NOVA IMS
Information Management School

MAA

Mestrado em Métodos Analíticos Avançados
Master Program in Advanced Analytics

An Initialization Technique for Geometric Semantic Genetic Programming based on Demes Evolution and Despeciation

Machine Learning for *Rare Diseases*: a Case Study

Illya Bakurov

Dissertaion submitted in partial fulfillment of the requirements for the degree of

Master of Science in Advanced Analytics

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

This thesis was prepared under the supervision of:

Leonardo Vanneschi

(lvanneschi@novaims.unl.pt)

Associate Professor

at NOVA IMS

and

Mauro Castelli

(mcastelli@novaims.unl.pt)

Auxiliary Professor

at NOVA IMS

An Initialization Technique for Geometric Semantic Genetic Programming based on Demes Evolution and Despeciation

This document was created using the (pdf)LaTeX processor, based in the "novathesis" template[1], developed at the Dep. Informática of FCT-NOVA [2].
[1] https://github.com/joaomlourenco/novathesis [2] http://www.di.fct.unl.pt

Acknowledgements

I would like to thank my two amazing supervisors, professors Leonardo Vanneschi and Mauro Castelli, for their wise words told in the right moments, impressive availability and understanding. Also, I would like to thank every teacher I had - each one of them directly or indirectly contributed to this work.

Special acknowledgments are to my parents who gave me life, education and always supported me from the first step (of my life).

Finally, I would like to thank *destiny* for all this journey called *life*. Namely I would like to thank for difficulties, strikes and failures - something we use to reference as *experience*. Simply because *experience* is a mean of improvement.

Abstract

Evolutionary Algorithms mimic *Darwin's Theory of Evolution* for Machine Learning. A set of candidate solutions, called *individuals*, are let to evolve in iterative manner exposed to adaptation through simulation of natural selection mechanism.

Genetic Programming (GP), is a supervised Machine Learning technique for automatic induction of computer programs from a set of training examples. Initializing the population is an important step for GP, and several strategies have been proposed so far. The issue is particularly important for Geometric Semantic Genetic Programming (GSGP), a sub-field of GP, where initialization is known to play a very important role.

In this thesis, an initialization technique inspired by the biological phenomenon of *demes despeciation* is proposed, i.e. the combination of *demes* from previously distinct species into a new population. In synthesis, the initial population for GP, or its variant GSGP, can be created using individuals from a set of separate sub-populations, or *demes*, some of which run standard GP and the others GSGP evolutionary algorithm for few generations. GSGP with this novel initialization technique is shown to outperform GSGP using traditional Ramped Half-and-Half (RHH) algorithm on six complex symbolic regression applications. More specifically, on all studied problems, the proposed initialization technique allows to generate solutions with comparable or even better generalization ability, and of significantly smaller size than with RHH algorithm.

Additionally, it is shown the practical application of the algorithm to solve a specific problem in context of an institutional collaboration with *Casa dos Marcos*, currently the first and unique resource center for *Rare Diseases* in Portugal, along with other (descriptive) techniques.

Keywords: Genetic Programming, Geometric Semantic Genetic Programming, Initialization Algorithms, Evolutionary Algorithms, Machine Learning, Data Mining, Hierarchical Clustering ...

Resumo

Algoritmos Evolutivos reproduzem os princípios da *Teoria de Evolução de Darwin* para a Aprendizagem Automática. Um conjunto de soluções candidatas, chamadas *indivíduos*, são evoluídas de forma iterativa e expostas à adaptação através da simulação do mecanismo de seleção natural.

Programação Genética (PG), é uma ramo na Aprendizagem Automática supervisionada para a indução automática de programas computacionais a partir de um conjunto de exemplos de treino. A inicialização da população é um passo importante para PG, e várias estratégias já foram propostas até agora. A questão é particularmente importante para Programação Genética em Geometria Semântica (PGGS), um sub-campo de PG, onde a inicialização é conhecida por desempenhar um papel muito importante.

Nesta tese, é propósta uma técnica de inicialização inspirada no fenômeno biológico de *despeciação de sub-populações*, isto é, a combinação de *sub-populações* de espécies previamente distintas numa nova população. Em síntese, a população inicial para PG, ou para a sua variante PGGS, pode ser criada através de indivíduos oriundos de um conjunto de subpopulações separadas, algumas das quais são evoluídas por um algoritmo de PG, outras PGGS, durante poucas gerações. PGGS com esta nova técnica de inicialização supera PGGS usando o algoritmo tradicional - Ramped Half-and-Half (RHH) - em seis aplicações complexas de regressão simbólica. Mais especificamente, em todos os problemas estudados, a técnica de inicialização proposta permite obter soluções com capacidade de generalização comparável ou mesmo melhor, e de tamanho significativamente menor do que com o algoritmo RHH.

Além disso, é proposta uma aplicação prática do algoritmo desenvolvido, para além de técnias descritivas, para resolver um problema específico no contexto da colaboração institucional com a *Casa dos Marcos*, atualmente o primeiro e o único centro de recursos para *Doenças Raras* em território nacional.

Palavras-chave: Programação Genética, Programação Genética em Geometria Semântica, Algoritmos Evolutivos, Aprendizagem Automática, Data Mining, ...

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The current document outlines two fundamental aspects of conducted work. One of designing, implementing and assessing a novel initialization algorithm for Genetic Programming (GP). Another of applying the proposed algorithm, besides descriptive tools, in context of institutional collaboration with Portuguese Association of Mental and Rare Disabilities resource center - *Casa dos Marcos*.

Genetic Programming (GP) with integrated semantic awareness [20] is becoming popular [37]. The need for semantics-based techniques derives from a simple observation about the nature of GP: unlike other forms of evolutionary computation, it relies on the execution, or interpreted execution, of programs in order to attain fitness values [3] (although sometimes fitness also relies on other factors, like program structure [11, 17, 31, 36]). Under this perspective, in terms of creating initial random programs to seed a GP run, a semantically diverse starting population may be desirable [3], rather than a starting population that is only syntactically diverse. For this reason, several methods based on semantics have been proposed to ensure population diversity, especially in the initial GP population.

Beadle studies methods for an effective population initialization based on semantic diversity in [3, 4], giving a very convincing and effective idea about the importance of the initialization of GP populations, clearly showing and motivating the importance of semantics in this part of the evolutionary process. Many other researchers had already recognized that having an initial set of programs with possibly many different semantics increases the power of GP, bestowing on the process a wider exploration power [20, 21, 24, 34].

More recently, Moraglio and colleagues introduced Geometric Semantic Genetic

Programming (GSGP) [27, 35], a version of GP where traditional crossover and mutation are replaced by new genetic operators, called *geometric semantic operators*, that have precise and known effects on the semantics of the individuals. GSGP has raised a remarkable interest in the GP community, in part because of its interesting property of inducing an unimodal error surface for any supervised learning problem [35]. Thanks to its efficient implementation presented in [7], GSGP has also allowed us to obtain relevant applicative results, some of which are summarized in [35]. Since its introduction, it was clear that an appropriate method for initializing the population may play a crucial role for GSGP. Thus, a significant investigation in GP was devoted to the definition of population initialization methods that could facilitate the search of GSGP, making it more effective. In this context, Pawlak and Krawiec recently introduced Semantic Geometric Initialization [30] and Oliveira and colleagues introduced the concept of *dispersion of solutions*, for increasing the effectiveness of geometric semantic crossover [28].

The work presented in this document is contextualized in this research track. We present a new initialization method and we show its usefulness for GSGP. Our initialization method is inspired by the biological concepts of *demes evolution* and *despeciation*. In Biology, demes are local populations, or subpopulations, of polytypic species that actively interbreed with one another and share distinct gene pools [39]. The term despeciation indicates the combination of demes of previously distinct species into a new population [33]. Despeciation, although not very common in Nature, is a well-known phenomenon, and, in some cases, it is recognized to fortify populations. In simple terms, our idea consists in seeding the initial population with good quality individuals that have been evolved for few generations in other populations, or demes. In other words, in our system, a population of N individuals will be initialized using the best individuals in N different demes, that have been left to evolve independently for few generations. In this work, to foster diversity, part of these demes runs standard GP while the remaining part runs GSGP. Given that the proposed system mimics the evolution of demes, followed by despeciation, from now on it will be called *Evolutionary Demes Despeciation Algorithm* (EDDA).

Raríssimas is the name of *Portuguese Association of Mental and Rare Disabilities* whose mission is to provide support to patients and their relatives. *Casa dos Marcos* is the head-project of *Raríssimas*, currently the first and unique *Resource Center for Rare Diseases* in Portuguese territory where a pioneering care model in Portugal and Europe was deployed. It is focused on rare disabilities, integrating social and health care in a single assistance. Within the scope of social care, *Casa dos Marcos* has an iccupational Activity Center, a Residential Home and an Autonomous Residence. In the field of health, the center has an Ambulatory Clinical Unit, an Integrated Continuing Care Unit and a Development and Rehabilitation Center, which includes an Early Intervention service.

Advanced Analytical tools can bee seen as an intermediary tool between simple values, the data, and the knowledge which, if extracted correctly and applied wisely, can empower decision-making. Practical application of such tools (namely state-of-the-art algorithms like EDDA), in fields of medicine can strongly empower decision-making processes in research, diagnosis, therapeutic treatments, etc.

Along with novel initialization technique, we expose in this document the methodological framework and results of its application in context of institutional collaboration with *Casa dos Marcos* (from now on it will be called *entity*). Provided a specific problem, a top-down *Data Mining* approach was employed as a guideline for its solution. More concretely, six predictive models were deployed using EDDA for GSGP to foresee the expected effect of an extremely specialized therapy on five particular and one global domains. Additionally, one descriptive model was developed to find natural groups among a set of extremely particular data instances - patients with rare, generally neurodegenerative, diseases. Finally a specially designed web-application to support internal decision-making processes was developed and provided to the entity.

It is worth to highlight European Patients' Forum (EPF) [14] position to stronger the framework for the protection of personal data concerning health in numerous contexts, namely in health-care, *e-Health* and research. According to EPF *"Patients' health and genetic data are sensitive information which requires a high level of protection to ensure they are not unnecessarily disclosed. At the same time, the smooth sharing of these data is absolutely crucial for the good functioning of health-care services, patient safety, and to advancing research"*. For this reason, some details regarding previously described models and web-application are presented fractionally in this document.

Current document is organized as follows. *Theoretical Background* in chapter 2 contextualizes the reader with scientific area where the work presented in current document was integrated. *Evolutionary Demes Despeciation Algorithm* in chapter 3 introduces a novel initialization technique. *Machine Learning for Rare Diseases, a Case Study* in chapter 4 presents the procedures and results of institutional collaboration with *Casa dos Marcos*, where the novel technique, presented in this document, was applied to solve a real-life problem. *R Shiny Web-Application for Rare Diseases* in chapter 5 provides a quick summary of R-Shiny web-application developed in scope of this collaboration. Finally, *Conclusions and Future Work* in chapter 6 presents the main conclusions of the work.

Chapter 2

Theoretical Background

The aim of this chapter is to introduce readers that are not familiar with the field of *Machine Learning* and *Evolutionary Computation* to the set of key concepts necessary to contextualize with procedures and results presented in this document. In section 2.1 the concepts of *Machine Learning* and its two main fields - *supervised* and *unsupervised* learning - are introduced. Section 2.2 exposes the most recent sub-fields of *Evolutionary Computation*, which were operated in this work - *Genetic Programming* and *Geometric Semantic Genetic Programming* (the later in sub-section 2.2.2).

2.1 Machine Learning

Machine Learning (ML) is a vast sub-field in *Computer Science* which main objective is to provide computers (*machines*) an amazing ability (previously only reserved to intelligent life begins like Humans) - the ability to learn [26].

Although there are several conceptually different ML techniques, these can be grouped in supervised and unsupervised learning.

2.1.1 Supervised Learning

Supervised learning aims to infer a function, provided a set of labeled training examples (*training data*), which reflects underlying relationship between input features and the output (called *target*). One of major concerns of this input/output mapping is the *generalization ability* - build a model that reflects general and persistent pattern between input and output features that will work not only for known information (stored in *training data*), but also for another, previously unseen (*unseen data*). Depending on level of measurement of the target, the learning task can be defined as regression (for continuous values) or classification (for discrete values).

2.1.2 Unsupervised Learning

Contrarily to supervised learning, where the target is known, in unsupervised learning *training data* do not have target values. Since the target is not known, the main concern of unsupervised learning is of inferring a function which describes hidden structure from unlabeled training data (a.k.a. clustering).

2.2 Genetic Programming

Genetic Programming is the most recent sub-field in Evolutionary Computation. It aims to evolve computer programs, among the space of all possible computer programs, which can solve a given optimization problem [20].This population-based process follows principles of *Darwin's Theory of Evolution* [12], and can be summarized by the following five steps, which are iterated during execution of the algorithm:

1. reproduction;
2. ability to adapt to environment (selection);
3. inheritability;
4. variability;
5. competition and survival;

Figure 2.1 provides visual representation of the process held by evolutionary algorithms, including GP.

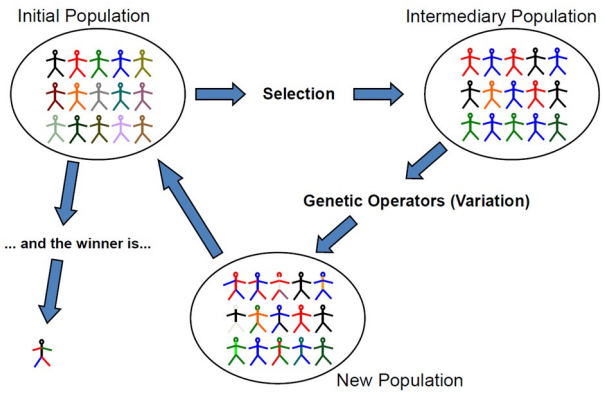


Figure 2.1: Graphical representation of iterative work-flow of an evolutionary algorithms, guided by principles of *Darwin's Theory of Evolution*

As we can see from figure 2.1, evolution of solutions starts with initialization of individuals that later will be evolved. Then, by applying selection mechanisms and

genetic operators, new individuals are created and transited to the next generation. This process iterates until reaching certain stopping criteria (maximum number of generations, for example).

In GP, individuals are computer programs composed (in a particular way) by specific elements of a given programming language. Commonly, individuals are represented in a tree-based structure. Consider the following two sets of program elements used to compose a computer program: $terminals = \{x1, x2, x3\}$ and $functions = \{+, -, *, /\}$. A possible individual resulting from composition of such elements is represented in figure 2.2.

