptors were used for the automatic assignment of the EC numbers, at the four levels of the EC hierarchy, from the structures of all reactants and products (as described in the KEGG reaction file) using Random Forests. Several partitions in training and test sets using different criteria were explored, to assess the robustness of the method.

Table 8.17 summarizes the results obtained with the different partitions. In all experiments, the reactions were represented in both directions. The results presented for the experiments with all data and for training sets were from the internal cross validation of RFs obtained by out-of-bag (OOB) estimation.

The first experiment (all data) was carried out with all the available data at each level of classification. An error of 6.19, 11.07, 13.25, and 16.69% were obtained for the EC class, subclass, sub-subclass and EC full number.

In partition 1 the training and test sets were selected using a 49×49 SOM. The SOM was trained with all reactions and then the test set was chosen by random selection of one reaction from each occuppied neuron, resulting in a training and a test set with 5855 and 1646 reactions respectively. Wrong predictions were obtained for 9.17, 15.95 and 18.60% of the test set for the first, second and third digit of the EC system respectively. Partition 1 is the only used where it is not guaranteed that the two entries for each reaction (corresponding to both directions) were included in the same set (training or test).

Partition 2 resulted from a different criterion to cover the reaction space as much as possible. In this case one reaction of each full EC number was randomly selected to the training set, and the remaining reactions labeled with the same full EC number were moved to the test set. With this partition every full EC number represented in the test set was represented in the training set. There are some EC numbers represented in the training set without reactions in the test set. The high similarity between training and test sets was reflected in the results. The results for the test sets were better than the OOB estimation for training sets. The results for the OOB estimations illustrate the ability of the model to classify reactions belonging to full EC numbers not available in the training.

The next partitions were designed to reduce the similarity between the training and test sets, and therefore to test the ability of the MOLMAP approach to assign EC numbers to reactions increasingly different from those provided during the training. In partition 3 the model was trained only with one reaction of each sub-subclass (350 reactions), which means that all reactions in the training set were different at the sub-subclass level, and all sub-subclasses were represented in the training set. Two test sets were used, the first including one reaction from each EC number not included in the training set (4896 reactions), and the second including all the reactions not included in the training set (7132 reactions). This means the first test set does not include any reaction with a full EC number available in the training set. For the class level, 80% and 82% of the test sets were correctly assigned despite the rather small size of the training set, and the fact that in the training set there is only one reaction of the same sub-subclass of each reaction in the test sets. This low level of similarity between training and test set is reflected in the results for the subclass level. The percentage of correct assignments decrease to 50% and 52% for the two test sets.

With partition 4 all reactions of the test set are from different sub-subclasses of the training set. This set was built by random selection of 40 reactions from the data set of 350 reactions used in partition 3 (with only one reaction of each sub-subclass). The test and training set were predicted with an error of ∼30%, and are of approximately the same accuracy as the OOB estimation for the training set of partition 3 (which has the same meaning, i.e. predicting the class with no cases of the same sub-subclass in the training).

Table 8.18: Confusion matrix for the classification of enzymatic reactions according the first digit of the EC number (test set of partition 2).

	EC 1	EC 2	EC 3	EC 4	EC 5	EC 6	% Uncorrect Predictions
EC 1	950	8	6	4	-	4	2.26
EC 2	10	612	22	8	-	-	5.85
EC 3	-	2	372	2	-	2	1.59
EC 4	2	3	18	103	-	-	18.25
EC 5	-	10	-	8	16	-	52.94
EC 6	-	4	-	-	-	70	5.41

The last partition has the main objective of check the possibility of assigning the full EC number. In partition 5 only full EC numbers with four or more reactions were considered. A data set with 1343 reactions corresponding to 110 different full EC numbers were obtained. One reaction of each full EC number was randomly selected to the test set (220 reactions), and the remaining were used as training set (1122 reactions). The full EC number was wrongly assigned in 19% of the reactions of the training set (OOB estimation), and could be correctly assigned for 86% of the test set.

Table 8.18 shows the confusion matrix obtained for the test set of partition 2 (2236 reactions).

The confusion matrix shows a higher prediction accuracy for the most represented classes (oxidoreductases, transferases and hydrolases) and for ligases, and the worst for lyases and isomerases with 18 and 53% of uncorrected predictions. Reactions catalysed by isomerases, EC 5, are more difficult to classify because usually no substantial structural changes occur in this type of reactions, and furthermore this class is the less represented in the data set. Reactions catalysed by lyases, EC 4, gave 18.25% of errors (23 reactions out of 126) with 18 reactions classified as hydrolases. This result shows that the patterns of reactions catalysed by lyases are in principle similar in many cases to hydrolases. Hydrolases present the higher prediction accuracy, with only 6 reactions badly classified. At the same time it is the class of reactions with the large number of false assignments (18 lyases and 22 transferases wrongly classified as hydrolases). This fact may indicate possible inconsistencies in EC numbers for the classification of these reactions.

Table 8.19 shows the relationship between the prediction accuracy and the probability of the prediction by RFs and supports its use as a measure of the reliability of the predictions.

The probability of each prediction reflects the number of votes obtained by each class. For the test set of partition 2, 2098 reactions out of 2236 (93.8%) were predicted with a probability higher than 0.5, with 2035 of these correctly classified at the class level. The number of reactions decreases to 1434 (64%) if we consider only reactions predicted with

Table 8.19: Relationship between the prediction accuracy and the probability associated to each prediction by RFs (test set of partition 2 and classification according the first digit of the EC number)

	Probability					
	≥0.5		≥0.7		≥0.9	
	Nr. reactions	Nr. Correct	N. reactions	Nr. Correct	Nr. reactions	Nr. Correct
EC 1	938	930 (99.2)	879	875 (99.5)	736	734 (99.7)
EC 2	596	584 (98.0)	549	543 (98.9)	392	390 (99.5)
EC 3	383	356 (93.0)	338	326 (96.5)	241	235 (97.5)
EC 4	97	84 (86.6)	45	43 (95.6)	18	18 (100)
EC 5	11	11 (100)	4	4 (100)	0	-
EC 6	73	70 (95.9)	64	64 (100)	47	47 (100)
Total	2098	2035 (97.0)	1879	1855 (98.7)	1434	1424 (99.3)

a probability higher than 0.9, but in this case almost all reactions were correctly classified (1424 out of 1434).

The results are similar for the same experiment performed at the subclass level. The percentages of correct predictions for each level of probability are almost the same. For probabilities higher than 0.5, 0.7 and 0.9 the reactions are correctly assigned in 97.2%, 98.5% and 99.2% of the cases at the subclass level, respectively. The main difference is in the number of reactions predicted with these levels of probability. In the first case 93.8%, 84.0% and 64.1% of the entire data set are predicted with probabilities higher than 0.5, 0.7 and 0.9 respectively, while in the second case the percentages of reactions predicted with these probabilities decrease to 81.8%, 84.0% and 64.1%.

The same experiment was performed for the tester set 1 of partition 3, at the subclass level (where only ∼50% of the reactions were correctly assigned). For probabilities higher than 0.5, 0.7 and 0.9 the subclass is correctly predicted in 84.2%, 85.8% and 89.5% of the cases, respectively. However, the number of reactions predicted with these probability values decreases considerably when compared with the other experiments. In this case the percentages of reactions predicted with a probability of 0.5, 0.7 and 0.9 is 21.1%, 12.0% and 3.3%. If the considered probability threshold is decreased to 0.4, then 32% of the data set is covered, with 79.8% of correct assignments at the subclass level.

8.3.7 RFs Assignment of EC Numbers from the Main Reactants and Products

MOLMAPs of reactions are difference MOLMAPs and therefore, in principle, require reaction equations to be balanced. However, reactions are often not balanced, and de-

Table 8.20: Classification of enzymatic reactions by RFs based on the main reactants and products.

Data sets		Uncorrect Predictions / %		
		1^{st}level	2^{nd}level	3^{rd}level
All data	MRP	23.48 (3850)	28.32 (3800)	32.76 (3632)
	MRP + FR	11.42 (11332)	14.81 (11242)	17.27 (10890)
Partition 1 (MRP)	Training	29.39 (2644)	36.71 (2596)	42.78 (2452)
	Test	21.56 (1206)	26.50 (1204)	37.37 (1180)
Partition 1 (MRP + FR)	Training	14.54 (7890)	20.01 (7802)	22.73 (7498)
	Test	11.30 (3442) 5.72 (2236 FR) 21.64 (1206 MRP)	14.97 (3440) 9.26 (2236 FR) 25.58 (1204 MRP)	22.49 (3392) 14.24 (2212 FR) 37.97 (1180 MRP)

MRP - MOLMAPS encoded based on the main reactants and products of each reaction.; FR - MOLMAPS encoded based on the full equation; Partition 1 (MRP) - Training set with one reaction of each EC number of the data set and test set with the remaining reactions, (the number of reactions are in parenthesis); Partition 1 (MRP + FR) - Training set with one reaction of each EC number of the data set and test set with the remaining reactions, data set with reactions encoded from the full reaction equation and only from the main reactants and products (the number of reactions are in parenthesis)

sirably a classification system should be able to classify reactions only from main reactants/products. We explored the possibility of training RF with MOLMAPs of reactions calculated only from the mains reactants and products (as described in the KEGG database), and the ability of RF trained with complete reactions to classify incomplete reactions.

Table 8.20 summarizes the results obtained for the assignment of EC numbers using different data sets of reactions.

The results show that the accuracy of the predictions is decreased in ca. 25% if only main reactants and products are used. Intermediate results are obtained if incomplete and complete reactions are mixed. In that case, if accuracies are separately calculated for the incomplete and complete reactions of the test set, the results for the complete reactions are similar to those obtained from RF trained only with complete reactions. The same happens with incomplete reactions. These observations suggest that the patterns of incomplete and complete reactions are processed independently by the RF, even for those belonging to a common classification. We also observed that RF trained with complete reactions yielded errors of 59% for the class level and 69% for the subclass and sub-subclass levels for test sets of incomplete reactions (including reactions in the test set that were available in the training set in the complete form). For the opposite experiment (training with incomplete reactions and testing with complete reactions) yielded errors of 39%, 52%

and 55% for the class, subclass and sub-subclass levels.

8.4 Conclusions

The results of unsupervised mapping of metabolic reactions show a general agreement with the EC classification. The general reasonable clustering of reactions according to the EC classification allowed for the SOMs to assign EC numbers at the class, subclass, and sub-subclass levels for reactions of independent test sets with accuracies up to 92%, 80%, and 70%, respectively. These numbers reflect the similarity between reactions within the first three levels of the EC hierarchy. Accuracy of predictions was correlated with the number of votes in consensus predictions, and with the Euclidean distance to the winning neuron of SOMs.

Similarity of reactions across sub-subclasses (within the same class) was assessed by test sets only including reactions of sub-subclasses not available in the training set - 68% of correct predictions were obtained for the class level. At the same time, experiments to predict the third digit of the EC number demonstrated the ability of MOLMAP descriptors to discriminate sub-subclasses.

The correspondence between chemical similarity of metabolic reactions and similarities in their MOLMAP descriptors was confirmed with a number of reactions detected in the same neuron but labeled with different EC classes. Such an exercise also demonstrated the possible application of the MOLMAP/SOM approach to the verification of internal consistency of classification in databases of metabolic reactions. Conveniently, the MOLMAP method avoids the assignment of reaction centers and atom-to-atom mapping previous to the classification of reactions.

The results demonstrate the possibility of applying Random Forests to the automatic assignment of EC numbers with better accuracy than unsupervised methods, such as Kohonen self-organizing maps used in previous studies. The accuracies of predictions reached 92%, 85%, and 83% for the class, subclass, and sub-subclass level if several examples of reactions belonging to the same sub-subclass are available in the training set. In the absence of information for reactions belonging to the same sub-subclass, accuracies drop to 70% in for the prediction of the class. This illustrates a heterogeneity of subclasses within the same subclass for a large number of cases.

This study also shows the possibility of training a single RF with complete reactions together with reactions represented only by their main reactants and products, and to still obtain accurate predictions for complete reactions, while predictions for incomplete reactions are ∼25% less accurate.

RF advantageously associate a probability to each prediction that correlated well with the observed accuracy for independent test sets.

Acknowledgements

The authors thank ChemAxon Ltd (Budapest, Hungary) for access to JChem and Marvin software, Kyoto University Bioinformatics Center (Kyoto, Japan) for access to the KEGG Database, and Molecular Networks GmbH (Erlangen, Germany) for access to the Biopath Database and PETRA software.

Chapter 9

Genome-Scale Classification of Pathways and Organisms

9.1 Introduction

The experiments presented in the last two Chapters allow the representation of enzymatic reactions by a numerical fixed length code that can be further processed by automatic learning methods. The MOLMAP approach represents chemical reactions based on the chemical bonds of the products and reactants. The same idea can be extended to the representation of other levels of metabolic information, such as metabolic pathways, or reactomes of organisms.

In the cases presented until now the pattern of neurons activated by the chemical bonds of a molecule is used as a representation of that molecule. Following the same concept, the pattern of neurons activated by the reactions of a metabolic pathway is a representation of the reactions involved in that pathway - a descriptor of the metabolic pathway. The same for reactomes of organisms where the pattern of neurons activated by the metabolic reactions of an organism is a representation of the reactions that take place in the metabolism of that organism - a descriptor of the organism based on their metabolic reactions.

In these experiments, Kohonen SOMs are used to map three different levels of information - maps of metabolic pathways and reactomes of organisms on top of maps of chemical reactions, and these on top of maps of chemical bonds.

9.2 Methodology and Computational Details

9.2.1 Data sets

Enzymatic reactions were extracted from the KEGG LIGAND database (release of November 2006) [37, 38, 40, 177] in MDL .mol format, and were pre-processed as was described

Table 9.1: Number of metabolic pathways in each type of metabolism.

Label	Type of Metabolism	Number of Metabolic Pathways
🟥	Carbohydrate metabolism	16
🟩	Lipid Metabolism	12
🟨	Xenobiotics Biodegradation and Metabolism	18
🟦	Metabolism of Amino Acid and other Amino Acid	24
🟧	Biosynthesis of Secondary Metabolites	12
🟪	Metabolism of Cofactors and Vitamins	10

in the previous Chapter. Differently, duplication of reactions and reactions differing only in stereochemical features were here also considered. Also reactions that were listed with more than one EC number, no assigned EC number or with incomplete EC numbers were used for training maps of reactions. The information concerning the reactions that participate in each metabolic pathway were extracted from the KEGG PATHWAY database. Only the pathways with more than 75% of the reactions encoded were used, and types of metabolism with a small number of metabolic pathways did not participate in the experiments. Table 9.1 summarizes the types of metabolism included and the number of metabolic pathways of each.

The KEGG classification was used, with metabolic pathways classified in 11 different types of metabolism. The experiments were performed with 92 metabolic pathways from six different types of metabolism (the “metabolism of amino acids” and the “metabolism of other amino acids” were considered in the same class).

The experiments with reactomes of organisms were performed with the lists of metabolic reactions for each organism extracted from the KEGG GENES database. Organisms were restricted to prokariotes fully sequenced, and classification was considered at the third and fourth taxonomy level. Table 9.2 presents the number of organisms used in each of the third level classes.

A total of 347 organisms were used in the experiments to classify organisms in terms of their third level of taxonomy. The 347 organisms were classified in 16 classes. Acidobacteria, Fusobacteria and Planctomyces included only one organism encoded.

Experiments were also performed at the fourth level of the taxonomic hierarchy. Table 9.3 shows the number of organisms in each of the fourth level classes.

A total of 340 organisms were used in these experiments. Third level classes were partitioned into other classes for the experiments with classification at the fourth level. This is the case for example of proteobacteria that were classified in six different classes at the fourth level. Classes with only one organism at the third level of classification were not used in the experiments at the fourth level. The 340 organisms used in this

Table 9.2: Number of organisms in each class at the third taxonomy level.

Label	Classification at the third taxonomy level	Number of Organisms
🟥 (A)	Prokariotes/Bacteria/Proteobacteria	165
🟦 (B)	Prokariotes/Bacteria/Acidobacteria	1
🟩 (C)	Prokariotes/Bacteria/Firmicutes	81
🟨 (D)	Prokariotes/Bacteria/Actinobacteria	23
🟦 (E)	Prokariotes/Bacteria/Fusobacteria/Fusobacteria	1
🟥 (F)	Prokariotes/Bacteria/Planctomyces/Planctomyces	1
🟧 (G)	Prokariotes/Bacteria/Chlamydia	10
🟪 (H)	Prokariotes/Bacteria/Spirochete	6
⬜ (I)	Prokariotes/Bacteria/Cyanobacteria	17
⬜ (J)	Prokariotes/Bacteria/Bacteroides	5
🟥 (K)	Prokariotes/Bacteria/Green Sulfur Bacteria	3
🟫 (L)	Prokariotes/Bacteria/Green Non Sulfur Bacteria	2
⬜ (M)	Prokariotes/Bacteria/Deinococcus-thermus	4
🟦 (N)	Prokariotes/Bacteria/Hyperthemophilic Bacteria	2
🟦 (O)	Prokariotes/Archaea/Euryarchaeota	21
🟦 (P)	Prokariotes/Archaea/Crenarchaeota	5

Table 9.3: Number of organisms in each class at the fourth taxonomy level.

Label	Classification at the fourth taxonomy level	Number of Organisms
🟥 (A’)	Prokariotes/Bacteria/Proteobacteria/Gamma/Enterobacteria	31
🟦 (B’)	Prokariotes/Bacteria/Proteobacteria/Gamma/Others	47
🟩 (C’)	Prokariotes/Bacteria/Proteobacteria/Beta	24
🟨 (D’)	Prokariotes/Bacteria/Proteobacteria/Epsilon	8
🟦 (E’)	Prokariotes/Bacteria/Proteobacteria/Delta	10
🟥 (F’)	Prokariotes/Bacteria/Proteobacteria/Alpha	45
🟧 (G’)	Prokariotes/Bacteria/Firmicutes/Bacillales	31
🟪 (H’)	Prokariotes/Bacteria/Firmicutes/Lactobacillales	27
⬜ (I’)	Prokariotes/Bacteria/Firmicutes/Clostridia	7
⬜ (J’)	Prokariotes/Bacteria/Firmicutes/Mollicutes	16
🟥 (K’)	Prokariotes/Bacteria/Actinobacteria/Actinobacteria	23
🟫 (L’)	Prokariotes/Bacteria/Chlamydia/Chlamydia	10
⬜ (M’)	Prokariotes/Bacteria/Spirochete/Spirochete	6
⬜ (N’)	Prokariotes/Bacteria/Cyanobacteria/Cyanobacteria	17
🟦 (O’)	Prokariotes/Bacteria/Bacteroides/Bacteroides	5
🟦 (P’)	Prokariotes/Bacteria/Green Sulfur Bacteria/Green sulfur Bacteria	3
🟦 (Q’)	Prokariotes/Bacteria/Deinococcus-thermus/Deinococcus-thermus	4
🟦 (R’)	Prokariotes/Archaea/Euryarchaeota/Euryarchaeota	21
🟦 (S’)	Prokariotes/Archaea/Crenarchaeota/Crenarchaeota	5

experiments were classified in 19 taxonomic classes at the fourth level of the hierarchy.

9.2.2 Kohonen Self-Organizing Maps (SOM)

SOMs with toroidal topology were used in this study for three independent tasks, the classification of chemical bonds for the generation of a molecular descriptor, the classification of metabolic reactions for the generation of a metabolic pathway or reactome descriptor, and the classification of metabolic pathways and organisms. A Kohonen SOM is an unsupervised method that projects multidimensional objects into a 2D surface (a map) which can reveal similarities between objects, mapped into the same or neighbor neurons. More details about this method can be found in Section 2.6.

Training was performed by using a linear decreasing triangular scaling function with an initial learning rate of 0.1 and an initial learning span of half the size of the map (except for maps of size 49×49, for which an initial learning span of 7 was used). The weights were initialized with random numbers that were calculated using as parameters the mean and standard deviation of the corresponding variables in the input data set. For the selection of the winning neuron, the minimum Euclidean distance between the input vector and neuron weights was used. The training was typically performed over 50, 75 or 100 cycles, with the learning span and the learning rate linearly decreasing until zero. SOMs were implemented throughout this study with an in-house developed Java application derived from the JATOON Java applets. [171]

9.2.3 Generation of MOLMAP Reaction Descriptors

The generation of MOLMAP molecular descriptors is based on a SOM that distributes chemical bonds through the grid of neurons. The chemical bonds are represented by topological and physico-chemical features (Tables 8.4 and 8.5). The SOM is trained with a diversity of bonds, taken from a diverse data set of molecules. The methodology is described in Section 8.2.3 of the last Chapter.

The bonds existing in a molecule can be represented as a whole by mapping all the bonds of that molecule onto the SOM previously trained with a diversity of bonds. The pattern of activated neurons is interpreted as a fingerprint of the available bonds in the molecule, and it was used as a molecular descriptor (MOLMAP). From this molecular descriptor a reaction MOLMAP is obtained by the difference between the molecular MOLMAPs of the products and reactants.

9.2.4 Generation of MOLMAP Metabolic Pathway Descriptors

MOLMAPs of reactions were used to derive MOLMAPs of metabolic pathways as follows:

- A Kohonen SOM is trained with all metabolic reactions available. The objects of this SOM are reactions.
- All reactions of a metabolic pathway are mapped on the trained Kohonen SOM. Each metabolic reaction activates one neuron.
- The pattern of activated neurons is a representation of the metabolic reactions available in that pathway - a fingerprint of chemical machinery of the pathway.
- The pattern of activated neurons is encoded numerically for computational processing. Each neuron is given a value equal to the number of times it was activated by metabolic reactions.
- The map (a matrix) is transformed into a vector by concatenation of columns. To account for the relationship between the similarity of metabolic reactions and proximity in the map, a value of 0.3 was added to each neuron multiplied by the number of times a neighbor was activated by a reaction. If an empty neuron is

a direct neighbor of more than one activated neuron, its value is the sum of the contributions from each activated neuron.

9.2.5 Generation of MOLMAP Reactome Descriptors

The methodology to generate MOLMAP reactome descriptors was the same used in the last Subsection. The difference was in the last step. Instead of mapping the reactions of a metabolic pathway, all reactions participating in the metabolism of an organism were mapped on the Kohonen SOM previously trained with reactions. The pattern of activated neurons by the metabolic reactions of that organism can be interpreted as a fingerprint of the organism based on its chemical reactions - a reactome MOLMAP descriptor. This reactome descriptor encodes, in a fixed length numerical code, the biochemical machinery of an organism.

9.2.6 Mapping of Metabolic Pathways and Organisms

The metabolic pathway descriptors and reactome descriptors can then be used to train new independent Kohonen SOMs. In the case of the metabolic pathways, the objects were metabolic pathways classified according to the types of metabolism. In the case of organisms the objects are reactomes of organisms and were classified at two taxonomic levels.

Figure 9.1 illustrates the three levels of classification of chemical information used to classify metabolic pathways. Kohonen SOMs were used in the three cases. In the first map the objects were chemical bonds, in the second chemical reactions, and finally in the last map the objects were metabolic pathways.

9.3 Results and Discussion

9.3.1 Mapping of Metabolic pathways

The metabolic pathway descriptors were used to train Kohonen SOMs to classify metabolic pathways according to the main types of metabolism (Figure 9.2). The obtained map shows well-defined regions for specific types of metabolism.

The map presents some formal conflicts between types of metabolism, but in some cases these correspond to chemical similarities between metabolic pathways classified in different types of metabolism. For example the riboflavin metabolism (metabolism of co-factors and vitamins), and the tetrachloroethene degradation (xenobiotics biodegradation and metabolism) were both classified as metabolism of co-factors and vitamins. In these two pathways there were no reactions with the same EC number. Another example is the mapping in the same neuron of the c5-branched dibasic acid metabolism (carbohydrate

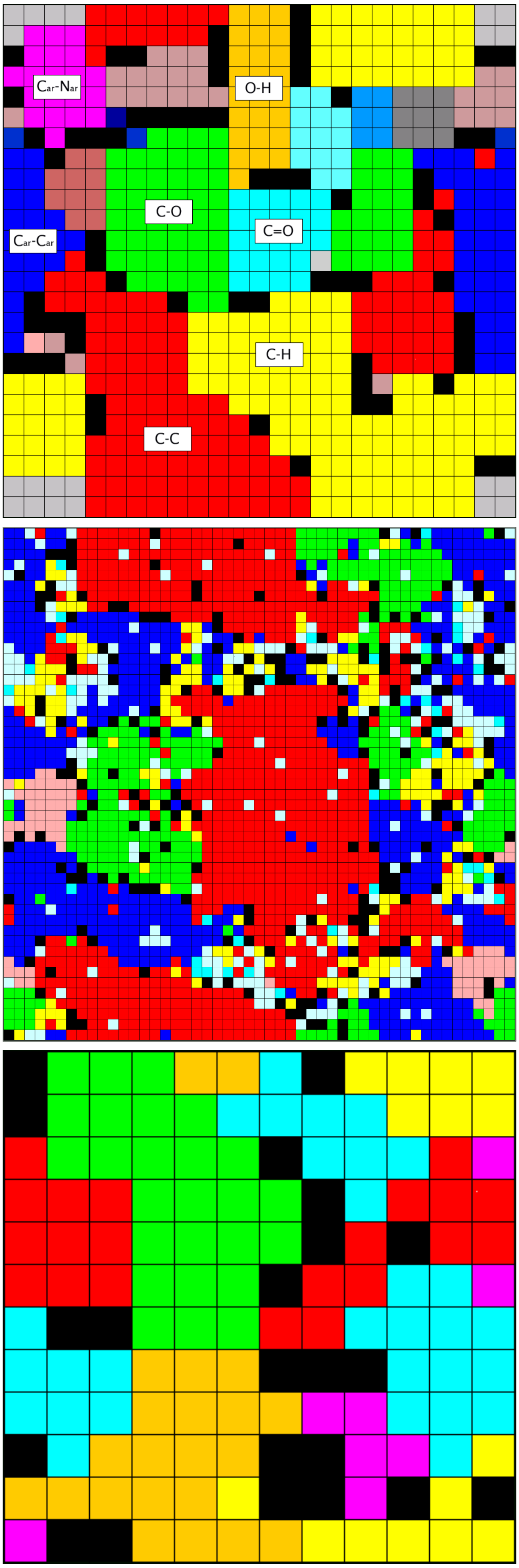


Figure 9.1: Toroidal surfaces of Kohonen SOMs representing the three codification levels of chemical information. The objects are chemical bonds, reactions and metabolic pathways in the first, second and third map, respectively.

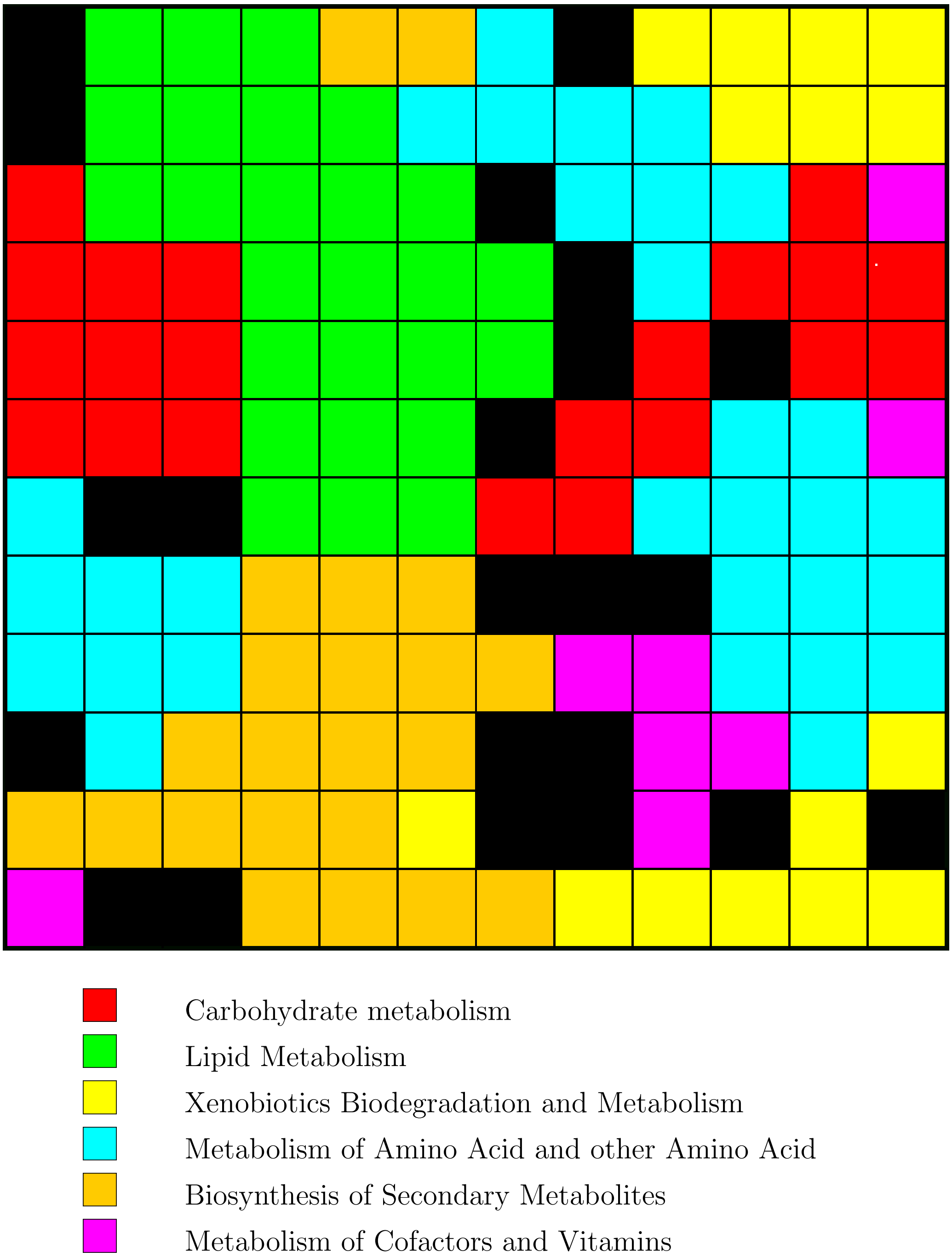


Figure 9.2: Toroidal surface of a 12×12 Kohonen SOM trained with 92 metabolic pathways encoded by metabolic pathway MOLMAPs of size 2401. After the training, each neuron was colored according to the type of metabolism of the metabolic pathways that was mapped onto it or onto its neighbors. Black neurons correspond to conflicts.

metabolism), and the valine, leucine and isoleucine metabolism (amino acid metabolism). In this example there was no repetition of reactions, but the two pathways presented six reactions with similar EC numbers. In these two cases, a few reactions of both pathways were encoded in the same position of the pathway descriptor. These reactions were often similar and shared the first three digits of the EC number, or had no assigned EC number but were found to be similar to reactions in the other pathway.

Two pathways of different metabolisms mapped in the same neuron do not imply that the entire pathways are similar. Equal or very similar parts of the pathways (two or three consecutive reactions) often suffice for both pathways to activate the same neuron.

These preliminary experiments suggest that this approach can be used in the future to compare different metabolic pathways and metabolic pathways of different organisms.

This methodology based on three levels of codification of chemical information allows the mapping and automatic screening of similarities between metabolic pathways of different types of metabolism even in cases where the reactions participating in the pathway did not share EC numbers, or have no assigned EC numbers.

9.3.2 Mapping of Organisms

The Kohonen SOM trained with the diversity of metabolic reactions allows to map the entire reactome of an organism. Figures 9.3 and 9.4 show the pattern of activated neurons by two different organisms - the reactome of the *Staphylococcus aureus N315* and the reactome of the *Bacillus licheniformis DSM13* (both cases are bacteria/firmicutes/bacillales).

The maps allows for an easy visualization of differences between the two organisms in the reactome space. The pattern of activated neurons by each organism consists in their reactome descriptor. It is to point out that the Kohonen SOM at this level is trained with all metabolic reactions (independently of the organism where they occur).

The next step was to use the reactome descriptors of all organisms available to train a Kohonen SOM. The organisms were classified in terms of their taxonomy. The reactome MOLMAP descriptors of a data set of prokariotes were used to train Kohonen SOMs to classify organisms at two different taxonomy levels.

Figures 9.5 and 9.6 show the Kohonen SOMs obtained.

The two maps showed a considerable clustering according to the taxonomy. The first Kohonen SOM was trained with reactome MOLMAP descriptors of 347 organisms classified at the third level of taxonomy (as listed in the KEGG database). Well-defined clusters of organisms according to their class can be observed. Clustering according to the previous level of classification can also be observed in some cases, for example with Euryarchaeota (class O) and Crenarchaeota (Class P), the only two classes of Archae organisms, mapped in the same region of the map.