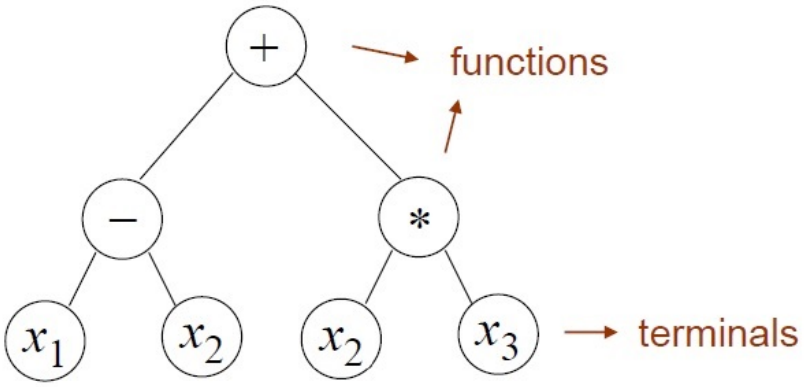


Figure 2.2: Example of a tree-based representation of a GP individual (taken from [25])

In other words, an individual evolved by means of GP can be a mathematical function like $f(x1, x2, x3) = x1 - x2 + x2 * x3$.

2.2.1 Initialization of the Population in Genetic Programming

Population initialization is the first step of an evolutionary algorithm. In GP, initialization consists in creation of almost random functions that later will be evolved. Koza described three initialization methods: Grow, Full, Ramped Half-and-Half [20].

As we already saw, individuals, represented in a tree-based structure, are composed by two sets: terminals (T) and functions (F). Starting for the root of the tree, elements (called *nodes*) are combined one after another in specific manner, until reaching a pre-defined tree depth (d). The choice of nodes among two sets, although based on a random event, follows a specific approach.

2.2.1.1 Grow Method

Starting at the root, the first node to be selected comes from F (with uniform probability). This avoids having trees composed by one single terminal. For depth between 1 and $d - 1$, nodes are selected regardless the set (with uniform probability). Once a

branch contains a terminal node, it is ended even if the maximum depth d has not been reached. Finally, nodes at depth d are chosen from T (with uniform probability).

By allowing selection of nodes regardless the set, trees are likely to have irregular shape, i.e. to contain branches of different lengths.

2.2.1.2 Full Method

Unlike Grow method, the Full method only chooses nodes from F until the maximum depth. After reaching maximum depth d, it chooses only nodes from T. The result is that every branch of the tree goes to the full maximum depth, which results in bushy trees.

Figure 2.3 represents how an individual could look like if it would be initialized by means of Full or Grow method.

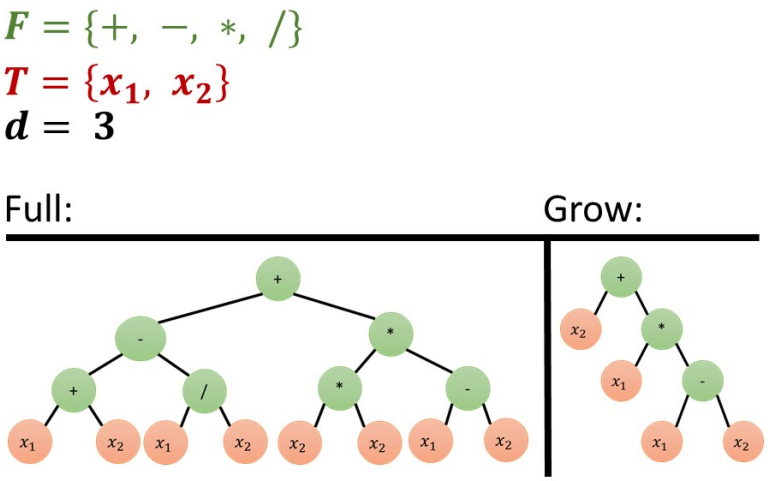


Figure 2.3: Visual representation of two possible individuals initialized by means of Full and Grow methods.

2.2.1.3 Ramped Half-and-Half Method

John Koza introduced Ramped Half-and-Half method in order to overcome drawbacks of previous initialization methods [20]. He pointed that population initialized with Grow or Full methods produces trees that are too similar, which floors the diversity in GP populations. The proposed technique is summarized my means of pseudo-code in figure 2.4

Let d be the maximum depth parameter and P the population size:

1. divide P in d groups;
2. in each group (g_i), set distinct maximum depth equal to 1, 2, (...), $d-1$, d;
3. $for(i = 1; c <= n; c++)$:
   a) initialize one half of individuals in group g_i with Full method;
   b) initialize one half of individuals in group g_i with Grow method;

Figure 2.4: Pseudo-code for Ramped Half-and-Half initialization method.

2.2.2 Geometric Semantic Genetic Programming

Even though the term semantics can have several different interpretations, it is a common trend in the Genetic Programming (GP) community (and this is the terminology we adopt here) to identify the semantics of a solution with the vector of its output values on the training data [27, 37]. Under this perspective, a GP individual can be identified by a point (its semantics) in a multidimensional space that we call semantic space (where the number of dimensions is equal to the number of observations in the training set, or fitness cases). The term Geometric Semantic Genetic Programming (GSGP) [35] indicates a recently introduced variant of GP in which traditional crossover and mutation are replaced by so called *geometric semantic operators*, which exploit semantic awareness and induce precise geometric properties on the semantic space. *Geometric semantic operators*, introduced by Moraglio et al. [27], are becoming more and more popular in the GP community [37] because of their property of inducing an unimodal error surface (characterized by the absence of locally suboptimal solutions on training data) on any problem consisting in matching sets of input data into known targets (like for instance supervised learning problems such as regression and classification). The proof of this property can be found in [27], while an intuitive explanation of it can be found in [35]. Here, we report the definition of *geometric semantic operators* as given by Moraglio et al. for real functions domains, since these are the operators we will use in the experimental phase. For applications that consider other types of data, the reader is referred to [27].

Geometric semantic crossover generates, as the unique offspring of parents $T_1, T_2 : \mathbb{R}^n \to \mathbb{R}$, the expression

$$T_{XO} = (T_1 \cdot T_R) + ((1 - T_R) \cdot T_2),$$

where T_R is a random real function whose output values range in the interval $[0, 1]$. Analogously, *geometric semantic mutation* returns, as the result of the mutation of an individual $T : \mathbb{R}^n \to \mathbb{R}$, the expression

$$T_M = T + ms \cdot (T_{R1} - T_{R2}),$$

where T_{R1} and T_{R2} are random real functions with codomain in $[0,1]$ and ms is a parameter called mutation step. Moraglio and co-authors show that geometric semantic crossover corresponds to geometric crossover in the semantic space (i.e. the point representing the offspring stands on the segment joining the points representing the parents) and geometric semantic mutation corresponds to box mutation on the semantic space (and thus induces a unimodal error surface on the above mentioned types of problem). As Moraglio and co-authors point out, *geometric semantic operators* create much larger offspring than their parents and the fast growth of the individuals' size rapidly makes fitness evaluation unbearably slow, making the system unusable. In [7], a possible workaround to this problem was proposed, consisting in an implementation of Moraglio's operators that makes them not only usable in practice, but also very efficient. This is the implementation used in this work.

Another recognized drawback of GSGP consists in the potential weakness of geometric semantic crossover. In fact, given its geometric properties (as we said above, this crossover generates an offspring whose semantics stands on the segment joining the points representing the parents in the semantic space), geometric semantic crossover has the possibility of generating the global optimum only if its semantics are "surrounded"by the semantics of the individuals in the population. In more formal terms, and using the terminology of [9, 28], geometric semantic crossover has the possibility of generating a globally optimal solution only if this solution lays within the semantic *convex hull* identified by the population. The need of overcoming this drawback has lead to several methods to properly initialize a population of GSGP, like for instance the one presented in [30].

Chapter 3

Evolutionary Demes Despeciation Algorithm

This chapter presents the new initialization method for GP individual and results of its comparison with current state-of-the-art algorithm. Section 3.1 describes the EDDA system. Section 3.2 presents our experimental study conducted over six symbolic regression problems, aimed at comparing EDDA with a GSGP system that uses a traditional initialization method like the Ramped Half-and-Half algorithm (RHH). In Section-3.3, we analyze and critically discuss those results, offering an interpretation for their good quality.

3.1 EDDA: The Proposed Initialization Method

EDDA works like a canonical GSGP system, with the only difference that the population, instead of being initialized with a "classical"initialization method like for instance Ramped-Half-and-Half (RHH) [20], is seeded using the best individuals in other populations (demes), after that these demes have been evolved for some generations.

The EDDA system has some clear analogies with the multi-layered GP system proposed by Lin and co-workers in [22, 23]. Nevertheless, EDDA has also some important differences compared to that system: for instance, Lin's system is specialized for multi-class classification, while EDDA can in principle be used to tackle any kind of problem (indeed, we use symbolic regression applications as test problems in this work). More specifically, Lin's system uses populations in different architectural layers to improve the ability of the system to discern the different classes. Furthermore, Lin's system does not use population layers for initializing the population, which is instead the main characteristic of EDDA.

We studied several variants of the EDDA system, with several types of demes,

left to evolve for several different number of generations. In the continuation of this paper, these variants will be indicated with the notation EDDA-n%, meaning that n% of the population was initialized using individuals from GSGP demes, while the remaining $(100 - n)$% was initialized using standard GP demes. For instance, EDDA-75% initializes the GSGP population of 100 individuals by evolving 75 GSGP demes and 25 standard GP demes. The initial population is composed by the best individuals found by all these demes.

Given a natural number n included between 0 and 100, EDDA-n%, where demes are left to evolve for m generations, works like in the pseudo-code of Figure 3.1.

EDDA-n% (evolving demes for m generations):

1. Create an empty population P of size N;
2. Repeat $N * (n/100)$ times:
   a) Create an empty deme;
   b) Randomly initialize this deme using classical initialization algorithm (RHH used here);
   c) Evolve individuals from 2.b) for m generations using GSGP;
   d) After finishing 2.c), select the best individual from the deme and store it in P;
3. Repeat $N * (1 - n/100)$ times:
   a) Create an empty deme;
   b) Randomly initialize this deme using classical initialization algorithm (RHH used here);
   c) Evolve individuals from 3.b) for m generations using standard GP;
   d) After finishing 3.c), select the best individual from the deme and store it in P;
4. Retrieve P and use it as the initial population of GSGP

Figure 3.1: Pseudo-code of the EDDA-n% system, in which demes are left to evolve for m generations.

In that pseudo-code, points 2.b), 2.c), 3.b) and 3.c) implement the *evolution of demes*, while points 2.d) and 3.d) implement the *despeciation* phase. In the former step, the different demes evolve independently; in the latter phase, individuals coming from different demes, and thus from different evolutionary dynamics and histories, are joined together in a new population (P in the pseudo-code). To evolve this new population, GSGP is preferred over standard GP because for almost all problems used in this paper (all except Istanbul), we know that GSGP outperforms standard GP [35].

The rationale behind the initialization method used by the EDDA system, based on the evolution of demes, is that it should generate an initial population composed by diverse, and at the same time good quality, genetic material. In fact, each individual in the initial GSGP population comes from a different evolution history, performed in a different deme. Since each individual in the initial GSGP population was the best individual in its deme, good quality should be ensured. Another source of diversity

is given by the fact that not all the demes run the same GP system. In this paper, demes running GSGP and demes running standard GP are used, but nothing prevents us from using other GP variants for evolving the different demes in the future, with the objective of fostering even more diversity. Our ambition is to give the EDDA system the ability of evolving individuals that are, at the same time, not as large as the ones of GSGP (and possibly of a contained size, like it happens in some versions of standard GP), but with the same good quality and generalization ability as the ones usually evolved by GSGP. Finally, a major advantage of this initialization method relies on the fact that it can be used on top of any GP version (not necessarily GSGP as in this paper), making it a very general and promising approach.

3.2 Experimental Study

3.2.1 Test Problems

In our experimental study, six real-life symbolic regression problems were considered. The first three of them (Bioavailability –%F–, Plasma Protein Binding level –PPB–, and Toxicity –LD50) are problems from the drug discovery area and their objective is to predict the value of a pharmacokinetic parameter, as a function of a set of molecular descriptors of potential new drugs. The remaining three problems are the Concrete problem, whose objective is to predict the concrete strength as a function of some observable characteristics of the material, the Energy problem, whose objective is to predict the energy consumption in particular geographic areas and in particular days, as a function of some observable features of those days, including meteorologic data, and the Istanbul problem, whose objective is to predict returns of the Istanbul Stock Exchange as a function of seven other international indexes. Table 3.1 reports, for each one of these problems, the number of features (variables) and of instances (observations) in the respective datasets. The table also reports a bibliographic reference for each one of the datasets, where the reader can find a more detailed description of these problems.

Table 3.1: Description of the test problems. For each dataset, the number of features (independent variables) and the number of instances (observations) have been reported.

Dataset	# Features	# Instances
Bioavailability (%F) [2]	241	206
Protein Plasma Binding Level (PPB) [2]	626	131
Toxicity (LD50) [2]	626	234
Concrete [6]	8	1029
Energy [8]	8	768
Istanbul [1]	7	536

3.2.2 Experimental Settings

The objective of this experimental study is to compare EDDA with a GSGP system that uses the RHH algorithm [20] to initialize the population. In these experiments, GSGP used populations of 100 individuals, allowed to evolve for 5200 generations. We have used this number of generations because, in order to have a fair comparison, we want to ensure that all the studied systems perform the same number of evaluations (included, in particular, all the fitness evaluations that are needed to evolve the demes before despeciation in the EDDA systems). This choice is justified further in the document. Tree initialization was performed with a maximum initial depth equal to 6 and no upper limit to the size of the individuals was imposed during the evolution. The used function set was $\{+, -, *, /, sin, cos, ln, exp\}$, where / and ln were protected as in [20]. Fitness was calculated as the root mean squared error (RMSE) between predicted and expected outputs. The terminal set contained the number of variables corresponding to the number of features in each dataset. Tournaments of size 5 were used to select the parents of the new generation. As suggested in [38], the probability of applying geometric semantic crossover and geometric semantic mutation was randomly drawn at the beginning of each generation. In other words, at the beginning of each generation, we draw a random number, with uniform distribution, in $[0,1]$. Let p be that random number. In that generation, the crossover rate is p and the mutation rate is $p-1$. Again following [38], also the mutation step ms of geometric semantic mutation was randomly generated, with uniform probability in $[0,1]$, at each mutation event. Survival was elitist as it always copied the best individual into the next generation.

For each test problem, EDDA-0%, EDDA-25%, EDDA-50%, EDDA-75% and EDDA-100% were studied. For each one of these variants, we studied a version in which each deme was evolved for 25 generations and a version in which each deme was evolved for 50 generations. In each case, at the end of the evolution of each deme, the best individual in the deme was put in the initial GSGP population. The number of individuals in each deme was equal to 100. The number of generations in which the final GSGP population of EDDA was left to evolve depends on the version: in all cases, we have guaranteed that the final total number of fitness evaluations was the same for all the studied GP systems (i.e. 520000 fitness evaluation for each run).

For all the considered test problems, 30 independent runs of each studied system have been executed. In each one of these runs, the data was split into a training and a test set, where the former contains 70% of the data samples selected randomly with uniform distribution, while the latter contains the remaining 30% of the observations. For each generation of each studied GP variant, the best individual on the training set has been considered, and its fitness (or error) on the training and test sets was stored. For simplicity, in the continuation, we will refer to the former as training error and to the latter as test error or unseen error (these two last terms are considered as synonymous, and will be used interchangeably in the continuation).

3.2.3 Experimental Results

The obtained results are reported in Tables 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7 (respectively for the %F, PPB, LD50, Concrete, Energy and Istanbul datasets). The first column indicates the computational method. For the EDDA methods, the "EDDA-n%"notation indicates that n% of the individuals in the initial GSGP population have been generated using GSGP demes (while the remaining $(100-n)$% were generated using standard GP demes). The second column (labeled "NGens") indicates the number of generations that the demes have been left to evolve before picking up the best individual and inserting it in the initial population. The remaining three columns show the median training error, median unseen error and median depth obtained by means of a given computation method and its parametrization.

To analyze the statistical significance of these results, a set of tests has been performed. The Kolmogorov-Smirnov test has shown that the data are not normally distributed and hence a rank-based statistic has been used. The Wilcoxon rank-sum test for pairwise data comparison has been used under the alternative hypothesis that the medians of a "EDDA-n%"method and GSGP are not equal, with a significance level $\alpha = 0.05$. In the columns reporting the median unseen error and the median tree depth, the presence of check-mark symbol near the result of "EDDA-n%"method means that the difference with GSGP is statistically significant, according to the Wilcoxon rank-sum test.

Table 3.2: *Bioavailability dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	**143.30**	168.27	6720
EDDA-0%	50	162.96	167.95 ✓	372.5 ✓
EDDA-25%	50	162.65	167.97 ✓	347 ✓
EDDA-50%	50	162.65	167.93 ✓	342.5 ✓
EDDA-75%	50	162.60	167.98 ✓	**340** ✓
EDDA-100%	50	162.62	167.94	347.5 ✓
EDDA-0%	25	147.05	167.93	3619.5 ✓
EDDA-25%	25	146.91	168.19	3639.5 ✓
EDDA-50%	25	146.85	**167.88** ✓	3614.5 ✓
EDDA-75%	25	146.86	168.05	3650 ✓
EDDA-100%	25	146.91	168.06	3635 ✓

Table 3.3: *Plasma Protein Binding Level dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	**0.02**	28.78	9831
EDDA-0%	50	16.57	27.60 ✓	411.5 ✓
EDDA-25%	50	16.38	29.11	400 ✓
EDDA-50%	50	17.23	28.64	406.5 ✓
EDDA-75%	50	17.65	**27.17** ✓	**399** ✓
EDDA-100%	50	21.40	28.14 ✓	427 ✓
EDDA-0%	25	0.05	27.99 ✓	5081.5 ✓
EDDA-25%	25	0.06	28.12 ✓	5068.5 ✓
EDDA-50%	25	0.05	27.69 ✓	5070 ✓
EDDA-75%	25	0.05	29.09	5083 ✓
EDDA-100%	25	0.06	27.80 ✓	5034 ✓

Table 3.4: *Toxicity dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	2083.30	1957.47	7148
EDDA-0%	50	1622.13	2074.63 ✓	387.5 ✓
EDDA-25%	50	**1571.42**	1458881491.66 ✓	372.5 ✓
EDDA-50%	50	1588.83	2083.87 ✓	385 ✓
EDDA-75%	50	1642.04	**1917.56** ✓	377.5 ✓
EDDA-100%	50	2047.27	2074.82 ✓	**364** ✓
EDDA-0%	25	1748.28	2094.63 ✓	4312 ✓
EDDA-25%	25	1750.98	1953.88	4353.5 ✓
EDDA-50%	25	1787.84	1968.50	4363.5 ✓
EDDA-75%	25	1779.24	4680.39 ✓	4329 ✓
EDDA-100%	25	1983.99	1929.52	4131 ✓

Table 3.5: *Concrete dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	**92.59**	105.58	6873.5
EDDA-0%	50	105.29	105.46 ✓	368.5 ✓
EDDA-25%	50	105.12	105.50	346.5 ✓
EDDA-50%	50	105.09	105.44 ✓	**340** ✓
EDDA-75%	50	105.12	105.47 ✓	349 ✓
EDDA-100%	50	105.10	**105.43** ✓	343.5 ✓
EDDA-0%	25	96.03	105.56	3697 ✓
EDDA-25%	25	95.98	105.47	3712 ✓
EDDA-50%	25	95.99	105.56	3717 ✓
EDDA-75%	25	95.94	105.59	3708 ✓
EDDA-100%	25	95.93	105.55	3716 ✓

Table 3.6: *Energy dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	**0.01**	15.16	9896
EDDA-0%	50	8.07	15.07 ✓	471.5 ✓
EDDA-25%	50	7.57	**15.05** ✓	**367** ✓
EDDA-50%	50	7.55	**15.05** ✓	367.5 ✓
EDDA-75%	50	7.58	15.07 ✓	368.5 ✓
EDDA-100%	50	7.57	15.08 ✓	368 ✓
EDDA-0%	25	0.04	15.23	4966.5 ✓
EDDA-25%	25	0.04	15.23	4957.5 ✓
EDDA-50%	25	0.04	15.20	4959 ✓
EDDA-75%	25	0.04	15.32 ✓	4953.5 ✓
EDDA-100%	25	0.04	15.17	4961 ✓

Table 3.7: *Istanbul dataset*.

Method	NGens	Med. Training Error	Med. Unseen Error	Med. Depth
GSGP	N/A	**0.9**	13.51	8518.5
EDDA-0%	50	18.39	12.64 ✓	429 ✓
EDDA-25%	50	18.13	**12.62** ✓	364.5 ✓
EDDA-50%	50	18.12	**12.62** ✓	369.5 ✓
EDDA-75%	50	18.11	12.63 ✓	362.5 ✓
EDDA-100%	50	18.12	12.65 ✓	**364** ✓
EDDA-0%	25	7.68	13.24 ✓	4078.5 ✓
EDDA-25%	25	7.57	13.24 ✓	4072.5 ✓
EDDA-50%	25	7.56	13.15 ✓	4081.5 ✓
EDDA-75%	25	7.52	13.25 ✓	4077 ✓
EDDA-100%	25	7.53	13.17 ✓	4086 ✓

From the tables, we can see that for all the studied test problems GSGP finds solutions with a smaller training error than EDDA. On the other hand, EDDA is consistently able to outperform GSGP on unseen data, at the same time also generating significantly smaller solutions. Also, we can see that, in general, using a uniform mix of individuals coming from GSGP and standard GP demes in the initial population is better than using individuals coming predominantly from one of the two types of demes. In fact, EDDA-50% is the method that returns the best results in the majority of the cases, outperformed in some cases by EDDA-75% or by EDDA-25%. Last but not least, independently from the considered test problem, evolving the demes for 50 generations, before seeding the initial GSGP populations, seems convenient compared to evolving them for 25 generations.

Figure 3.2 reports the evolution plots of the final GSGP population of EDDA, after seeding from the demes, for the versions that have returned the best results for each test problem (results highlighted in bold in tables 3.2, 3.3, 3.4, 3.5, 3.6 and 3.7). Given

that the demes in plot (a) were evolved for 25 generations, then GSGP was allowed to run for 2700 generations. The demes in the other plots were evolved for 50 generations, and thus GSGP only executed for 200 generations (in order to reach the same final total number of fitness evaluations).

More precisely, figure 3.2 reports the best results for each test problem, where the term "best results"is based on the median unseen error. More specifically, Figure 3.2 reports one plot for each studied test problem. The plot shows the best run of the best EDDA method on that problem, where the "best run"is the one which shows the smallest test error on the last generation and the "best EDDA method"is the one which shows the smallest median test error and, in the case the median test error is equal, the smallest median tree depth.

The plots in figure 3.2 (at page 20) shows that even though, as expected, the evolution on test set is less remarkable than the one on training set, EDDA is not overfitting on training data (while still improving the training error) for any of the studied test problems. In fact, no deterioration of the test error is visible. On the other hand, the final GSGP evolution in the EDDA systems generally leads to a further slight improvement of the test error, compared to the one of the individuals coming from the demes. Thin horizontal lines in the plots of figure 3.2 should help notice that the curves of the unseen error (gray curves) are, in most of the cases, decreasing.

3.3 Discussion

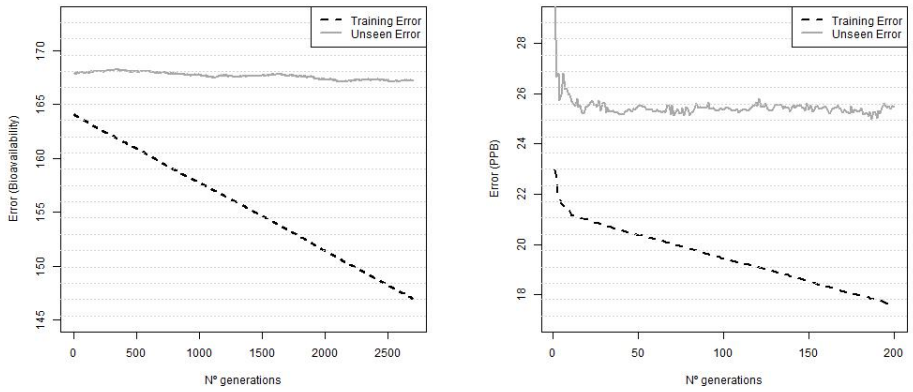
The results presented in the previous section are very satisfactory. Indeed, developing a system with better or comparable performance to GSGP on unseen data, and able to generate significantly smaller solutions, was our objective and it has been completely achieved. In this section, we give our interpretation for these good results. We believe that, in order to explain these results, one should analyse the main difference between the EDDA system and GSGP that uses RHH algorithm to initialize the population. The two methods, in fact, are similar (in the end, also the single demes in EDDA are initialized using RHH), but they have one fundamental difference: each individual in EDDA is evolved for a smaller total number of generations.

Let us consider, for instance, the case in which demes run for 50 generations (analogous considerations could be done in the case that the demes run for 25 generations). An individual is evolved for 50 generations inside its deme and then, if it is the best in its deme, it migrates in the GSGP population and it is evolved for 200 further generations. Thus, when an EDDA run terminates, each individual in the population was evolved for a total number of generations equal to 250. On the other hand, when a run of GSGP with RHH terminates, each individual was evolved for 5200 generations (in order to have the same total number of fitness evaluations for the two systems).

Considering the case where demes run for 25 generations, an individual is evolved for 25 generations inside its deme and then, if it is the best in its deme, it migrates in

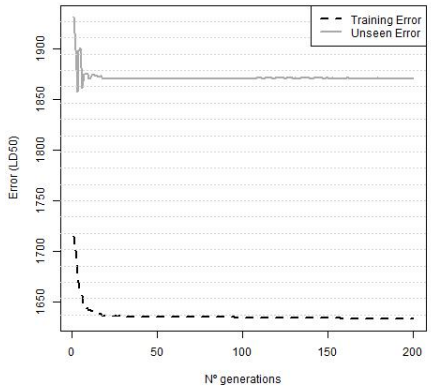
the GSGP population and it is evolved for 2700 further generations. Analogously, if demes run for 50 generations, an individual that migrates in the GSGP population, is evolved for 200 further generations. Thus, when an EDDA run terminates, each individual in the population was evolved for a total number of generations equal to 2725 or 250, depending on the number of generations in the despeciation phase. On the other hand, when a run of GSGP with RHH terminates, each individual was evolved for 5200 generations (in order to have the same total number of fitness evaluations for all the studied systems). Since the total number of generations to evolve an individual is higher using GSGP, EDDA is outperformed in the training set. Exactly for the same reason, when demes run for 25 generations the results are better on the training set compared to running them for 50 generations.

In other words, each individual in EDDA is evolved for a smaller number of generations, and the remaining computational effort is spent in generating a larger number of individuals, looking for the "good"ones. We could informally say that each individual undergoes a shorter, but "better"evolution, and the process, at least in the evolution of the demes, is much more selective (only the best individuals in the demes survive). We hypothesize that this allows EDDA to have less learning time to overfit, and we believe that this is one of the reasons for the success of EDDA. On the other hand, experimental results in our possession show us that "just"evolving individuals for 250 generations would not be enough to obtain the good results that we have obtained with EDDA. Another reason for the success of EDDA, which does not have to be forgot, is the fact that, in the despeciation phase, individuals of very good quality are joined and evolved together in the same population. Last but not least, those individuals are not only of good quality, but also very diverse from each other, coming from different evolutionary histories in different demes, and having been evolved using different algorithms. Summarizing, we identify four main reasons for the success of EDDA: (1) in EDDA, each individual is evolved for a smaller number of generations (and thus it has less opportunities of overfitting training data); (2) EDDA is a much more selective process than a usual evolutionary algorithm (only the best in each deme survives); (3) in EDDA, good individuals are joined and evolved together, thanks to despeciation; (4) besides being composed by only good quality individuals, the population of EDDA obtained after despeciation is very diverse. In synthesis, we could describe the evolution of EDDA with the maxim **"less evolution, but better evolution"**.