The second map was trained with reactome MOLMAP descriptors of 340 organisms

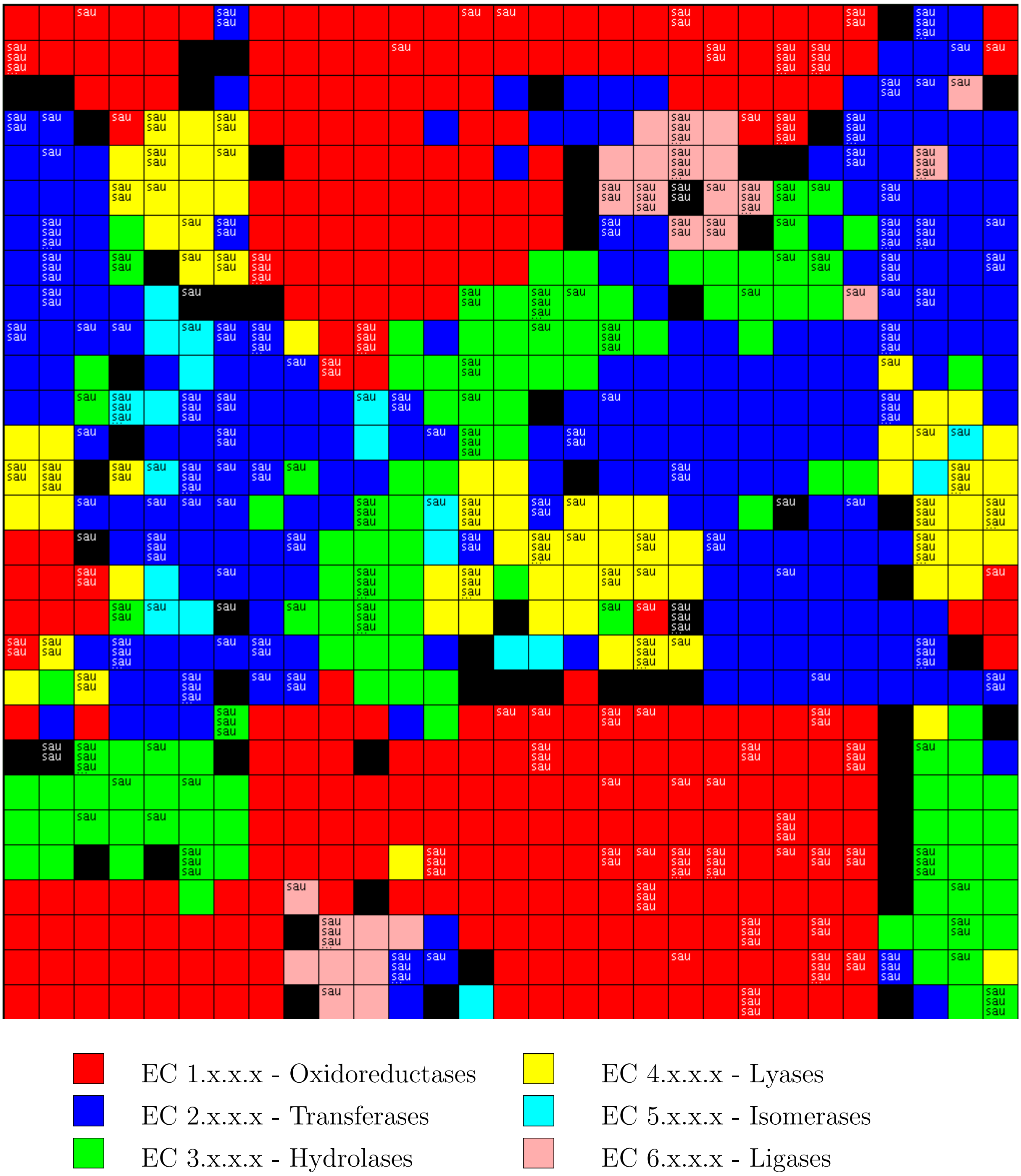
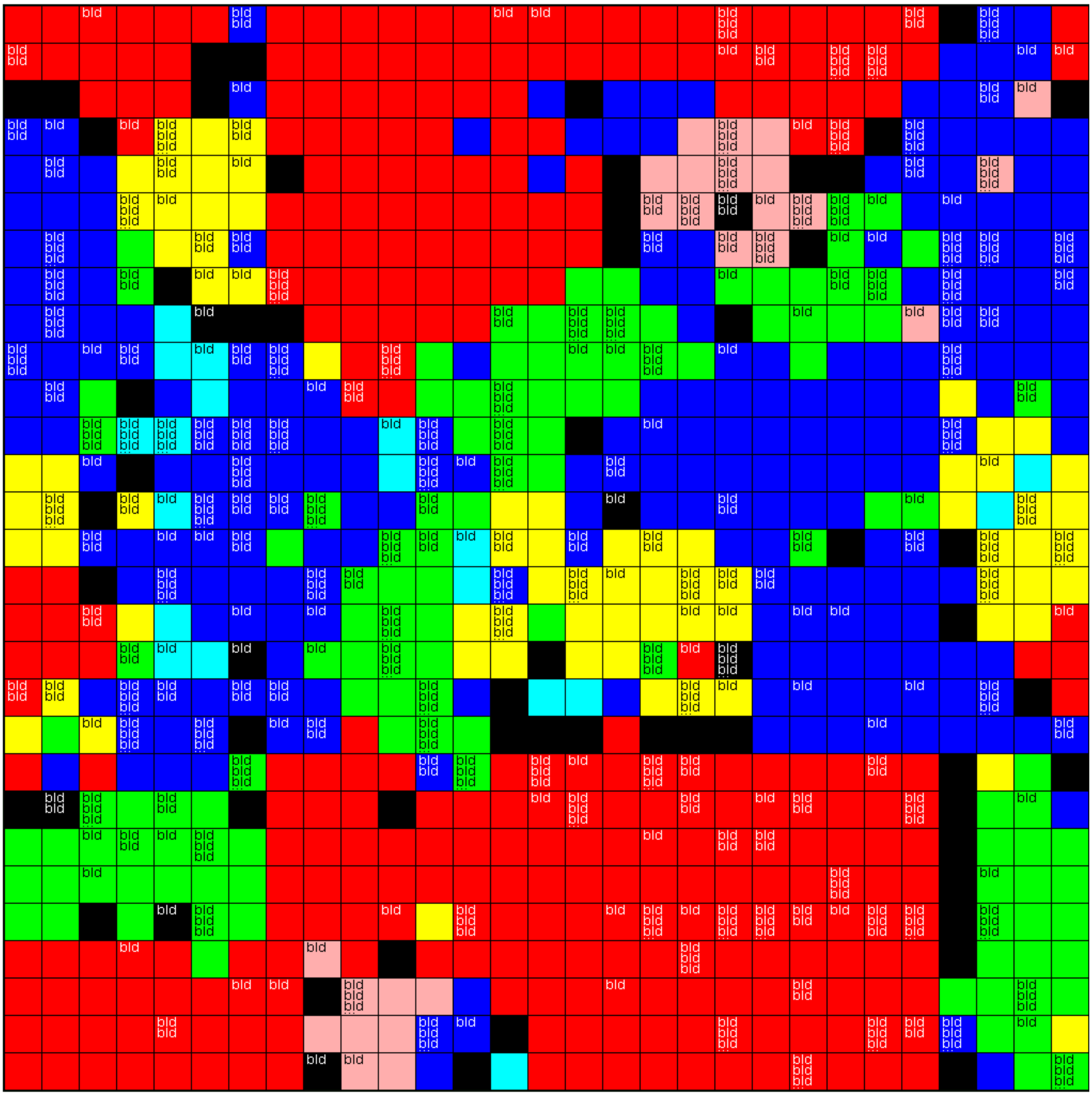


Figure 9.3: Toroidal surface of a 29×29 Kohonen SOMs trained with enzymatic reactions encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. The color of the neurons correspond to the six EC main classes. The black neurons are conflicts. The map shows the pattern of activated neurons by the reactome of *Staphylococcus aureus N315*.



🟥 EC 1.x.x.x - Oxidoreductases

🟦 EC 2.x.x.x - Transferases

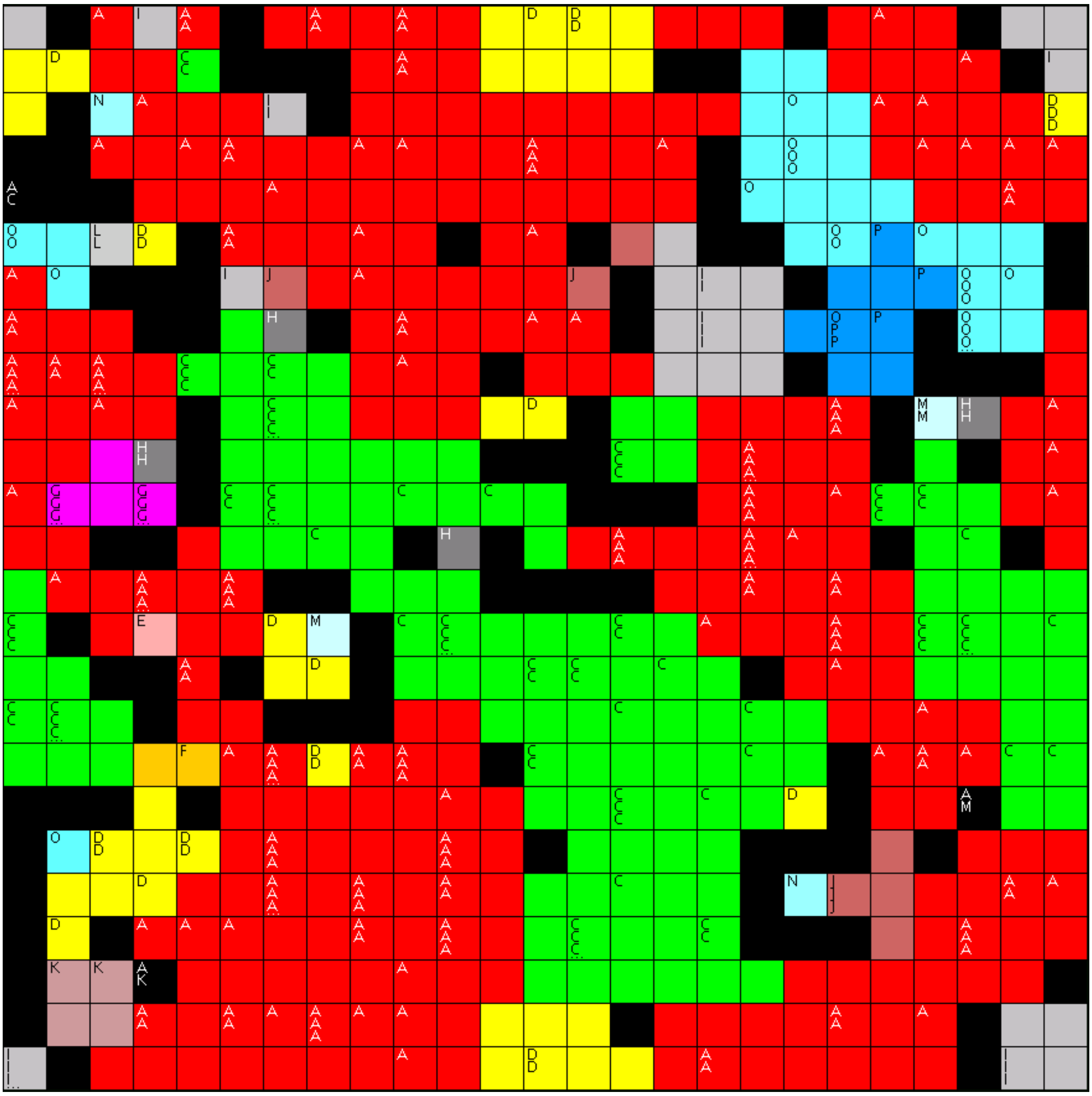
🟩 EC 3.x.x.x - Hydrolases

🟨 EC 4.x.x.x - Lyases

🟦 EC 5.x.x.x - Isomerases

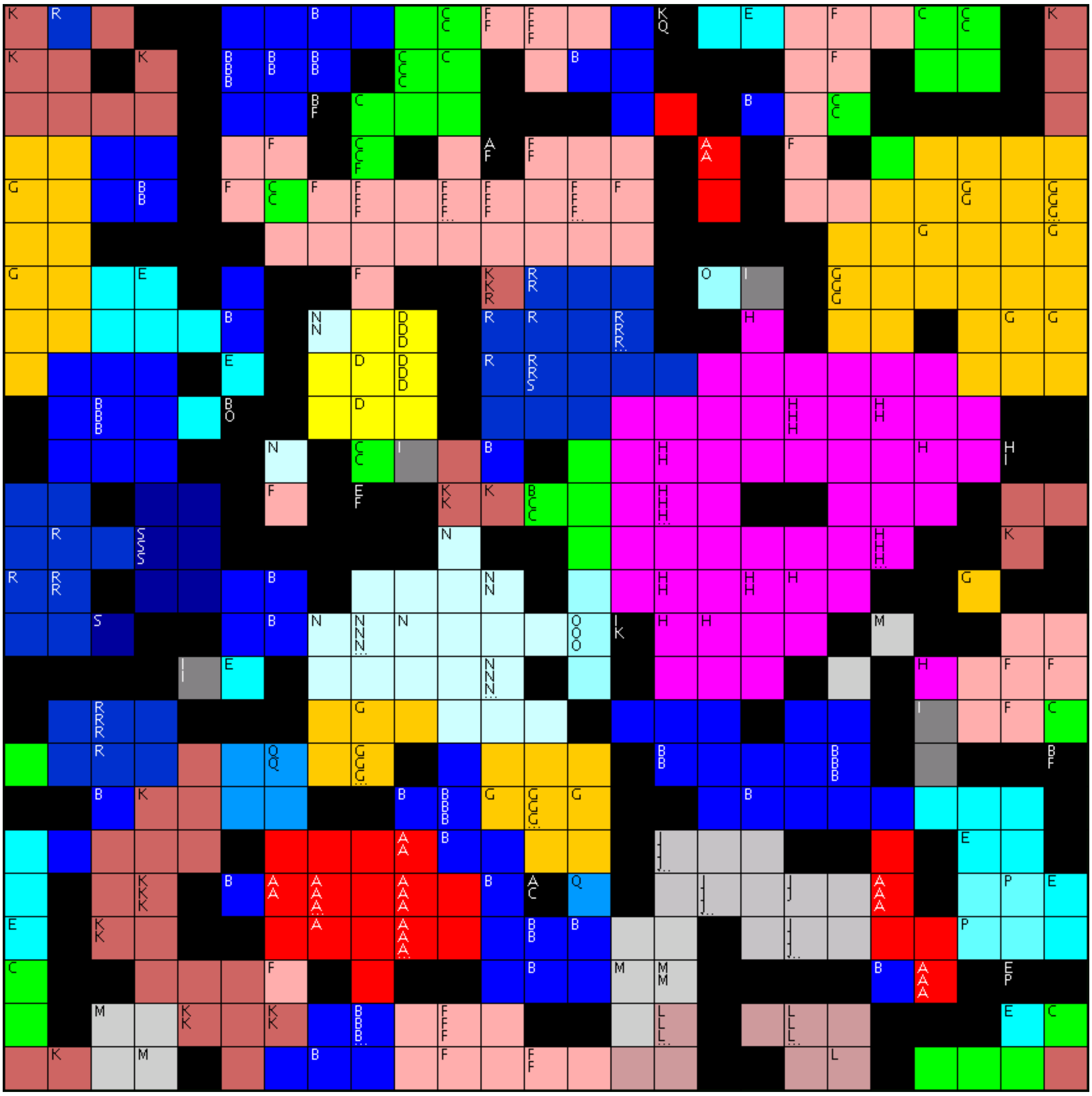
🟥 EC 6.x.x.x - Ligases

Figure 9.4: Toroidal surface of a 29×29 Kohonen SOMs trained with enzymatic reactions encoded by MOLMAPs of size 625 using topological and physico-chemical descriptors. The color of the neurons correspond to the six EC main classes. The black neurons are conflicts. The map shows the pattern of activated neurons by the reactome of *Bacillus licheniformis DSM13*.



🟥	(A)	Proteobacteria	⬜	(I)	Cyanobacteria
🟦	(B)	Acidobacteria	⬜	(J)	Bacteroides
🟩	(C)	Firmicutes	🟫	(K)	Green Sulfur Bacteria
🟨	(D)	Actinobacteria	🟫	(L)	Green Non Sulfur Bacteria
🟦	(E)	Fusobacteria	⬜	(M)	Deinococcus-thermus
🟥	(F)	Planctomyces	⬜	(N)	Hyperthemophilic Bacteria
🟧	(G)	Chlamydia	🟦	(O)	Euryarchaeota
🟪	(H)	Spirochete	🟦	(P)	Crenarchaeota

Figure 9.5: Toroidal surface of a 25×25 Kohonen SOM trained with reactomes of organisms encoded by reactome MOLMAPs of size 841. After the training, each neuron was colored according to the third taxonomic level of the organism in the training set that were mapped onto it or onto its neighbors. The black neurons are conflicts.



🟥 (A')	Gamma/Enterobacteria	(I')	Clostridia	(Q')	Deinococcus-Thermus
🟦 (B')	Gamma/Others	(J')	Mollicutes	(R')	Euryarchaeota
🟩 (C')	Beta	(K')	Actinobacteria	(S')	Crenarchaeota
🟨 (D')	Epsilon	(L')	Chlamydia		
(E')	Delta	(M')	Spirochete		
(F')	Alpha	(N')	Cyanobacteria		
🟧 (G')	Bacillales	(O')	Bacteroides		
🟪 (H')	Lactobacillales	(P')	Green Sulfur Bacteria		

Figure 9.6: Toroidal surfaces of a 25×25 Kohonen SOM trained with reactomes of organisms encoded by reactome MOLMAPs of size 841. After the training, each neuron was colored according to the fourth taxonomy level of the organism in the training set that were mapped onto it or onto its neighbors. The black neurons are conflicts.

classified at the fourth taxonomy level in 19 classes. A considerable clustering of objects according their classes at the fourth level of the taxonomy can be observed. Clustering according to the third level can also be observed, as was expected and the first map suggested. For example the class A of the first map (proteobacteria) had 165 organisms and occupied a considerable part of the map. Here this class is classified in six different classes (A'-F') that appear mapped in close regions of the map. The Bacillales (class G') and Lactobacillales (class H') both Firmicutes form two clusters in contiguous regions.

It is to point out that the good results obtained concerning the clustering of organisms according to their taxonomic classification can be affected by the way the genomes were annotated. In fact a genome being 100% sequenced does not mean that the reactome is 100% completed (we are not using all reactions that occur in an organism). It is possible in some cases that the biochemical machinery of two organisms appear as very similar as a consequence of the similarities between genomes - genomes are often annotated by comparison with similar genes in other organisms.

9.4 Conclusions

The extension of the MOLMAP concept for the encoding of other levels of metabolic information enabled the comparison of different pathways, the automatic classification of pathways, and a classification of organisms in terms of taxonomy based on their biochemical machinery.

The three levels of classification (from bonds to metabolic pathways) allowed to map and perceive chemical similarities between metabolic pathways even for pathways of different types of metabolism and pathways that do not share similarities in terms of EC numbers.

The results show that it is possible to extend the fundamental concept of encoding chemical reactions using local properties of the molecules to other levels of metabolic information such as the encoding of metabolic pathways and the encoding of reactomes of organisms.

Acknowledgements

The authors thank ChemAxon Ltd (Budapest, Hungary) for access to JChem and Marvin software and Kyoto University Bioinformatics Center (Kyoto, Japan) for access to the KEGG Database.

Part III

Mapping of Potential Energy Surfaces by Neural Networks

This part of the Thesis is devoted to the experiments concerning the mapping of Potential Energy Surfaces (PES) using Neural Networks (NNs). PES are crucial to the study of reactive and nonreactive chemical systems by Monte Carlo (MC) or Molecular Dynamics (MD) simulations. Ideally, PES should have the accuracy provided by *ab initio* calculations and be set up as fast as possible. Recently, NNs turned out as a suitable approach for estimating PES from *ab initio*/DFT energy data sets as an alternative to the common methods based on fitting analytical functions to quantum mechanical energies. The main objective of this work is to develop an approach to accurate PES by NNs regarding their use in molecular simulations.

Chapter 10

Fundamental Concepts

This Chapter introduces the fundamental concepts of molecular simulation and statistical thermodynamics concerned with the experiments reported in the next Chapters. Section 10.1 reviews the basic concepts of molecular simulation, including a short description of MC and MD methods. A brief discussion on interaction potentials and PES is also presented in Section 10.2. Section 10.3 touches upon the main concepts of the Density Functional Theory (DFT). Finally, Section 10.4 overviews the application of NNs to the map of PES.

10.1 Molecular Simulation

Molecular simulation is based on theoretical models from classical, statistical and quantum mechanics. The research in this field began in the 50s of the XX century with the development of the MD [186–188] and MC [189] methods when the first simulations of simple systems were performed. Presently, they are currently used to study a large variety of chemical and physical systems. This is a consequence of the advances in the underlying theoretical models, and of the spectacular growth of computer technology. Until the 80s of the XX century, the simulation of the most complex systems was restricted to some research groups that had access to supercomputers. Today, any PC has enough processing power to perform simulations of systems with considerable size and complexity.

Molecular simulations allow the determination of very many physico-chemical properties, even those that are difficult to estimate by experimental or first principles methods, being limited more by technical aspects than by conceptual ones. The two most important technical issues are:

- the processing power - mainly related to the velocity and number of available processors
- the memory capacity - related to the virtual memory available for handling the

input data to perform the calculations and to the storage of the simulation results for further analysis.

In the last years, several software packages for molecular simulation have been put forward which greatly facilitate the work of the computational chemist. Of course, when the research is focused on complex and unexplored chemical systems, not contemplated in the software packages, the computational chemist has to modify the programs or write them from the beginning.

The molecular simulation methods are essentially based on statistical mechanics, which links the microscopic properties of the molecules to the thermal, structural and dynamical properties of the respective molecular assemblies.

From a molecular model, characterized by well-defined molecular interactions, it is possible to study the behavior of a system, with a given number of molecules, using MC and MD methods at preset conditions of temperature, pressure, density or chemical potential. The number of molecules used to perform the simulations is much smaller than the number of molecules in a macroscopic system (10^{23}), but it can be increased at the pace of computer power. Presently, it is common to perform simulations with $\sim 10^6$ molecules contrasting with the $\sim 10^2$molecules in the first molecular simulations. Even so, to simulate bulk systems it is necessary to apply some kind of boundary conditions in order to attenuate the considerable surface effects (not present in macroscopic systems) introduced by the use of relatively small number of molecules in the simulation models. We shall refer to boundary conditions in Section 10.1.4.

If the simulation results reproduce experimental data, then they do shed light into the molecular mechanisms and may suggest new experiments. Additionally, computer simulation is certainly an invaluable route to assess properties that are difficult, or even impossible, to obtain experimentally. Furthermore, molecular simulation also explores and validates molecular theories by a comparison of theoretical and simulation results, eliminating eventual ambiguities that can turn out from a direct comparison between theoretical and experimental results.

The simulations presented in Chapter 11, to assess the reliability of the NNs-PES for argon, were carried out by molecular dynamics.

10.1.1 Molecular Dynamics Method

The molecular dynamics method was developed by Alder and Wainwright in 1957-1959 [186, 187, 190] who applied it to a system of hard spheres. In the 1960s, the method was extended to realistic potentials by Rahman [191] and by Verlet [192].

The method generates, sequentially in time, an ensemble of microstates (a microstate is defined by the molecular positions and momenta, in a classical description, or by a wave function, in a quantum description) through the numerical integration of Newton's equations of motion, for each molecule in the model:

$$m_i \frac{d^2 \boldsymbol{r}_i}{dt^2} = \boldsymbol{F}_i \tag{10.1}$$

where m_i, $\boldsymbol{r}_i$, and $\boldsymbol{F}_i$ are, respectively, the mass, the position vector and the total force acting on molecule i. In the case of a quantum description the time-dependent Schrödinger equation has to be considered instead of Newton's equations. In what follows, however, we shall only be concerned with classical methods.

The *time averages* of the mechanical properties are calculated over the generated microstates, that is, over the trajectory of the system in the phase space.

The integration starts from a set of initial positions and momenta and takes into account the forces acting on each molecule due to the other molecules in the system.

MD was developed in the context of the microcanonical ensemble, (E, V, N), and invariably carried out, until the 1980s, in this ensemble (apart from a heuristic technique to maintain the temperature constant [193]). Following the work of Anderson [194] the method was then extended to the isoenthalpic-isobaric (H, p, N) [195, 196], canonical (T, V, N) [195, 197, 198] and isothermal-isobaric (T, p, N) [195, 199, 200] ensembles, where E, V, N, H, p or T are the environmental thermodynamic variables kept constant during the simulations, namely, the total energy, volume, number of molecules, enthalpy, pressure or absolute temperature. The extension of the method to the last ensembles was a considerable methodological achievement since the respective environmental conditions are more adequate for comparison with experimental results. In the 1990s, MD was extended to situations allowing the exchange of molecules of the system with the surroundings [201–203].

10.1.2 Monte Carlo Method

Contrary to the pure MD method, which is fully deterministic, the Monte Carlo (MC) method is a stochastic technique. It was introduced by Metropolis et al. [189] in 1953.

The original method randomly generates an ensemble of microstates in the configurational part of phase space, without any sequence in time, and performs the averages (in this context, called *ensemble averages* to distinguish them from the MD time averages) of the mechanical properties over that ensemble. The calculations start from an initial microstate and take into account the interaction energy of the molecules in the model. Typically, picking up *just one molecule* of a microstate, *o,* and giving it displacements in random directions and orientations a new microstate, *n*, is generated.

Such generation is carried out so that the microstates become distributed according to the appropriate probability distribution function, for the chosen ensemble which has a predefined set of environmental constraints. This is realized by accepting or rejecting the new configuration, n, obtained from the old configuration o, with the following transition probability:

$$\pi\,(o{\rightarrow}n) = min\left(1,\ \frac{\rho(n)}{\rho(o)}\right) \tag{10.2}$$

where ρ is the ensemble probability function and min is the minimum intrinsic function. When the new configuration is rejected, the old configuration o is counted as a repeated one.

In the case of the canonical ensemble (TVN) [190, 204, 205]:

$$\pi\,(o \rightarrow n) = min\left(1,\ \frac{\exp\left(-U(n)/k_BT\right)}{\exp\left(-U(o)/k_BT\right)}\right) \tag{10.3}$$

where $U()$ is the potential energy of the respective configuration and k_B is the Boltzmann constant.

The method was extended to the isothermal-isobaric ensemble [190, 206] and to the grand-canonical ensemble [190, 207] by introducing the appropriate Boltzmann factors in the transition probabilities. Another important extensions which allow the direct study of phase-equilibria, chemical reactivity and associating systems are the Gibbs ensemble [208–211] and the Gibbs-Duhem integration [204, 212] methods.

An essential characteristic of the mentioned techniques is the assumption of the separability of the kinetic and potential energies of the Hamiltonian. The kinetic part can be analytically integrated in the probability density function and the sampling is only carried out on the configurational-phase subspace, as already referred to above. Some MC methods, however, allow the sampling of the momenta space [213].

As far as equilibrium properties are concerned MC and MD are equivalent. However, as in MC the microstates are not generated in a time sequence, that is, there is no integration of motion equations, the dynamic properties are not directly measured by MC, although some kind of stochastic dynamics may be worked out [214]. On the contrary, as MD is fully deterministic the dynamic properties can be calculated from the trajectory of the system in phase space.

10.1.3 Limitations of the Methods

The classical MD and MC methods, described in the last Sections, require well-defined, simultaneous, positions and velocities of the molecules. This requirement seems to violate the Heisenberg's uncertainty principle which is universally valid, in particular at the atomic level. However, it is assumed that the classical translational motion is approached when the de Broglie's thermal wavelenght of a molecule is less than the average distance between neighbouring molecules. This condition is observed for the most part of the systems at normal thermodynamic conditions (except for liquid helium and hydrogen at very low temperatures). For molecules with internal degrees of freedom the classical approximation is also valid when the rotational energy spacings are small compared with k_BT and the molecules are mainly in the ground vibrational level. A correct treatment of vibration, which is mainly a quantum phenomenon, generally requires special quantum simulation approaches. Yet, in a first order approximation, it can be treated in a classical way, by the introduction of harmonic potentials to describe the stretch and bending energies, and integrating the corresponding Newton's motion equations. This approach is typical in molecular mechanics calculations.

10.1.4 Boundary Conditions

As said before, for the simulation of bulk systems it is necessary to apply boundary conditions in order to attenuate the surface effects implicit in models with a relatively small number of molecules.

Cubic periodic conditions are commonly used [190]. This consists of enclosing N molecules in a cubic box (simulation box) whose volume, V, is chosen according to a predefined density. The simulation box behaves as though it was part of an infinite system by surrounding it with periodically repeated images of itself.

The evaluation of the potential energy and forces must consider not only the particles in the simulation box but also their images (*minimum image approximation*) in order to eliminate surface effects and approach a bulk system. When the interactions between molecules are short-ranged, that is, when they can be neglected after ∼3 molecular diameters (the cut-off distance, r_c) the calculation is carried out using the the *minimum image approximation with truncation*: given a molecule i, the simulation box is translated so that is centred on i. Then the molecule i only interacts explicitly with the molecules or their images within the sphere of radius r_c centred on molecule i. The distance r_c must be less than half of the simulation box side length. After the cut-off distance, an uniform distribution of molecules is assumed and long range corrections are analytically calculated.

This approach can not be applied in the study of ionic or high polar systems where the effect of the electrostatic interactions extends over many molecular diameters. In these cases, all images have to be accounted for. Ewald's sum [190, 215] is the typical method to take into account the long range interactions. The simulations are performed with the minimum image approximation without truncation and the electrostatic interactions are calculated by means of two rapidly convergent series, one in the real space and the other in the reciprocal space.

In those approaches, the density of the system is maintained by assuming that when a molecule leaves the simulation box through a wall, an image enters the opposite wall with the same velocity.

For some simulations it is inappropriate to use standard periodic boundary conditions in all directions. For example, when studying the adsorption and self-assembly of molecules onto a surface, the use of periodic boundary conditions for motion perpendicular to the surface it is clearly inappropriate. Rather, the surface is modeled as a true boundary, for example by explicitly including the atoms in the surface. The opposite side of the box must still be considered; when a molecule strays out of the box top side it is reflected back into the simulation box. Yet, usual periodic boundary conditions apply to motion parallel to the surface.

Periodic boundary conditions are not always applied in computer simulations. Some systems, such as liquid droplets or atomic and molecular clusters, inherently have boundary surfaces. The aim of their study is precisely to assess the influence of the surfaces in such small systems and to trace out the differences relatively to the corresponding bulks.

Finally, in other cases, the use of periodic boundary conditions would require a prohibitive number of atoms to be included in the simulations. This particularly arises in the study of the structural and conformational behaviour of macromolecules such as proteins and protein-ligand complexes. The first simulations of these systems ignored all solvent molecules due to the limited computational resources then available. This corresponds to the unrealistic situation of simulating an isolated protein in vacuum, and then comparing the results with experimental data obtained in solution. Such calculations can lead to considerable problems. A vacuum boundary tends to minimize the surface area and may distort the shape of the molecule if it is non-spherical. As computer power has increased it has become possible to incorporate explicitly some solvent molecules and thereby simulate a more realistic system. The simplest way to do this is to surround the molecule with a "skin" of solvent molecules. If the skin is sufficiently deep then the system is equivalent to a solute molecule inside a solvent "drop". Thus, the molecule-vacuum effects are transferred to the solvent-vacuum interface and more realistic results for the solute might be expected.

10.1.5 Computer Algorithms

A MD algorithm in the microcanonical, (E, V, N), ensemble consists of the following steps:

1. Assign initial positions $\boldsymbol{r}_i(0)$ to the molecules in the simulation box. The positions of a suitable lattice are generally chosen to initiate a calculation. Alternatively, the stored positions of a previous run can also be used.

2. Assign initial velocities $\boldsymbol{v}_i(0)$ to the molecules corresponding to a pre-defined temperature so that the total linear momentum is zero.

3. Calculate the potential energy, $U(\boldsymbol{r}^N)$.

4. Derive the force acting on each molecule:

$$\boldsymbol{F}_i = -\nabla_i U(\boldsymbol{r}^N) \tag{10.4}$$

where $U(\boldsymbol{r}^N)$ is the potential energy function of the system.

5. Integrate Newton's equations of motion for each one of the N molecules, taking into account boundary conditions if necessary. A commonly used numerical algorithm to perform the integration is Verlet's algorithm [190] in the so-called leap-frog version:

$$\boldsymbol{v}_i\left(t+\frac{\Delta t}{2}\right) = \boldsymbol{v}_i\left(t-\frac{\Delta t}{2}\right) + \boldsymbol{F}_i\left(t\right)\frac{\Delta t}{m_i} \tag{10.5}$$

$$\boldsymbol{r}_i\left(t+\Delta t\right) = \boldsymbol{r}_i\left(t\right) + \boldsymbol{v}_i\left(t+\frac{\Delta t}{2}\right)\Delta t \tag{10.6}$$

$$\boldsymbol{v}_i\left(t\right) = \left[\boldsymbol{v}_i\left(t+\frac{\Delta t}{2}\right) + \boldsymbol{v}_i\left(t-\frac{\Delta t}{2}\right)\right]/2 \tag{10.7}$$

where Δt is the integration time-step. The value of the integration time-step must be less than the molecular relaxation times and is of order 10^{-16}-10^{-14} seconds, depending on the type of molecules. A new microstate is then generated.

6. Go to 3.

7. Let the system evolve in time during n_e time-steps (equilibration run), accumulating the time averages of the mechanical properties over the $n_e$microstates. As soon as the system reaches equilibrium, that is, when the cumulative averages cease to show up significant drifts, store the last microstate of this run and reset to zero the variables for the cumulative averages.

8. Start a new run (production run) from the last microstate of 7). Let the system evolve in time for further n_p time-steps, a proper number to obtain good statistics, and calculate the final time averages of the properties over the respective microstates.

This algorithm keeps the volume and number of molecules constant. Furthermore, after the assignment of the initial positions and velocities the total energy must remain constant during the simulation. All the other properties will, however, fluctuate.