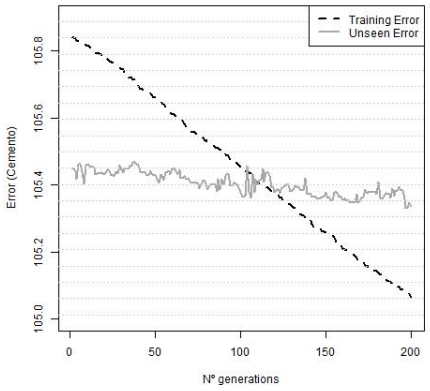


(a)

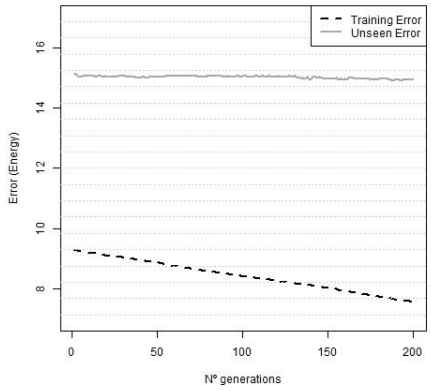
(b)



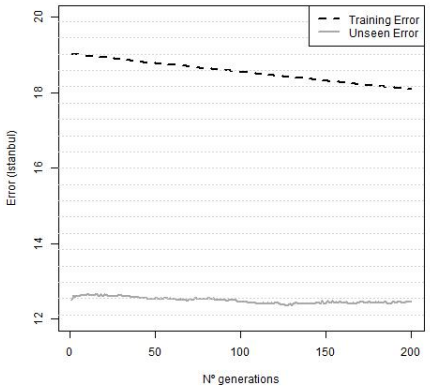
(c)



(d)



(f)



(f)

Figure 3.2: Plot (a): evolution of EDDA-50%, with demes running 25 generations, on Bioavailability problem. Plot (b): evolution of EDDA-75%, for 50 generations, on PPB problem. Plot (c): evolution of EDDA-75%, for 50 generations, on Toxicity problem. Plot (d): evolution of EDDA-100%, for 50 generations, on Concrete problem. Plot (e): evolution of EDDA-25%, for 50 generations, on Energy problem. Plot (f): evolution of EDDA-25%, for 50 generations, on Istanbul problem.

Chapter 4

Machine Learning for *Rare Diseases*: a Case Study

In this chapter, methodological framework and results of its application in scope of institutional collaboration with *Casa dos Marcos* are presented. Sections are organized as follows. In section 4.1 we describe essential needs and requirements of the entity. Section 4.2 describes the methodology we used to approach the requirements of the entity. In section 4.3 we describe the data that was provided by the entity. In section 4.4 data preprocessing phase is presented. Section 4.5 exposes descriptive analysis process and obtained results. In Section 4.6 we present predictive modeling process and obtained results.

4.1 Problem Definition

Analytics can bee seen as an intermediary tool between simple values stored in cells of a spreadsheet, the data, and the knowledge which, if extracted correctly and applied wisely, can empower decision-making processes. The global requirement of the entity, if abstracted from some problem-specific details, is not extremely different from the one of a financial institution (for example). Both need to support their decision-making processes by means of tangible and objective facts.

The aim of this section is to describe the problem which was proposed to solve and specific context where the it is inserted.

4.1.1 Pedia Suit Protocol

Pedia Suit Protocol, from now on it will be called *therapy*, is a therapeutic approach for individuals with neurological disorders such as cerebral palsy, developmental delay,

traumatic brain injury, autism and other conditions that affect motor development and/or cognitive functions. It is composed by a personalized treatment program which combines specific and intensive exercises that help minimize pathological reflexes and promote the establishment of new, correct and functional movements.

The therapy is one of therapeutic treatments performed by the entity to assist patients with rare, mostly neurodegenarative, diseases. It lasts for 4 weeks, with daily sessions of 2 or 4 hours each. Moreover, since it requires assistance of specialized physiotherapists, it represents significant costs for the patients - price ranges from 1300 to 2500 EUR, according to the number of hours per session.

4.1.2 Gross Motor Function Measure

The *Gross Motor Function Measure* (GMFM) is a clinical tool designed to evaluate change in gross motor function in children with cerebral palsy, traditionally by means of 88 measures (GMFM-88) [16]. Items on the GMFM-88 span large set of activities (from now on these will be called objective-areas): lying and rolling (17 measures), sitting (20), crawling and kneeling (14), standing (13) and walking, running and jumping (24).

GMFM international standard was adopted by the entity to motorize evolution in terms of gross motor function in patients who attend the therapy. Each patient is evaluated trough GMGM-88 exactly twice: right before and after the therapy. All 88 measures are recorded in ordinal scale ranging form 0 to 3 (inclusively), where:

- 0 means "does not initiate";
- 1 means "initiates";
- 2 means "partially completes";
- 3 means "completes";

At the end, global assessment measures are calculated: one total and five local scores (particular to each objective-area), measured in percentage points.

4.1.3 List of Requirements

Provided the context where the problem is inserted, a detailed list of requirements formulated with the entity is proposed to characterize it. As result of a meeting, where careful and objective deliberation took place, the following list of requirement was agreed:

- understand if, in general terms, the therapy has a positive effect on the patients;
- identify if there is variability among GMFM-88 measures before/after the therapy;

- identify the most volatile GMFM-88 measures before/after the therapy;
- if therapy has an effect, foresee the expected outcome of the therapy;
- identify which variables evidence higher influence on the outcome of the therapy.

An important aspect was taken in consideration during collaboration: since every patient is, in some sense, "unique", he/she is expected to show "unique" GMFM-88 assessment as well receiving a specialized therapeutic approach. Nevertheless, according requirement of the entity, this study will not focus on discrimination between different patient-adapted therapies neither their (patients) diagnosis (which, probably, can be inaccurate or not available at all). The therapy must be seen as a whole process, abstracted from therapeutic details and adjusted to the needs of every (special) patient.

4.2 Methodology and Soft-Ware

The nature of this collaboration is assumed to be long-lasting and recurrent. As such, generalized methodology implemented in top of a flexible and reusable technological structure had to be implemented.

4.2.0.1 Methodological Approach

To address requirements of the entity, a multidisciplinary approach was taken, counseled by *SAS Data Mining* process organization (Sample, Explore, Modify, Model and Access) [29]:

- the data file provided by the entity came in an antagonistic format. It was manipulated in several ways to extract the traditional configuration of an analytical base table (ABT);
- once desirable ABT format was obtained, exploratory analysis was conducted where several visualization tools were employed to prospect the distribution, correctness and completeness of the data. Relationships, trends, and anomalies were explored to gain understanding of the data;
- provided information from exploration phase, data modification took place to impute missing, replace anomalous values, create, select, and transform input variables to foster modeling phase;
- state-of-the-art supervised-learning algorithm (EDDA) and traditional unsupervised-learning algorithm (Hierarchical Clustering) were applied to predict outcome of the therapy and provide insightful vision over patients by means of agglomerative procedures;
- assessment of modeling results was performed by evaluating *numerical* and *empirical* usefulness and reliability of the findings from data mining process.

4.2.1 Technological Framework

Methodological approach was implemented using the following a multi-software technological framework, summarized as follows:

- Java: given that EDDA is implemented in Java, development of predictive models took place using this technological solution;
- Microsoft SQL Server: in order to find the most suitable parameterization for predictive models, performance of several parameters was carefully studied. Given that model development of EDDA generates significant amount of data, regarding both initialization and main evolutionary processes, stored in unstructured .txt files, it had to be stored and analyzed in an efficient way. SQL Server was used to rapidly load unstructured information into structured tables in SQL data base using *bulk insert*. Then it was used in combination with R, explained in the following item;
- R: one of R's most incredible features is the ability to work with a variety of tools and data sources. R was used to connect to Microsoft SQL Server, so that data could be extracted directly from the database by means of SQL queries. The remaining data mining process was conducted in R.
- Microsoft Excel: Microsoft Excel is incredibly intuitive and simple tool, specially to perform explanatory analysis, tables manipulation and results organization. It was actively used in combination with R as an auxiliary tool.

Figure 4.1 provides visual representation of multi-software technological framework that was used to implement methodological approach.

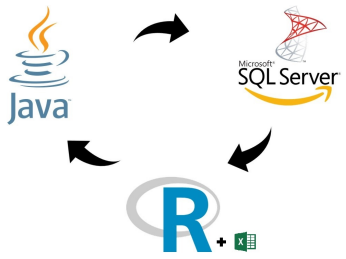


Figure 4.1: Multi-software technological framework.

4.3 Operating Data

The entity provided one data file (.xlsx) composed by twenty-seven data instances of different patients who attempted the therapy. Twenty-six of these were evaluated through traditional GMFM-88, while the remaining patient was evaluated using 66

measures (GMFM-66). The former was excluded from the analysis in preprocessing phase. Each patient attempted the therapy at least once and, as it is expected, for each therapy there are two GMFM evaluations (one before another after).

Besides GMFM and summary indicators (5), the entity also provided socio-demographic attributes: birth date, diagnosis, number of therapies attempted and gender. A total of ninety-seven deployable attributes were provided.

4.3.1 Making Small Data Bigger

Provided a data set with only twenty-seven data instances, it is a risky task to develop a predictive model with, at least, ninety-seven predictors. Given this limitation, the following decision was taken: instead of considering one data instance per patient, consider one per therapy. As a result, forty-one data instances became available for the analysis.

The fact that one patient can attend more than one therapy, could lead to a fair thought: it looks inaccurate to consider different therapies of the same patient as independent records. However, the problem here is not so linear.

Most of the time, patients present neurodegenerative diseases - hereditary and sporadic conditions which are characterized by progressive nervous system dysfunction [13]. Neurodegenerative diseases are incurable and debilitating conditions that result in progressive degeneration and/or death of nerve cells causes problems with movement or mental functioning [15].

Given neurodegenerative nature of diseases, there will be a unique and unpredictable deviation in terms of GMFM in between subsequent therapies taken by a given patient. Additionally, such patient is expected to react differently to similar therapies performed on different time-periods due to disease progression status. In other words, if a given patient attends the therapy now, the effect is expected to be different from the one performed in a year; this is likely to happen because the disease during one year will progress in unpredictable manner. Moreover, GMFM of such patient after one year is expected to vary in negative fashion.

There are other factors that influenced the decision stated in above. For example, it happens that the entity does not control patients progress in between two successive therapies which also are not regularly distributed over time.

4.4 Preprocessing Phase

According to [5], *"Data is the key to unlock the creation of robust and accurate models (...) However, data is often inadequate on its own and needs to be cleaned, polished, and molded into a much richer form"*. In fact, well-elaborated preprocessing can not only reduce modeling time, but also qualify its results.

The data provided by the entity was in an antagonistic format. Additionally, several data quality issues were identified: missing values and anomalous values (out of admissible range).

4.4.1 Set-up Analytical Base Table (ABT)

Before leveraging data preprocessing, a key aspect was carried - the spreadsheet was structured into an Analytical Base Table (ABT) format, where one record represents a single therapy. It happens that records in the spreadsheet provided by the entity were organized in an unusual way. One record was storing at least two sets of GMFM-88, beside socio-demographic information. For a patient who took one therapy, there was a record with 192 columns; for a patient who took two therapies, the record contained 378 columns; etc.

In order to correctly arrange the ABT, an iterative algorithm was developed. It decomposes every record of original spreadsheet into at least two records (one pair per therapy), preserving unique identification and replicating the set of socio-demographic variables for each new record. To maintain unique identification and ease future analysis, other variables were added: one to track if the record belongs to GMFM before or after the therapy ($beforeTherapy$), another to track order of the therapy ($gmfmOrder$). Besides this, summary measures GMFM measures were computed as in [16]: one for global score and five for local scores regarding each objective-area.

The rationale behind an automated ABT construction resides on future prospect: the nature of this collaboration is long-lasting and recurrent. In this concrete case, a generalized algorithm for transformation of a spreadsheet into an admissible ABT was implemented - and it holds for any size of the spreadsheet (any number of patients - rows, and therapies - columns), with a single constraint: preservation of the original format of the spreadsheet. The same approach was taken in consideration all along the process.

By the end of this process, an ABT containing 41 data instances and 104 features was extracted, already including 6 target features.

4.4.2 Data Quality Issues

In total, there were four missing and one out of admissible bounds values (all regarding GMFM measures). Their treatment was managed through collaborative channel: the entity was contacted with a request for clarification regarding problematic cases and all five values were replaced according to entity's information.

4.5 Descriptive Modeling

A descriptive model can be defined as an abstraction of the data that summarizes and describes its relevant features [18].

Cluster analysis is a generic name for an array of methods that partition multivariate data set in natural groups. Data instances in each of these groups are aimed to be as similar as possible to each other and as different as possible from those data instances who belong to other groups. Presently, several clustering methods were proposed with different algorithmic approaches: Hierarchical Clustering (HC), k-means, Self-Organizing Maps (SOM), etc.

In this work, we conducted a cluster analysis with intention of finding and characterizing unexpected natural groups of patients. Assuming that patients who belong to the same group will exhibit similar behavior towards therapy, detection and characterization of such homogeneous structures will allow the entity to develop therapeutic strategies personalized to particular needs of each group.

4.5.1 Variable Selection

Variable selection is an important step in cluster analysis and it is strongly tied to two concepts: *curse of dimensionality* and *discrimination ability*. According to [18], *curse of dimensionality* can be explained by the following sentence: *"the amount of data we need often increases exponentially with dimensionality if we are to maintain a specific level of accuracy (...)"*. Provided a multidimensional dataset with 104 features and only 41 data instances, the way as variable selection is conducted may strongly influence accuracy of cluster analysis. Moreover, not only the size of extracted subset matters, it should be small enough to conduct a reliable analysis, but also the quality of retained features - these must contain enough information to create natural and distinct groups of data observations which are meaningful to exist *(discrimination ability)*.

In this work, feature selection was performed based on volatility before/after the therapy measured by means of *Absolute-Rate-of-Change* (ARC). This has to do with one of requirements of the entity: identify the most volatile GMFM-88 measures before/after the therapy.

4.5.1.1 Absolute Rate of Change (ARC)

Absolute-Rate-of-Change (ARC) is defined as the sum of absolute differences between values taken before and after the therapy, divided by number of data instances.

Consider **X={0, 1, 2, 3,}** as one of GMFM-88 measures. Let **Xb** denote **X** before the therapy and **Xa** after. The first step is to create a frequency table I.1 for **X**, where **Xb** and **Xa** will represent, for each level of **X**, the number of patients recorder before and after the therapy.

Then absolute differences are computed between values of **Xb** and **Xa**, for the same level.

Finally, differences are summed, divided by number of data instances and multiplied by 100. Considering the example in table I.2:

level	Xb	Xa
0	5	6
1	0	2
2	2	1
3	34	32

Table 4.1: Frequency table for **X={0, 1, 2, 3,}**

level	Xb	Xa	|Xb − Xa|
0	5	6	1
1	0	2	2
2	2	1	1
3	34	32	2

Table 4.2: Frequency table with absolute differences

$$(\frac{1+2+1+2}{41}) * 100 \approx 14.63$$

As such, ARC for feature **X** is approximately 14.63%. Figure 4.2 plots GMFM-88 ranked according to ARC, measured in percentage.

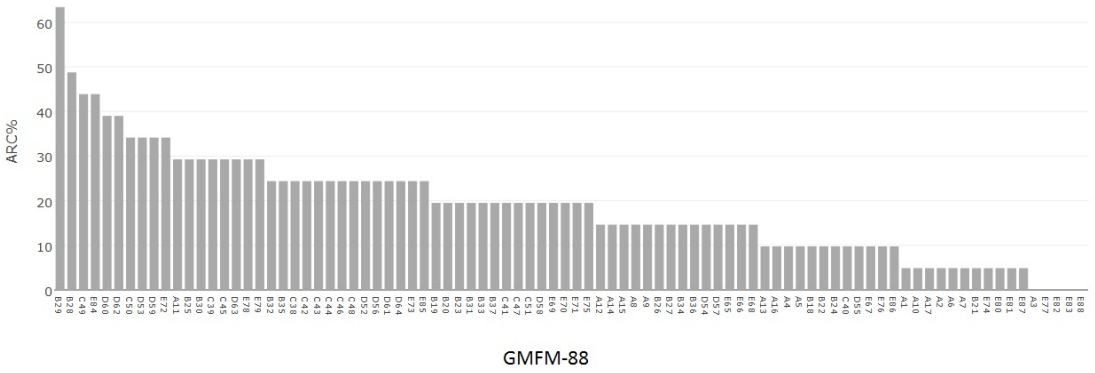


Figure 4.2: ARC(%) calculated for all GMFM-88

ARC is proposed as a simple and intuitive measure of impurity for ordinal features that are measured twice, before and after the therapy. Higher the ARC of a given feature, higher the interest it represents for clustering purpose. In other words, it is expected that features which were highly affected by therapy may present higher discrimination ability for cluster analysis.

Initially we considered a set of ten top most volatile features. From these, six were excluded based on iterative analysis of Spearman Correlation Matrix. The features set that was selected to perform cluster analysis has the following nomenclature:

- B29: can sit on the left side maintaining arms free for at least five seconds;
- C49: can kneel on the right knee maintaining arms free for at least ten seconds;

- D62: can have a controlled sit on the floor with maintaining arms free;
- E72: walks forward ten steps, carrying a "large object"with two hands;

It is worth to note that features B29 and C49 do not have apparently different behavior from their reciprocal features (B28 and C50, respectively). The reason why these variables were included instead their reciprocals were not added in feature set has to do with exclusion based on Spearman Correlation Coefficient. In other words, B29 and C49 could be practically replaced by their reciprocals, which cuts the significance of being "left"or "right".

4.5.2 Dissimilarity Measure

Provided a set of features, the next step in cluster analysis is to define how to measure multidimensional data instances in an appropriate way. To assess dissimilarity between data instances measured in ordinal scale, Spearman Dissimilarity was used (SD) as described in [32].

Consider $\mathbf{s(i, j)}$ as Spearman's Rank-Order Correlation Coefficient between clusters **i** and **j**. Spearman Dissimilarity, defined as $\mathbf{d(i, j)}$, will be computed as:

$$d(i, j) = 1 - |s(i, j)|$$

4.5.3 Clustering Algorithm

In this work we used Hierarchical Clustering (HC), due to its algorithmic characteristics and ABT at hand - which was small enough to operate the distance matrix in memory. The characteristics that attracted us to choose HC are the following:

- **flexibility of working in non-Euclidean spaces** by specifying different distance measures;
- **possibility to handle clusters of varied shapes** by specifying different linkage methods;
- **no need to specify initial number of clusters** (k);
- **the method is deterministic**, i. e. independent on a random event. Whenever HC algorithm is executed, under the same parameters, the result is going to be the same;
- solutions with many clusters are nested within those with few. This means that **observations do not change configuration when cutting tree with different** k as it happens with, for example, k-means;

HC iteratively agglomerates or divides clusters in multidimensional space. Consider SD between any two clusters, $\imath$ and $\jmath$, defined as $d(\imath,\jmath)$. Let **D** be the matrix that stores pair-wise dissimilarities between **n** clusters. Let **link** be a pre-defined linkage method. The agglomerative approach can be described as in figure 4.3.

1. for(**c=1**; **c<n**; **c++**):
   a) choose **min** ($d(\imath,\jmath)$) from **D**;
   b) merge clusters $\imath$ and $\jmath$ into a single new cluster **k**;
   c) considering **link**, calculate a new set of dissimilarities from **k** to remaining clusters and store them in **D'**;
   d) **D** = **D'**;

Figure 4.3: Pseudo-code for Hierarchical Clustering Algorithm

HC allows perform several linkage methods and results of cluster analysis may strongly depend on the chosen method. For example, *Single Linkage*, a.k.a. nearest neighbor, defines the distance between two clusters by their two closest members. Since linkage is local, a long chain of points can be merged to the same cluster without regard to the overall shape. Contrarily, *Complete Linkage*, a.k.a. furthest neighbor, defines the distance between two clusters by their two farthest members. Since linkage is not local, this method tens to create well separated and compact clusters with small diameters.

hclust function from *stats* package in *R* [19], provides implementation for eight different linkage methods. To select most appropriate one, two-way approach was taken. First, *Cophenetic Correlation Coefficient* (CCC) was computed for all eight methods. Second, methods which exhibited highest CCC with original data observations were compared by means of their *dendrograms*.

4.5.3.1 Cophenetic Correlation Coefficient (CCC)

Cophenetic Correlation Coefficient (CCC) can be defined as a linear correlation coefficient between the *Cophenetic Dissimilarity* (CD) obtained from the clustering solution and the original dissimilarities between data instances (we use Spearman Dissimilarity) [10]. CD between two observations $\imath$ and $\jmath$ is computed from the clustering *dendrogram* by height of the link at which those two observations were first merged.

Informally, CCC is a measure of how faithfully HC algorithm represents the dissimilarities among observations. In this work, CCC is used to assess performance of different linkage methods.

Formally:

$$\frac{\sum_{\imath<\jmath}\left(d(\imath,\jmath)-\bar{d}\right)(z(\imath,\jmath)-\bar{z})}{\sqrt{\sum_{\imath<\jmath}\left(d(\imath,\jmath)-\bar{d}\right)^2\sum_{\imath<\jmath}(z(\imath,\jmath)-\bar{z})^2}},$$

where $d(\imath, \jmath)$ is SD between original data instances $\imath$ and $\jmath$, $z(\imath, \jmath)$ is CD between data instances $\imath$ and $\jmath$, $\bar{d}$ and $\bar{z}$ are average values of SD and CD respectively.

4.5.4 Analysis and Validation of Results

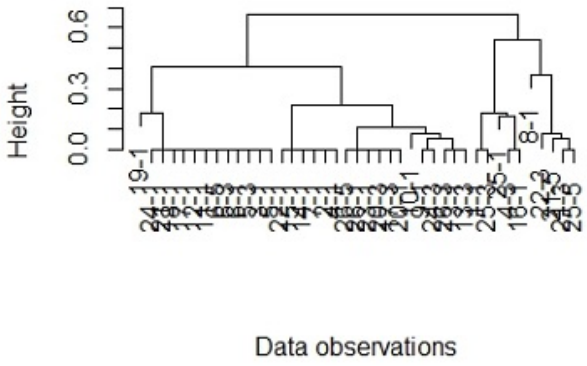
Table 4.3 presents accuracy of eight linkage methods available in *hclust* function [19], sorted in descending order of of CCC. By analyzing it, one can observe that *Average*, *Median* and *Ward* linkage methods exhibit the highest accuracy.

Method	**CCC**
Average	0.77
Median	0.74
Ward	0.73
Centroid	0.69
Complete	0.68
Mcquitty	0.67
Ward (2)	0.66
Single	0.60

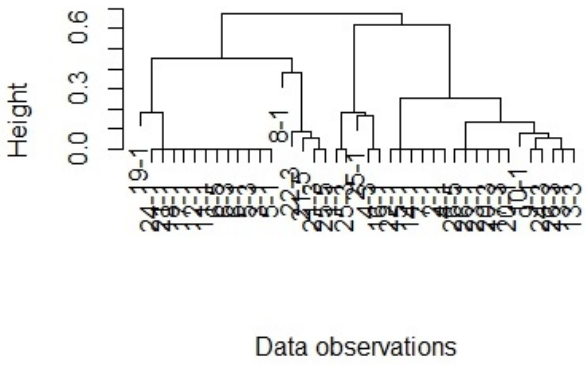
Table 4.3: Accuracy of eight HC linkage methods measured by means of CCC (results are presented in descending order)

Figure 4.4 (next page) exhibits *dendrograms* of three most accurate linkage methods.

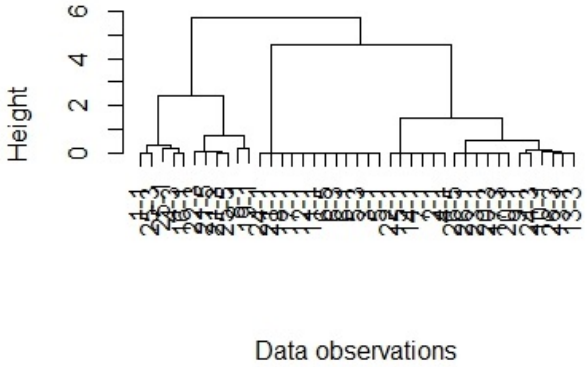
From the analysis of figure 4.4, although *Ward* linkage method exhibits third highest CCC, by looking to its *dendrogram* one can argue that it provides purer clusters. The following statement can be highlighted by the analysis of figure 4.5, which exhibits multidimensional representation of data instances (patients) by means of a heat map.



(a)



(b)



(c)

Figure 4.4: Plot (a): *Average* linkage method *dendrogram*. Plot (b): *Median* linkage method *dendrogram*. Plot (c): *Ward* linkage method *dendrogram*.

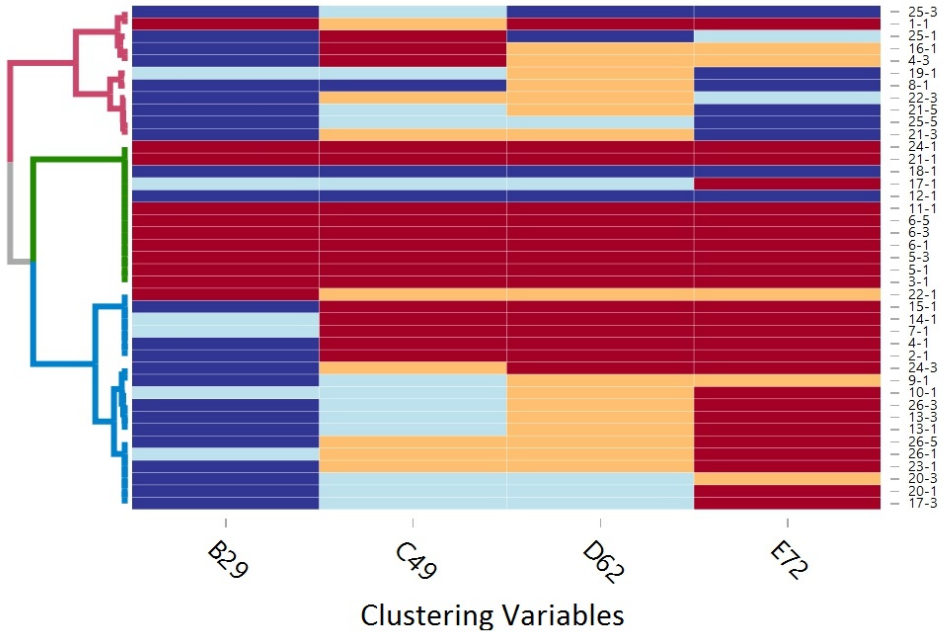


Figure 4.5: Heat map for clustering with *Ward* linkage method.

From the analysis of heights of the *dendrogram* produced by HC using *Ward* linkage method, it is clear that there are expected to be three natural groups of data instances (in our context, patients). Table 4.4 summarizes each group of patients by presenting the *mode* of every clustering variable.

Cluster ID	B29	C49	D62	E72
1	3	2	1	3
2	3	2	1	0
3	0	0	0	0

Table 4.4: Summary statistics of three clusters of patients (mode).

4.5.4.1 Characterization of Clusters

Provided figure 4.5 and table 4.4, broad description of patients belonging to each cluster can be provided.

Patients in cluster 1 can be characterized as the most fit group - for B29 and E72 they "complete"the measurements. The worst GMFM is D62, for which the majority of patients in the cluster only "initiate"the measurement. The situation is not as poor as with C49 where the majority of them "partially complete"the measurement.

Patients in cluster 2 can be characterized as the second most fit group. Compared to patients in group 1, they present equal performance in regard to B29, C49 and D62 measurements. The only difference is in regard E72 where the majority of patients "does not initiates"the measurement.

Patients in cluster 3 can be characterized as the least fit group. The majority of patients who belong to this group present the worst performance for all GMFM in consideration - they "do not initiate"the measurement for B29, C49, D62 and E72.

4.6 Predictive Modeling

The aim of a predictive model is to foresee a value of one variable (*target*) based on known values of other variables. In classification, the variable being predicted is categorical, while in regression the variable is quantitative [18].

In this section we expose details regarding practical application of EDDA, to fulfill entity's requirement, and its results. Given that the outcome of therapy can be assessed trough six possible perspectives, task of predicting its expected outcome turned out to be a one-to-six *matrioska*. In other words, to predict the outcome of therapy as a whole, six different models had to be deployed: one to predict total score and five to predict the score for each objective-areas. These models present set of relevant advantages in terms of practical application. More than simply providing prediction of a value:

- these may be a reference for efficiency of the therapy. Assuming that models provide accurate results, if, after attempting the therapy, the final evaluation of a given patient will highly differ from expected values, this may suggest that therapeutic approach at hand (its intensity, aim, coordination, etc.), was inadequate for that particular patient;
- hypothesizing lack of internal resources to address needs of all the patients. Models may help to decide which of them may attempt the treatment primarily. For example, give priority to those patients who may be in a worse condition and whose expected rehabilitation will be higher;
- models may be used as a simple informative tool, answering a very simple question: "will this treatment help my relative and by how much?". People who take their relatives to such therapy have hope in improvement. This hope can be partially concertized by providing characterization of expected values for the improvement, after the therapy.