The extension of the algorithm to other ensembles requires the definition of the correct constraints and modification of the motion equations. For the simulation of systems consisting of polyatomic molecules rotational and vibrational degrees of freedom have also to be considered [190].

A Monte Carlo algorithm is similar to the MD one. Yet, there are neither the integration of motion equations (steps 4 and 5) nor the calculation of forces. The most part of MC methods do not sample the momenta space. As such, the assignment of molecular velocities (step 2) and their random alteration is also not performed in general.

Metropolis's algorithm generates a chain of configurations (called Markov's chain) where each one is obtained from the last generated configuration, accordingly the transition probabilities indicated in Section 10.1.2.

Considering the system in a configuration (o), defined by the molecular positions and orientations, the algorithm can be described as follows:

- Calculate the potential energy of that configuration, $U(o)$;
- Select a molecule and displace it to a new random position and orientation, inside the simulation box. A new configuration "n" is then obtained. Calculate the respective potential energy, $U(n)$;
- If $\Delta U = U(n) - U(o) < 0\,(\Leftrightarrow P(n) \geq P(o))$ the new configuration is accepted and becomes an effective member of the chain;
- If $\Delta U \geq 0$ the acceptance or rejection is decided based on $exp(-\beta\Delta U)$ by:

  1. generate a random number, *rand*, between 0 and 1.
  2. if $exp(-\beta\Delta U) > rand$ the new configuration is accepted
  3. if $exp(-\beta\Delta U) \leq rand$ the new configuration is rejected and the previous configuration is considered as a repeated member of the chain (transition $o \rightarrow o$)

- Repeat the process the number of times needed to obtain a good statistics, and calculate the desired properties as ensemble averages over the members of the chain.

10.1.6 Properties Calculation in Molecular Simulation

The calculation of equilibrium properties is equivalent in MD and MC. However, as in MC the configurations are not generated in a time sequence (there is no integration of the motion equations) the dynamic properties can not be directly measured by this method. On the contrary, MD is a fully deterministic method allowing the calculation of the dynamic properties from the trajectory of the system in phase space.

Here, it will be only indicated the calculation of the properties determined in Chapter 11.

The total energy of the system is given by:

$$E = \left\langle \sum_{i=1}^{N} \frac{1}{2} m_i \boldsymbol{v}_i^2 + U\left(\boldsymbol{r}^N\right) \right\rangle \tag{10.8}$$

where the brackets mean the ensemble or time averages.

If an effective pair potential [190] is used, the potential energy is pairwise additive and simply given by:

$$U\left(\boldsymbol{r}^N\right) = \sum_{i<j} u\left(r_{ij}\right) \tag{10.9}$$

where $u(\boldsymbol{r}_{ij})$ is the effective pair potential and $r_{ij} = |\boldsymbol{r}_i - \boldsymbol{r}_j|$.

If a MC method without sampling of the momenta space is used, the kinetic energy of each microstate is not explicitly calculated and the average kinetic energy is given by $(3/2)Nk_BT$.

The pressure of the system is given by the virial theorem [190]:

$$p = \left\langle \left(\frac{1}{3V}\right) \left(\sum_{i=1}^{N} \frac{p_i^2}{2m_i} + \sum_{i=1}^{N} \sum_{j>i}^{N} \boldsymbol{r}_{ij} \boldsymbol{F}_{ij}\right) \right\rangle \tag{10.10}$$

where V is the volume and $\boldsymbol{F}_{ij}$ is the force on molecule i due to molecule j.

If a MC method without momenta sampling is used, Nk_BT/V gives the average kinetic part of the pressure.

The temperature, T, of the system is given by the equipartition theorem [190]:

$$T = \frac{1}{3Nk_B} \left\langle \sum_{i=1}^{N} \frac{p_i^2}{2m_i} \right\rangle \tag{10.11}$$

Second order properties of the system can be measured from the fluctuations of the instantaneous properties [190]. The heat capacity at constant volume is given by:

$$C_v = \frac{3N}{2} k_B + \frac{\langle U^2 \rangle - \langle U \rangle^2}{k_B T^2} \tag{10.12}$$

where U is the potential energy.

One important structural property is the radial distribution function, rdf. This function gives the local density of the system at a distance r of a molecule taken as origin. It reflects the space correlations between the molecules. Indeed, when $r \to \infty$, the radial distribution function must approach the bulk density of the system. The normalized radial distribution function is given by:

$$g\left(r\right) = \left\langle \frac{V \Delta n\left(r\right)}{N 4\pi r^2 \Delta r} \right\rangle \tag{10.13}$$

where $\Delta n\left(r\right)$ is the number of molecules in the spherical shell of volume $4\pi r^2 \Delta r$ at a distance r from the molecule taken as origin.

Dynamical properties can also be measured by MD. The velocity autocorrelation function is defined by:

$$Z\left(t\right)=\left\langle \boldsymbol{v}_{i}\left(t\right)\boldsymbol{v}_{i}\left(0\right)\right\rangle \qquad (10.14)$$

where $\boldsymbol{v}_i$ is the velocity vector of molecule i. The average is taken over the time and the number of molecules.

The self-diffusion coefficient can also be obtained from $Z(t)$:

$$D=\frac{1}{3}\int_{0}^{\infty}Z\left(t\right)dt \qquad (10.15)$$

10.2 Potential Energy Surfaces

The classical simulation methods pressupose, as we have seen, the knowledge of the molecular potential energy functions, generally designated by potential energy surfaces (PES), a concept based on the Born-Oppenheimer approximation [216] which assumes the separability of the electronic and nuclear terms of the Hamiltonian of the system, due to the very high velocities and low masses of the electrons relatively to the velocities and masses of the nuclei.

A PES is a function of the coordinates of the nuclei of the molecules, in a convenient referential, describing the intra- and intermolecular interactions. It underlies the motions of the nuclei in the electronic fields, thereby shaping the translational, rotational and vibrational properties of the systems. Strictly, a PES is a solution of the electronic Schrödinger's equation. By introducing the PES in the nuclear Schrödinger's equation the translational, rotational and vibrational properties should come out. This is a formidable task, generally impossible to accomplish entirely for macroscopic real system. Thus, it is inevitable to resort to approximations.

The first approximation is to shape a PES by means of phenomenological and quantum-based terms. The second approach is to use classical mechanics, instead of the nuclear Schrödinger's equation, to describe the motions of the nuclei in the force field derived from the PES, as it is done, for example, in the classical molecular dynamics method.

In this context, the Lennard-Jones (LJ) potential, concerned with the van der Waals (vdW) interactions, is a typical and useful one-dimensional PES for monoatomic or spherical symmetric molecules, which have been used in the work of the next Chapter:

$$u(r_{ij})=4\varepsilon\left\{\left(\frac{\sigma}{r_{ij}}\right)^{12}-\left(\frac{\sigma}{r_{ij}}\right)^{6}\right\} \qquad (10.16)$$

where ε is the depth of the potential well, σ is approximately the molecular diameter, and r_{ij} is the distance between molecules i and j. While the attractive term, in r^{-6}, has a well-founded quantum mechanical root, the repulsive term, in r^{-12}, is just a phenomenological one. However, if the parameters are adjusted in order to reproduce certain experimental properties, LJ turns out a good *effective* pair potential for the study of many systems of chemical interest. *Effective,* means that the total potential energy can be approached in a pairwise form:

$$U\left(\boldsymbol{r}^N\right) = \sum_{i<j} u\left(r_{ij}\right) \tag{10.17}$$

that is, many-body terms are explicitly neglected, although they are implicitly taken into account, on average, by means of the parametrization based on experimental results.

Moreover, LJ takes part as an effective site-site potential, for example, in the following more general effective PES, extensively used in energy minimizations by molecular mechanics, classical simulations by MD and MC, and by Quantum Mechanics/Molecular Mechanics (QM/MM) hybrid methods [217]:

$$\begin{aligned} U \quad \left(r^N\right) = & \sum_{bonds} \frac{1}{2} k_l \left(l - l_e\right)^2 + \sum_{bends} \frac{1}{2} k_\theta \left(\theta - \theta_e\right)^2 \\ & + \sum_{dihedrals} \frac{V_n}{2} k_\chi \left(1 - cos\left(n\left(\chi - \chi_e\right)\right)\right) \\ & + \sum_{vdW} \left(\frac{c_{ij}^{12}}{r_{ij}^{12}} - \frac{c_{ij}^{6}}{r_{ij}^{6}}\right) + \sum_{coulombic} \frac{q_i q_j}{4\pi\varepsilon r_{ij}} \end{aligned} \tag{10.18}$$

where l, θ and χ are, respectively, the bond length, the bond angle and the dihedral angle (the subscript $'e'$ refers to the equilibrium values), and the k's are the respective force constants; V_n is the rotational barrier height and n the periodicity of the rotation; r_{ij} and c_{ij} are, respectively, the distance between the molecular sites i and j, and the vdW parameters; the $q's$ are the partial charges at the sites and ϵ the dielectric constant (to take into account average solvent effects).

Such a PES (called a *force-field* in the context of molecular mechanics) contains terms describing bond stretching, bond angle bending, dihedral torsions (intramolecular), Coulombic and vdW interactions. The last two terms (non-bonding interactions), are present not only inside a given molecule but also between different molecules (intermolecular). Some versions also contain additional terms describing, for example, hydrogen bonds and out-of-plane molecular distortions.

The parameters of a force-field are generally chosen to make the function fit experimental or *ab initio*/DFT data representative of a molecular class. Presently, various versions of force-fields are included in commercial and academic packages [217]. Differently from first principles methods, electronic effects are not explicitly considered in this kind of PES, although they are implicitly taken into account, on average, by the parametrization procedures. Therefore, bond formation or bond breaking (chemical reactions) and molecular properties which depend on subtle electronic details are not reproducible by such effective force fields. Thus, their study requires appropriate quantum methods. Among them is the Car-Parrinello method [218] that combines density functional theory and *ab initio* molecular dynamics.

Also, in the particular case of the interactions between organic molecules and metallic surfaces, quantum calculations are of the utmost importance to determine the respective PES. To this end, a convenient number of single point potential energies is firstly worked out. Then, three strategies can be used: i) fitting of analytical functions to the quantum results; ii) interpolation of the quantum results by special methods; or iii) mapping the whole PES by neural networks whose learning and test procedures are based on the quantum results.

As for the fitting of analytical functions, the most common methods use power series in an appropriate coordinate system, local functions such as splines and semiempirical potentials with adjustable parameters to reproduce experimental and theoretical results. The London-Eyring-Polanyi-Sato (LEPS) function to fit *ab initio* data [219–221] is an example. This methodology, however, have some disadvantages. It may not be able to reproduce, in general, the most subtle features of PES for complex systems from a limited number of energy points. Moreover, finding suitable analytical functions for complex systems, with many degrees of freedom, appears a non-trivial or even impossible task.

A recent method that does not require the previous knowledge of the analytical form is the corrugation reducing procedure [222]. It consists of two parts: the decomposition of the total molecule surface interaction into two contributions (one part strongly corrugated and the other weakly corrugated); and the interpolation, separately, of the two contributions using cubic splines and symmetry adapted functions. This method was used successfully to study diatomic molecules reacting on metal surfaces. Despite the good results, it is difficult to extend the method to problems with more than six degrees of freedom.

Another method that also does not require the knowledge of a priory analytical forms is the modified Shepard Interpolation [223–227]. Basically, it uses classical trajectory simulations to provide an iterative scheme for improving the PES. The surface is then given by interpolation of local Taylor expansions and an iterative procedure places new points only in regions of the configuration space that are important for dynamics. This method was used in gas phase reactions [223, 226, 227] and a modified one in gas surface reactions [224, 225].

Finally, another recent strategy to approach PES with similar or better results than the above referred to methods is the application of neural networks (NNs) [228–237] that shall be reviewed in Section 10.4. It also does not require the knowledge of analytical functions and it is applied in Chapter 12 to the interactions, calculated by DFT, of ethanol and Au(111) interfaces.

10.3 Density Functional Theory

The Density Functional Theory (DFT), is a powerful quantum-mechanical method for the calculation of molecular properties, particularly suitable to deal with electronic correlations. It is based on two theorems demonstrated by Hohenberg and Kohn in 1964 [238], whose fundamental statements are:

- The electronic density univocally determines all the properties of a given system.
- The energy calculated from an electronic density is the lowest, if the chosen density corresponds to the ground state of the system.

These theorems imply that the wave function of the system, which depends on 4N coordinates (three spatial and one for the spin, where N is the number of electrons), can be replaced by the electronic density which only depends on three variables. This replacement suggests that the wave function has more information than the necessary to fully describe the system [239].

In a molecular system the energy becomes a functional of the electronic density:

$$E\left[\rho\right] = K\left[\rho\right] + E_{ee}\left[\rho\right] + E_{Ne}\left[\rho\right] \tag{10.19}$$

where ρ is the electron density, E is the total energy, K is the kinetic energy, E_{ee} is the interaction energy between electrons, E_{Ne} is the interaction energy between the electrons and the atomic nuclei.

The last functionals can be divided in two groups, one dependent on the configuration of the nuclei and number of electrons, and the other universal, always calculated in the same way independently of the system under study. The “dependent” part corresponds to E_{Ne} and can be written as:

$$E_{Ne}\left[\rho\right] = \int \rho\left(\boldsymbol{r}\right) V_{Ne} d\boldsymbol{r} \tag{10.20}$$

where V_{Ne} is the interaction potential nuclei-electrons and r is the position vector. The independent part, F_{HK} , is designated by Hohenberg-Kohn’s functional and is written as

$$F_{HK}\left[\rho\right] = K\left[\rho\right] + E_{ee}\left[\rho\right] \tag{10.21}$$

Applying the variational principle, implied in the second theorem, the energy of the ground state can be written as

$$E_0 = \min_{\rho \to \rho_0} \left(F_{HK}\left[\rho\right] + \int \rho\left(\boldsymbol{r}\right) V_{Ne} d\boldsymbol{r} \right) \tag{10.22}$$

The universality of the F_{HK} functional is an important aspect of the DFT theory. This functional can be applied in the same way to the hydrogen atom or to systems of large dimension such as polymers and biomolecules. Unfortunately, the explicit form of this functional is unknown. Indeed, the knowledge of the analytical expression of the F_{HK} functional would allow the exact resolution of Shrödinger's equation for all chemical systems.

Hohenberg-Kohn's theorems, however, do not give any hint on the calculation of the ground state energy from the electronic density and on the functionals forms, or even, how to find the functionals.

Kohn and Sham [240], in 1965, proposed a solution to this problem. Their approach is based on a non-interacting distribution of electrons moving in a effective potential. They introduced the following formula for the universal functional, F_{HK}:

$$F_{HK}\left[\rho\right] = K_s\left[\rho\left(\boldsymbol{r}\right)\right] + J\left[\rho\left(\boldsymbol{r}\right)\right] + E_{xc}\left[\rho\left(\boldsymbol{r}\right)\right] \tag{10.23}$$

where $K_s\left[\rho\left(\boldsymbol{r}\right)\right]$ is the kinetic energy of the non-interacting system:

$$K_S\left[\rho\right] = -\frac{1}{2}\sum_{i}^{N}\left\langle\varphi_i \mid \nabla^2 \mid \varphi_i\right\rangle \tag{10.24}$$

and $J\left[\rho\left(\boldsymbol{r}\right)\right]$ is the classical electron interaction energy:

$$J\left[\rho\left(\boldsymbol{r}\right)\right] = \frac{1}{2}\sum_{i}^{N}\sum_{j}^{N}\int\int\left|\varphi_i\left(\boldsymbol{r}_1\right)\right|^2\frac{1}{r_{12}}\left|\varphi_j\left(\boldsymbol{r}_2\right)\right|^2 d\boldsymbol{r}_1 d\boldsymbol{r}_2 \tag{10.25}$$

and $E_{xc}\left[\rho\left(\boldsymbol{r}\right)\right]$ is the exchange-correlation (XC) energy whose explicit form is unknown. This term includes the non-classical electrostatic contributions, the correlation and exchange energies, and a small part of the kinetic energy.

The functions φ_i, $i = 1, 2, 3, ..., N$ are called the Kohn-Sham orbitals and are calculated from the resolution of the monoelectronic equations:

$$\hat{F}_{KS}\left(1\right)\varphi_i\left(1\right) = \varepsilon_{i,KS}\varphi_i\left(1\right) \tag{10.26}$$

where $\hat{F}_{KS}$ is the Kohn-Sham operator:

$$\hat{F}_{KS} = -\frac{1}{2}\nabla_1^2 + \sum_{j=1}^{N}\hat{J}_j\left(1\right) + V_{xc}\left(1\right) - \sum_{\alpha}\frac{Z_\alpha}{r_{1\alpha}} \tag{10.27}$$

The last term of this equation corresponds to the functional E_{Ne} of equation 10.19. The exchange-correlation potential, V_{xc}, is defined as the functional derivative of E_{xc} :

$$V_{xc} = \frac{\delta}{\delta\rho\left(\boldsymbol{r}_1\right)}E_{xc}\left[\rho\left(\boldsymbol{r}_1\right)\right] \tag{10.28}$$

The Kohn-Sham orbitals, φ_i , do not have a physical meaning. They are only used in the calculation of the electronic density through the expression:

$$\rho = \sum_{i=1}^{N} |\varphi_i|^2 \tag{10.29}$$

It is noteworthy that the wave function in DFT is not a Slater determinant of spin-orbitals (as in the Hartree-Fock theory), that is, there is no molecular wave function in DFT. Thus, the energies of the Kohn-Sham orbitals can not be interpreted as molecular orbital energies.

Since the $E_{xc}\left[\rho\left(\boldsymbol{r}\right)\right]$functional is not explicitly known, the DFT methods are based on various approximations to this term. Presently, there are three main levels of approach to this functional:

The *local density approximation* (LDA) or *local spin density approximation* (LSDA) [241, 242] is based on the idea of a uniform electron gas, a homogeneous arrangement of electrons moving against a positive background charge distribution that makes the total system neutral. These methods have been used since the beginning of the computational chemistry [243] with similar or inferior accuracy to Hartree-Fock (HF) methods.

The *generalized gradient approximation* (GGA) [241, 242, 244, 245] gives better results than the previous ones. These methods express the exchange and correlation functionals as a function of the electronic density and its first derivative. The most important practical feature of GGA functionals is that they depend not only in the value of the electronic density itself, but also on its derivative (gradient) with respect to the position in space. Several types of functionals have been proposed, some of which employ empirical parameters determined by fitting to experimental data.

Finally, the *hybrid functionals* are a combination of the LSDA and the GGA functional forms, with an exchange contribution that comes partly from an exchange functional and partly from the HF theory (where the exchange energy is calculated exactly). The appropriate mixing of the three terms is determined by fitting to experimental data [246]. The hybrid functionals present a high accuracy and reliability but are more expensive computationally than GGA due to the calculation of the HF exchange energy.

In the last years the computer hardware continues to follow the Moore's law (doubling the performance-price ratio every 18 months) making DFT methods well suited to use in "inexpensive", compared with the price of the first supercomputers, cluster-type computers. With the improvements in the DFT methods it appears that the scaling with system size have been solved making possible to apply these methods to molecular systems of dimensions much greater than the allowed for *ab initio* methods, in particular, when the electronic correlation plays an important role in the chemistry of the species under study. It is worth mentioning that, from a purist standpoint, DFT is not classified as an *ab initio* method, due to the empirical nature of the approximations necessary to calculate the exchange-correlation functional.

Chapter 12 presents experiments using data for the interaction of the ethanol/Au(111) calculated using DFT methods, implemented in the GAUSSIAN98 software [247]. The B3LYP method [248] was used. This method is a hybrid method based on *ab initio* and DFT approximations. It uses the HF exchange potential and the DFT exchange functionals, namely the exchange-correlation functional developed by Lee et al. [249], and implemented and parametrized by Beck [246, 248].

10.4 Overview of Potential Energy Surfaces by Neural Networks

As mentioned before the approach to PES, from *ab initio*/DFT data, is of high interest in molecular simulations. Section 10.2 presented some of the approaches to fit *ab initio*/DFT data to analytical functions. However, these methodologies show some disadvantages. They may not be able to reproduce, in general, the most subtle features of PES for complex systems from a limited number of energy points. Moreover, finding suitable analytical functions for complex systems, with many degrees of freedom, appears a non-trivial or even impossible task.

A recent strategy to map PES, with similar or better results than the referred to methods, is the application of neural networks (NNs) [228–237].

No et al [234] used a Feed-Forward Neural Network (FFNN) to fit a PES for the water dimer using as training points the energy values obtained by *ab initio* methods. Gassner et al. [230] applied FFNNs to reproduce the three-body interaction energy in the system H_2O-Al^{3+}-H_2O, and compared the radial distribution function obtained by MC simulations using an analytical potential and the PES generated by FFNNs. They show that FFNNs can be used to reproduce the dimer energies in the whole conformational space.

Prudente and Neto [235] also applied FFNNs to map PES, and the transition dipole moment of HCl^+, from the electronic energies calculated by *ab initio* methods. The NNs-PES was used in the study of the photodissociation process. They show that for multi-dimensional surfaces the results of FFNNs can be better than those of splines. A similar work was done by Bittencourt et al [228] for OH^-, also from *ab initio* data. In this, the properties evaluated and used to test the accuracy of the NNs fittings were the vibration levels and the transition probabilities between the $A^2\sum^+$ and $X^2\Pi$ electronic states.

Cho et al. [229] used an NN to build a polarizable force field for water. The PES implemented is a combination of a empirical potential (the TIP4P model) and a non-empirical one (a FFNN was used to reproduce the many body interaction energy surface from *ab initio* calculations). The new force field was applied in MC simulations and some structural and thermodynamic properties were compared with experimental data, showing that FFNNs could be used in the development of a force field to treat many-body interactions. In the same way, but with different goals, Rocha Filho et al. [237] used NNs to map *ab initio* data for the ground state of H^{3+} .

The investigation of reactions in surfaces has also been performed by NNs. For example, Lorenz et al. [232] built continuous PES for H_2 interacting with the Pd (100) surface covered by potassium. More recently, the same authors [231] applied NNs to fit six dimensional PES for the H_2 dissociation on clean and sulfur covered Pd (100) surfaces. The models were used to describe reaction rates for the dissociative adsorption in those surfaces and showed that a description of dissociation reactions with NNs are orders of magnitude smaller than those of “on the fly” *ab initio* dynamics.

Witkoskie and Doren [250] applied NNs to represent PES of some prototypical examples of one, two and three dimensions. The authors tried to achieve the optimal number of neurons and NNs configuration as well as the amount of data needed to map each one of these simple cases.

Toth et al. [251], instead of using energy points obtained from *ab initio* calculations, took diffraction data on liquids to obtain the pair potentials. NNs were then trained with known pair interaction- simulated structure factors of one-component systems.

Advantageously, NNs do not require analytical functions, that is, neither the functional type (polynomial, exponential, logarithmic, etc) nor the adjustable parameters need to be given.

An important question is whether or not the mapping provided by NNs can be more accurate than other approximation schemes based on fitting analytical functions or special interpolation techniques. That depends, of course, on the complexity of the PES and the number of single point energies available to a given system.

The next two Chapters present results concerning the mapping of PES by NNs in two distinct situations. Chapter 11, reports ensembles of Feed-Forward Neural Networks (EnsFFNNs) and Associative Neural Networks (ASNNs) trained for mapping PES which are also represented by well-known analytical potential functions. The accuracy of the NNs predictions was assessed by comparison of the simulation results from NNs and the analytical PES. The parametrized Lennard-Jones (LJ) potential for argon was used to test the models. MD simulations were performed using the tabular potential energies, predicted by the NNs, to work out thermal, structural and dynamical properties which compare well with the values obtained from the LJ analytical function [252].

Chapter 12 has the main objective of assessing an alternative method to map multidimensional PES for the interaction of ethanol and Au (111) interfaces regarding the simulation of the adsorption and self-assembly of alkylthiols solvated by ethanol [253].

Chapter 11

NNs to Approach Potential Energy Surfaces: Application to a MD Simulation

11.1 Introduction

A recent strategy to approach PES with similar or better results than other mapping methods is the application of neural networks (NNs) [228–237] as reviewed in Section 10.4.

Although NNs seem a suitable approach for estimating PES from *ab initio*/DFT energy data sets, the accuracy of the properties determined by MC and MD simulation methods from NNs generated PES has not yet, to our knowledge, been systematically analyzed in terms of the minimum number of energy points required for training different NN models.

The main goal of the work presented in this Chapter is to train NNs for reproducing PES represented by well-known analytical potential functions, and then to assess the accuracy of the method by comparing the simulation results obtained from NNs and analytical PES. Ensembles of feed-forward neural networks (EnsFFNNs) [63,64] and associative neural networks (ASNNs) [65,66] were the machine learning methods used to estimate the full energy surface. Differently from other authors, the experiments reported here use these methods instead of single Feed-Forward Neural Networks (FFNNs). In other problems of modeling and fitting, the advantages and greater generalization ability of these two methods over single FFNNs have been demonstrated. Although the two methods are supervised learning techniques, the EnsFFNNs is a memory-less method (after the training all information about the input patterns are stored in the NN weights but there is no explicit storage of the data in the system) while the ASNNs is a combination of memory-less and memory-based methods (the data used to build the models are also stored in a "memory" and the predictions are corrected based on some local approximations of the stored examples). The advantages of these two methods over the common FFNNs will be demonstrated in this work.

Training sets with different number of points, from 15 differently parameterized Lennard-Jones (LJ) potentials, are used and argon is taken to test the NNs. MD simulations are performed using the tabular potential energies, predicted by NNs, for working out thermal, structural and dynamic properties which are then compared with the values obtained from the LJ analytical function.

Advantageously, NNs do not require an analytical function on which the model should be built, i.e. neither the functional type (polynomial, exponential, logarithmic, etc) nor the number and position of the parameters in the model function need to be given.

An important question is whether or not the mapping provided by NNs can be more accurate than other approximation schemes based on fitting analytical functions and special interpolation techniques. That depends, of course, on the complexity of the PES and the number of single energy points obtained for a given system. The work presented here does not address such question, since the experiments are based on a simple analytical potential function from which an arbitrary number of energy points can always be worked out. As far as molecular simulations are concerned when a new method or theory is proposed the usual way is to test it with LJ PES, from which a huge amount of physical properties has been calculated along the past years. If the test results are accurate enough then one can proceed towards more complex and multi-dimensional PES. This is just the spirit of the present work.

The next Section contains the methodology and computational details. Section 11.3 discusses the NNs results and their comparison with the ones from the Lennard-Jones potential. Finally Section 11.4 presents the concluding remarks.

11.2 Methodology and Computational Details

The experiments here described involve three main steps. In the first, six training sets with different numbers of energy points were generated for 15 differently parameterized LJ potentials:

$$u(r) = 4\varepsilon \left\{ \left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6} \right\} \tag{11.1}$$

where ε is the depth of the potential well, σ is approximately the molecular diameter, and r is the distance between particles.

For each of these training sets, a test set was generated with the parameters for argon (argon points were not used in the training set). The EnsFFNNs and ASNNs are the machine learning methods used to set up the relation between the input (parameters σ, ε and the distance between particles, r) and the output (potential energy). In the second step, the NN-generated PES of argon, in tabular form, was used in MD simulations. The thermal, structural, and dynamic properties evaluated for the system were compared with the ones obtained from the analytical PES for argon. To work out the dynamics of the system from the tables predicted by NNs it is necessary to perform interpolations, which introduce some errors. As such, the dynamics based on the analytical potential has also been calculated from tables directly constructed from the analytical potential for the same number of points. In this way, when the results from the two calculations are compared, the errors due to the interpolations are practically eliminated. In the third step experiments were performed to assess the ability of an ASNN trained with a small number of points to improve the generated PES by using different memories with more points, or by retraining with more points. Also, the interpolation ability of the NN with a different set of descriptors and different sets of curves in the training set was analyzed.

11.2.1 Data set of PES Generated by LJ Potential to Train and Test NN

The two methods used to learn the relationship between the LJ parameters and the potential energy are supervised learning methods. Therefore they need a set of pairs to learn: the inputs and the targets. In our case we have three inputs: the depth of the potential well, ε, the distance at which the potential is equal to 0, σ, and the distance between particles, r.

The output is the potential energy, $u(r)$. The LJ parameters used for 16 substances [254, 255] are listed in Table 11.1.

Six independent training sets, with different number of energy points, were generated with the analytical expression for the 15 PES. The PES used to train and test the models were generated between $0.5\times\sigma$ and $2.5\times\sigma$ with different intervals for the five training sets and with an interval of $\sim 7.246\times10^{-4}$ Å for the test set. These values were chosen because in MD simulations the potential energy calculated with distances between particles shorter than $0.5\times\sigma$ and larger than $2.5\times\sigma$ have no influence on the simulation results. The number of objects in each training set was 1,312, 674, 518, 442, 390, and 267.

Because of the steep varying nature of the first part of the potential, and the requirement that the input and output of NNs is normalized between 0.1 and 0.9, each PES of the training set was divided in one part with r between $0.5\times\sigma$ and σ , and a second part with r from σ to $2.5\times\sigma$. Accordingly, each data set was divided into two subsets, and each one was used to separately train NNs. Each network makes predictions for one part of the potential. After this division, the training sets have the following number of points (in each part of the curve): 1,312 (315+997), 674 (162+512), 518 (162+356), 442 (162+280), 390 (162+228), and 267 (96+171).

Table 11.1: Lennard-Jones parameters for the substances used in the training and test of NNs.

	ε / K	σ / pm
He	10.22	258.0
C_2H_2	209.11	463.5
C_2H_4	200.78	458.9
C_2H_6	216.12	478.2
C_6H_6	377.46	617.4
CCl_4	378.86	624.1
CO_2	201.71	444.4
F_2	104.29	357.1
Kr	154.87	389.5
N_2	91.85	391.9
O_2	113.27	365.4
Xe	213.96	426.0
Ne	35.7	279
CH_4	137	382
Cl_2	296.27	448.5
Ar	111.84	362.3

11.2.2 Feed-Forward Neural Networks

FFNNs [1] were defined with three input neurons, one hidden layer of neurons, and one output neuron. In the input layer and in the hidden layer, an extra neuron (called bias) with the constant value of one was also added. Section 2.8 presents the full description about FFNNs.

The networks were trained with the ASNN program from Igor Tetko [65,66] to predict the potential energy for each pair of particles with a given separation, taking as input the distance between the particles (r) and the LJ parameters (ε and σ). Corrections were performed on the weights during the training (learning) using the Levenberg-Marquardt algorithm [59, 256]. The number of neurons in the hidden layer was optimized for each case, generally in the range 10-25. Before the training, the whole training set is randomly partitioned into a learning set with 50% of the objects and a validation set with the other 50%. Full cross-validation of the entire training set was performed using the leave-one-out method (LOO). The activated function used is the logistic function and each input and output variable was linearly normalized between 0.1 and 0.9 on the basis of the training set. The maximum number of iterations used in the training was set to 5,000 or 10,000. The training was stopped when there was no further improvement in the root mean squared error (RMSE) for the validation set [56]. After the training, the results were calculated for the learning set, the validation set and for the LOO method.

11.2.3 Ensemble of Feed-Forward Neural Networks

An EnsFFNN is made of several independently trained FFNNs that contribute to a single prediction [63, 64]. The final prediction for an object, in our case the potential energy, is the average of the outputs from all FFNNs of the ensemble.

This methodology smoothes random fluctuations in the individual predictions of individual FFNNs.

11.2.4 Associative Neural Networks

An Associative Neural Network (ASNN) [65,66] is a combination of a memory-less (ensemble of Feed-Forward Neural Networks) and a memory-based method (K-Nearest Neighbor [67] technique). Full details about this methodology can be found in Chapter 2.8.

The EnsFFNNs is combined with a memory into a so-called Associative Neural Network (ASNN) [65, 66]. In this work, the memory consists of a list of potential energy points, represented by their three descriptors, and their calculated value using the LJ potential. The ASNN scheme is employed for composing a prediction of the potential energy from: *a)* the outputs from the EnsFFNNs, and *b)* the data in the memory. When a query point in a PES is submitted to an ASNN, the following procedure takes place to obtain a final prediction of the potential energy:

1. The descriptors of the energy point are presented to the ensemble, and a number of output values are obtained from the different FFNNs of the ensemble - the output profile of the query energy point.

2. The average of the values in the output profile is calculated. This is the uncorrected prediction of the potential energy for the query point in the PES.

3. Every potential energy point of the memory is presented to the ensemble to obtain an output profile.

4. The memory is searched to find the k-nearest neighbors of the query energy point. The search is performed in the output space, i.e. the nearest neighbors are the potential energy points with the most similar output profiles (calculated in Step 3) to the query energy point (calculated in Step 1). Similarity is here defined as the Spearman correlation coefficient between output profiles.

5. For each of the KNN energy points, an (uncorrected) prediction is also obtained-the average of its output profile.

6. The uncorrected predictions for the KNN energy points (calculated in Step 5) are compared with their potential energy values in the memory. The mean error is computed.

7. The mean error computed in Step 6 is added to the uncorrected prediction of the query energy point (computed in Step 2) to yield the corrected prediction for the query point.

The experiments here described were carried out with the ASNN program [257].

11.2.5 Molecular Dynamics Simulations

Using the six different training sets (each containing a different number of points) six EnsFFNN and six ASNN models were trained and separately applied to generate the PES of argon (external tables with 1,003 points). These 12 external tabular PES were used in MD simulations with cubic periodic boundary conditions. The MD program reads and interpolates the potential energy generated by NNs from the external tables. The simulation results were compared with the ones from the LJ analytical function.

The simulations were carried out in the NVT ensemble (damped-force method [190]) at different thermodynamic conditions, with a time step of 1.0×10^{-14}s for the numerical integration of Newton's equations of motion (using Verlet's leap-frog algorithm [190]) and 50,000 time steps for the equilibration and production runs. Different regions of the phase diagram of argon were tested.

11.3 Results and Discussion

The discussion on training and testing the EnsFFNNs and ASNNs is followed by the analysis of the results from MD simulations using the PES generated by the two NN models and the LJ analytical function. The improvement of the accuracy on the NN-generated PES using NN retrained, or different memories, was investigated, as well as the interpolation ability of the NN trained with different set of descriptors and different sets of potential energy curves in the training set.

11.3.1 Generation of PES by ENSFFNNs and ASNNs (Training and Testing)

As mentioned earlier, the PES used to train NNs were divided into two parts ($0.5\times\sigma < r < \sigma$, and $\sigma < r < 2.5\times\sigma$). For the first part, three training sets were used with 315, 162, and 96 energy points. The results of EnsFFNNs and ASNNs for the first part of the PES are displayed in the Table 11.2.

Table 11.2: Mean absolute error (MAE) of training and test sets, using EnsFFNNs and ASNNs, for different training sets in the first part of PES ($0.5 \times \sigma < r < \sigma$).

NN Type	Data sets	MAE[a] / K		
		315	162	96
EnsFFNNs	Learning	231.21 (0.10)[b]	511.33 (0.19)	2680.79 (1.13)
	Validation	728.74 (0.67)	2753.77 (1.83)	8928.56 (5.17)
	LOO	737.15 (0.68)	2772.58 (1.96)	9579.33 (5.77)
	Test	352.30 (1.05)	649.52 (2.84)	4074.20 (4.90)
ASNNs	Learning	790.59 (0.26)	511.33 (0.19)	2219.78 (1.31)
	Validation	1188.12 (0.76)	2753.77 (1.8)	8567.78 (4.68)
	LOO	1196.52 (0.77)	2772.58 (1.96)	9288.02 (5.27)
	Test	812.51 (0.99)	649.52 (2.84)	2008.17 (3.76)

$MAE = \frac{\sum_{i=1}^{n}|y_{cal} - y_{exp}|}{n}$ where y_{calc} is the output of the NN and y_{exp} the target.
[a] 315, 162, 96 are the number of energy points used in training.
[b] Values in parentheses indicate the percentages of error.

For the second part of the curve (r between σ and 2.5× σ) six different training sets were considered and the results are displayed in Table 11.3.

NNs with different number of neurons in the hidden layer were used for each training set. For the data sets with 997, 512, and 356 energy points a hidden layer with 25 neurons were used, while 15 neurons were used for the data set with 280 energy points, and 10 hidden neurons for the data sets with 228 and 171 energy points. The results with fewer neurons were not much different. In the case of the training set with 512 points, a MAE of 0.092 K was obtained for the test set by an ASNN with 25 hidden neurons. With 20, 15, 10, and 5 neurons in the hidden layer, MAE of 0.126, 0.167, 0.168, and 0.212 K were obtained, respectively.

As expected, higher MAE were observed in the first part of the curve, although always lower than 5% for the test set. Tables 11.2 and 11.3 show that the results using ASNNs are in general better than those using only EnsFFNNs, particularly for the test set, and for the NNs trained with fewer points.

Although the results for the test set corresponding to the second part of the curve are considerably better using 997 points, the PES generated by NNs for argon (test) has an acceptable MAE when a smaller number of energy points were used in the training. For example, the model trained with 280 points shows a MAE of 0.135 K for the test set against 0.050 K of the model trained with 997 points. Reducing the number of points below 280 resulted in a significant increase of the MAE. This trend is more visible when going from 228 to 171 energy points. A difference of nearly 60 energy points causes an increase in MAE of almost 300%. This transition is considered as the limit of the minimum

Table 11.3: Mean Absolute Error (MAE) for training and test sets using EnsFFNNs and ASNNs trained with different sets for the second part of the PES ($\sigma < r < 2.5 \times \sigma$).

NN Type	Data sets	MAE[a] / K					
		997	512	356	280	228	171
EnsFFNNs	Learning	0.116	0.191	0.487	0.601	0.650	0.899
	Validation	0.281	0.760	1.337	1.724	1.853	2.786
	LOO	0.269	0.836	1.476	1.809	2.114	3.092
	Test	0.096	0.116	0.221	0.277	0.578	1.022
ASNNs	Learning	0.109	0.183	0.406	0.562	0.713	0.796
	Validation	0.237	0.682	1.109	1.409	1.450	2.469
	LOO	0.227	0.752	1.238	1.497	1.638	2.758
	Test	0.050	0.092	0.118	0.135	0.219	0.639

$MAE = \frac{\sum_{i=1}^{n} |y_{cal} - y_{exp}|}{n}$ where y_{calc} is the output of the NN and y_{exp} the target.
[a] 997, 512, 356, 280, 228, 171 are the number of energy points used in training.

number of energy points required for ASNN to learn the relationship between the input and the potential energy. Figure 11.1 displays the PES for argon generated by the LJ function and by the ASNN trained with 442 (162+280) energy points.

Errors for the test set are often lower than those obtained for the training set, particularly for the second part of the curve. This can be understood considering that all the points of the test set are from the argon PES, a curve well within the space of the training set. Curves in the training set with maximum and minimum LJ parameters are more difficult to learn.

The overlap of the two PES in all regions is good. The region with worse results is near where the potential is zero (this was the region of the transition where the subdivision of the curve in two parts was made). For a set of 25 points the average error in this region is 8%. The other parts of the curve present an error considerably smaller (less than 1%). For example, near the minimum of the potential the average error is 0.09% for 25 energy points and in the final part of the curve the average error is 0.59% also considering 25 energy points. Tables 11.4, 11.5 and 11.6 shows the absolute error for 25 energy points for each one of the regions cited earlier.

The region that presents the highest absolute errors (up to 43%) is for values of r where the potential is near zero. Absolute errors of 2.522 and 2.290 K were obtained for distances between particles of 0.35650 and 0.35158 Å. In the minimum of the potential the errors are ~0.1 K and ~0.1% and for the final part of the curve the absolute errors are less than 0.03 K.

Table 11.4: Absolute error of the potential energy of argon generated by ASNN, trained with 442 (162 + 280) energy points, compared with the analytical PES ($0.35359 < r < 0.37101$).

r / Å	u_{anal}/K	u_{pred}/K	$Error/K$
0.35360	81.212	80.298	0.914 (1.13%)
0.35433	73.000	72.910	0.091 (0.12%)
0.35505	65.080	65.896	0.816 (1.25%)
0.35578	57.442	59.624	2.182 (3.80%)
0.35650	50.077	52.599	2.522 (5.04%)
0.35723	42.977	44.264	1.288 (3.00%)
0.35795	36.132	37.163	1.031 (2.85%)
0.35868	29.535	29.522	0.013 (0.04%)
0.35940	23.178	23.641	0.463 (2.00%)
0.36013	17.053	17.871	0.818 (4.79%)
0.36085	11.154	13.285	2.131 (19.11%)
0.36158	5.471	7.761	2.290 (41.85%)
0.36230	0.000	2.247	2.247 (-)
0.36302	-5.267	-3.006	2.261 (42.93%)
0.36375	-10.337	-8.117	2.220 (21.48%)
0.36447	-15.216	-12.962	2.254 (14.82%)
0.36520	-19.910	-17.781	2.129 (10.69%)
0.36592	-24.424	-22.452	1.972 (8.07%)
0.36665	-28.765	-26.973	1.792 (6.23%)
0.36737	-32.938	-31.346	1.593 (4.84%)
0.36810	-36.949	-35.569	1.380 (3.73%)
0.36882	-40.802	-39.643	1.158 (2.84%)
0.36955	-44.503	-43.572	0.931 (2.09%)
0.37027	-48.056	-47.355	0.701 (1.46%)
0.37100	-51.467	-50.791	0.676 (1.31%)

Values in parentheses indicate the percentages of error. r, distance between particles; u_{anal}, potential generated by analytical function; u_{pred}, potential generated by ASNN; $Error$, absolute error.

Table 11.5: Absolute error of the potential energy of argon generated by ASNN, trained with 442 (162 + 280) energy points, compared with the analytical PES ($0.39780 < r < 0.41521$).

r / Å	u_{anal}/K	u_{pred}/K	$Error/K$
0.39781	-109.606	-109.756	0.151 (0.14%)
0.39853	-109.980	-110.126	0.145 (0.13%)
0.39925	-110.317	-110.455	0.139 (0.13%)
0.39998	-110.616	-110.746	0.131 (0.12%)
0.40070	-110.879	-110.883	0.004 (0.00%)
0.40143	-111.108	-111.102	0.006 (0.01%)
0.40215	-111.303	-111.286	0.017 (0.02%)
0.40288	-111.466	-111.475	0.009 (0.01%)
0.40360	-111.599	-111.596	0.003 (0.00%)
0.40433	-111.701	-111.686	0.015 (0.01%)
0.40505	-111.775	-111.746	0.028 (0.03%)
0.40578	-111.820	-111.779	0.041 (0.04%)
0.40650	-111.839	-111.921	0.082 (0.07%)
0.40723	-111.833	-111.901	0.069 (0.06%)
0.40795	-111.801	-111.845	0.044 (0.04%)
0.40867	-111.745	-111.776	0.031 (0.03%)
0.40940	-111.667	-111.526	0.141 (0.13%)
0.41012	-111.566	-111.749	0.183 (0.16%)
0.41085	-111.444	-111.614	0.170 (0.15%)
0.41157	-111.301	-111.459	0.158 (0.14%)
0.41230	-111.139	-111.284	0.145 (0.13%)
0.41302	-110.958	-111.091	0.133 (0.12%)
0.41375	-110.758	-110.879	0.121 (0.11%)
0.41447	-110.541	-110.651	0.110 (0.10%)
0.41520	-110.307	-110.614	0.307 (0.28%)

Values in parentheses indicate the percentages of error. r, distance between particles; u_{anal}, potential generated by analytical function; u_{pred}, potential generated by ASNN; $Error$, absolute error.

Table 11.6: Absolute error of the potential energy of argon generated by ASNN, trained with 442 (162 + 280) energy points, compared with the analytical PES ($0.88907 < r < 0.90648$).

r / Å	u_{anal}/K	u_{pred}/K	$Error/K$
0.88908	-2.039	-2.013	0.026 (1.26%)
0.88981	-2.029	-2.005	0.024 (1.16%)
0.89053	-2.019	-1.998	0.022 (1.07%)
0.89126	-2.009	-1.990	0.020 (0.97%)
0.89198	-2.000	-1.982	0.017 (0.87%)
0.89271	-1.990	-1.975	0.015 (0.77%)
0.89343	-1.980	-1.967	0.013 (0.67%)
0.89416	-1.971	-1.960	0.011 (0.57%)
0.89488	-1.961	-1.952	0.009 (0.46%)
0.89561	-1.952	-1.945	0.007 (0.36%)
0.89633	-1.942	-1.938	0.005 (0.25%)
0.89705	-1.933	-1.930	0.003 (0.14%)
0.89778	-1.924	-1.923	0.001 (0.03%)
0.89850	-1.915	-1.894	0.021 (1.09%)
0.89923	-1.905	-1.887	0.019 (0.98%)
0.89995	-1.896	-1.880	0.017 (0.87%)
0.90068	-1.887	-1.873	0.014 (0.76%)
0.90140	-1.878	-1.866	0.012 (0.65%)
0.90213	-1.869	-1.859	0.010 (0.53%)
0.90285	-1.860	-1.852	0.008 (0.42%)
0.90358	-1.851	-1.846	0.006 (0.30%)
0.90430	-1.842	-1.839	0.003 (0.18%)
0.90503	-1.834	-1.833	0.001 (0.06%)
0.90575	-1.825	-1.826	0.001 (0.07%)
0.90647	-1.816	-1.820	0.003 (0.19%)

Values in parentheses indicate the percentages of error. r, distance between particles; u_{anal}, potential generated by analytical function; u_{pred}, potential generated by ASNN; $Error$, absolute error.

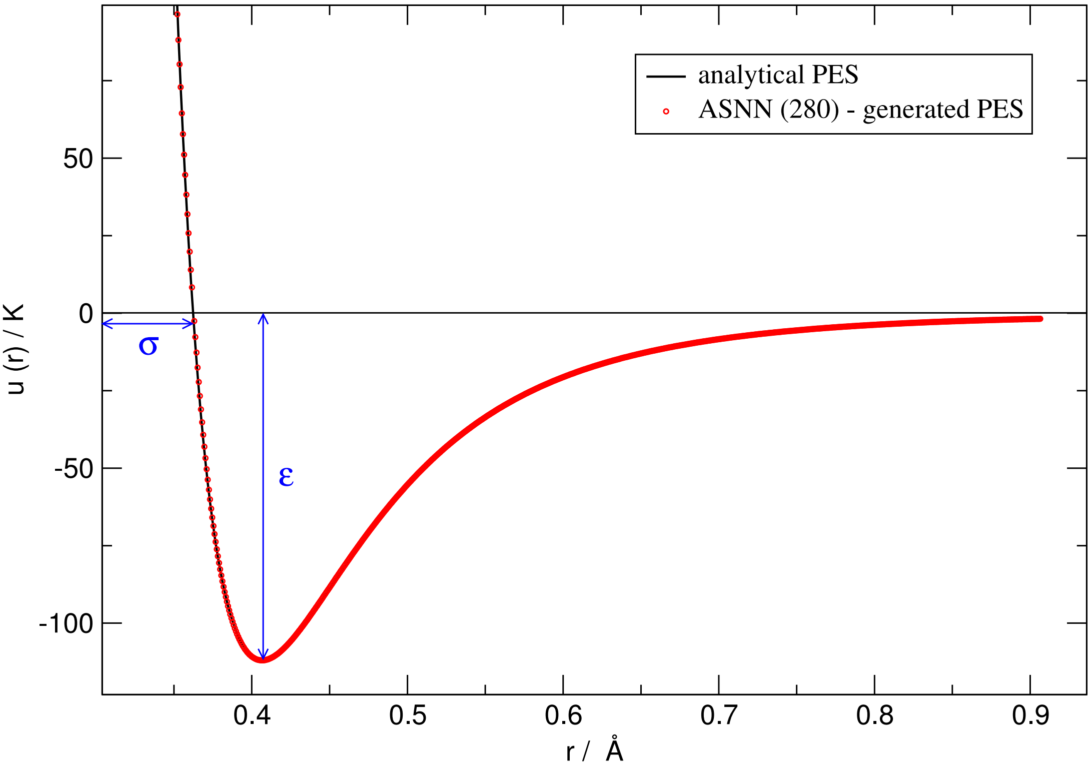


Figure 11.1: Potential energy of argon generated by ASNN, trained with 442 (162+280) energy points, compared with the analytical PES. ε and σ are the parameters of the LJ potential [see Eq. 11.1].

11.3.2 Properties from Molecular Dynamics Simulations

The simulation results are collected in Tables 11.7 and 11.8. The errors are measured relatively to the results based on the tabular PES of argon directly constructed from the analytical function.

The first general conclusion, from the comparison of the two different methods, is that ASNNs give slightly better results than EnsFFNNs.

The energies have errors less than 1% for the first five training sets, only for the training set with 267 energy points errors of 2-3% were obtained. The heat capacities are also in good accordance with the values worked out from the LJ analytical function. The pressure is, as expected, the property that presents the most irregular behavior.

The ASNN model that gives the better results (error 0.01%) for the pressure is the one trained with 442 (162+280) energy points. The pressure is the most difficult property to obtain with a high accuracy even with a great number of points in the training set. Also, from the simulation results obtained by the different training sets we can conclude that the number of points used and the overall fit of the curve are not crucial for obtaining the best results. Errors in some regions of the curve are more significant than the global MAE for the whole curve. On the whole, the model that gives the best results is the ASNN trained with 442 points. For five of the six properties evaluated this model gives

Table 11.7: Thermal and dynamic properties evaluated in MD simulations (NVT ensemble, N=256, T=1.0, ρ=0.85) using the PES of argon generated by EnsFFNNs.

Properties	Analytical	NN generated PES					
	Function	1312	674	518	442	390	267
E	-4.322	-4.359 (0.85)	-4.306 (0.38)	-4.312 (0.22)	-4.305 (0.40)	-4.358 (0.82)	-4.449 (2.93)
K	1.500	1.500 (0.00)	1.500 (0.00)	1.500 (0.00)	1.500 (0.00)	1.500 (0.00)	1.500 (0.00)
U	-5.822	-5.859 (0.63)	-5.806 (0.28)	-5.812 (0.17)	-5.805 (0.30)	-5.858 (0.61)	-5.949 (2.18)
p	1.936	1.930 (0.33)	1.978 (2.15)	1.993 (2.94)	2.008 (3.70)	1.933 (0.15)	1.805 (6.78)
Cv	2.519	2.576 (2.29)	2.519 (0.03)	2.593 (2.94)	2.546 (1.10)	2.536 (0.69)	2.587 (2.71)
DC / cm^2s^{-1}	2.635E-5	2.638E-5 (0.11)	2.505E-5 (4.95)	2.670E-5 (1.33)	2.575E-5 (2.28)	2.639E-5 (0.16)	2.661E-5 (0.97)

T, temperature; ρ, density; E, total energy; K, kinetic energy; U, potential energy; p, pressure; Cv, heat capacity; (all in reduced units). DC, diffusion coefficient.

Within the parentheses are the errors of the properties using NN-generated PES relatively to the results using the analytical function. 1,312, 674, 518, 442, 390, 267 are the number of energy points used in training.

Table 11.8: Thermal and dynamic properties evaluated in MD simulations (NVT ensemble, N=256, T=1.0, ρ=0.85) using the PES of argon generated by ASNNs.

Properties	Analytical	NN generated PES					
	Function	1312	674	518	442	390	267
E	-4.322	-4.341 (0.44%)	-4.326 0.09%)	-4.334 (0.28%)	-4.324 (0.03%)	-4.346 (0.55%)	-4-422 (2.31%)
K	1.500	1.500 (0.00%)	1.500 (0.00%)	1.500 (0.00%)	1.500 (0.00%)	1.500 (0.00%)	1.500 (0.00%)
U	-5.822	-5.841 (0.33%)	-5.826 (0.07%)	-5.834 (0.21%)	-5.824 (0.03%)	-5.846 (0.41%)	-5.846 (0.41%)
p	1.936	1.922 (0.71%)	1.928 (0.40%)	1.923 (0.67%)	1.936 (0.01%)	1.903 (1.71%)	1.847 (4.62%)
Cv	2.519	2.564 (1.78%)	2.559 (1.59%)	2.527 (0.35%)	2.544 (1.02%)	2.544 (1.02%)	2.570 (2.03%)
DC / cm^2s^{-1}	2.635E-5	2.581E-5 (2.04%)	2.582E-5(2.03%)	2.700E-5(2.48%)	2.607E-5(1.07%)	2.673E-5(1.44%)	2.555E-5 (3.05%)

T, temperature; ρ, density; E, total energy; K, kinetic energy; U, potential energy; p, pressure; Cv, heat capacity; (all in reduced units). DC, diffusion coefficient.

Within the parentheses are the errors of the properties using NN-generated PES relatively to the results using the analytical function. 1,312, 674, 518, 442, 390, 267 are the number of energy points used in training.

Table 11.9: Thermal and dynamic properties evaluated in MD simulations (NVT ensemble, N=256) in different regions of the phase diagram of argon using the PES of argon generated by ASNN (trained with 442 energy points).

Properties	ρ=0.818;T=0.799 (near triple point)	ρ=0.850;T=1.0 (liquid pocket)	ρ=0.850;T=1.5 (fluid)
E	-4.644 (0.02%)	-4.324 (0.03%)	-3.091 (0.05%)
K	1.199 (0.00%)	1.500 (0.00%)	2.250 (0.00%)
U	-5.843 (0.02%)	-5.824 (0.03%)	-5.341 (0.03%)
p	0.224 (2.23%)	1.936 (0.01%)	4.612 (0.08%)
Cv	2.538 (0.01%)	2.544 (1.02%)	2.400 (0.22%)
$DC\ /\ cm^2s^{-1}$	2.431E-5 (1.08%)	2.607E-5 (1.07%)	4.563E-5 (2.29%)

T, temperature; ρ, density; E, total energy; K, kinetic energy; U, potential energy; p, pressure; Cv, heat capacity; (all in reduced units). DC, diffusion coefficient.
Within the parentheses are the errors of the properties using NN-generated PES relatively to the results using the tabular PES generated using the analytical function.

better results than the ASNN trained with 1,312 energy points. However, the results are similar for the first five tested cases with errors lower than 2.5%. In the last experiment, with only 267 energy points, a significant increase in the errors of some properties was observed.

In the next step it is checked out whether the thermodynamic conditions at which the simulations are performed have influence on the accuracy of the results. To this end, MD calculations were performed at different points of the argon phase diagram. The chosen points were: one near to the triple point and the liquid-gas coexistence, other well inside the homogeneous liquid pocket and a third in the fluid region at a supercritical temperature. The simulations were performed using the PES generated with the ASNN trained with 442 energy points. The obtained results are displayed in Table 11.9.

Figures 11.2 and 11.3 show radial distribution functions (rdf) and velocity autocorrelation functions (vcf), respectively, in excellent agreement with those from LJ analytical function.

11.3.3 Improvement of the Accuracy on the NN-Generated PES using Different Memories and NN Retraining

With the ASNN methodology it is possible to incorporate new data in the memory after the training is finished, making it possible to improve predictions with new data without the need to retrain the NNs. This has a potential application for the simulation of PES, in which *ab initio* data is hard to obtain, and can become only gradually available. To test this possibility, a series of experiments were performed with the second part of the curves in which: *a)* an ASNN was initially trained with a small number of points, and

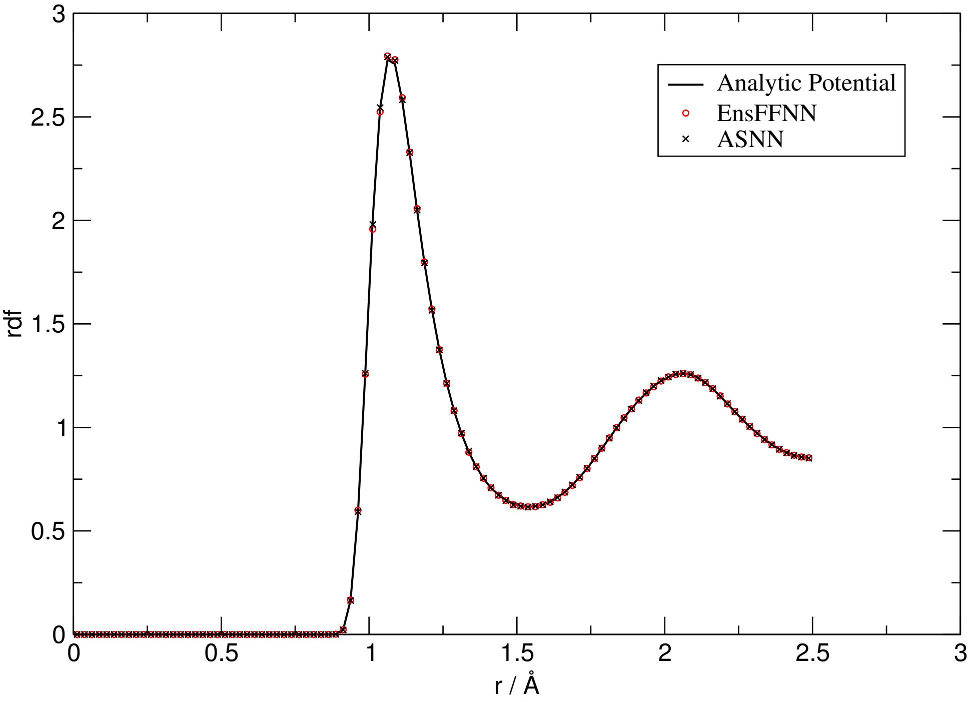


Figure 11.2: Radial distribution functions for argon obtained in MD simulations using the PES generated by NNs (EnsFFNN and ASNN trained with 442 energy points) compared with the rdf from the analytical function.

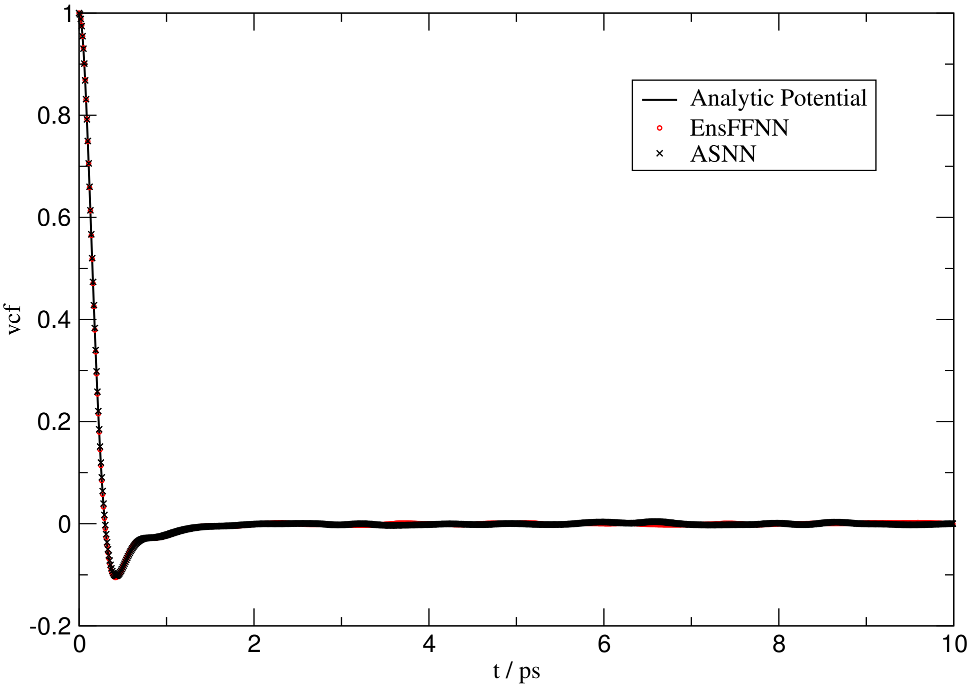


Figure 11.3: Velocity autocorrelation functions for argon obtained in MD simulations using the PES generated by NNs (EnsFFNN and ASNN trained with 442 energy points) compared with the vcf from the analytical function.

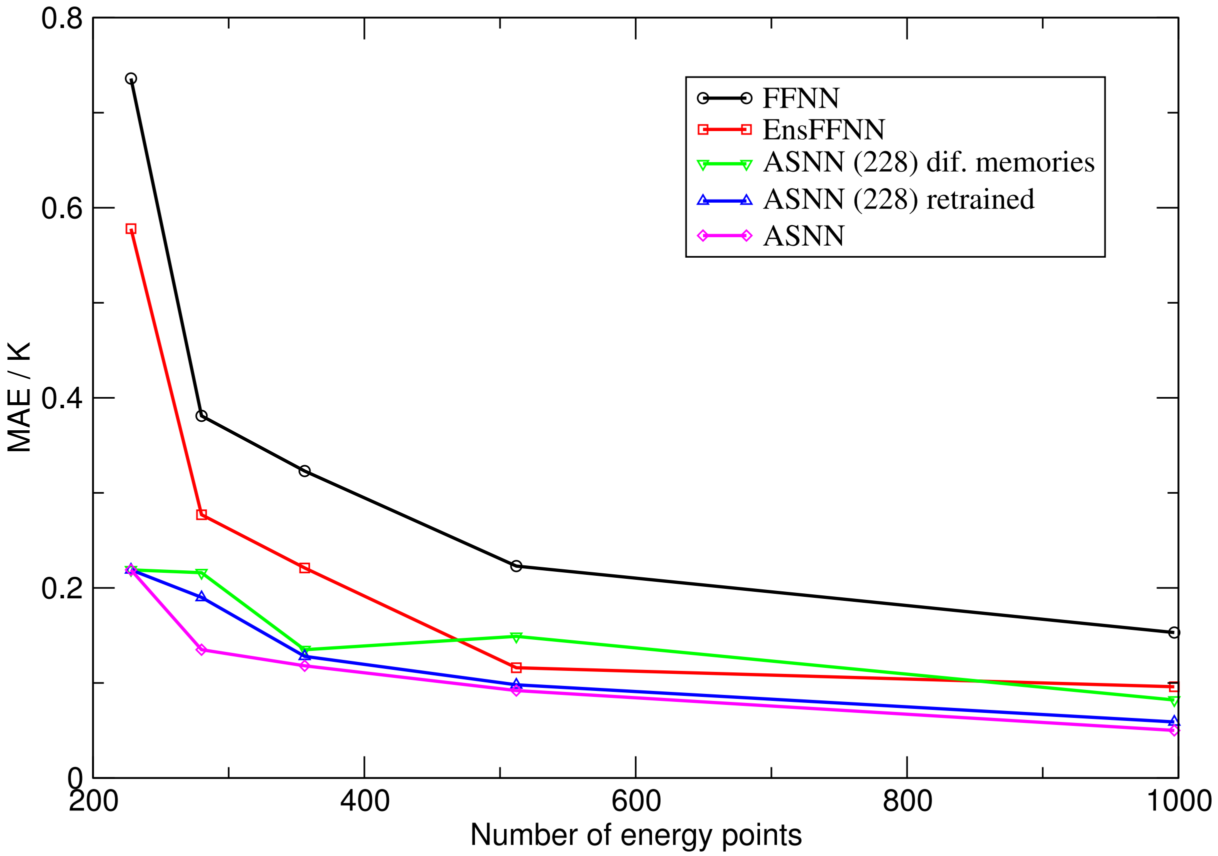


Figure 11.4: Mean Absolute Error *vs* the number of energy points used in the train or in the memory. FFNN, single FFNN trained with different number of energy points; EnsFFNN, EnsFFNN trained with different number of energy points; EnsFFNN (228) dif. memories, EnsFFNN trained with the 228 energy points and using memories with the number of points indicated at the x axis; ASNN (228) retrained, ASNN initially trained with the 228 energy points and then further trained with the other training sets; ASNN, ASNN trained with training sets including different number of points.

then more points were added to the memory without retraining the NNs; *b)* an ASNN was first trained with a small number of points and was later further trained with the larger training sets. The results obtained with these experiments are presented in Figure 11.4, and are compared to those obtained by FFNNs, EnsFFNNs, and ASNNs using different number of points in the training sets.

By analysis of the Figure 11.4, a remarkable difference between single FFNN, EnsFFNN, and ASNN could be seen. The use of EnsFFNN shows considerable advantages over single FFNN. When the comparison is only made between EnsFFNN and ASNN a small difference was observed for training sets with 512 and 997 energy points, but a great difference when the NN are trained with a small number of energy points (356, 280, 228). ASNN exhibits a superior performance to EnsFFNN, particularly when the training set is small. The use of the correlation measure, on the ASNN, made possible to determine neighborhood relations and to use this information to correct predictions, thus explaining the improved predictions. The method not only use global knowledge about the PES, that was stored in the NN weights of the ensemble, but also local knowledge retrieved from the most similar objects in the memory. The experiments made with the ASNN trained with 228 energy points but using as memory the other training sets shows that it is possible to train an ASNN with a small number of energy points and improve the accuracy of the predictions using memories with more data. The results were, in general, better than the ones by EnsFFNN trained with all data and only slightly worse than those obtained by the ASNN trained with all data. Relatively to the retrained ASNN taking as starting point the model trained with 228 energy points, Figure 11.4 shows that the results were similar to the ones obtained by ASNN trained from the beginning with all data in each training set.

11.3.4 Learning PES without the LJ Parameter ε

In the experiments reported so far, the LJ parameters ε, σ and the distance between particles, r, were used as input of the NNs to approach the potential energy, $u(r)$. In real complex situations, however, we do not have PES parameters but the geometry and single point energies calculated at different distances. As such, a different representation that encode the geometry of the systems has to be devised to use as descriptors. In the simple case of LJ potential the only geometric descriptor is σ, which is directly related to the polarizabilities of the atoms by second-order perturbation theory, and turns out to be approximately equal to the molecular diameter. Thus, it is an intrinsic molecular property, not a subproduct as the parameter ε.

An exploratory experiment was performed to check if NNs can learn from only two inputs: σ and the distance between particles, r. A MAE of 0.376 K was obtained with ASNNs, for the test set, which allows MD simulations in agreement with the ones using the analytical function. For complex systems, more descriptors should be employed to encode the geometry, topology, and physico-chemical features of the molecules.