This section is further divided in two sub-sections: Exploratory Analysis in subsection 4.6.1, which describes cross-validation technique and EDDA parameterization, and Prediction in sub-section 4.6.2, which describes construction of deployable predictive models.

4.6.1 Exploratory Analysis

In order to accurately solve an optimization problem, EDDA has to be tuned in a proper way. Technically, the algorithm has to executed several times and with using different parameters prior to building the final individual in order to conclude:

- for the initialization:
  - proportion of GSGP individuals;
  - maturation period (number of generations needed to evolve demes);
- for the main evolutionary process:
  - number of generations.

4.6.1.1 Experimental Settings

In our benchmarks (presented in sub-sub-section 4.6.1.2), tree initialization was performed with a maximum initial depth equal to 5 and no upper limit to the size of individuals was imposed during the evolution. Operating function set was $\{+, -, *, /, sin, cos, \sqrt{}\}$, where $/$, $\sqrt{}$ and ln were protected as described in sub-section 3.2.2. Fitness was calculated as *Root Mean Squared Error* (RMSE) between predicted and expected outputs. The number of variables in terminal set was corresponding to number of features. Tournament selection pressure of 10% was used to select the parents. Similarly to subsection 3.2.2, the probability of applying geometric semantic crossover and geometric semantic mutation was randomly drawn at the beginning of each generation. The mutation step was randomly generated, with uniform probability in $[0, 1]$, at each mutation event. Survival was elitist.

For each benchmark, EDDA-0%, EDDA-25%, EDDA-50%, EDDA-75% and EDDA-100% were studied. For each one of these variants, we studied a version in which each deme was evolved for 5, 10, 20 and 40 generations. In each case, at the end of the evolution of each deme, the best individual in the deme was put in the initial population of the main algorithm.

For all the considered algorithm executions, populations of 200 individuals were used (both in initialization and main algorithm), and 41 independent runs of EDDA were performed. This value is justified in next paragraph. Given restricted number of data instances and high dimensionality of the problem, each initial population contained a set of 98 individuals only composed by one distinct terminal - one input feature.

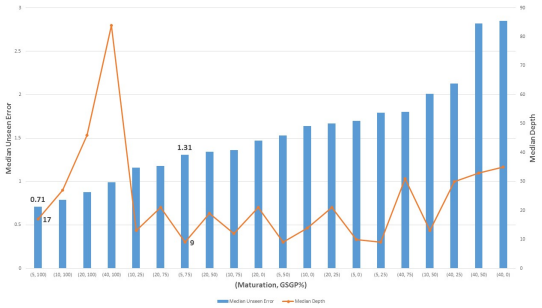
Cross-Validation.

The particular features of current problem leaded to take a special decision regarding cross-validation method. Given that there were only 41 data instances, Leave-One-Out (LOO) method was applied. More concretely, at the beginning of each run one different data instance is left out to assess generalization, while the remaining $n-1$ are used to train the model. That is why each algorithm execution was iterated for 41 runs.

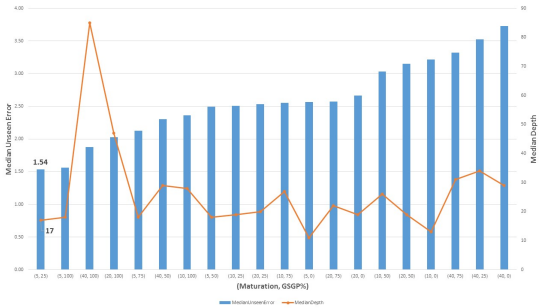
4.6.1.2 Benchmarking Results

Under concretely defined experimental environment, twenty different benchmarks were conducted for each one of six targets. In total, 120 different benchmarks were performed.

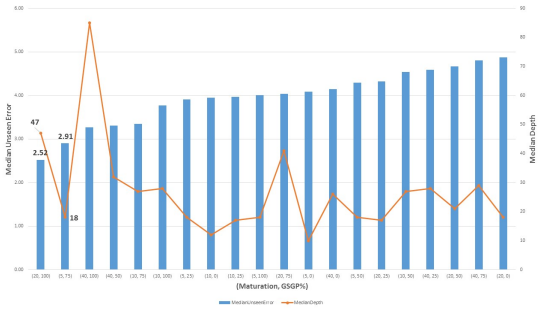
Figure 4.6 resumes, for each of six targets, twenty conducted benchmarks to select initialization parameters. One sub-figure represents results of twenty benchmarks for a given target. For every benchmark, 41 runs were executed. The best individual at the end of each run was described by means of its depth and unseen error. Then, every benchmark was summarized through *median* depth and unseen error over 41 runs. The blue bars represent *median* unseen error of best individuals at the end of 41 runs on y axis with the scale on the right. The orange line represents their corresponding *median* depth,on y axis with the scale on the left. Plots are ordered by *median* unseen error.



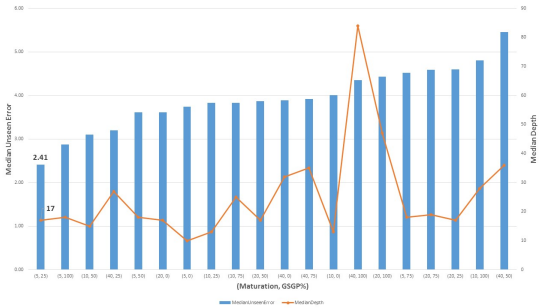
(a) Benchmark for objective-area A



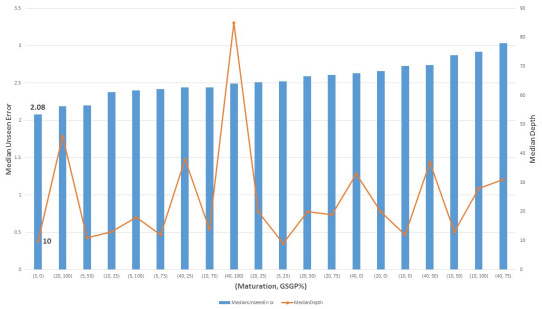
(b) Benchmark for objective-area B



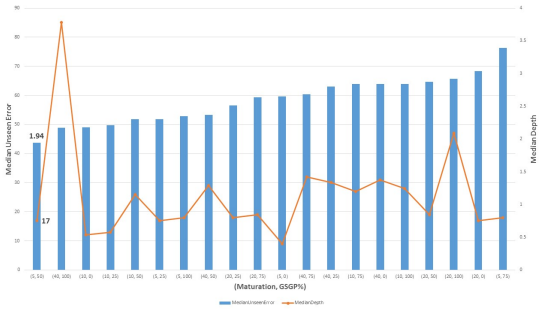
(c) Benchmark for objective-area C



(d) Benchmark for objective-area D



(e) Benchmark for objective-area E



(f) Benchmark for global score

Figure 4.6: Plot (a): performance assessment of 20 different parameter-sets for prediction of total score for objective-area A. Plot (b): assessment of parameter-sets for objective-area B. Plot (c): assessment of parameter-sets for objective-area C. Plot (d): assessment of parameter-sets for objective-area D. Plot (e): assessment of parameter-sets for objective-area E. Plot (f): assessment of parameter-sets for prediction of global score.

From the analysis of figure 4.6, the following parameterization was concluded:

- based on analysis of the plot (a), it was decided to opt parameterization provided by seventh benchmark (5, 75) due to its noticeably lower *median* depth in regard to parameterizations with better generalization.

- from plot (b), the parameterization provided by first benchmark (5, 25) was chosen because it exhibited best combination of summary statistics - lowest *median* unseen error and depth.

- from plot (c), the parameterization provided by second benchmark (5, 75) was chosen since it exhibited significantly lower *median* at almost no penalty in terms of generalization.

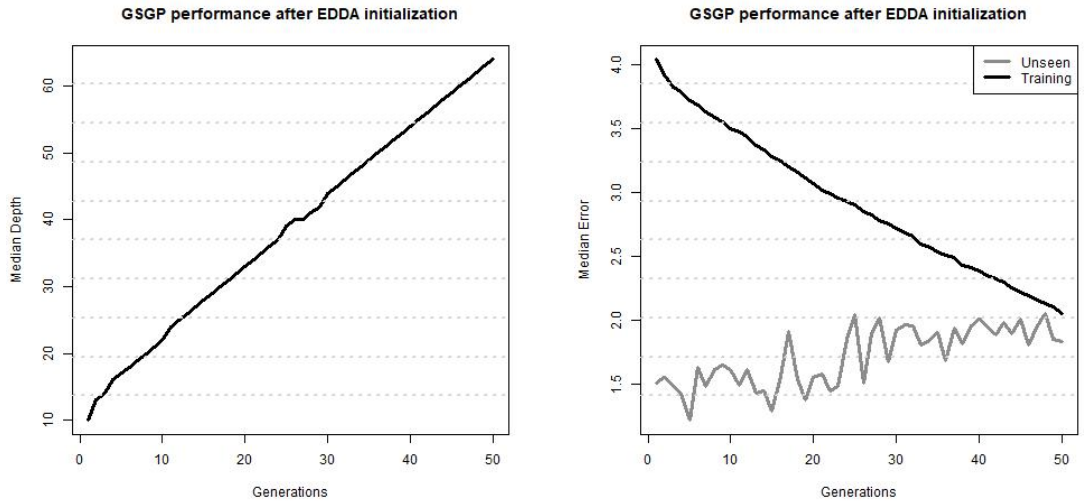
- from plot (d), the based on analysis of the plot, it was decided to opt parameterization provided by the first benchmark (5, 25) since it exhibited the best combination of summary statistics.

- from plot (e), the parameterization provided by first benchmark (5, 0) was chosen because it exhibited the best combination of summary statistics.

- based on analysis of the plot (f), it was decided to opt parameterization provided by the first benchmark (5, 50) since it exhibited the best combination of summary statistics.

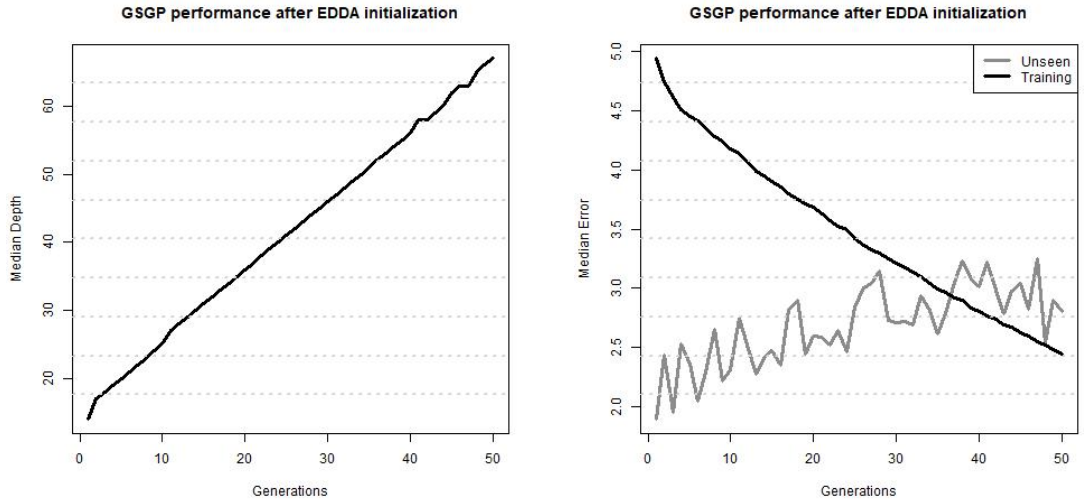
After selecting parameterization for initialization, detailed analysis on the evolutionary process was conducted. Figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12 provide visualization of the main evolutionary processes executed with chosen parameterization. Plot on right-hand-side exhibits the growth of best individuals in population, measured at each generation, by means of *median* depth calculated over 41 runs. Plot on left-hand-side exhibits the quality of best individuals in population, measured at each generation, by means of *median* training and unseen errors calculated over 41 runs. The main evolutionary process was conducted for 100 generations.
Deeper analysis of figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12 shows that construction of individuals should take place, at most at the fifth generation. Further evolution, in median terms, does not seems to be worthy because after 5-10 generations unseen error do not decrease or starts to increase.

Analysis of figures 4.7, 4.8, 4.9, 4.10, 4.11 and 4.12 concluded proper parameterization for main evolutionary process: 5 generations. Having both initialization and main algorithm parameters clearly defined, the next step was to construct individuals (predictive models) and assess their generalization ability.



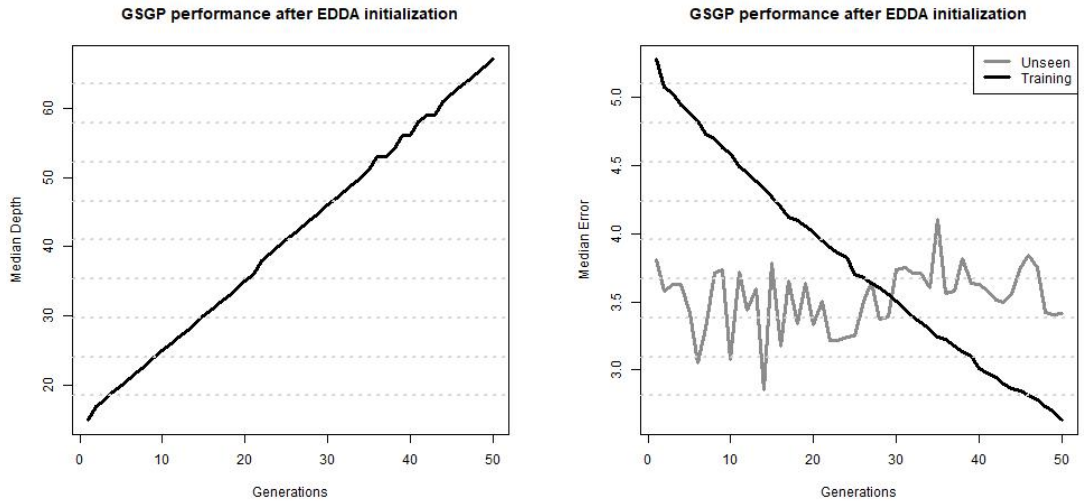
(a) (b)

Figure 4.7: Prediction of objective-area A. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.