11.3.5 Testing the Ability to Interpolate

Finally two experiments were made with the second part of the curves to verify the ability of the NN models to interpolate. From the training set with 512 energy points, the most similar PES to argon (the PES of O_2) was deleted yielding a training set with 482 energy points. The new training set was then used to train an EnsFFNN and an ASNN, and predictions were obtained for the test set (argon). The obtained MAE was almost unchanged. With the EnsFFNN a MAE of 0.150 K was obtained against 0.116 K with all PES, and with the ASNN a MAE of 0.093 K was obtained against 0.092 K. Next, the two most similar curves to argon were deleted from the training set (O_2 and the F_2 curves) yielding a training set with 453 energy points. In this case the results were slightly affected. With EnsFFNN a MAE of 0.412 K was obtained for the test set, and

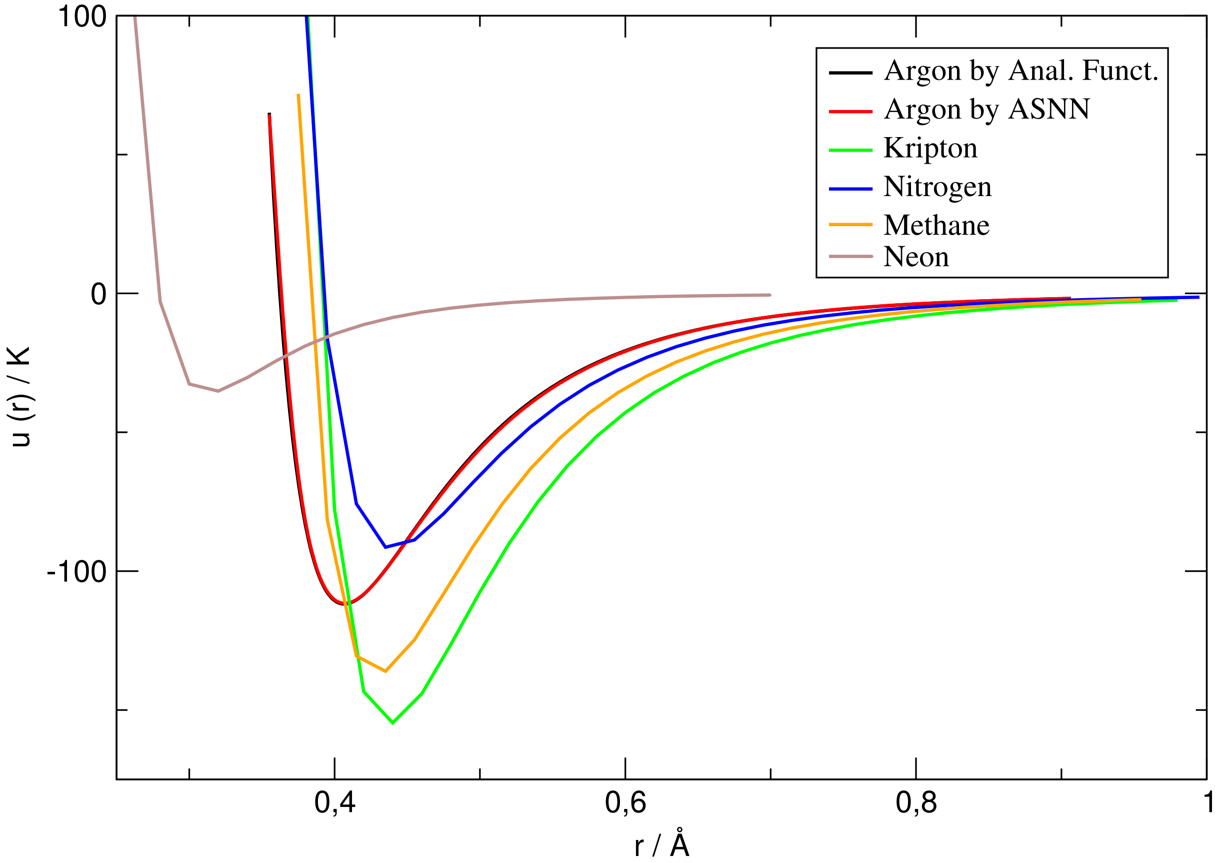


Figure 11.5: Potential energy of argon by analytical function and generated by ASNN trained with 453 energy points (without the two most similar curves of the initial training set) and comparison with the four most similar curves of the training set.

the ASNN yielded a MAE of 0.463 K. Even in this last experiment, it must be noted that good results in MD simulations could be obtained using NN-generated PES with a similar MAE (the experiment with the small number of energy points in Table 11.7 still yielded acceptable results in MD simulations). Figure 11.5 shows the analytical function of argon compared with the one generated by ASNN trained with 453 energy points. It also shows some energy curves for different substances used in the training. The remarkable fit of the argon PES obtained by the NNs clearly demonstrates the ability of the method to interpolate from available distant data.

11.4 Conclusions

The present results suggest that, at least for LJ type potentials, NNs could be trained to generate accurate PES to be used in molecular simulations. The ASNN method gives better results than the single FFNNs and EnsFFNNs and could be a useful method to generate PES of more complex systems capable of taking into account the most subtle features of complex systems in contrast to single FFNNs and other common methods of fitting.

It is possible to train an ASNN with a small data set and later improve the accuracy using different memories as new data become available. Retraining of EnsFFNNs and ASNNs from a previous model, using more data, is also possible, but computationally more expensive, yielding similar results to the model trained in only one step with all the data. This is important for cases when the calculation of PES is made by *ab initio* methods. These present results show that the availability of similar curves in the training set greatly helps to make accurate predictions, as was expected. They also show a remarkable ability of the NN models to interpolate between distant curves yielding good potentials to be used in molecular simulations.

Finally, it is noteworthy that the present study-approximating one-dimensional potential functions by different NNs-does not mean, at all, that neural mappings should replace the direct use of LJ potentials. The purpose of this work was just to analyze, systematically, the accuracy and technicalities of different NNs using the typical test potential (LJ) in molecular simulations. Our main motivation, however, is to approach multidimensional PES to simulate the adsorption and self-assembly of solvated organic molecules on noble-metal electrodes, for which good analytical functions are, in general, inexistent. Work along these lines is in progress and will be reported soon.

Acknowledgements

The assistance of Dr. Yuri Binev with the use of ASNN is gratefully acknowledged. The authors also thank Dr. Igor Tetko who made available the software to implement EnsFFNNs and ASNNs.

Chapter 12

Mapping Potential Energy Surfaces by NNs. The Ethanol/Au(111) Interface

12.1 Introduction

The structure and dynamics of electrode / solution interfaces are of great importance in the domain of electrochemistry. The modification of metallic surfaces properties by the adsorption of molecules allows, for example, photovoltaic, biosensing, and corrosion protection developments.

The adsorption and spontaneous organization of organic molecules on metallic surfaces giving rise to films of organized and stable monolayers is known as Self-Assembled Monolayers (SAMs). SAMs can be produced using different types of molecules and substrates. Typically, alkane chains (with ten or more methylene units) and a thiol (SH) head group are used with Au surfaces due to the well-known chemical affinity between sulfur and gold. These species have the advantage and singular characteristic of creating dense monolayers when the thiol molecules adsorb onto gold with the tail chains pointing outwards the surface. Moreover, it is possible to functionalize the tail chain, after the formation of the SAMs, by the chemical insertion of specific functional groups or molecules.

To investigate the mechanisms involved in the adsorption and self-assembly of solvated organic molecules on metallic electrodes by Monte Carlo or Molecular Dynamics simulations, the determination of the Potential Energy Surfaces of the systems is crucial. They should describe the interactions between the molecular species present in the liquid phase as well the interactions between those species and the electrodes. Our current interest is the study of the adsorption of alkylthiols, solvated by ethanol, on gold electrodes and the understanding of the physi- and chemisorption mechanisms. To this end, our group has recently proposed an analytical force field, based on Density Functional Theory calculations, for the interaction of ethanol with Au (111) surfaces. A preliminary test of the force field has also been carried out by MC simulations [258].

A function that matches DFT data provides, on one hand, a topographical visualization of the surface features, which may not be evident from a coarse-grained quantum mechanical study. On the other hand, it is a suitable input for simulation work. A good representation of PES should smoothly connect the asymptotic as well as the most interactive regions of the configuration space. It should accurately represent the true potential energy in the regions for which experimental or theoretical results are available and predict the interaction energies for the regions where such data is not available. Fitting analytical functions to the energies of a set of suitable configurations of the system is, of course, one of the standard approaches to obtain PES. The London-Eyiring-Polanyi-Sato (LEPS) functions, many-body expansions, splines and semiempirical potentials with adjustable parameters to reproduce experimental and theoretical results, are commonly used [221, 259].

It appears that as the complexity of the systems increases, the development of accurate analytical functions becomes a non-trivial task. In the last years, neural networks (NNs) turned out as an alternative way for mapping PES from *ab initio*/DFT energy data sets [228–237]. In such approximation, there are no *a priori* guesses of analytical functions and the results can come out in tabular form. Moreover, once the networks are well trained, they are able to produce, as output, any required number of energy points for numerical interpolations with similar or better accuracy than other representation methods.

This Chapter presents the mapping of the Potential Energy Surfaces (PES) for the ethanol/Au(111) interface by Neural Networks (NNs). The main objective is to assess an alternative to our analytical force field, in order to map multidimensional PES for the interaction of ethanol and Au (111) surface regarding the simulation of the adsorption and self-assembly of alkylthiols solvated by ethanol.

Interaction energies, calculated from Density Functional Theory (DFT), for the adsorption of the ethanol on Au(111) surfaces are used to train Ensembles of Feed-Forward Neural Networks (EnsFFNNs) [63, 64]. They show a greater generalization ability over single FFNNs in problems of modeling and fitting. EnsFFNNs, a supervised learning technique, are an extension of single FFNNs and a memory-less method (after the training, all information about the input patterns are stored in the NN weights without the explicit storage of the data in the system).

The distance of the ethanol molecule to the surface, two angles describing the molecular orientation relatively to the surface, and three binary descriptors encoding the gold adsorption sites, are the input to the NNs. The training sets contain energy values at different distances, for seven molecular orientations and three adsorption sites. The models are assessed by: *a)* internal cross validation; *b)* Leave-One-Out procedure (LOO); and *c)* external test sets corresponding to orientations not used in the training procedure.

The results are compared with the ones obtained from an analytical force field recently proposed [258] to match the DFT data. It is shown that NNs can be trained to map PES with a similar or better accuracy than analytical representations. This is a relevant point, particularly in simulations by Monte Carlo (MC) or Molecular Dynamics (MD), which require an extensive screening of the interaction sites at the interface, turning the development of analytical functions a non-trivial task as the complexity of the systems increases.

The next Section 12.2 contains the methodology and computational details. Section 12.3 discusses the NNs results and their comparison with the ones from the analytical force field. Finally Section 12.4 presents the main conclusions of the work.

12.2 Methodology and Computational Details

The training has been performed with the DFT data of Fartaria et al. [258] used to set up the analytical force field. Additional DFT energy points have been calculated, at six orientations, to test the accuracy of the model in regions of the PES not considered by the training set.

EnsFFNNs is the machine learning method used to set up the relationship between the input (two orientation angles, the distance between the ethanol oxygen atom and the plane of the first layer of the Au (111) surface, and three binary descriptors to encode the adsorption sites) and the output (potential energy). The models are tested with different internal data sets (internal cross validation and LOO procedure) and external data sets.

The results are compared with the values from DFT calculations and the ones from the analytical function developed. The full NNs-PES for each site in 3D representation is presented and analysed. A model is also trained with a different training and test sets, and with all the available data being compared with the one obtained from the initial training set. The interpolation ability of the NN is analysed.

12.2.1 DFT Calculations and Analytical Force Field

The theory level chosen to calculate the interaction energy of ethanol - Au (111) was the hybrid B3LYP method [246, 248] with the LanL1MB basis set [260] applied to the gold atoms and the 6-31G basis set [261] for the H, C and O atoms. The calculations have been performed by the Gaussian 98 package [247].

A cluster of 14 Au atoms, to represent the gold surface, and one ethanol molecule, in the optimized gas-phase geometry, have been used to model the ethanol - Au (111) PES. The size of the gold cluster was chosen with a compromise between the consistency of the interaction energy and the computation time. Figure 12.1 shows the three adsorption sites, top (*Top*), hollow 1 (*H1*) and hollow 2 (*H2*), that have been selected to study the

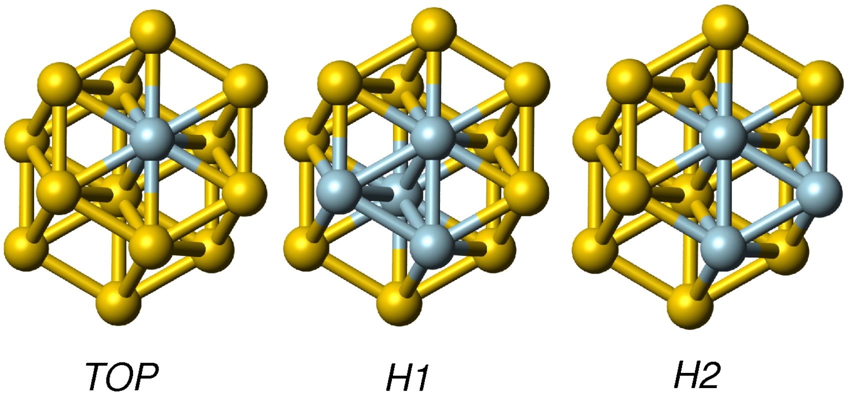


Figure 12.1: Surface sites, top, hollow1 and hollow2, chosen to set up the ethanol-Au surface interaction.

ethanol-surface interaction.

In the *Top* site, the oxygen atom of ethanol approaches the surface directly over a gold atom of the first layer; the *H1* site corresponds to a hcp (hexagonal closed packed) site and the approach of the ethanol is made in the direction of the centre of a triangle formed between three gold atoms of the first layer with a gold atom of the second layer at the centre; and the *H2* site corresponds to a fcc (face centred cubic) site and the ethanol approach is made in the direction of the centre of a triangle formed between three gold atoms of the first layer.

The ethanol - Au $(111)_{14}$ cluster interaction energy, as a function of the distance and orientations of ethanol molecule to the Au (111) surface, is calculated by:

$$\begin{aligned} U_{ethanol-Au(111)_{14}}(r,\alpha,\beta) &= U_{Au(111)+ethanol}(r,\alpha,\beta) \\ &\quad -U_{Au(111)_{14}} - U_{ethanol} \end{aligned} \tag{12.1}$$

where $U_{Au(111)+ethanol}$ is the energy of the system composed by the ethanol molecule and the cluster; $U_{Au(111)}$ and $U_{ethanol}$ are the energies of the isolated cluster and ethanol molecule; r is the distance from the ethanol oxygen atom to the plane of the first layer of the Au(111) surface; α is the angle between the O-H bond and the normal to the surface and β is the angle between the plane H-O-C and the plane H-O-*normal to the surface* (both angles in degrees). The orientations for the ethanol molecule have been selected to span a wide range. Figure 12.2 shows the orientations used in the training, and Figure 12.3 displays the molecular orientations used to test the models.

For each orientation, several values of r, along the interval 0-10Å, have been chosen and the $U_{ethanol-Au(111)_{14}}$ evaluated. Full details of the DFT calculations are described elsewhere [258].

The following analytical function was fitted to the above mentioned DFT results:

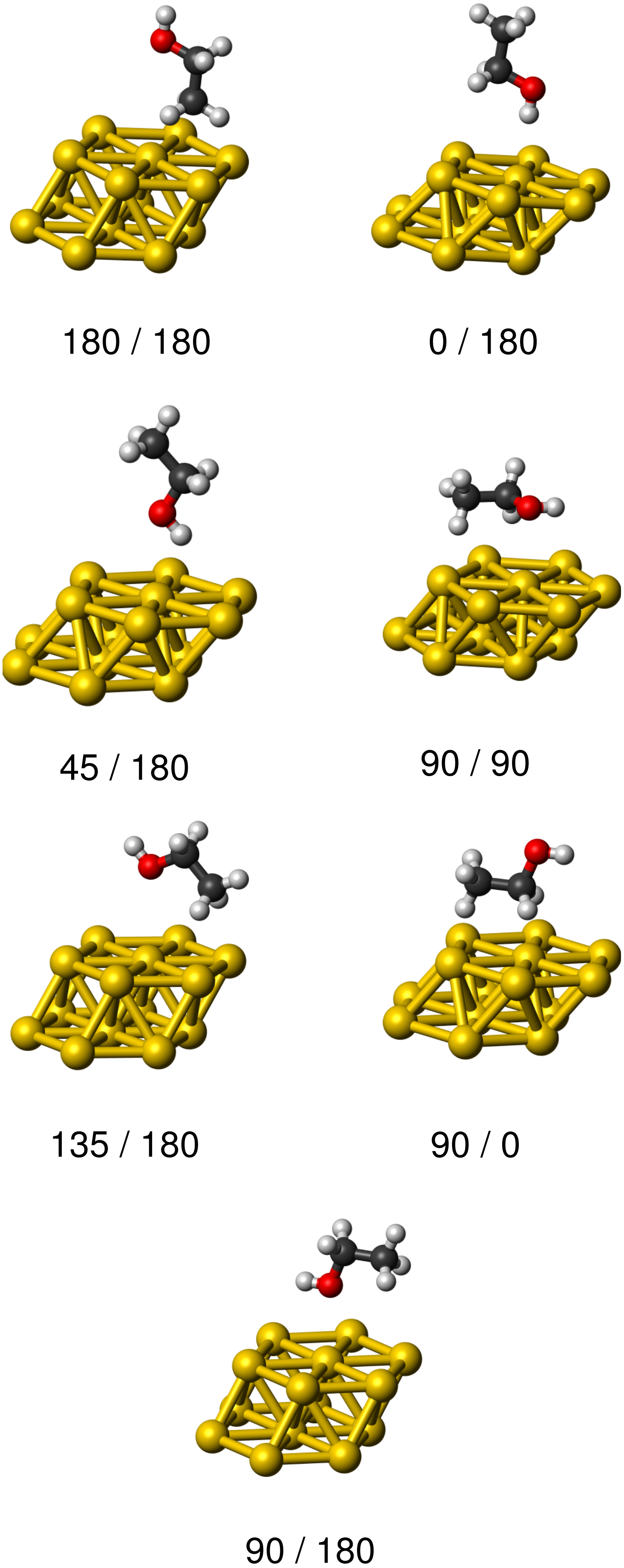
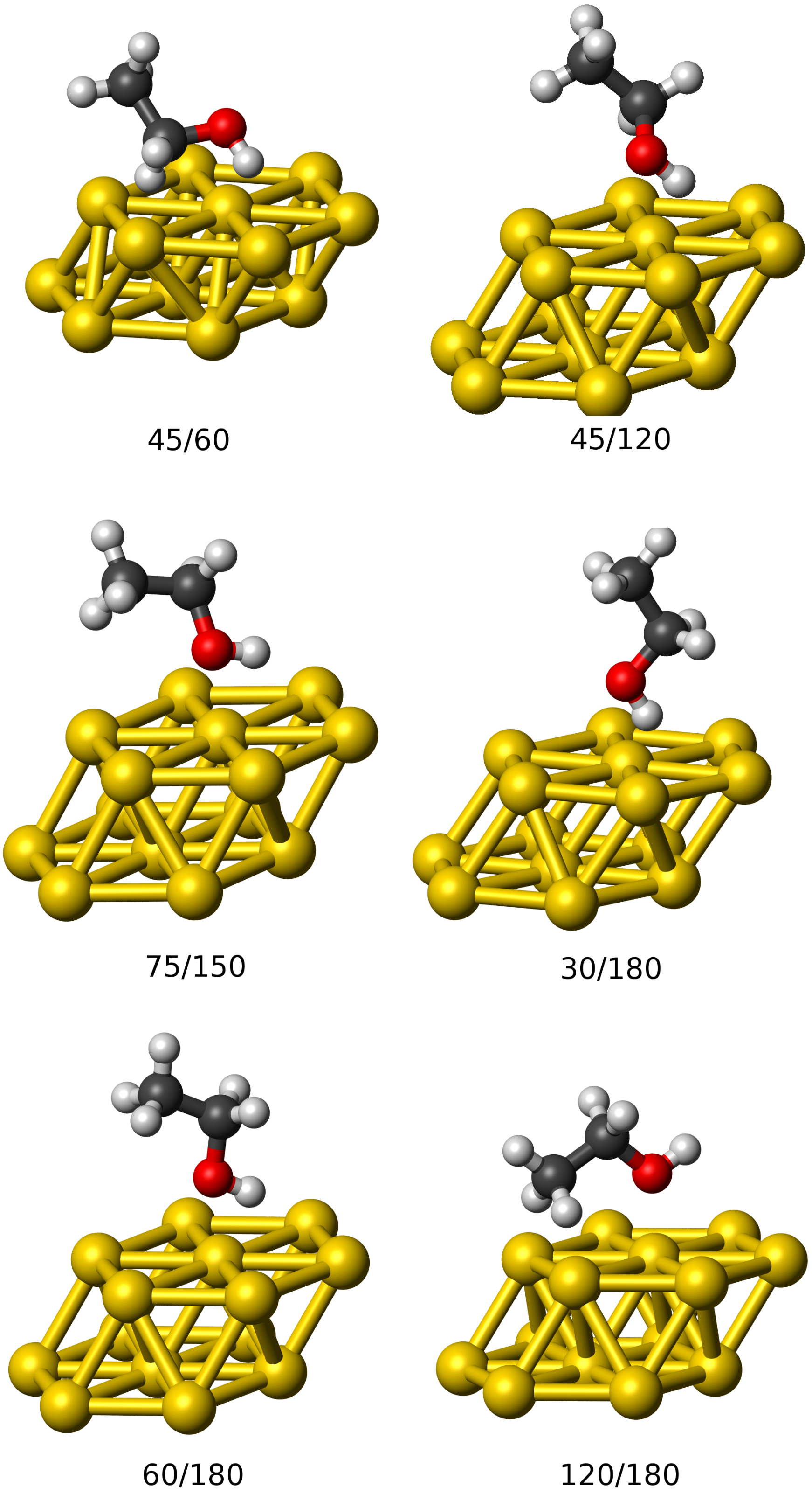


Figure 12.2: Selected orientations of the ethanol molecule relative to the Au(111) surface used in the training set. The angles are in degrees.



45/60

45/120

75/150

30/180

60/180

120/180

Figure 12.3: Selected orientations of the ethanol molecule relative to the Au(111) surface used in the test set. The angles are in degrees.

$$\begin{aligned} U_{EtOH-Au} &= U_H\left(r_{H-Au}\right) + \left(1 + B_0 cos\left(\theta/rad\right)^{20}/r^3_{o-Au}\right) U_O\left(r_{O-Au}\right) \\ &+ U_{CH_2}\left(r_{CH_2-Au}\right) + U_{CH_3}\left(r_{CH_3-Au}\right) + V\left(r_{O-Au},\, \theta,\, \phi\right) \end{aligned} \tag{12.2}$$

where

$$\begin{aligned} U_i\left(r_{i-Au}\right) &= A_{0,\,i} exp\left[A_{1,\,i}\left(r_{i-Au} + A_{2,\,i}\right)\right] \\ &-A_{3,\,i} exp\left[A_{4,\,i}\left(r_{i-Au} + A_{5,\,i}\right)\right] \end{aligned} \tag{12.3}$$

is the site-site interaction energy, and

$$\begin{aligned} V\left(r_{O-Au},\, \theta,\, \phi\right) &= C_0 sin\left(\theta/rad\right)^6 \\ &\times exp\left[\left(r sin\left(\theta/rad\right) - C_1\right)^2/C_2\right] \\ &\times\left(C_3 - C_4 cos\left(3\left(\phi/rad - C_5\right)\right) + \right. \\ &\left. + C_6 cos\left(6\left(\phi/rad - C_5\right)\right)\right) \end{aligned} \tag{12.4}$$

is the contribution due to the surface symmetry around a *Top* site. It is noteworthy that the cosine term with exponent 20, in Equation 12.2, can be expressed by other forms (e.g. $\exp\left(-10x^2\right)$).

The function expresses the sum of the interactions between each gold atom and the sites of the ethanol molecule further modulated by two angular contributions, one related to the *Top* site surface symmetry and the other to the O-Au relative direction. The ethanol molecule was described by a united atom model with the H, O, CH_2 and CH_3 as the interaction sites. The variables used are: the distances, r_i, from each Au atom to each of the ethanol sites; the angle, θ, between the r_{O-Au} vector and the normal to the surface; and the angle, ϕ, between the projection of the r_{O-Au} vector on the surface plane and a reference surface vector beginning in a *Top* site and directed to a *H1* site. The fit to the DFT results was performed by means of genetic algorithms, using an Au(111) double layered electrode with 74 Au atoms in order to minimize border effects from the surface.

The overall fitting quality is good. We shall return to it in Section 12.3. The function parameters and the details of the fitting procedure can be seen elsewhere [258].

12.2.2 Feed-Forward Neural Networks

FFNNs [1] were implemented with six input neurons, one hidden layer of neurons, and one output neuron. Full details about FFNNs can be found in Section 2.8.

The networks were trained using the ASNN program of Igor Tetko [65, 66] taking as input the distance between the ethanol oxygen atom and the plane of the first layer of the Au (111) surface (r), two angles to describe the orientation of the ethanol to the surface (α and β) and three binary descriptors to encode the gold adsorption sites (*Top*, *H1* and *H2*). Corrections of the weights during the training procedure were performed by the Levenberg-Marquardt algorithm [59, 256] and the number of neurons in the hidden layer was optimized. Before the training, the whole training set was randomly partitioned into a learning and validation set, each one with 50% of the objects. Full cross-validation of the entire training set was performed by the leave-one-out method (LOO). The logistic activation function was used (a sigmoidal) and each input and output variable was linearly normalized between 0.1 and 0.9 on the basis of the training set. The maximum number of iterations in the training was 5,000 or 10,000. The training stopped when no further improvement in the root mean squared error (RMSE) for the validation set [56] was observed. After the training, the results were calculated for the learning set, validation set, LOO method and test sets.

12.2.3 Ensembles of Feed-Forward Neural Networks

An EnsFFNN consists of several independently trained FFNNs, each one contributing with a single prediction [63, 64]. The final prediction for an object, in our case the potential energy, is the average of the outputs from all FFNNs of the ensemble. This methodology smoothes out the random fluctuations in the individual FFNNs predictions. The experiments were carried out with the ASNN program [257].

12.3 Results and Discussion

The impact of the number of hidden neurons, and the size of the ensembles, on the accuracy of the models is analysed in terms of the root mean square error:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} \left[(Y_{calc} - Y_{exp})^2 \right]}{n}} \tag{12.5}$$

where Y_{calc} is the predicted value, Y_{exp} is the target value and n is the number of objects.

The RMSE is calculated for the training set and for the different internal and external test sets. After the optimization of these parameters, a discussion of the NN-PES is presented. The minima of the DFT-calculated potential energy for each molecular orientation and adsorption site are compared with the values from the NNs and from the analytical function at the same distances. The three preferential orientation curves for each site are analysed for the test set. The obtained models are used to generate the 3D representation of the PES for each site. The interpolation ability of different EnsFFNNs is also investigated with EnsFFNNs trained with a different partition of the data and with all data available. The comparison between the DFT data, the predictions of the different EnsFFNNs, and the analytical function are analysed in terms of the RMSE, the correlation coefficient of the predictions relatively to the DFT data,and the mean absolute error (MAE):

$$MAE = \frac{\sum_{i=1}^{n} |Y_{calc} - Y_{exp}|}{n} \tag{12.6}$$

where Y_{calc} is the predicted value, Y_{exp} is the target value and n is the number of objects.

12.3.1 Impact of the Number of Hidden Neurons and Networks in the Ensemble

Ensembles of 15 FFNNs were evaluated for the training, internal validation and external test sets, using different numbers of neurons in the hidden layer: 2-10, 15, 20 and 25. Then EnsFFNNs were trained with different number of networks (1, 5, 10, 15, 20, 25, 50, 75, 100, 150, 200).

Figure 12.4 shows a decrease in the RMSE for the test set from ~8 kJ mol^{-1} for networks with 2 neurons, to a RMSE of ~5 kJ mol^{-1} for networks with more than 5 hidden neurons. The results for the training and other internal validation sets correlate with those for the external test set. Reduction of the RMSE values is more pronounced up to 5 hidden neurons. Training with more than 10 hidden neurons does not indicate improvements and increases computational requirements. Thus a compromise of 8 hidden neurons has been chosen for the experiments. As for the impact of the ensemble size, the use of ensembles with more than 10 networks has not shown a significant improvement in the predictions.

12.3.2 Mapping of PES by EnsFFNNs

The minima of the potential energy for each orientation of the ethanol molecule and adsorption sites, from the DFT calculations, and their comparison with the values from the EnsFFNNs and the analytical function at the same distances are presented in Table 12.1 for the points of the training set.

The EnsFFNNs training results are, in general, in good agreement with those from DFT. Only two cases present absolute errors higher than 1kJ mol^{-1}: 1.8 kJ mol^{-1} for the

Table 12.1: Energy minima of the training set for each orientation and adsorption site. In parenthesis is the absolute error.

Ethanol orientation		Distance	Potential Energy -U/kJ mol^{-1}		
α/degrees	β/degrees	O-surface r/Å	DFT	Anal. Func.	EnsFFNNs
Top site					
0	180	3.54	7.76	8.7 (0.94)	7.96 (0.20)
135	180	4.07	5.87	6.7 (0.83)	6.15 (0.28)
180	180	6.0	1.37	1.2 (0.17)	1.78 (0.41)
45	180	3.3	9.04	15.0 (5.96)	9.13 (0.09)
90	0	5.5	2.32	1.5 (0.82)	2.38 (0.06)
90	180	2.67	19.03	19.25 (0.22)	18.17 (0.86)
90	90	3.54	11.77	10.4 (1.37)	11.95 (0.18)
H1 site					
0	180	3.3	13.14	10.7 (2.44)	12.93 (0.21)
135	180	4.29	4.34	5.5 (1.16)	3.96 (0.34)
180	180	6.04	1.35	0.9 (0.45)	1.51 (0.16)
45	180	2.87	16.42	13.3 (3.12)	14.61 (1.81)
90	0	5.57	2.35	1.4 (0.95)	2.46 (0.11)
90	180	2.87	12.8	15.5 (2.7)	13.91 (1.11)
90	90	3.87	5.99	7.1 (1.11)	6.03 (0.04)
H2 site					
0	180	3.3	10.46	9.8 (0.66)	10.32 (0.14)
135	180	4.2	5.08	5.3 (0.22)	4.96 (0.12)
180	180	6.04	1.48	1.0 (0.48)	1.42 (0.06)
45	180	3.08	11.82	12.7 (0.88)	11.39 (0.43)
90	0	5.37	2.33	1.4 (0.93)	2.37 (0.04)
90	180	2.67	15.07	14.7 (0.37)	14.86 (0.21)
90	90	3.87	5.75	7.1 (1.35)	6.03 (0.28)

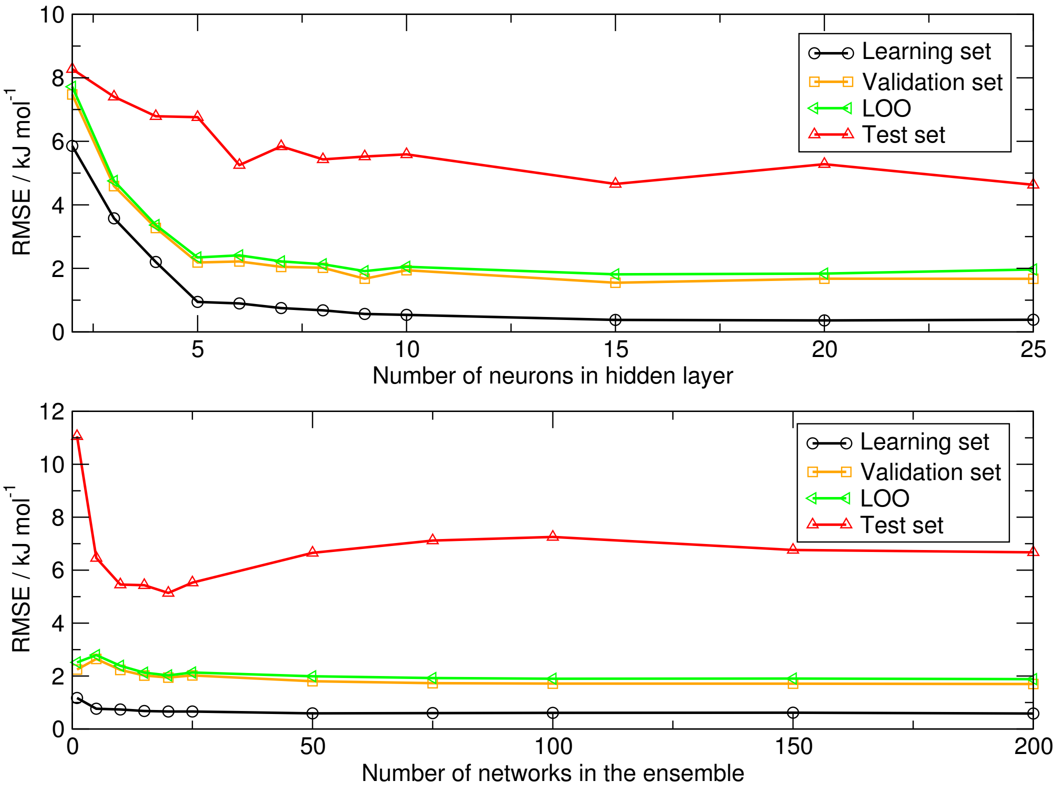


Figure 12.4: Root Mean Square Error of the different data sets for networks with different number of neurons in the hidden layer (first graphic) and ensembles of different sizes (second graphic).

45/180 orientation and 1.1 kJ mol^{-1} for the 90/180 orientation both on *H1* site. [56]

As for the comparison with the values obtained from the analytical function the EnsFFNNs provide, in general, more accurate predictions for the energy minima. Moreover, it should be emphasized that while the analytical function shows an evident discrepancy, relatively to the order of the DFT binding energies, for the referred to 45/180 and 90/180 orientations on the *H1* site, the neural networks predict the right DFT energy order.

Figure 12.5 displays the whole NNs PES, from the training set, for the three most attractive orientations on each site and their comparison with the DFT data, as a complement of Table 12.1 (that only presents the values of the potential energy minima).

Very accurate predictions are obtained for the all orientations in the three sites. The NNs predicted curves are in good agreement with the DFT data both in the repulsive and attractive part of the curves.

Table 12.2 presents the results for the energy minima from the test set, whose elements, as already said, have not participated in the training. The results show, in general, the same level of accuracy as those from the training set, except for the 45/60 and 45/120 orientations that exhibit considerable errors at all the chosen sites. The value for the 45/60 orientation at the *Top* site is even a non-physical one, since it predicts a positive

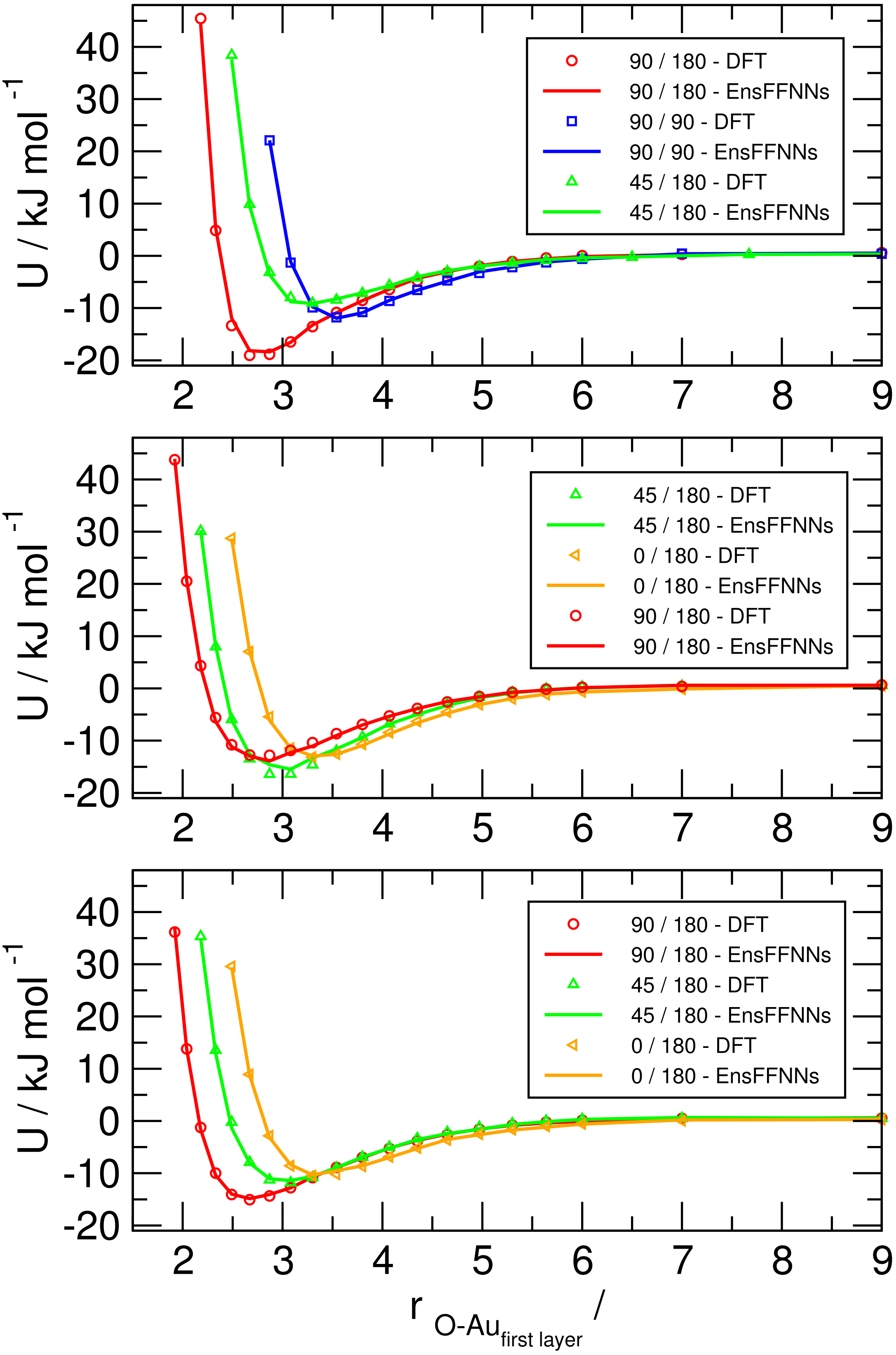


Figure 12.5: Ethanol-Au $(111)_{14}$ potential energy curves by DFT and EnsFFNNS for the *Top*, *H1* and *H2* sites (from top to bottom, respectively) from the training set.

Table 12.2: Energy minima of the test set for each orientation and adsorption site. In parenthesis is the absolute error.

Ethanol orientation		Distance	Potential Energy -U/kJ mol^{-1}		
α/degrees	β/degrees	O-surface r/Å	DFT	Anal. Func.	EnsFFNNs
Top site					
45	60	3.79	7.0	8.78 (1.78)	-2.70 (9.70)
45	120	3.37	9.81	14.14 (4.33)	5.65 (4.16)
75	150	2.81	15.91	18.85 (2.94)	17.09 (1.18)
30	180	3.37	7.75	13.36 (5.61)	8.10 (0.35)
60	180	3.17	11.52	16.63 (5.11)	11.93 (0.41)
120	180	3.17	11.84	12.13 (0.29)	11.49 (0.35)
H1 site					
45	60	3.79	6.69	7.20 (0.51)	3.07 (3.62)
45	120	3.17	12.5	13.03 (0.53)	7.48 (5.02)
75	150	2.81	12.58	15.46 (2.88)	12.82 (0.32)
30	180	3.17	15.59	12.76 (2.83)	14.86 (0.73)
60	180	2.81	16.71	14.34 (2.37)	14.68 (2.03)
120	180	3.58	6.64	10.43 (3.79)	6.43 (0.21)
H2 site					
45	60	4.02	4.44	6.55 (2.11)	2.11 (2.33)
45	120	2.99	10.61	12.09 (1.48)	3.96 (6.65)
75	150	2.81	13.78	14.50 (0.72)	13.24 (0.54)
30	180	3.17	10.73	12.19 (1.46)	10.54 (0.19)
60	180	2.81	13.09	13.37 (0.28)	12.59 (0.50)
120	180	3.37	8.55	10.61 (2.06)	7.88 (0.67)

binding energy.

The bad results for those orientations are presumably due to the fact that the β angle range is not well covered in the training set. From the seven orientations used in it only two have a β angle different of 180: the 90/0 and the 90/90 orientations. To set up models allowing more accurate predictions for all the PES regions it is essential that the training set covers a wider range of possible molecular orientations. This issue will be reanalysed in the following Subsection through the generation of a PES by training with the same set as before but with different initial weights, using a different training and test set and training with all available data without an external test set.

The comparison with the results from the analytical function shows that, except for the 45/60 and 45/120 orientations, the EnsFFNNs provide, in general, more accurate predictions for the energy minima: a maximum absolute error of ∼2 kJ mol^{-1} from the

NNs against ∼6 kJ mol^{-1} from the function.

Figure 12.6 displays the whole NNs PES, from the test set, for the three most attractive orientations on each site and their comparison with the DFT data, as a complement of Table 12.2 (that only presents the values of the potential energy minima).

In all presented orientations the most considerable deviations from DFT data are always in the repulsive part of the curves. Despite this deviations this part of the curves is less important if we take into consideration that, at 298 K, k$_B$T is only 2.4 kJ mol^{-1}, suggesting that the probability of the high repulsive parts of the PES becoming sampled, during a simulation, is very low.

The 120/180 orientation on the *Top* site presents an average deviation from the DFT data, of ∼6 kJ mol^{-1}, at the repulsive part of the curve. For the other two orientations the predictions are in good accordance with DFT data over the entire curves. On the *H1* site, very accurate predictions are obtained for the 30/180 and 75/150 orientations, and for the 60/180 orientation a deviation, of ∼2 kJ mol^{-1}, is observed in the attractive part of the curve. For *H2* site the NNs predicted curves are in good agreement with the DFT data.

After the training and testing, 83509 potential energy points have been predicted by the EnsFFNNs for distances of the ethanol to the surface between 2 and 8 Å, with an interval of 0.1 Å, and for α and β angles between 0 and 180°, with an interval of 5°. Each site is treated separately. Figure 12.7 displays an example of PES for the *H1* site in a 3D representation.

The cutting plane at α= 45° includes energy minima for the *H1* site, for example at β=180° and r=2.9 Å (see Table 12.1). The isoenergetic lines show the energy dependence on the distance and the β angle. The surfaces corresponding to the lines 0, 5 and 10 kJ mol^{-1} are not represented just to avoid a heavy picture.

Figure 12.8 illustrates PES projections for minima energy regions. The first, second and third columns correspond to the *Top*, *H1* and *H2* sites, respectively. The first, second and third row correspond to fixing the distances, the α and the β angles, respectively.

The examples of the 3D plots and their projections show the smooth and well-behaved PES predicted by the EnsFFNNs, allowing straightforward energy interpolations. This is a relevant aspect regarding their possible use in Monte Carlo or molecular dynamics simulations. As we shall see in the next Subsection, such representations also give a good visualization of the improvements introduced in the PES by changing the training and test sets.

12.3.3 Different training and test sets.

The failure or less accurate predictions of the NNs-PES in a few surface regions, mentioned above, are presumably not due to a limitation of the networks learning but to the

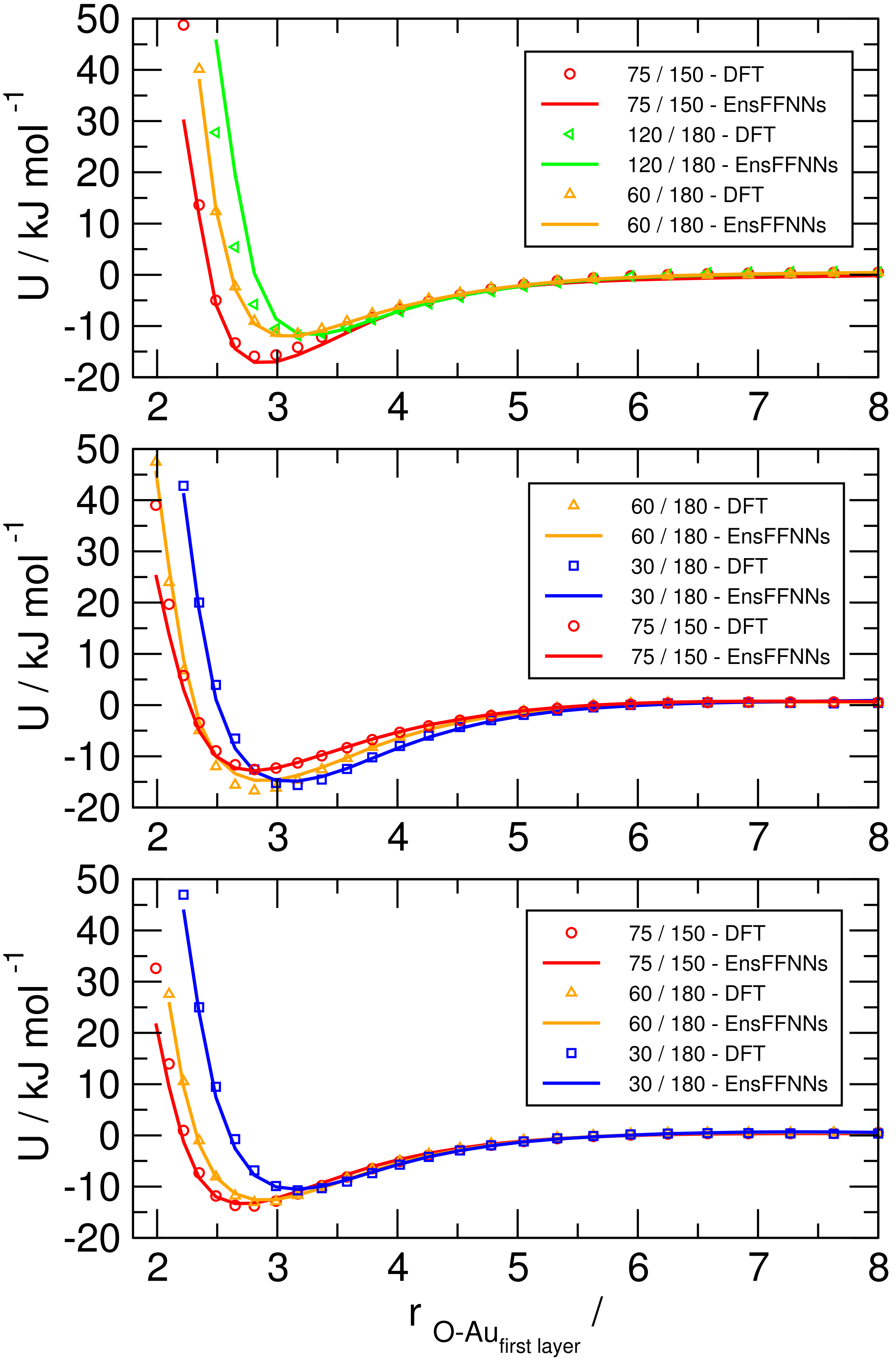


Figure 12.6: Ethanol-Au $(111)_{14}$ potential energy curves by DFT and EnsFFNNS for the *Top*, *H1* and *H2* sites (from top to bottom, respectively) from the test set.

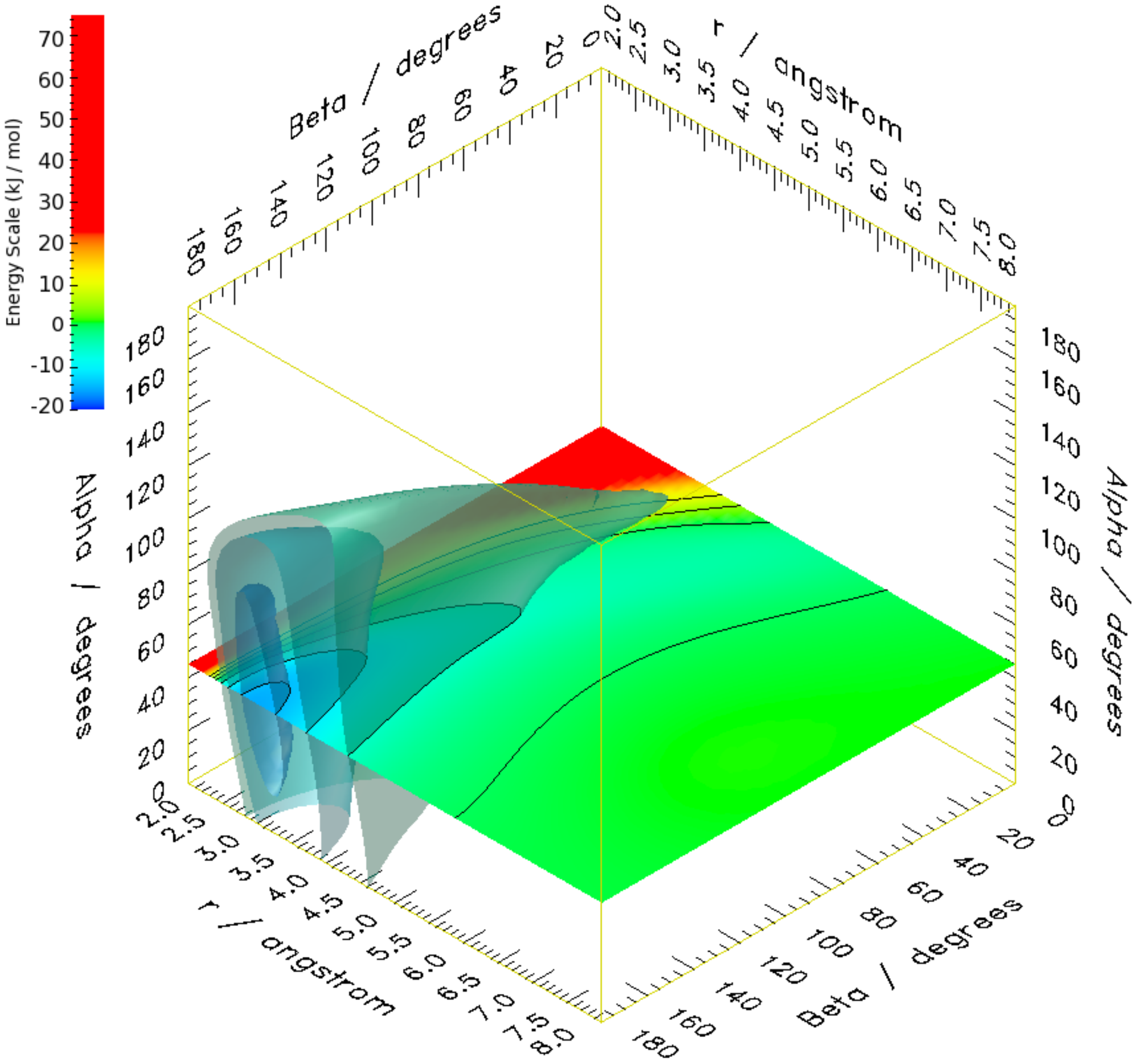


Figure 12.7: PES of ethanol over the *H1* site. Three isoenergetic surfaces are displayed at -14, -10 and -5 kJ mol^{-1}. The plane cuts the surfaces at α= 45°. The isoenergetic lines correspond to the energies -14, -10, -5, 0, 5 and 10 kJ mol^{-1}. The colored bar represents the energy scale.

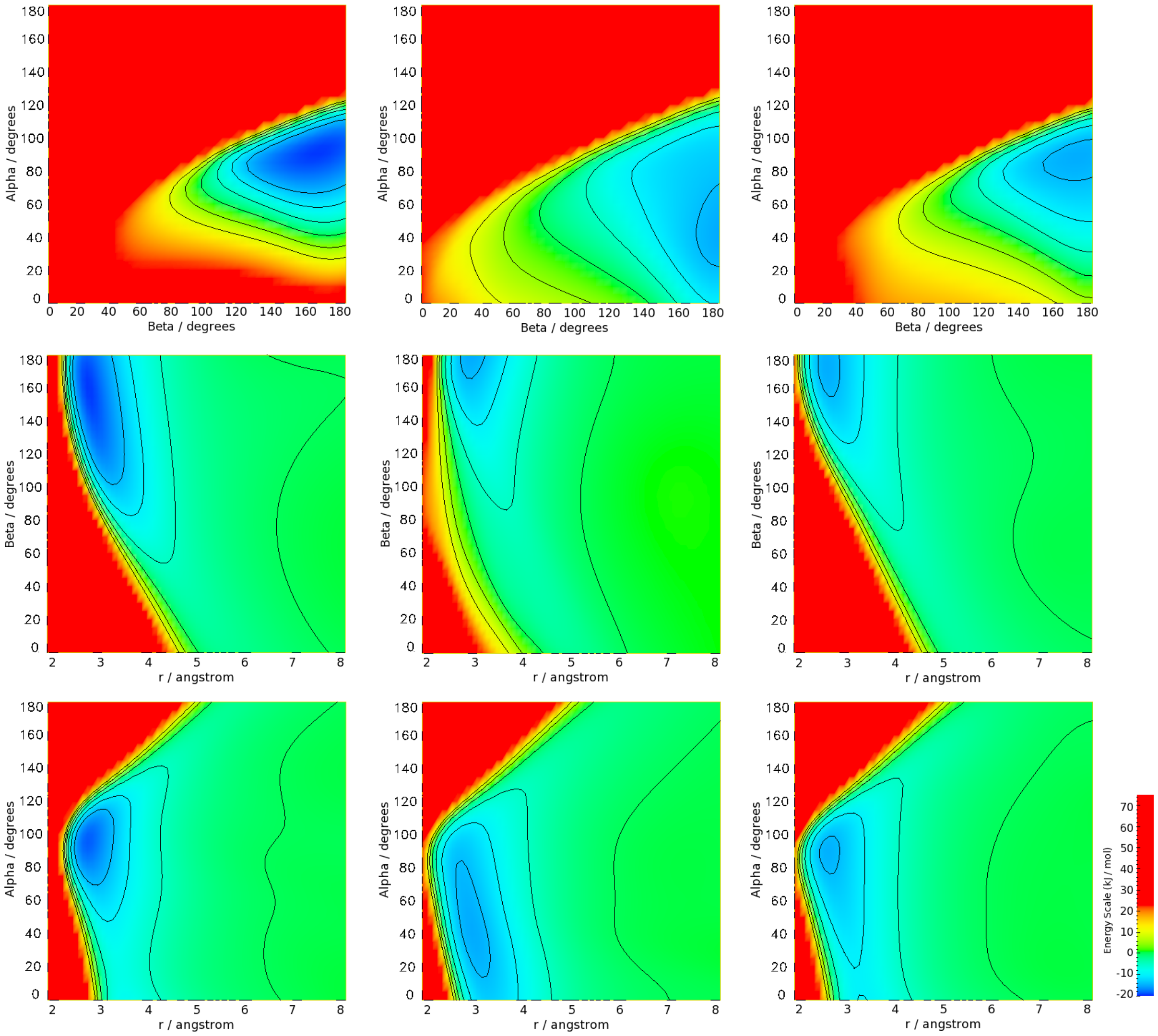


Figure 12.8: PES projections for the *Top* (first column), *H1* (second column) and *H2* (third column) sites. First row, from the left to right, projections at r = 2.8, 3.0, 2.7 Å; second row, from left to right, projections at α= 90°, 45°, 90°; third row, β=180° for all sites. The isoenergetic lines correspond to potential energy of -14, -10, -5, 0, 5 and 10 kJ mol^{-1}. The colored bar represents the energy scale.

available information: a limited data set of potential energy points. In fact, the NNs ability of mapping multidimensional data has already been shown, for the most part of the orientations, by the accurate predictions obtained from an external data set using a limited number of orientations in the training set.

In order to evaluate the impact of the available data in the NNs learning, and consequently in the accuracy of the predicted PES, further experiments have been carried out: *i)* using the same training and test sets as before but with different (random) training parameters; *ii)* training with a new partition of training and test sets; and *iii)* training with all available data for (13 orientations for each site corresponding to 780 energy points) without using an external test set.

The energy minima at each orientation are presented in Table 12.3 together with the results of the last Section.

From Table 12.3 the following conclusions can be drawn. A different ensemble of FFNNs (EnsFFNNs^2 trained with the same data set but different random training parameters) may yield better predictions for some orientations, namely 45/60 and 45/120, but the performance degrades for other tests (the overall RMSE for the validation and test sets) (see Table 12.4). However, if it is only considered the predictions for the energy minima of each orientation the EnsFFNNs^2 results are more accurate: a RMSE of 1.5 kJ mol^{-1} against 3.36 kJ mol^{-1} for the test set. Also, the correlation coefficients between the DFT results for the energy minima and the EnsFFNNs^2 are improved from 0.85 to 0.91 (Table 12.4).

A similar situation was observed by using a different partition of the data set into training and test sets. The RMSE are improved from 3.36 kJ mol^{-1} to 1.83 kJ mol^{-1} for the test set (Table 12.4). The new orientations used in the training help the NNs to learn much better the interactions at 45/60 and 45/120 orientations. Yet, the binding energies at orientations 180/180 in the sites *H1* and *H2* and 45/60 in *H1* site are less accurately predicted. These results indicate that the training sets did not cover well enough the configuration space, leading to some instability in the results, as well as large errors for some orientations. Even though, excellent predictions were achieved for most of the orientations in the independent test set.

Figure 12.9 shows isoenergetic surfaces for the three sites trained with the initial set and with all available data.

The isoenergetic surfaces allow the visualization of the improvement for the regions not so well represented with the first training set. When the training uses all data available, the surfaces cover, in general, wider regions in the configurational space. As already referred to before, it is very important to cover as much as possible of the configurational space in the training in order to obtain accurate predictions.

In the course of the present experiments, Associative Neural Networks have also been probed. An ASNN is a combination of a memory-less (after the training all information

Table 12.3: Energy minima of the training and test sets for each orientation and adsorption site obtained from DFT calculations, analytical function and different EnsFFNNs. In parenthesis is the absolute error.

Orientation		Distance	Potential Energy $-U/kJmol^{-1}$					
α/deg.	β/deg.	O-surf. r/Å	DFT	Anal. Funct.	EnsFFNNs^1	EnsFFNNs^2	EnsFFNNs^3	EnsFFNNs^4
Top site								
0	180	3.54	7.76	8.7 (0.94)	7.96 (0.20)	8.68 (0.92)	7.92 (0.16)	8.21 (0.45)
135	180	4.07	5.87	6.7 (0.83)	6.15 (0.28)	5.84 (0.03)	5.78 (0.09)	6.02 (0.15)
180	180	6.0	1.37	1.2 (0.17)	1.78 (0.41)	1.47 (0.10)	2.14 (0.77)*	1.57 (0.20)
45	180	3.3	9.04	15.0 (5.96)	9.13 (0.09)	9.68 (0.64)	9.20 (0.16)	9.34 (0.30)
90	0	5.5	2.32	1.5 (0.82)	2.38 (0.06)	1.82 (0.50)	2.31 (0.01)	2.23 (0.09)
90	180	2.67	19.03	19.25 (0.22)	18.17 (0.86)	16.82 (2.21)	17.85 (1.18)	17.93 (1.10)
90	90	3.54	11.77	10.4 (1.37)	11.95 (0.18)	11.53 (0.24)	11.81 (0.04)	11.25 (0.52)
45	60	3.79	7.0	8.78 (1.78)*	-2.70 (9.70)*	5.15 (1.85)*	5.94 (1.06)*	6.83 (0.17)
45	120	3.37	9.81	14.14 (4.33)*	5.65 (4.16)*	12.29 (2.48)*	9.68 (0.13)	9.88 (0.07)
75	150	2.81	15.91	18.85 (2.94)*	17.09 (1.18)*	16.89 (0.98)*	15.77 (0.14)	15.75 (0.16)
30	180	3.37	7.75	13.36 (5.61)*	8.10 (0.35)*	8.30 (0.55)*	7.82 (0.07)*	7.97 (0.22)
60	180	3.17	11.52	16.63 (5.11)*	11.93 (0.41)*	12.41 (0.89)*	11.44 (0.08)	11.49 (0.03)
120	180	3.17	11.84	12.13 (0.29)*	11.49 (0.35)*	9.72 (2.12)*	12.28 (0.44)*	11.61 (0.23)
H1 site								
0	180	3.3	13.14	10.7 (2.44)	12.93 (0.21)	13.36 (0.22)	12.92 (0.22)	13.15 (0.01)
135	180	4.29	4.34	5.5 (1.16)	3.96 (0.34)	4.11 (0.23)	4.23 (0.11)	4.23 (0.11)
180	180	6.04	1.35	0.9 (0.45)	1.51 (0.16)	1.36 (0.01)	3.44 (2.09)*	1.38 (0.03)
45	180	2.87	16.42	13.3 (3.12)	14.61 (1.81)	14.81 (1.61)	15.74 (0.68)	15.23 (1.19)
90	0	5.57	2.35	1.4 (0.95)	2.46 (0.11)	2.69 (0.34)	2.42 (0.07)	2.37 (0.02)
90	180	2.87	12.80	15.5 (2.7)	13.91 (1.11)	12.93 (0.13)	12.94 (0.14)	13.49 (0.69)
90	90	3.87	5.99	7.1 (1.11)	6.03 (0.04)	6.00 (0.01)	5.71 (0.28)	5.68 (0.31)
45	60	3.79	6.69	7.20 (0.51)*	3.07 (3.62)*	6.35 (0.34)*	1.48 (5.21)*	6.45 (0.24)
45	120	3.17	12.50	13.03 (0.53)*	7.48 (5.02)*	11.74 (0.76)*	12.26 (0.24)	12.2 (0.30)
75	150	2.81	12.58	15.46 (2.88)*	12.82 (0.32)*	13.99 (1.41)*	12.72 (0.14)	12.88 (0.30)
30	180	3.17	15.59	12.76 (2.83)*	14.86 (0.73)*	14.41 (1.18)*	15.10 (0.49)*	14.9 (0.69)
60	180	2.81	16.71	14.34 (2.37)*	14.68 (2.03)*	15.22 (1.49)*	15.84 (0.87)	15.6 (1.11)
120	180	3.58	6.64	10.43 (3.79)*	6.43 (0.21)*	5.87 (0.77)*	5.72 (0.92)*	6.37 (0.27)
H2 site								
0	180	3.3	10.46	9.8 (0.66)	10.32 (0.14)	10.20 (0.26)	10.37 (0.09)	10.30 (0.16)
135	180	4.2	5.08	5.3 (0.22)	4.96 (0.12)	5.06 (0.02)	4.94 (0.14)	4.78 (0.30)
180	180	6.04	1.48	1.0 (0.48)	1.42 (0.06)	1.44 (0.04)	3.4 (1.92)*	1.45 (0.03)
45	180	3.08	11.82	12.7 (0.88)	11.39 (0.43)	11.72 (0.10)	11.77 (0.05)	11.58 (0.24)
90	0	5.37	2.33	1.4 (0.93)	2.37 (0.04)	2.47 (0.14)	2.38 (0.05)	2.43 (0.10)
90	180	2.67	15.07	14.7 (0.37)	14.86 (0.21)	14.67 (0.40)	14.68 (0.39)	15.20 (0.13)
90	90	3.87	5.75	7.1 (1.35)	6.03 (0.28)	5.84 (0.09)	5.58 (0.17)	5.74 (0.01)
45	60	4.02	4.44	6.55 (2.11)*	2.11 (2.33)*	6.32 (1.88)*	4.10 (0.34)*	4.50 (0.06)
45	120	2.99	10.61	12.09 (1.48)*	3.96 (6.65)*	7.08 (3.53)*	10.36 (0.25)	10.26 (0.35)
75	150	2.81	13.78	14.50 (0.72)*	13.24 (0.54)*	14.00 (0.22)*	13.58 (0.20)	13.65 (0.13)
30	180	3.17	10.73	12.19 (1.46)*	10.54 (0.19)*	10.52 (0.21)*	10.75 (0.02)*	10.64 (0.09)
60	180	2.81	13.09	13.37 (0.28)*	12.59 (0.50)*	12.79 (0.30)*	12.80 (0.29)	12.73 (0.36)
120	180	3.37	8.55	10.61 (2.06)*	7.88 (0.67)*	7.35 (1.20)*	7.15 (1.40)*	7.97 (0.58)

EnsFFNNs^1- Training with 366 energy points, tested with 414 energy points (results from Table 1 and Table 2); EnsFFNNs^2- Same training and test set as EnsFFNNs^1, different initial weights; EnsFFNNs^3 - Training with a new partition in training and test set (training with 536 energy points, test with 244 energy points); EnsFFNNs^4 - Training with all data available.

For all orientations the results from the test set are marked with *;

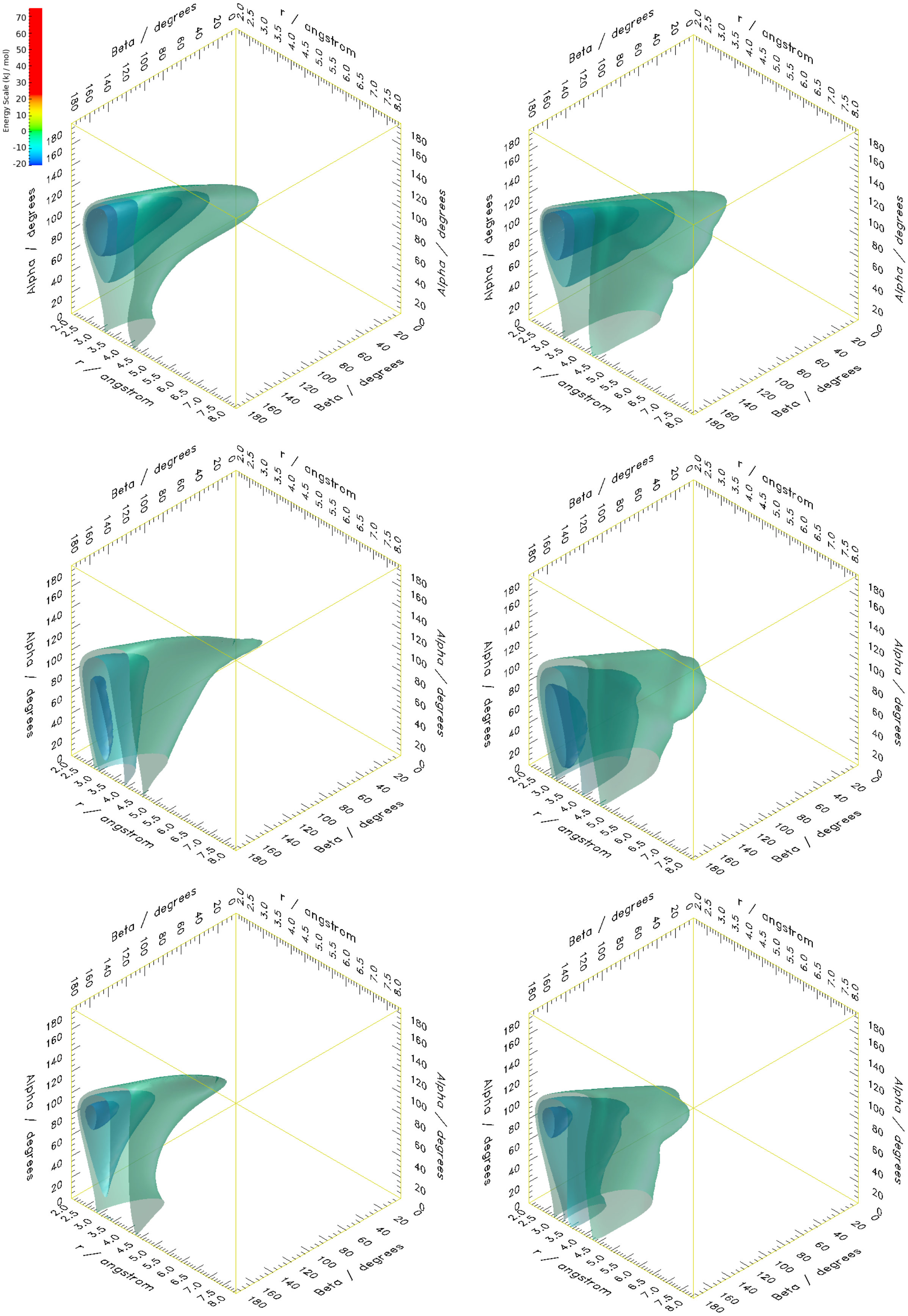


Figure 12.9: PES from training with the first set (first column) and with all available data (second column) for the *Top* (first row), *H1* (second row) and *H2* (third row) sites. Isoenergetic surfaces at: -14, -10 and -5 kJ mol^{-1}. The colored bar represents the energy scale.