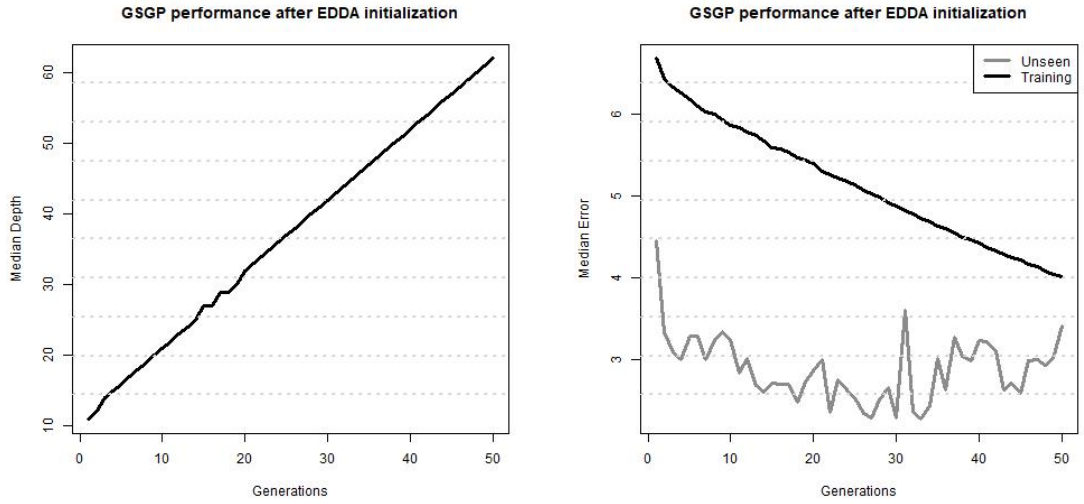
(a) (b)

Figure 4.8: Prediction of objective-area B. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.



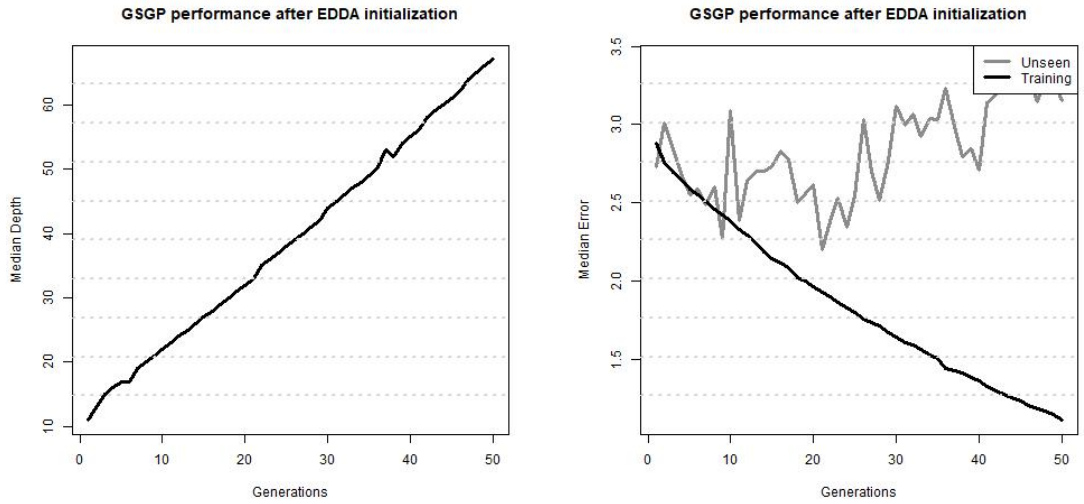
(a) (b)

Figure 4.9: Prediction of objective-area C. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.



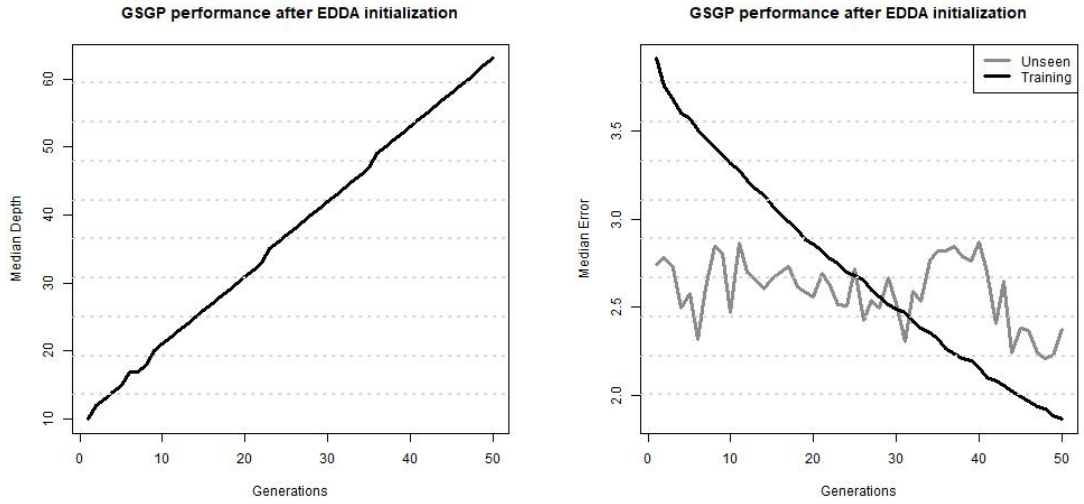
(a) (b)

Figure 4.10: Prediction of objective-area D. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.



(a) (b)

Figure 4.11: Prediction of objective-area E. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.



(a) (b)

Figure 4.12: Prediction of global score. Plot (a): growth of best individuals in the population (*median* depth). Plot (b): *median* unseen error of the best individuals.

4.6.2 Prediction

In this sub-section, six final predictive models are presented, their generalization ability and rank of input features according to predictive worth in each model.

Generalization ability was assessed by calculating *Mean Absolute Deviation* (MAD) between individuals semantics and corresponding target vector. This measure was chosen in spite of traditional RMSE in order to facilitate interpretation by the *entity*. Efficiency of solutions was assessed by calculating *Sample Standard Deviation* (SD) of MAD. Formally:

1. compute the average of absolute deviations between individuals semantics and target vector: $(|\hat{y}_0 - y_0| + |\hat{y}_1 - y_1| + |\hat{y}_2 - y_2| + (...) + |\hat{y}_n - y_n|)/n = MAD$
2. compute a measure of dispersion of absolute deviations from MAD: $\sqrt{[(|\hat{y}_0 - y_0| - MAD)^2 + (...) + (|\hat{y}_n - y_n| - MAD)^2]/(n-1)} = s$.

Worth of input features was computed as their frequency in a given individual. Features with higher frequency are expected to contribute more for prediction of the target. In other words, most frequent features are expected to be more important than less frequent.

Objective-Area A: at the end of 41 runs, representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (+ (* (+ (* (+ (- X71 (- (- X53 X91) (/ (* (^(1/2) (+ (/ (^(1/2) X28)
    X33) (sin() X23))) (+ (* X13 X1) X48)) X30))) (* C-1.0 (- (LF (/ (- (*
    (+ X74 X65) C-1.0) X95) X74)) (LF (* X24 (- X11 X57))))))) (LF (sin() X87
    ))) (* (- C1.0 (LF (sin() X87))) (+ (+ (- (/ (+ X5 C-0.5) X53) (+ (- (
    cos() X57) X77) (+ X61 (- (/ (/ (- X32 X53) (^(1/2) (- (- X60 X91) (*
    X90 C0.75)))) (sin() X7)) X91)))) (/ (/ X57 (- (- (/ X52 (sin() X85))
    X15) (* (* X2 X42) (+ (- X52 X20) X53)))) X64)) (* C-1.0 (- (LF X28) (LF
     X42)))))) (LF (cos() (* X9 C-1.0)))) (* (- C1.0 (LF (cos() (* X9 C-1.0)
    ))) (+ (+ (* (+ (/ (cos() (cos() X32)) (sin() (- (+ (sin() (+ (^(1/2)
    X53) X40)) X13) (sin() X51)))) X91) (LF (sin() X11))) (* (- C1.0 (LF (
    sin() X11))) (+ X91 (/ (+ X38 (* (cos() X52) X79)) (* X15 (sin() X13))))
    )) (* C-1.0 (- (LF X28) (LF X12)))))) (* C-1.0 (- (LF (/ (+ X89 X4)
    (^(1/2) (- X88 X71)))) (LF X22)))) (* C-1.0 (- (LF X53) (LF X2))))
```

Table 4.5 summarizes general assessment measures of the individual presented in above. When predicting total score for objective-area A, the individual, in average terms, faults in 1.76 percentage points. In 66.2 percent of cases, the real score (lets represent it by θ) will lay in $\lfloor \theta - 2.6 \, , \, \theta + 2.6 \rfloor$ prediction interval. The depth of the individual, if represented in a tree-based structure (see section 2.2), is 17.

Figure 4.13 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It can be seen that GMFM features with the highest importance in prediction of total score for objective-area A are: C51, total score for objective-area A, A11, A9, B26, C50 and D55.

MAD	s	depth
1.76	2.6	17

Table 4.5: Characteristics of individual for prediction of objective-area A

C51 stands for "the ability of crawling 10 steps to the front, hands free". A11 stands for "lifting the head up and extending the chest, resting on the forearms". A9 stands for "while lying, rolls over left the side". B26 stands for "while sitting, touches a toy placed 45 degrees right-hand-side behind and goes back". C50 stands for "maintains arms free for 10 seconds on the left knee". D55 stands for "keeps the left foot lifted for 3 seconds, holding a bench". It can be inferred that directed investment for the improvement in these measures may result in higher total score for objective-area A.

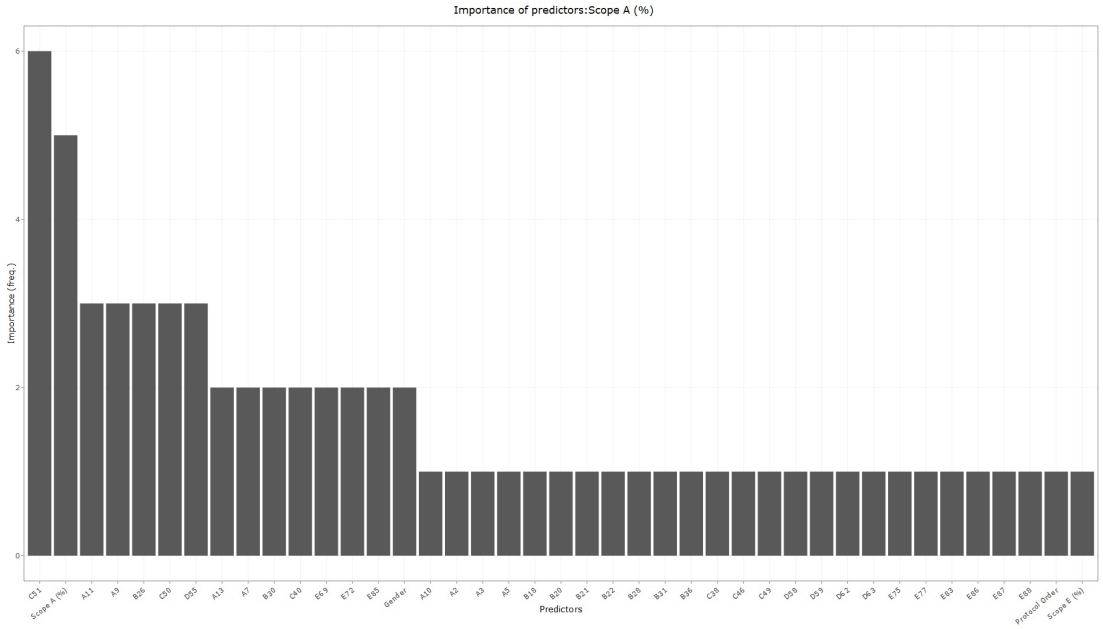


Figure 4.13: Worth plot of input features for prediction of objective-area A.

Objective-Area B: at the end of 41 runs representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (+ (* (+ (* (+ (+ (+ (^(1/2) (/ (- (- (sin() X67) X14) (cos() (^(1/2)
     X66))) (- (sin() (/ (^(1/2) (- X55 X56)) X63)) X79))) (* (cos() C0.5)
    (/ X26 (- X26 (* X28 (^(1/2) (sin() X42)))))))) X92) (* C-1.0 (- (LF (+
    (- X28 C0.25) C-0.75)) (LF (cos() X33))))) (LF (sin() (+ (* X63 X27)
    X64)))) (* (- C1.0 (LF (sin() (+ (* X63 X27) X64)))) (+ (* (+ (+ (/ (+
    (^(1/2) X67) (^(1/2) X34)) (sin() (^(1/2) (^(1/2) (sin() X34))))) (+ (
    cos() (^(1/2) X22)) (cos() (* X65 X77)))) (+ (sin() (/ X1 (cos() X83)))
     (- (+ X69 (+ X92 X13)) (+ (- X30 (+ X89 X85)) (cos() X3))))) (LF (cos
    () X29))) (* (- C1.0 (LF (cos() X29))) (+ (+ (+ (+ (+ (* X92 (LF (cos()
     (* X16 X92)))) (* (- C1.0 (LF (cos() (* X16 X92)))) X91)) (* C-1.0 (-
    (LF (- X33 X67)) (LF (+ (^(1/2) X28) (^(1/2) (- X37 X64)))))))) (* C-1.0
     (- (LF (- C0.0 (/ (* X35 X78) X3))) (LF (/ (+ X57 X86) (sin() X39)))))
    ) (* C-1.0 (- (LF (* (cos() (* (- X58 C-0.75) (sin() (^(1/2) X48))))
    X84)) (LF (* (sin() X48) X24))))) (* C-1.0 (- (LF (^(1/2) (- (sin() X80
    ) (/ X92 (^(1/2) (sin() X34)))))) (LF (+ (cos() X52) (cos() (+ X52 X49)
    ))))))))))) (LF C-0.25)) (* (- C1.0 (LF C-0.25)) (+ (* (+ (* (+ (+ (+ (+
     (* (+ (* X91 (LF X50)) (* (- C1.0 (LF X50)) (- (^(1/2) (- (+ (* X57
    X70) (+ X86 X0)) (cos() (^(1/2) X76)))) (* (* (+ (sin() X46) (- X68 X44
    )) (- (sin() X8) (- X69 X47))) (/ (cos() (- X29 X20)) (cos() (- X69 C
    -1.0)))))))) (LF (cos() X23))) (* (- C1.0 (LF (cos() X23))) (+ (* X92 (
    LF X80)) (* (- C1.0 (LF X80)) X91)))) (* C-1.0 (- (LF (sin() X24)) (LF
    X96)))) (* C-1.0 (- (LF (/ (- X15 (+ (- (sin() X25) X49) (cos() (sin()
    X13)))) (cos() (* (^(1/2) (^(1/2) X31)) X28)))) (LF (+ (cos() (sin() (+
     (- X83 X2) X86))) X84))))) (* C-1.0 (- (LF (sin() (^(1/2) (^(1/2) (cos
    () X72))))) (LF X96)))) (LF C0.5)) (* (- C1.0 (LF C0.5)) (+ (+ (- X71
    (* (^(1/2) (^(1/2) C-0.75)) (/ (* (cos() X65) (+ X8 (* X33 X60))) X59))
    ) (+ (+ (- (/ (+ (^(1/2) X13) (/ (sin() (/ (cos() X4) (- X56 X28))) (
    cos() (* X33 X92)))) (^(1/2) (sin() X37))) (cos() (* X56 X58))) X92)
    X65)) (/ (* (cos() (sin() (^(1/2) X77))) X23) X35)))) (LF X59)) (* (-
    C1.0 (LF X59)) (+ (+ (+ (^(1/2) (/ (- (- (sin() X67) X14) (cos()
    (^(1/2) X66))) (- (sin() (/ (^(1/2) (- X55 X56)) X63)) X79))) (* (cos()
     C0.5) (/ X26 (- X26 (* X28 (^(1/2) (sin() X42)))))) X92) (* C-1.0 (-
    (LF X21) (LF X5)))))))) (* C-1.0 (- (LF (+ (sin() (/ X50 (+ (* X69 X10)
     X43))) X86)) (LF (* X2 (* (+ X92 (/ X52 X20)) X37)))))) (* C-1.0 (- (
    LF (/ X26 (- X48 X12))) (LF (cos() (/ (+ X28 X18) (- (/ X54 X5) (+ X30
    (sin() X39)))))))))
```