Table 12.4: MAE, RMSE and correlation coefficient for training, validation and test sets for different ensembles and analytical function.

		MAE / kJ mol^{-1}			RMSE / kJ mol^{-1}			Correlation Coefficient		
		Learn	Valid	Test	Learn	Valid	Test	Learn	Valid	Test
For all energy points	EnsFFNNs^1	0.21	0.70	2.15	0.38	1.67	4.84	0.9996	0.9915	0.8939
	EnsFFNNs^2	0.24	0.67	1.83	0.44	1.78	6.37	0.9994	0.9903	0.8131
	EnsFFNNs^3	0.14	0.36	1.83	0.22	0.94	4.94	0.9998	0.9969	0.9047
	EnsFFNNs^4	0.21	0.45	-	0.52	1.30	-	0.9990	0.9721	-
Only for energy minima	Anal. func.	1.29	-	2.28	1.83	-	2.78	0.9789	-	0.7749
	EnsFFNNs^1	0.35	0.58	2.13	0.54	0.77	3.36	0.9955	0.9917	0.8506
	EnsFFNNs^2	0.39	0.44	1.23	0.67	0.76	1.5	0.9939	0.9925	0.9117
	EnsFFNNs^3	0.24	0.37	1.23	0.35	0.50	1.83	0.9985	0.9970	0.9035
	EnsFFNNs^4	0.29	0.43	-	0.42	0.60	-	0.9972	0.9390	-

MAE - Mean absolute error; RMSE - Root mean square error; EnsFFNNs^1- Training with 366 energy points, tested with 414 energy points (results from Table 1 and Table 2); EnsFFNNs^2- Same training and test set as EnsFFNNs^1, different initial weights; EnsFFNNs^3 - Training with a new partition in training and test set (training with 536 energy points, test with 244 energy points); EnsFFNNs^4 - Training with all data available.

about the input patterns are stored in the NN weights of the networks) and a memory-based method (the data used to build the models are also stored in a ”memory” and the predictions are corrected based on some local approximations of the stored examples). The results obtained by ASNNs are not presented and discussed here because they are similar to the ones from EnsFFNNs. Nonetheless, this is an important methodology to take into account - it allows to incorporate new data in the memory after the training is finished, making it possible to improve predictions with new data without the need to retrain the NNs. It has a potential application in the course of our future experiments, when DFT data for new orientations and sites will become gradually available and kept in the ”memory” of the system.

12.4 Conclusions

The present results indicate that NNs can be trained to map PES with suitable accuracy to be used in molecular simulations, particularly for the ethanol - Au (111) interactions. Indeed, once the networks are well trained they are able to produce, as output, any required number of energy points for numerical interpolations, with similar or better accuracy than other mapping methods.

EnsFFNNs give better results than single FFNNs, showing their capability of taking into account the most subtle features of this type of interactions when the model was tested for orientations that did not participate in the learning procedure.

The NNs-PES have to be tested in molecular simulations. The test will be carried out using the tabular potential energies, predicted by the NNs, for working out thermal, structural and dynamical properties to be compared with the preliminary Monte Carlo simulation values already obtained from the analytical function [258]. One point has already turned out, however, in the present work: in general, the NNs can reproduce the DFT results with a better accuracy than the analytical function as far as the probed interaction sites are concerned.

Work is in progress regarding a much finer screening, by DFT, of the different gold interaction sites and ethanol orientations. Indeed, before a full simulation test can be performed, based on the NNs data, it is necessary to obtain DFT results for sites other than the *Top*, *H1* and *H2*. Such results will certainly increase the accuracy of the network mappings, using different memories as the new data is becoming available.

Finally, it should be mentioned that the representation of metallic surfaces by cluster models has been a common approximation in order to minimise the heavy computational requirements. We have commented on that elsewhere [258, 262]. Yet, nowadays, accurate periodic DFT methods can be implemented in relatively low cost computer networks. We will consider them in future applications to surface models. Nevertheless, the conclusions of the present paper do not depend on the choice of the system and model but rather indicate that NNs offer an alternative to be used in Monte Carlo and molecular dynamics simulations. As for molecular dynamics, which usually require continuously differentiable potentials, the interpolation routines may, however, slow down the simulations.

Acknowledgements

The assistance of Dr. Yuri Binev with the use of ASNN software is gratefully acknowledged. The authors also thank Dr. Igor Tetko who made available the software to implement EnsFFNNs and ASNNs.

Part IV

Conclusions and Future Work

Chapter 13

Conclusions and Future Work

This Thesis presents the application of automatic learning methods in the resolution of two main chemical problems: a) the classification of organic and metabolic reactions, and b) the mapping of Potential Energy Surfaces. In the course of the experiments presented here three main objectives were achieved:

- The development of a methodology for the representation of chemical reactions based on NMR data of the products and reactants of the reaction
- The development of a methodology for the representation of chemical reactions based on the physico-chemical and topological features of chemical bonds
- The NN mapping of PES of a complex system (interaction ethanol/Au(111)

A methodology is also suggested for the representation of metabolic pathways and organisms based on chemical reactions (an extension of the MOLMAP approach).

NMR-based classification of chemical reactions. The automatic classification of photochemical and enzymatic reactions, from the differences between ^{1}H NMR spectra of reactants and products, was demonstrated with a high level of accuracy. Two different automatic learning methods, Kohonen SOMs and Random Forests, were used. The use of ensembles of Kohonen SOMs allowed for an improvement in the results in comparison with individual SOMs. The results using Random Forests were similar to the ones from ensembles of SOMs. The obtained models for classification of photochemical reactions were tested with a subset of reactions represented with a combination of experimental and simulated spectra. Predictions with the same quality were obtained by Random Forests and a decrease of the accuracy level was observed in the predictions from ensembles of SOMs. These results suggest that a model can be built using simulated data and applied to experimental data.

The preliminary results of the experiments using mixtures of reactions to train Kohonen SOMs and FFNNs indicate that it is possible to infer the reaction types with two reactions occurring simultaneously.

The proposed method has some limitations related to the use of ^{1}H NMR data as input. The method can not be applied to reactions without hydrogen atoms in the neighbourhood of the reaction centre, and is limited by the sensitivity of their chemical shifts to the changes resulting from the reaction.

The presented results show the possibility of linking reaction and NMR data for automatic classification of reactions.

Despite the good results obtained with the classification of single chemical reactions, and the preliminary results with the identification of reactions in mixtures, probably the input given to NNs can be optimized. At this stage the input of NNs was simply the difference between the ^{1}H NMR pseudo-spectra of the products and reactants. As the SPINUS software now simulates full spectra (including coupling constants and peak shapes) full spectra can be used instead. In future work some chemometric methods such as Principal Component Analysis (PCA) [263], should be tested for the pre-processing of the spectra. Such an approach could reveal information about the most relevant regions of the NMR spectra for the characterization of each type of reaction. It could also possibly increase the accuracy of the predictions for the classification of mixtures. PCA and Partial Least Squares (PLS) [263] could also be tried as alternatives to Kohonen SOMs and Random Forests as machine learning methods.

Genome-scale classification of enzymatic reactions from their reaction equation. The presented results demonstrate the ability of MOLMAPs to be used in the codification of genome-scale data sets of metabolic reactions. The results show a general compatibility between the MOLMAP approach and the EC classification system. Two different automatic learning methods were used to classify metabolic reactions encoded by the MOLMAP approach. The experiments were carried out with Kohonen SOMs, an unsupervised learning technique, and with Random Forests, a supervised learning technique. The two techniques allow a different analysis of the data sets of enzymatic reactions and different conclusions.

The results of the unsupervised mapping reveal a general agreement with the EC classification system as is shown by the general reasonable clustering of reactions according to the three different levels of the EC hierarchy. Kohonen SOMs were able to assign the first, second and third digit of the EC numbers for reactions belonging to independent test sets with accuracies of 92%, 80% and 70%, respectively. It is to point out that in the learning procedure no information concerning the classification of the reactions was given. The model learns only with the information of the reaction descriptors. Two different measures of prediction reliability were considered taking into account the relation between the prediction accuracy and the number of votes for the predicted class in the ensemble of Kohonen SOMs, and the relation between prediction accuracy and the Euclidean distance of the reaction MOLMAP to the winning neuron. The experiments performed to predict the third digit of the EC number demonstrated the ability of MOLMAP descriptors to discriminate sub-subclasses.

The use of Kohonen SOMs to map genome-scale data sets of enzymatic reactions allows for the analysis of the correspondence between chemical similarity of metabolic reactions and similarities in their MOLMAP descriptors. With this analysis it is possible to detect a number of very similar reactions in the same neuron, but labeled with different EC classes. These cases demonstrated the possible application of the MOLMAP/SOM approach to the verification of internal consistency of classification in databases of metabolic reactions.

The possibility of applying Random Forests for the automatic assignment of EC numbers with better accuracy than Kohonen SOMs was demonstrated. EC numbers were correctly assigned in 95%, 90%, 85% and 86% of the cases at the class, subclass, sub-subclass and full EC number level, respectively, for independent test sets.

The experiments with Random Forests also demonstrated the possibility of assigning EC numbers only from the main reactants and products of the reaction. The level of accuracy decreases ∼25% with comparison with the ones obtained from the complete reaction.

Experiments with metabolic pathways and organisms show that the MOLMAP / SOM concept can be extended to the representation of other levels of metabolic information. The approach enabled the comparison of different pathways, the automatic classification of pathways, and a classification of organisms based on their biochemical machinery. It is shown that the three levels of classification (classification of chemical bonds, classification of reactions and classification of metabolic pathways) allowed mapping and perceiving chemical similarities between metabolic pathways even for pathways of different types of metabolism and pathways that do not share EC numbers. The map trained with organisms encoded on the basis of their reactomes reveals similarities between organisms and clustering that are correlated with the taxonomic classification.

The MOLMAP reaction descriptors present some advantages over other reaction representation methods:

- represent the reaction on the basis of physico-chemical and topological features of the reactants and products;
- are a fixed length numerical representation that eases the processing by automatic learning methods;
- allow the comparison of reactions with different number of bonds involved;
- do not require ranking of bonds;
- avoid the assignment of reaction centres and atom-to-atom mapping previous to the classification of reactions.

The presented experiments open the way to a host of new investigations on the diversity analysis of metabolic reactions and comparison of metabolic pathways. The presented work can be linked in the future to other applications in the domain of chemoinformatics and bioinformatics. At the same time several optimizations can be performed.

Concerning the MOLMAP reaction descriptors a possible improvement is related to the properties describing the chemical bonds. Experiments with other sets of descriptors, e.g. descriptors obtained from semi-empirical quantum chemistry methods, can be carried out to achieve a set of descriptors that can better distinguish the different types of bonds. In a different direction, the encoding of the pattern of activated neurons in the map trained with chemical bonds, used to build the MOLMAP, can be changed. In the experiments here performed, the frequency of activation of each neuron was used to obtain the MOLMAP of a compound, and in some experiments a value of 0.3 was added to the frequency count of a neuron each time a neighbor was activated. This value of 0.3 could be changed, or could be made proportional to the similarity between a neuron and the activated neuron.

MOLMAP reaction descriptors could also integrate information about the reaction centre, when this is available. Although one of the advantages of the method is to avoid the explicit assignment of reaction centres, this information can be useful to improve the accuracy of the predictions. The reaction MOLMAP can be built using only the bonds of the reaction center, or a combination of the reaction center with the reaction MOLMAP built with all the bonds. Specifically the classification of enzymatic reactions only from the main reactants and products could possibly be improved with the inclusion of the information about the reaction centre.

With the trained models that allow accurate assignments of, at least, the first three digits of the EC number, one future project is the implementation of a web interface for automatic assignment of enzymatic reactions. The model behind the interface will be built using a genome-scale data set of metabolic reactions that has been used in the experiments presented in this Thesis. The output will be the prediction for the first three digits of the EC number (sub-subclass) for the submitted reaction, and a list of the three most similar reactions in the data set (including the similarity measure as determined by the RFs and a link to the corresponding entry at the KEGG web site). Another application related to the web interface will be the mapping of a query reaction on a previously trained Kohonen SOM. In this case the web interface will be able to retrieve the most similar reactions based on previous examples hitting the winning neuron. The trained map with all reactions available could also incorporate information on organisms and metabolic pathways, to yield similar reactions occurring in specific organisms or pathways.

Mapping PES by Neural Networks. In the first experiments concerning the mapping of PES with NNs the results suggest that for LJ type potentials NNs could be trained to generate accurate PES to be used in molecular simulations.

This first preliminary study, concerning the experiments to map PES with NNs, is the approximation of one-dimensional potential functions by EnsFFNNs and ASNNs. It is to emphasize that the objective of this study is not replace the LJ potentials by the neural mappings. The purpose was just to analyze, systematically, the accuracy and limitations of different NNs using the typical test potential (LJ) in molecular simulations.

The obtained results show some interesting aspects that could be used in other more complex applications. The experiments with ASNNs demonstrate that it is possible to perform a train with a small data set and later improve the accuracy using different memories as new data become available. Other possibility, although computationally more expensive, is the retraining from the previous implemented models using more data. It is also shown that the availability of similar curves in the training set are essential for NNs to make accurate predictions for curves that do not participate in the training.

The experiments performed show that with a training set that covers well the object space, the NN models present a remarkable ability to interpolate between distant curves yielding good potentials to be used in molecular simulations.

This first set of experiments consisted of an assessment of NNs methodologies to be used in more complex systems. The main motivation was to approach multidimensional PES to simulate the adsorption and self-assembly of solvated organic molecules on noble-metal electrodes, for which good analytical functions are, in general, inexistent.

The results for the chosen system, ethanol/Au(111) surface, indicate that NNs can be trained to map PES with suitable accuracy to be used in molecular simulations. After the training the NNs are able to produce, as output, any required number of energy points for numerical interpolations, with similar or better accuracy than other mapping methods. The presented results are in general more accurate than the ones from a previously implemented analytical function for the three interaction sites investigated.

As was verified in the first set of experiments with LJ potentials, EnsFFNNs give better results than single FFNNs, showing their capability of taking into account the most subtle features of this type of interactions when the model was tested for orientations that did not participate in the learning procedure.

The NNs-PES for the ethanol/Au(111) surface have to be tested in molecular simulations. The test will be carried out using the tabular potential energies, predicted by the NNs, for working out thermal, structural and dynamical properties to be compared with the preliminary Monte Carlo simulation values already obtained from the analytical function.

Although the agreement between the DFT data and the NN predictions the quality of the models has to be improved for some regions of the PES. A much finer screening, by DFT, of the different gold interaction sites and ethanol orientations are needed and essential to obtain a more reliable method. Indeed, before a full simulation test can be performed, based on the NNs data, it is necessary to obtain DFT results for sites other than the *Top*, *H1* and *H2*. Such results will certainly increase the accuracy of the network mappings and can be used for assessing the usefullness of the ASNN methodology to increase the accuracy without retraining.

The full screening of other adsorption sites will be essential to encode all the Au(111) surface. This encoding will allow a prediction of the potential energy for each given orientation, distance and localization of the molecule relative to the surface. With this objective achieved, the model will be able to predict complete potential energy surfaces to be used in molecular simulations.

The results for LJ potentials and preliminary results for ethanol/Au(111) interaction show that the ASNN method gives better results than the single FFNNs and EnsFFNNs, and could be a useful method to generate PES of more complex systems capable of taking into account the most subtle features of this systems in contrast to single FFNNs and other common methods of fitting.

The development of the correct encoding of the molecule relatively to the surface will permit the application of this method to other systems of organic molecules adsorbed on metallic surfaces.

Bibliography

[1] J. Zupan and J. Gasteiger. *Neural Networks in Chemistry and Drug Design.* Wiley-VCH, Weinheim, 1999.

[2] J. Gasteiger. *Introduction*, In *Chemoinformatics: A Textbook.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, Berlin, 2003, pp 1–14.

[3] K. Brown. *Annu. Rep. Med. Chem.* 1998, *33*, 375–384.

[4] G. Paris. Meeting of the American Chemical Society, quoted by W. Warr at http://www.warr.com/warrzone.htm, 1999.

[5] A. Tropsha. *Application of Predictive QSAR Models to Database Mining*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 437–456.

[6] H. Kubinyi. *Quantitative Structure-Activity Relationships in Drug Design*, In *Encyclopedia of Computational Chemistry.* P. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner (Eds.). John Wiley & Sons, Chichester, 1999, Vol. 3, pp 2309–2330.

[7] J. Gasteiger and T. Engel (Eds.). *Applications*, In *Chemoinformatics: A Textbook.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, pp 487–618.

[8] P. C. Jurs. *Quantitative Structure-Property Relationships*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 3, pp 1314–1335.

[9] P. Selzer. *Correlations Between Chemical Structure and Infrared Spectra*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 3, pp 1349–1367.

[10] C. Steinbeck. *Correlations Between Chemical Structure and NMR Data*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 3, pp 1368–1377.

[11] C. Steinbeck. *Computer-Assisted Structure Elucidation*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 3, pp 1378–1406.

[12] D. J. Abraham. *Drug Discovery in Academia - A Case Study*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 457–484.

[13] C. M. W. Ho. *In Silico Lead Optimization*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 199–220.

[14] G. Marshall. *Introduction to Chemoinformatics in Drug Discovery. A Personal View*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 1–22.

[15] T. I. Oprea. *Chemoinformatics in Lead Discovery*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 25–42.

[16] C. L. Cavallaro, D. M. Schnur, and A. J. Tebben. *Molecular Diversity in Lead Discovery*, In *Chemoinformatics in Drug Discovery. Methods and Principles in Medicinal Chemistry.* T. I. Oprea (Ed.). Wiley-VCH, Weinheim, 2005, Vol. 23, pp 175–198.

[17] M. A. Ott. *Cheminformatics and Organic Chemistry. Computer-Assisted Synthetic Analysis*, In *Cheminformatics Developments - History, Reviews and Current Research.* J. H. Noordik (Ed.). IOS Press, Amsterdam, 2004, pp 83–110.

[18] J. Kelder, M. Wagener, and M. Timmers. *Cheminformatics and Drug Design*, In *Cheminformatics Developments - History, Reviews and Current Research.* J. H. Noordik (Ed.). IOS Press, Amsterdam, 2004, pp 110–128.

[19] G. Grethe. *Analysis of Reaction Information*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 4, pp 1407–1427.

[20] T. I. Oprea. *Chemoinformatics and the Quest for Leads in Drug Discovery*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 4, pp 1509–1531.

[21] H. Kubinyi. *QSAR in Drug Design*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 4, pp 1532–1554.

[22] M. C. Nicklaus. *Pharmacophore in Drug Discovery*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 4, pp 1687–1711.

[23] L. Chen. *Reaction Classification and Knowledge Acquisition*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 1, pp 348–388.

[24] J. Gasteiger. *Representation of Chemical Reactions*, In *Chemoinformatics: A Textbook.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, pp 169–202.

[25] B. Rost, J. Liu, D. Przybylski, R. Nair, K. O. Wrzeszczynski, H. Bigelow, and Y. Oman. *Prediction of Protein Structure through Evolution*, In *Handbook of Chemoinformatics: From Data to Knowledge.* J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, Vol. 4, pp 1789–1811.

[26] B. E. Shaknovich and J. Max Harvey. Quantifying structure-function uncertainty: A graph theoretical exploration into the origins and limitations of protein annotation. *J. Mol. Biol.* 2004, *337*, 933–949.

[27] A. E. Todd, C. A. Orengo, and J. M. Thornton. Evolution of function in protein superfamilies, from a structural perspective. *J. Mol. Biol.* 2001, *307*, 1113–1143.

[28] P. C. Babbitt. Definitions of enzyme function for the structural genomics era. *Curr. Opin. Chem. Biol.* 2003, *7*, 230–237.

[29] T. I. Oprea, A. Tropsha, J.-L. Faulon, and M. D. Rintoul. Systems chemical biology. *Nat. Chem. Biol.* 2007, *3*(8), 447–450.

[30] C. M. Dobson. Chemical space and biology. *Nature.* 2004, *432*, 824–828.

[31] C. Lipinski and A. Hopkins. Navigating chemical space for biology and medicine. *Nature.* 2004, *432*, 855–861.

[32] B. R. Stockwell. Exploring biology with small organic molecules. *Nature.* 2004, *432*, 846–854.

[33] B. Shoichet. Virtual screening of chemical libraries. *Nature.* 2004, *432*, 862–865.

[34] R. R. Breaker. Natural and engineered nucleic acids as tools to explore biology. *Nature.* 2004, *432*, 838–845.

[35] C. A. Orengo, F. M. Pearl, J. E. Bray, Todd A. E, A. C. Martin, L. Lo Conte, and J. M. Thornton. The CATH Database provides insights into protein structure/function relationships. *Nucleic Acids Res.* 1999, *27*, 275–279.

[36] S. Freilich, R. V. Spriggs amd R. A. George, B. Al-Lazikani, M. Swindells, and J. M. Thornton. The complement of enzymatic sets in different species. *J. Mol. Biol.* 2005, *349*, 745–763.

[37] M. Kanehisa and S. Goto. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* 2000, *28*, 27–30.

[38] M. Kanehisa, S. Goto, S. Kawashima, Y. Okuno, and M. Hattori. The KEGG resource for deciphering the genome. *Nucleic Acids Res.* 2004, *32*, D277–D280.

[39] M. Kanehisa, S. Goto, M. Hattori, K. F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: New developments in KEGG. *Nucleic Acids Res.* 2006, *34*, D354–D357.

[40] M. Kanehisa. A database for post-genome analysis. *Trends Genet.* 1997, *13*, 375–376.

[41] M. Kotera, Y. Okuno, M. Hattori, S. Goto, and M. Kanehisa. Computational assignement of the EC numbers for genomic-scale analysis of enzymatic reactions. *J. Am. Chem. Soc.* 2004, *126*, 16487–16498.

[42] A. Gutteridge, M. Kanehisa, and S. Goto. Regulation of metabolic networks by small molecule metabolites. *BMC Bioinformatics.* 2007, *8*, 88.

[43] T. Kadowaki, C. E. Wheelock, M. Hattori, S. Goto, and M. Kanehisa. Structure-activity relationships and pathway analysis of biological degradation processes. *J. Pestic. Sci.* 2006, *31*, 273–281.

[44] T. Kadowaki, C. E. Wheelock, T. Adachi, T. Kudo, S. Okamoto, N. Tanaka, K. Tonomura, G. Tsujimoto, H. Mamitsuka, S. Goto, and M. Kanehisa. Identification of endocrine disruptor biodegradation by integration of structure-activity relationship with pathway analysis. *Environ. Sci. Technol.* 2007, *41*, 7997–8003.

[45] M. Oh, T. Yamada, M. Hattori, S. Goto, and M. Kanehisa. Systematic analysis of enzyme-catalyzed reaction patterns and prediction of microbial biodegradation pathways. *J. Chem. Inf. Model.* 2007, *47*, 1702–1712.

[46] G. Lekishvili. *The Data*, In *Chemoinformatics: A Textbook*. J. Gasteiger and T. Engel (Eds.). Wiley-VCH, New York, 2003, pp 203–226.

[47] L. Brillouin. *Science and Information.* Academic Press, New York, 2nd edition, 1962.

[48] M. A. Fischler and O. Firschein. *Intelligence: The Eye, the Brain, and the Computer.* Addison-Wesley, Reading, 1987.

[49] T. M. Mitchell. *Machine Learning.* McGraw-Hill, Singapore, 1997.

[50] P. R. Schleyer, N. L. Allinger, T. Clark, J. Gasteiger, P. A. Kollman, H. F. Schaefer III, and P. R. Schreiner. *Encyclopedia of Computational Chemistry*, Vol. 3. John Wiley & Sons, Chichester, 1999.

[51] S. Haykin. *Neural Networks - A Comprehensive Foundation.* Prentice Hall, New Jersey, 1999.

[52] D. W. Patterson. *Artificial Neural Networks - Theory and Applications.* Prentice Hall, Singapore, 1996.

[53] J. A. Freeman. *Simulating Neural Networks with Mathematica.* Addison Wesley, USA, 1994.

[54] H. M. Cartwright. *Aplications of Artificial Intelligence in Chemistry.* Oxford, Oxford, 1993.

[55] T. L. Isenhour and P. C. Jurs. Some chemical applications of machine intelligence. *Anal. Chem.* 1971, *43*, 20A–35A.

[56] M. Bishop. *Neural Networks for Pattern Recognition.* Oxford University Press, Oxford, 1995.

[57] C. T. Kelley. *Iterative Methods for Optimization.* SIAM Press, Philadelphia, 1999.

[58] K. Madsen, H. B. Nielsen, and O. Tingleff. *Methods for Non-Linear Least Squares Problems.* Technical University of Denmark, Denmark, 2004.

[59] W. H. Press, S. A. Teukolsky, W. T. Vetterlung, and B. P. Flannery. *Numerical Recipes in C.* 2nd edition, Cambridge University Press, New York, 1994.

[60] M. T. Hagan and M. B. Menhaj. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. on Neural Networks.* 1994, *5*(6), 989–993.

[61] K. Levenberg. A method for the solution of certain nonlinear problems in least squares. *Quart. Appl. Math.* 1944, *2*, 164–168.

[62] An Algorithm for least squares estimation of nonlinear parameters. D. w. marquardt. *SIAM J.* 1963, *11*, 431–441.

[63] D. K. Agrafiotis, W. Cedeno, and V. S. Lobanov. On the use of neural network ensembles in QSAR and QSPR. *J. Chem. Inf. Comput. Sci.* 2002, *42*, 903–911.

[64] T. G. Dietterich. *Ensemble Learning*, In *The Handbook of Brain Theory and Neural Networks.* M. A. Arbib (Ed.). MIT Press, Cambridge, MA, 2002, pp 405–408.

[65] I. V. Tetko. Neural network studies. 4. Introduction to associative neural networks. *J. Chem. Inf. Comput. Sci.* 2002, *42*, 717–728.

[66] I. V. Tetko. Associative neural network. *Neural Processing Letters.* 2002, *16*, 187–199.

[67] B. Dasarthy. *Nearest Neighbor (NN) Norms.* IEEE Computer Society Press, Washington, DC, 1991.

[68] P. Gramatica, E. Papa, A. Marrocchi, L. Minuti, and A. Taticchi. Quantitative structure-activity relationship modeling of polycyclic aromatic hydrocarbon mutagenicity by classification methods based on holistic theoretical molecular descriptors. *Ecotoxicol. Environ. Saf.* 2007, *66*, 353–361.

[69] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees.* Chapman Hall/CRC, Boca Raton, FL, 2000.

[70] L. Breiman. Random forests. *Machine Learn.* 2001, *45*, 5–32.

[71] V. Svetnik, A. Liaw, C. Tong, J. C. Culberson, R. P. Sheridan, and B. P. Feuston. Random forest: A classification and regression tool for compound classification and QSAR modeling. *J. Chem. Inf. Comput. Sci.* 2003, *43*, 1947–1958.

[72] Z. Debeljak, A. Skrbo, I. Jasprica, A. Mornar, V. Plecko, M. Banjanac, and M. Medic-Saric. QSAR study of antimicrobial activity of some 3-nitrocoumarins and related compounds. *J. Chem. Inf. Model.* 2007, *47*, 918–926.

[73] Q.-Y. Zhang and J. Aires de Sousa. Random forest prediction of mutagenicity from empirical physicochemical descriptors. *J. Chem. Inf. Model.* 2007, *47*, 1–8.

[74] D. S. Palmer, N. M. O'Boyle, R. C. Glen, and J. B. O. Mitchell. Random forest models to predict aqueous solubility. *J. Chem. Inf. Model.* 2007, *47*, 150–158.

[75] T. S. Schroeter, A. Schwaighofer, S. Mika, A. Ter Laak, D. Suelzle, U. Ganzer, N. Heinrich, and K. R. Mueller. Estimating the domain of applicability for machine learning QSAR models: A study on aqueous solubility of drug discovery molecules. *J. Comput. Aided Mol. Des.* 2007, *21*, 485–498.

[76] R. Put and Y. V. Heyden. Review on modelling aspects on reversed-phase liquid chromatographic quantitative structure-retention relationships. *Anal. Chim. Acta.* 2007, *602*, 164–172.

[77] T. Hancock, R. Put, D. Coomans, Y. Vander Heyden, and T. Everingham. A performance comparison of modern statistical techniques for molecular descriptor

selection and retention prediction in chromatographic QSRR studies. *Chemom. Intell. Lab. Syst.* 2005, *76*, 185–196.

[78] M. Pardo and G. Sberveglieri. Random forests and nearest shrunken centroids for the classification of sensor array data. *Sens. Actuators, B.* 2008, *131*, 93–99.

[79] A. Koike. Comparison of methods for chemical-compound affinity prediction. *SAR QSAR Environ. Res.* 2006, *17*, 497–514.

[80] C. L. Bruce, J. L. Melville, S. D. Pickett, and J. D. Hirst. Contemporary QSAR classifiers compared. *J. Chem. Inf. Model.* 2007, *47*, 219–227.

[81] D. Plewczynski, S. A. H. Spieser, and U. Koch. Assessing different classification methods for virtual screening. *J. Chem. Inf. Model.* 2006, *46*, 1098–1106.

[82] J. M. Hettick, M. L. Kashon, J. E. Slaven, Y. Ma, J. P. Simpson, P. D. Siegel, G. N. Mazurek, and D. N. Weissman. Discrimination of intact mycobacteria at the strain level: A combined MALDI-TOF MS and biostatistical analysis. *Proteomics.* 2006, *6*, 6416–6425.

[83] P. J. Mazzone, J. Hammel, R. Dweik, J. Na, C. Czich, D. Laskowski, and T. Mekhail. Diagnosis of lung cancer by the analysis of exhaled breath with a colorimetric sensor array. *Thorax.* 2007, *62*, 565–568.

[84] D. Donald, T. Hancock, D. Coomans, and Y. Everingham. Bagged super wavelets reduction for boosted prostate cancer classification of seldi-tof mass spectral serum profiles. *Chemom. Intell. Lab. Syst.* 2006, *82*, 2–7.

[85] R. Korner and J. Apostolakis. Automatic determination of reaction mappings and reaction center information. 1. The imaginary transition state energy approach. *J. Chem. Inf. Model.* 2008, *48*, 1181–1189.

[86] W. Theilheimer. *Synthetic Methods of Organic Chemistry, 50 annual vols.* Karger, Basel, 50 annual vols. edition, 1946-1996.

[87] A. T. Balaban. Chemical graphs. 3. Reactions with cyclic 6-membered transition states. *Rev. Roum. Chim.* 1967, *12*, 875–902.

[88] J. B. Henrickson. The variety of thermal pericyclic reactions. *Angew. Chem. Int. Ed.* 1974, *13*, 47–76.

[89] J. F. Arens. Formalism for the classification and design of organic reactions. *Rec. Trav. Chim. Pays-Bas.* 1979, *98*, (a) 155–161, (b) 395–399, (c) 471–500.

[90] G. Vladutz. *Information Storage & Retrieval.* 1963, *1*, 117–146.

[91] S. Fujita. Description of organic reactions based on imaginary transition structures. 1. Introduction of new concepts. *J. Chem. Inf. Comput. Sci.* 1986, *26*, 205–212.

[92] S. Fujita. Description of organic reactions based on imaginary transition structures. 6. Classification and enumeration of 2-string reactions with one common node. *J. Chem. Inf. Comput. Sci.* 1987, *27*, 99–104.

[93] N. S. Zefirov and S. S. Tratch. An approach to systematization and design of organic reactions. *Acc. Chem. Res.* 1987, *20*, 237–243.

[94] N. S. Zefirov and S. S Tratch. Symbolic equations and their applications to reaction design. *Anal. Chim. Acta.* 1990, *235*, 115–134.

[95] N. S. Zefirov. SYMBEQ program and its applications in computer-assisted reaction design. *J. Chem. Inf. Comput. Sci.* 1994, *34*, 994–999.

[96] J. B. Hendrickson. Comprehensive system for classification and nomenclature of organic reactions. *J. Chem. Inf. Comput. Sci.* 1997, *37*, 852–860.

[97] I. Ugi and J. Dugundji. An algebraic model of constitutional chemistry as a basis for chemical computer programs. *Top. Curr. Chem.* 1973, *39*, 19.

[98] I. Ugi, M. Wochner, E. Fontain, J. Bauer, B. Gruber, and R. Karl. *Concepts and Applications of Molecular Similarity*, In *Concepts and Applications of Molecular Similarity.* M. A. Johnson and G. M. Maggiora (Eds.). Wiley, New York, 1990, pp 239–288.

[99] J. Gasteiger and C. Jochum. EROS - A computer program for generating sequences of reactions. *Top. Curr. Chem.* 1978, *74*, 93–126.

[100] J. Gasteiger M. G. Hutchings, B. Christoph, L. Gann, C. Hiller, P. Low, M. Marsili, H. Saller, and K. Yuki. A new treatment of chemical reactivity: Development of EROS, an expert system for reaction prediction and synthesis design. *Top. Curr. Chem.* 1987, *137*, 19–73.

[101] J. Bauer, R. Herges, E. Fontain, and I. Ugi. IGOR and computer-assisted innovation in chemistry. *Chimia.* 1985, *39*, 43–53.

[102] C. Jochum, J. Gasteiger, and I. Ugi. The principle of minimum chemical distance and the principle of minimum structure change. *Naturforsch.* 1982, *37b*, 1205–1215.

[103] E. Fontain and K. Reitsam. The generation of reaction networks with RAIN. 1. The reaction generator. *J. Chem. Inf. Comput. Sci.* 1991, *31*, 96–101.

[104] C. S. Wilcox and R. A. Levinson. *Artificial Intelligence Applications in Chemistry,* In *Am. Chem. Soc. Symposium Series.* T. H. Pierce and B. A. Hohne (Eds.). 1986, Vol. 306.

[105] H. Gelernter, J. R. Rose, and C. Chen. Building and refining a knowledge base for synthetic organic chemistry via the methodology of inductive and deductive machine learning. *J. Chem. Inf. Comput. Sci.* 1990, *30*, 492–504.

[106] J. Eiblmaier, G. Grethe, H. Kraut, and P. Loew. Linking reaction information from different sources in 224th National Meeting of the American Chemical Society, Boston, 2002.

[107] Infochem. Classify - the InfoChem reaction classification program version 2.9. to be found at http://www.infochem.de/en/downloads/documentations.shtml, 2008.

[108] N. M. O'Boyle, G. L. Holliday, D. E. Almonacid, and J. B. O. Mitchell. Using reaction mechanism to measure enzyme similarity. *J. Mol. Biol.* 2007, *368*, 1484–1499.

[109] A. Varnek, D. Fourches, F. Hoonakker, and V. P. Solov'ev. Substructural fragments: An universal language to encode reactions, molecular and supramolecular structures. *J. Comput.-Aided Mol. Des.* 2005, *19*, 693–703.

[110] J. R. Rose and J. Gasteiger. HORACE: An automatic system for the hierarchical classification of chemical reactions. *J. Chem. Inf. Comput. Sci.* 1994, *34*, 74–90.

[111] H. Satoh, O. Sacher, T. Nakata, L. Chen, J. Gasteiger, and K. Funatsu. Classification of organic reactions: Similarity of reactions based on changes in the electronic features of oxygen atoms at the reaction sites. *J. Chem. Inf. Comput. Sci.* 1998, *38*, 210–219.

[112] L. Chen, J. Gasteiger, and J. R. Rose. Automatic extraction of chemical knowledge from organic reaction data: Addition of carbon-hydrogen bonds to carbon-carbon double-bonds. *J. Org. Chem.* 1995, *60*, 8002–8014.

[113] L. Chen and J. Gasteiger. Knowledge discovery in reaction databases: Landscaping organic reactions by a self-organizing neural network. *J. Am. Chem. Soc.* 1997, *119*, 4033–4042.

[114] L. Chen and J. Gasteiger. Organic reactions classified by neural networks: Michael additions, Friedel-Crafts alkylations by alkenes, and related reactions. *Angew. Chem. Int. Ed. Engl.* 1996, *35*, 763–765.

[115] J. Gasteiger, M. Reitz, and O. Sacher. Multidimensional exploration into biochemical pathways in Proceedings of the Beilstein-Institut Symposium "The Chemical Theatre of Biological Systems, 2004.

[116] O. Sacher. *PhD Thesis, University of Erlangen-Nuremberg, can be found at: http://www2.chemie.uni-erlangen.de/services/dissonline/data/dissertation/OliverSacher/html/.* PhD thesis, 2001.

[117] Daylight. Daylight Theory Manual, Daylight version 4.9, release date 02/01/08, Daylight Chemical Information Systems, Inc., http://www.daylight.com/dayhtml/doc/theory, 2008.

[118] J.-L. Faulon, D. P. Visco, Jr., and R. S. Pophale. The signature molecular descriptor. 1. Using extended valence sequences in QSAR and QSPR studies. *J. Chem. Inf. Comput. Sci.* 2003, *43*, 707–720.

[119] J.-L. Faulon, M. Misra, S. Martin, K. Sale, and R. Sapra. Genome scale enzyme-metabolite and drug-target interaction predictions using the signature molecular descriptor. *Bioinformatics.* 2008, *24*, 225–233.

[120] L. Ridder and M. Wagener. SyGMa: Combining expert knowledge and empirical scoring in the prediction of metabolites. *ChemMedChem.* 2008, *3*, 821–832.

[121] J. B. Henrickson. A systematic organization of synthetic reactions. *J. Chem. Inf. Comput Sci.* 1979, *19*, 129–136.

[122] Q.-Y. Zhang and J. Aires de Sousa. Structure-based classification of chemical reactions without assignment of reaction centers. *J. Chem. Inf. Model.* 2005, *45*, 1775–1783.

[123] D. A. R. S. Latino and J. Aires de Sousa. Genome-scale classification of metabolic reactions: A chemoinformatics approach. *Angew. Chem. Int. Ed.* 2006, *45*, 2066–2069.

[124] D. A. R. S. Latino, Qing-You Zhang, and J. Aires de Sousa. Genome-scale classification of metabolic reactions and assignment of EC numbers with self-organizing maps. *Bioinformatics.* 2008, (doi:10.1093/bioinformatics/btn405).

[125] D. A. R. S. Latino and J. Aires de Sousa. Assignment of EC numbers to enzymatic reactions with MOLMAP reaction descriptors and random forests. *J. Chem. Inf. Model.* 2008, (in preparation).

[126] D. A. R. S. Latino and J. Aires de Sousa. Linking databases of chemical reactions to NMR data: An exploration of ^{1}H NMR - based reaction classification. *Anal. Chem.* 2007, *79*, 854–862.

[127] T. E. Moock, D. L. Grier, W. D. Hounshell, G. Grethe, K. Cronin, J. G. Nourse, and J. Theodosiau. Similarity searching in the organic reaction domain. *Tetrahedron Comput. Methodol.* 1988, *1*, 117–128.

[128] S. S. Tratch and N. S. Zefirov. A hierarchical classification scheme for chemical reactions. *J. Chem. Inf. Comput. Sci.* 1998, *38*, 349–366.

[129] S. Boyer, C. H. Arnby, L. Carlsson, J. Smith, V. Stein, and R. C. Glen. Reaction site mapping of xenobiotic biotransformations. *J. Chem. Inf. Model.* 2007, *47*, 583–590.

[130] J. Gasteiger. Modeling chemical reactions for drug design. *J. Comput. Aided Mol. Des.* 2007, *52*, 21–33.

[131] The program PETRA can be tested on the website: http://www2.chemie.uni-erlangen.de/software/petra and is developed by Molecular Networks GmbH, http:www.mol-net.de.

[132] J. Gasteiger, M. Marsili, M. G. Hutchings, H. Saller, P. Low, and K. Rafeiner. Models for the representation of knowledge about chemical reactions. *J. Chem. Inf. Comput. Sci.* 1990, *30*, 467–476.

[133] J. Gasteiger. Physicochemical effects in the representation of molecular structures for drug designing. *Mini Rev. Med. Chem.* 2003, *3*, 789–796.

[134] V. Simon, J. Gasteiger, and J. Zupan. A combined application of 2 different neural network types for the prediction of chemical reactivity. *J. Am. Chem. Soc.* 1993, *115*, 9148–9159.

[135] H. Saller and J. Gasteiger. Calculation of the charge distribution in conjugated systems by a quantification of the resonance concept. *Angew. Chem. Int. Ed. Engl.* 1985, *24*, 687–689.

[136] J. Gasteiger and M. Marsili. Iterative partial equalization of orbital electronegativity - A rapid access to atomic charges. *Tetrahedron.* 1980, *36*, 3219–3228.

[137] H. G. Hutchings and J. Gasteiger. Residual electronegativity - An empirical quantification of polar influences and its application to the proton affinity of amines. *Tetrahedron Lett.* 1983, *24*, 2541–2544.

[138] H. G. Hutchings and J. Gasteiger. Quantification of effective polarisability. Applications to studies of x-ray photoelectron spectroscopy and alkylamine protonation. *J. Chem. Soc.-Perkin Trans. 2.* 1984, pp 559–564.

[139] S. Gupta, S. Matthew, P. M. Abreu, and J. Aires de Sousa. QSAR analysis of phenolic antioxidants using MOLMAP descriptors of local properties. *Bioorg. Med. Chem.* 2006, *14*, 1199–1206.

[140] S. Gupta and J. Aires de Sousa. Comparing the chemical spaces of metabolites and available chemicals: Models of metabolite-likeness. *Mol. Diversity.* 2007, *11*, 23–36.

[141] T. Kohonen. *Self-Organization and Associative Memory.* Springer, Berlin, 1988.

[142] Enzyme commission site: http://www.chem.qmul.ac.uk/iubmb/enzyme/.

[143] V. Hatzimanikatis, C. Li, J. A. Ionita, and L. J. Broadbelt. Metabolic networks: Enzyme function and metabolite structure. *Curr. Opin. Struc. Biol.* 2004, *14*, 300–306.

[144] W. Tian and J. Skolnick. How well is enzyme function conserved as a function of pairwise sequence identity? *J. Mol. Biol.* 2003, *333*, 863–892.

[145] D. Devos and A. Valencia. Practical limits of function prediction. *Proteins.* 2000, *41*, 98–107.

[146] R. Y. Pinter, O. Rokhlenko, E. Yeger-Lotem, and M. Ziv-Ukelson. Alignment of metabolic pathways. *Bioinformatics.* 2005, *21*, 3401–3408.

[147] A. J. Barrett, C. R. Canter, C. Liebecq, G. P. Moss, W. Saenger, N. Sharon, K. F. Tipton, P. Vnetianer, and V. F. G. Vliegenthart. *Enzyme Nomenclature.* Academic Press, San Diego, 1992.

[148] K. Tipton and S. Boyce. History of the enzyme nomenclature system. *Bioinformatics.* 2000, *16*, 34–40.

[149] M. L. Green and P. D. Karp. Genome annotation errors in pathway databases due to semantic ambiguity in partial EC numbers. *Nucleic Acids Res.* 2005, *33*, 4035–4039.

[150] M. T. Alam and M. K. Alam. *Chemometric Analysis of NMR Spectroscopy Data: A Review*, In *Annual Reports on NMR Spectroscopy.* Academic Press, London, 2004, Vol. 54, pp 41–80.

[151] M. E. Bollard, E. G. Stanley, J. C. Lindon, J. K. Nicholson, and E. Holmes. NMR-based metabonomic approaches for evaluating physiological influences on biofluid composition. *NMR Biomed.* 2005, *18*, 143–162.

[152] J. R. Espina, J. P. Shockcor, W. J. Herron, B. D. Car, Contel N. R, P. J. Ciaccio, J. C. Lindon, E. Holmes, and J. K. Nicholson. Detection of in vivo biomarkers of phospholipidosis using NMR-based metabonomic approaches. *Magn. Reson. Chem.* 2001, *39*, 559–565.

[153] E. Holmes, A. Nicholls, J. C. Lindon, S. C. Connor, J. C. Connelly, J. S. Haselden, S. J. P. Damment, M. Spraul, P. Neidig, and J. K. Nicholson. Chemometric models for toxicity classification based on NMR spectra of biofluids. *Chem. Res. Toxicol.* 2001, *13*, 471–478.

[154] E. Holmes, J. K. Nicholson, A. W. Nicholls, J. C. Lindon, S. C. Connor, S. Polley, and J. Connelly. The identification of novel biomarkers of renal toxicity using automatic data reduction techniques and PCA of proton NMR spectra of urine. *Chemom. Intell. Lab. Syst.* 1998, *44*, 245–255.

[155] E. Holmes, A. W. Nicholls, J. C. Lindon, S. Ramos, M. Spraul, P. Neidig, S. C. Connor, J. Connelly, S. J. P. Damment, J. Haselden, and J. K. Nicholson. Development of a model for classification of toxin-induced lesions using 1H NMR spectroscopy of urine combined with pattern recognition. *NMR Biomed.* 1998, *11*, 235–244.

[156] J. C. Lindon, J. K. Nicholson, E. Holmes, and J. R. Everett. Metabonomics: Metabolic processes studied by NMR spectroscopy of biofluids. *Concepts Magn. Reson.* 2000, *12*, 289–320.

[157] J. K. Nicholson, J. C. Lindon, and E. Holmes. 'Metabonomics': Understanding the metabolic responses of living systems to pathophysiological stimuli via multivariate statistical analysis of biological NMR spectroscopic data. *Xenobiotica.* 1999, *29*, 1181–1189.

[158] K. Ott, N. Aranibar, B. Singh, and G. Stockton. Metabonomics classifies pathways affected by bioactive compounds. Artificial neural network classification of NMR spectra of plant extracts. *Phytochemistry.* 2003, *62*, 971–975.

[159] J. C. Lindon, E. Holmes, and J. K. Nicholson. Pattern recognition methods and applications in biomedical magnetic resonance. *Prog. NMR Spectrosc.* 2001, *39*, 1–40.

[160] O. Beckonert, J. Monnerjah, U. Bonk, and D. Leibfritz. Visualizing metabolic changes in breast-cancer tissue using 1H-NMR spectroscopy and self-organizing maps. *NMR Biomed.* 2003, *16*, 1–11.

[161] Q. N. Van, J. R. Klose, D. A. Lucas, D. A. Prieto, B. Luke, J. Collins, S. K. Burt, G. N. Chmurny, H. J. Issaq, T. P. Conrads, T. D. Veenstra, and S. K. Keay. The

use of urine proteomic and metabonomic patterns for the diagnosis of interstitial cystitis and bacterial cystitis. *Dis. Markers.* 2003, *19*, 169–183.

[162] T. F. Bathen, T. Engan, J. Krane, and D. Axelson. Analysis and classification of proton NMR spectra of lipoprotein fractions from healthy volunteers and patients with cancer or chd. *Anticancer Res.* 2000, *20*, 2393–2408.

[163] J.-Y. Shey and C.-M. Sun. Liquid-phase combinatorial reaction monitoring by conventional ^{1}H NMR spectroscopy. *Tetrahedron Lett.* 2002, *43*, 1725–1729.

[164] I. Vallikivi, I. Jarving, T. Penk, N. Samel, V. Tougu, and O. Parve. NMR monitoring of lipase-catalyzed reactions of prostaglandins: Preliminary estimation of reaction velocities. *J. Mol. Catal. B: Enzym.* 2004, *32*, 15–19.

[165] M. Maiwald, H. H. Fisher, Y. K. Kim, K. Albert, and H. Hasse. Quantitative high-resolution on-line NMR spectroscopy in reaction and process monitoring. *J. Magn. Reson.* 2004, *166*, 135–146.

[166] S. Kalelkar, E. R. Dow, J. Grimes, M. Clapham, and H. Hu. Automated analysis of proton NMR spectra from combinatorial rapid parallel synthesis using self-organizing maps. *J. Comb. Chem.* 2002, *4*, 622–629.

[167] Y. Binev and J. Aires de Sousa. Structure-based predictions of ^{1}H NMR chemical shifts using feed-forward neural networks. *J. Chem. Inf. Comput. Sci.* 2004, *44*, 940–945.

[168] Y. Binev, M. Corvo, and J. Aires de Sousa. The impact of available experimental data on the prediction of ^{1}H NMR chemical shifts by neural networks. *J. Chem. Inf. Comput. Sci.* 2004, *44*, 946–949.

[169] SPINUS can be accessed at http://www.dq.fct.unl.pt/spinus.

[170] SDBS can be acessed at http://riodb01.ibase.aist.go.jp/sdbs/cgi-bin/creindex.cgi.

[171] J. Aires de Sousa. JATOON: Java tools for neural networks. *Chemom. Intell. Lab. Syst.* 2002, *61*, 167–173.

[172] R Development Core Team. *R: A Language and Environment for Statistical Computing.* Vienna, 2004.

[173] Fortran original by Leo Breiman and Adele Cutler, R port by Andy Liaw and Mathew Wiener, http://www.stat.berkeley.edu/users/breiman/, 2004.

[174] F. B. Da Costa, Y. Binev, J. Gasteiger, and J. Aires de Sousa. Structure-based predictions of ^{1}H NMR chemical shifts of sesquiterpene lactones using neural networks. *Tetrahedron Lett.* 2004, *45*, 6931–6935.

[175] The JATOON applets are available at http://www.dq.fct.unl.pt/staff/jas/jatoon.

[176] J. Gasteiger. Automatic estimation of heats of atomization and heats of reaction. *Tetrahedron*. 1979, *35*, 1419–1426.

[177] S. Goto, T. Nishioka, and M. Kanehisa. LIGAND: Chemical database for enzyme reactions. *Bioinformatics*. 1998, *14*, 591–599.

[178] EC 2.5.1.53 in Enzyme Commission site: http://www.chem.qmul.ac.uk/iubmb/enzyme/ec2/5/1/53.html.

[179] EC 4.2.99.16 in Enzyme Commission site: http://www.chem.qmul.ac.uk/iubmb/enzyme/ec4/2/99/16.html.

[180] V. Atalay and R. Cetin-Atalay. Implicit motif distribution based hybrid computational kernel for sequence classification. *Bioinformatics*. 2005, *21*(8), 1429–1436.

[181] J. H. Ward. Hierarchical grouping to optimize an objective function. *J. Am. Statist. Assoc.* 1963, *58*, 236–244.

[182] F. Murtagh. *Review of Fast Techniques for Nearest Neighbour Searching*. Physica-Verlag, Vienna, COMPSTAT edition, 1984.

[183] A. El-Hamdouchi and P. Willet. Hierarchic document clustering using Ward's method. In Proceedings of the 9th International Conference on Research and Development in Information Retrieval, pp 149–156, 1986.

[184] L. A. Kelley, S. P. Gardner, and M. J. Sutcliffe. An automated approach for clustering an ensemble of NMR-derived protein structures into conformationally-related subfamilies. *Protein Eng.* 1996, *9*, 1063–1065.

[185] J. Aires de Sousa and J. Gasteiger. Chirality and its application to the prediction of the preferred enantiomer in stereoselective reactions. *J. Chem. Inf. Comput. Sci.* 2001, *41*(2), 369–375.

[186] B. J. Alder and T. E. Wainwright. Phase transition of hard sphere system. *J. Chem. Phys.* 1957, *27*, 1208–1209.

[187] B. J. Alder and T. E. Wainwright. Studies of Molecular Dynamics. I. General method. *J. Chem. Phys.* 1959, *31*, 459–466.

[188] H. E. Stanley. *Introduction to Phase Transitions and Critical Phenomena*. Oxford University Press, 1971.

[189] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Equation of state calculations by fast computing machines. *J. Chem. Phys.* 1953, *21*, 1087–1092.

[190] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids.* Claredon Press, Oxford, 1987.

[191] A. Rahman. Correlations in motion of atoms in liquid argon. *Phys. Rev.* 1964, *136*, A405–A411.

[192] L. Verlet. Computer experiments on classical fluids. I. Thermodynamical properties of Lennard-Jones molecules. *Phys. Rev.* 1967, *159*, 98–103.

[193] L. V. Woodcock. Isothermal molecular dynamics calculations for liquid salts. *Chem. Phys. Lett.* 1971, *10*, 257–261.

[194] H. Andersen. Molecular Dynamics simulations at constant pressure and/or temperature. *J. Chem. Phys.* 1980, *72*, 2384–2393.

[195] D. Brown and J. H. R. Clarke. A comparison of constant energy, constant temperature and constant pressure ensembles in molecular dynamics simulations of atomic liquids. *Mol. Phys.* 1984, *51*, 1243–1252.

[196] J. M. Haile and H. W. Graben. Molecular dynamics simulations extended to various ensembles. I. Equilibrium properties in the isoenthalpic-isobaric ensemble. *J. Chem. Phys.* 1980, *73*, 2412–2419.

[197] W. G. Hoover. Constant-pressure equations of motion. *Phys. Rev. A.* 1986, *34*, 2499–2500.

[198] S. Nose. A molecular dynamics method for simulations in the canonical ensemble. *Mol. Phys.* 1984, *52*, 255–268.

[199] D. J. Evans and G. P. Morriss. Isothermal-isobaric molecular dynamics. *Chem. Phys.* 1983, *77*, 63–66.

[200] S. Melchionna, G. Ciccotti, and B. L. Hollian. Hoover NpT dynamics for systems varying in shape and size. *Mol. Phys.* 1993, *78*, 533–544.

[201] T. Cagin and B. M Pettitt. Molecular dynamics with a variable number of molecules. *Mol. Phys.* 1991, *72*, 169–175.

[202] T. Cagin and B. M Pettitt. Dynamic simulations of water at constant chemical-potential. *J. Chem. Phys.* 1992, *96*, 1333–1342.

[203] L. F. Vega, K. S. Shing, and L. F. Rull. A new algorithm for molecular dynamics simulations in the grand-canonical ensemble. *Mol. Phys.* 1994, *82*, 439–453.

[204] Daan Frenkel and Berend Smit. *Understanding Molecular Simulations: From Algorithms to Applications.* Academic Press, 2nd edition, 2002.

[205] F. Mandl. *Statistical Physics.* John Wiley & Sons, Ltd, 2nd edition, 1988.

[206] I. R. McDonald. Monte Carlo calculations for one- and two-component fluids in the isothermal-isobaric ensemble. *Chem. Phys. Lett.* 1969, *3*, 241–243.

[207] D. J. Adams. Grand canonical ensemble Monte-Carlo for a Lennard-Jones fluid. *Mol. Phys.* 1975, *29*, 307–311.

[208] A.Z. Panagiotopoulos, N. Quirke, M. Stapleton, and D.J. Tildesley. Phase equilibria by simulation in the Gibbs ensemble - Alternative derivation, generalization and application to mixture and membrane equilibria. *Mol. Phys.* 1988, *63*, 527–545.

[209] A. Z. Panagiotopoulos. Direct determination of phase coexistence properties of fluids by Monte-Carlo simulation in a new ensemble. *Mol. Phys.* 1987, *61*, 813–826.

[210] J. K. Johson, E. A. Muller, and K. E. Gubbins. Reactive canonical Monte-Carlo. A new simulation technique for reacting or associating fluids. *J. Phys. Chem.* 1994, *98*, 6413–6419.

[211] B. Smit and D. Frenkel. Calculation of the chemical-potential in the Gibbs ensemble. *Mol. Phys.* 1989, *68*, 951–958.

[212] D. A. Kofke. Direct evaluation of phase coexistence by molecular simulation via integration along the saturation line. *J. Chem. Phys.* 1993, *98*, 4149–4162.

[213] F. M. S. S. Fernandes and J. P. P. Ramalho. Hypervolumes in microcanonical Monte Carlo. *Comput. Phys. Commun.* 1995, *90*, 73–80.

[214] K. Binder. *Monte Carlo Methods in Statistical Physics.* Springer-Verlag, Berlin, 2nd edition, 1986.

[215] D. J. Adams and G. S. Dubey. Taming the Ewald sum in the computer simulation of charged systems. *J. Comput. Phys.* 1987, *72*, 156–176.

[216] D. Hirst. *A Computational Approach to Chemistry.* Blackwell Scientific Publications, Oxford, 1990.

[217] A. Hinchliffe. *Molecular Modelling for Beginners.* John Wiley & Sons, Chichester, 2003.

[218] D. K. Remler and P. A. Madden. Molecular dynamics without effective potentials via the Car-Parrinello approach. *Mol. Phys.* 1990, *70*, 921–966.

[219] J. Dai and J. Z. H. Zhang. Quantum adsorption dynamics of a diatomic molecule on surface: Four-dimensional fixed-site model for H_2 on Cu(111). *J. Chem. Phys.* 1995, *102*, 6280–6289.

[220] A. Forni, G. Wiesenekker, E. J. Baerends, and G. F. Tantardini. A dynamical study of the chemisorption of molecular hydrogen on the Cu(111) surface. *J. Phys. Condens. Matter.* 1995, *7*, 7195–7207.

[221] J. N. Murrell, S. Carter, S. C. Farantos, P. Huxley, and A. J. C. Varandas. *Molecular Potential Energy Functions.* John Wiley & Sons, London, 1984.

[222] H. F. Busnengo, A. Salin, and W. Dong. Representation of the 6D potential energy surface for a diatomic molecule near a solid surface. *J. Chem. Phys.* 2000, *112*, 7641–7651.

[223] R. P. A. Bettens and M. A. Collins. Learning to interpolate molecular potential energy surfaces with confidence: A bayesian approach. *J. Chem. Phys.* 1999, *111*, 816–826.

[224] C. Crespos, M. A. Collins, E. Pijper, and G. Kroes. Multi-dimensional potential energy surface determination by modified Shepard interpolation for a molecule-surface reaction: H_2 + Pt(111). *J. Chem. Phys. Lett.* 2003, *376*, 566–575.

[225] C. Crespos, M. A. Collins, E. Pijper, and G. Kroes. Application of the modified Shepard interpolation method to the determination of the potential energy surface for a molecule-surface reaction: H_2 + Pt(111). *J. Chem. Phys.* 2004, *120*, 2392–2404.

[226] J. Ischtwan and M. A. Collins. Molecular potential energy surfaces by interpolation. *J. Chem. Phys.* 1994, *100*, 8080–8088.

[227] M. J. T. Jordan, K. C. Thompson, and M. A. Collins. The utility of higher order derivatives in constructing molecular potential energy surfaces by interpolation. *J. Chem. Phys.* 1995, *103*, 9669–9675.

[228] A. C. P. Bittencourt, F. V. Prudente, and J. D. M. Vianna. The fitting of potential energy and transition moment functions using neural networks: Transition probabilities in OH. *Chem. Phys.* 2004, *297*, 153–161.

[229] K.-H. Cho, K. T. No, and H. A. Scheraga. A polarizable force field for water using an artificial neural network. *J. Mol. Struct.* 2002, *641*, 77–91.

[230] H. Gassner, M. Probst, A. Lauenstein, and K. Hermansson. Representation of intermolecular potential functions by neural networks. *J. Phys. Chem. A.* 1998, *102*, 4596–4605.

[231] S. Lorenz, M. Sheffler, and A. Grob. Descriptions of surface chemical reactions using a neural network representation of the potential-energy surface. *Phys. Review B.* 2006, *73*, 115431.

[232] S. Lorenz, A. Grob, and M. Sheffler. Representing high-dimensional potential-energy surfaces for reactions at surfaces by neural networks. *Chem. Phys. Lett.* 2004, *395*, 210–215.

[233] S. Manzhos and T. Carrington Jr. A random-sampling high dimensional model representation neural network for building potential energy surfaces. *J. Chem. Phys.* 2006, *125*, 084109.

[234] K. T. No, B. H. Chang, S. Y. Kim, M. S. Ihon, and H. A. Scheraga. Description of the potential energy surface of the water dimer with an artificial neural network. *Chem. Phys. Lett.* 1997, *271*, 152–156.

[235] F. V. Prudente and J. J. S. Neto. The fitting of potential energy surfaces using neural networks. Application to the study of the photodissociation processes. *Chem. Phys. Lett.* 1998, *287*, 585–589.

[236] L. M. Ralf, M. Malshe, M. Hagan, D. I. Doughan, M. G. Rockley, and R. Komanduri. Ab initio potential-energy surfaces for complex, multichannel systems using modified novelty sampling and feedforward neural networks. *J. Chem. Phys.* 2005, *122*, 084104.

[237] T. M. Rocha Filho, Z. T. Oliveira, L. A. C. Malbouisson, R. Gargano, and J. J. S. Neto. The use of neural networks for fitting potential energy surfaces: A comparative case study for the H_3^+ molecule. *Int. J. Quantum Chem.* 2003, *95*, 281–288.

[238] P. Hohenberg and W. Kohn. Inhomogeneous electron gas. *Phys. Rev.* 1964, *136*, B864–B871.

[239] I. N. Levine. *Quantum Chemistry.* Prentice-Hall, Inc., 4th edition, 1991.

[240] W. Kohn and L. J. Sham. Self-consistent equations including exchange and correlation effects. *Phys. Rev.* 1965, *140*, A1133–A1138.

[241] Wolfram Koch and Max C. Holthausen. *A Chemist's Guide to Density Functional Theory.* Wiley-VCH, 2001.

[242] John P. Perdew and Stefan Kurth. *1. Density Functionals for Non-relativistic Coulomb Systems in the New Century*, In *A Primer in Density Functional Theory.* Carlos Fiolhais, Fernando Nogueira, and Miguel A. L. Marques (Eds.). Spriger-Verlag Berlin Heidelberg, 2003.

[243] J. C. Slater. A simplification of the Hartree-Fock method. *Phys. Rev.* 1951, *81*, 385–390.

[244] J. P. Perdew, K. Burke, and M. Ernzrhof. Generalized gradient approximation made simple. *Phys. Rev. Lett.* 1996, *77*, 3865–3868.

[245] D. A. Becke. Density-functional exchange-energy approximation with correct asymptotic behavior. *Phys. Rev. A.* 1988, *38*, 3098–4000.

[246] A. D. Becke. Density-functional thermochemistry. III. The role of exact exchange. *J. Chem. Phys.* 1993, *98*, 5648–5652.

[247] M. J. Frisch, G. W. Trucks, H. B. Schlegel, G. E. Scuseria, M. A. Robb, J. R. Cheeseman, V. G. Zakrzewski, J. A. Montgomery, R. E. Stratmann, J. C. Burant, S. Dapprich, J. M. Millam, A. D. Daniels, K. N. Kudin, M. C. Strain, O. Farkas, J. Tomasi, V. Barone, M. Cossi, R. Cammi, B. Mennucci, C. Pomelli, C. Adamo, S. Clifford, J. Ochterski, G. A. Petersson, P. Y. Ayala, Q. Cui, K. Morokuma, D. K. Malick, A. D. Rabuck, K. Raghavachari, J. B. Foresman, J. Cioslowski, J. V. Ortiz, B. B. Stefanov, G. Liu, A. Liashenko, P. Piskorz, I. Komaromi, R. Gomperts, R. L. Martin, D. J. Fox, T. Keith, M. A. Al-Laham, C. Y. Peng, A. Nanayakkara, C. Gonzalez, M. Challacombe, P. M. W. Gill, B. G. Johnson, W. Chen, M. W. Wong, J. L. Andres, M. Head-Gordon, E. S. Replogle, and J. A. Pople. *Gaussian 98.* Gaussian Inc., Pittsburgh PA, 1998.

[248] A. D. Becke. A new mixing of Hartree-Fock and local density-functional theories. *J. Chem. Phys.* 1993, *98*, 1372–1377.

[249] C. Lee, W. Yang, and R. G. Parr. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron-density. *Phys. Rev. B.* 1988, *37*, 785–789.

[250] J. B. Witkoskie and D. J. D. J. Doren. Neural network models of potential energy surfaces: Prototypical examples. *J. Chem. Theory Comput.* 2005, *1*, 14–23.

[251] G. Toth, N. Király, and A. Vrabecz. Pair potentials from diffraction data on liquids: A neural network solution. *J. Chem. Phys.* 2005, *123*, 174109.

[252] D. A. R. S. Latino, F. F. M. Freitas, J. Aires de Sousa, and F. M. S. S. Fernandes. Neural networks to approach potential energy surfaces. Application to a Molecular Dynamics simulation. *Int. J. Quantum Chem.* 2007, *107*(11), 2120–2132.

[253] D. A. R. S. Latino, R. P. S. Fartaria, F. F. M. Freitas, J. Aires de Sousa, and F. M. S. S. Fernandes. Mapping potential energy surfaces by neural networks. The ethanol / Au (111) interface. *J. Electroanal. Chem.* 2008, (in press).

[254] F. Cuadros, J. Cachadina, and W. Ahmuda. Determination of Lennard-Jones interaction parameters using a new procedure. *Molecular Engineering.* 1996, *6*, 319–325.

[255] J. O. Hirschfelder, C. F. Curtiss, and R. B. Bird. *Molecular theory of gases and liquids.* Wiley, New-York, 1954.

[256] A. J. Shepherd. *Second Order Methods for Neural Networks.* Springer-Verlag, London, 1997.

[257] VCCLAB, Virtual Computational Chemistry Laboratory, b) http://www.vcclab.org, 2005.

[258] R. P. S. Fartaria, F. F. M. Freitas, and F .M. S. Silva Fernandes. A force field for simulating ethanol adorption on Au(111) surfaces. A DFT study. *Int. J. Quantum Chem.* 2007, *107*(11), 2169–2177.

[259] M. Y. Ballester and A. J. C. Varandas. Double many-body expansion potential energy surface for ground state HSO_2. *Phys. Chem. Chem. Phys.* 2005, *7*, 2305–2317.

[260] P. J. Hay and W. R. Wadt. Ab initio effective core potentials for molecular calculations. Potentials for the transition metal atoms Sc to Hg. *J. Chem. Phys.* 1985, *82*, 270–283.

[261] W. J. Hehre, R. Ditchfield, and J. A. Pople. Self-consistent molecular orbital methods. XII. Further extensions of gaussian-type basis sets for use in molecular orbital studies of organic molecules. *J. Chem. Phys.* 1972, *56*, 2257–2261.

[262] R. S. Neves, A. J. Motheo, R. P. S. Fartaria, and F. M. S. S. Fernandes. Modelling water adsorption on Au(210) surfaces. I. A force field for water-Au interactions by DFT. *J. Electroanal. Chem.* 2007, *609*, 140–146.

[263] R. G. Brereton. *Chemometrics: Data Analysis for the Laboratory and Chemical Plant.* John Wiley & Sons, 2004.