Table 4.6 summarizes general assessment measures of the individual presented in above:

MAD	s	depth
2.25	3.09	20

Table 4.6: Characteristics of individual for prediction of objective-area B

Figure 4.14 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It is clearly seen that GMFM features with highest importance in prediction of total score for objective-area B are total score

for objective-area B, B26 and B24. B26 stands for "while sitting, touches a toy placed 45 degrees right-hand-side behind and goes back". B24 stands for "can sit on mat for 3 seconds arms free". Features B31, D54, D61, E65, E67 and E84 also exhibit relatively significant importance.

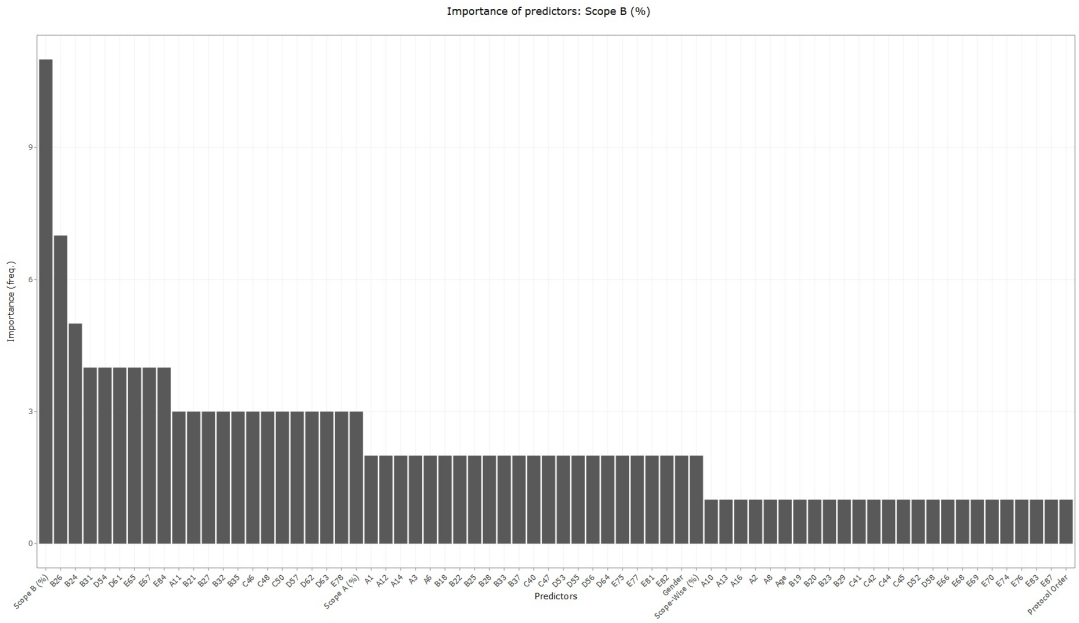


Figure 4.14: Worth plot of input features for prediction of objective-area B.

Objective-Area C: at the end of 41 runs representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (+ (* (+ (+ (* (- X93 (* (- (+ (sin() (- X0 X55)) X49) X4) X20)) (LF
    (+ (* (+ X31 (* (/ X77 X89) X49)) X34) (- (- (- X35 (- X71 X49)) X24) (
    cos() C0.0))))) (* (- C1.0 (LF (+ (* (+ X31 (* (/ X77 X89) X49)) X34) (-
     (- (- X35 (- X71 X49)) X24) (cos() C0.0))))) (+ (* (^(1/2) (* (* X25
    X55) (+ X28 C0.0))) (^(1/2) (* (- X24 X48) (^(1/2) X55)))) X93))) (* C
    -1.0 (- (LF C0.25) (LF X19)))) (LF (sin() X81))) (* (- C1.0 (LF (sin()
    X81))) (+ (+ (* (+ (- (* (sin() X37) X33) X51) (- (* X37 (cos() (+ X29
    X70))) (- (* (sin() X41) (- X3 (- (/ X22 C-0.75) X54))) (+ (+ X96
    (^(1/2) (* (/ C0.0 X34) (^(1/2) X80)))) (+ C0.5 X31))))) (LF (- X73 (*
    X15 (sin() (/ (cos() X22) X11)))))) (* (- C1.0 (LF (- X73 (* X15 (sin()
    (/ (cos() X22) X11)))))) (- X93 (^(1/2) (/ (- (- (- (* X54 X7) (* X5 X58
    )) (+ (* C-0.5 (sin() (/ X73 X35))) (* X18 X22))) X41) X48))))) (* C-1.0
     (- (LF (- (+ (^(1/2) (+ (^(1/2) X59) X59)) (cos() X18)) X81)) (LF (sin
    () (cos() X88)))))))) (* C-1.0 (- (LF X81) (LF X45)))) (* C-1.0 (- (LF
    X87) (LF (- (cos() (- (sin() (- X10 C0.25)) X87)) (cos() X70))))))
```

Table 4.7 summarizes general assessment measures of the individual presented in above:

MAD	s	depth
3.22	2.41	16

Table 4.7: Characteristics of individual for prediction of objective-area C

Figure 4.15 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It is clearly seen that GMFM features with highest importance in prediction of total score for objective-area C are C47, B20 and E79. Interestingly, the two most worthy variables for prediction of objective-area C are total scores for objective-areas C and B. C47 stands for "crawls down 4 steps on stairs on hands, knees and feet". B20 stands for "while lying, rolls to the left-hand-side and sits". E79 stands for "while standing, kicks a ball with left foot".

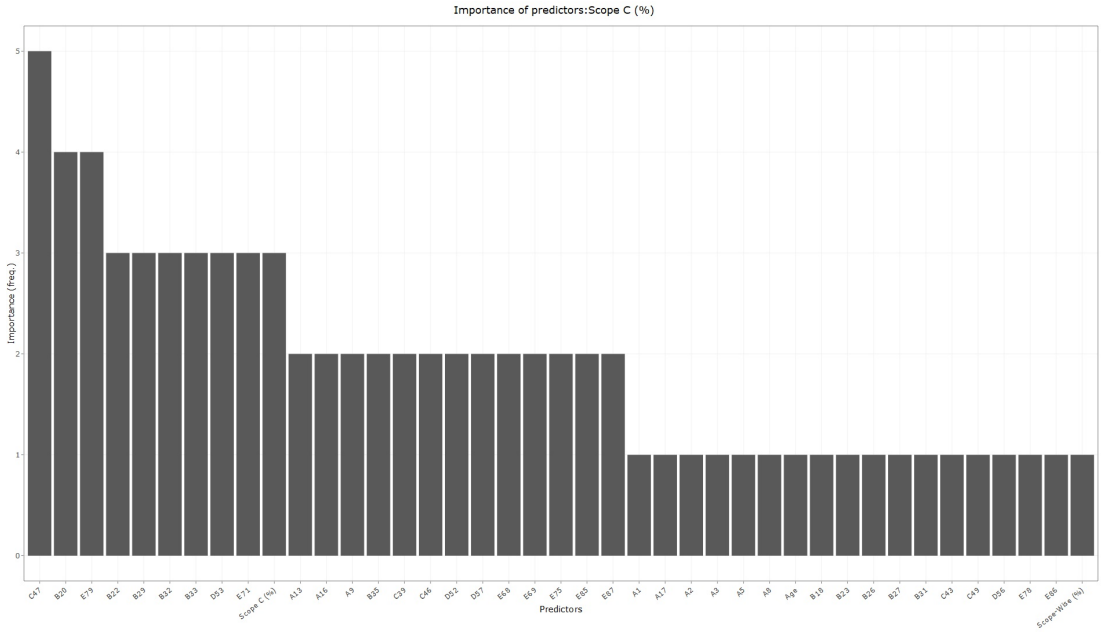


Figure 4.15: Worth plot of input features for prediction of objective-area C.

Objective-Area D: at the end of 41 runs representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (+ (+ (+ (+ (/ (* X4 (* (/ X70 X65) X12)) (+ X51 (+ (^(1/2) (cos() X93
    )) X87))) X94) (* C-1.0 (- (LF (^(1/2) (* (/ (* X79 C1.0) (+ (+ X70 X37)
     (cos() X90))) X66))) (LF (sin() (sin() (- (^(1/2) (- X86 X15)) (sin()
    X14)))))))) (* C-1.0 (- (LF (* X5 (+ X96 (+ X94 X76)))) (LF (/ X18 (/ (
    sin() X40) X39)))))) (* C-1.0 (- (LF X8) (LF (cos() (+ X54 (- (+ (/ X67
    C0.5) (^(1/2) X53)) X8))))))) (* C-1.0 (- (LF (+ (^(1/2) X44) X95)) (LF
    (- X58 X14))))) (* C-1.0 (- (LF C0.5) (LF (* X8 (* (sin() X49) (/ X90
    X74)))))))
```

Table 4.8 summarizes general assessment measures of the individual presented in above:

MAD	s	depth
2.92	3.54	13

Table 4.8: Characteristics of individual for prediction of objective-area D

Figure 4.16 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It is clearly seen that GMFM features with highest importance in prediction of total score for objective-area D are A6, A12, E68, E88 and total score for objective-area D. A6 stands for "reaching a toy with right arm while lying". A12 stands for "can put body weight on right arm and fully extend the left arm". E68 stands for "walking 10 steps forward with hand support". E88 stands for "jumping over 15cm stair without support". It worth to notice that the total score for objective-area D, measured before the therapy, exhibits on of the highest worth measures for obvious reason - high correlation with the target.

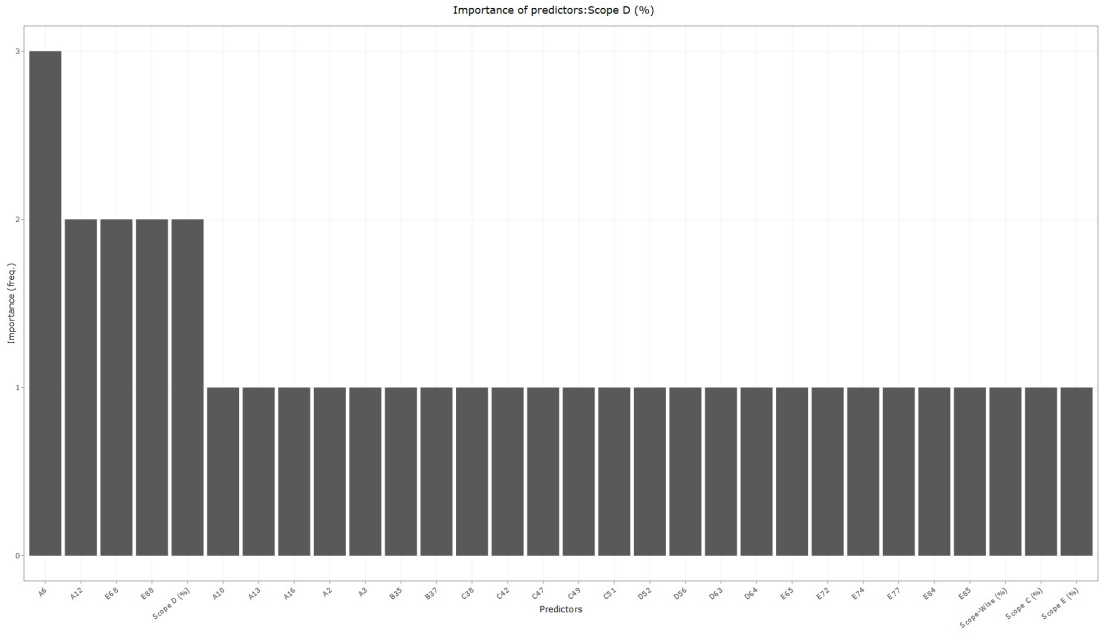


Figure 4.16: Worth plot of input variables for prediction of objective-area D.

Objective-Area E: at the end of 41 runs representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (+ (+ (* (+ (- X95 (/ (+ (- X2 X19) (- X77 X71)) (^(1/2) (- X64 (sin()
     (cos() (* (^(1/2) X3) (/ (+ (^(1/2) X6) (- X69 X33)) (sin() (sin() X57)
    ))))))))) (* C-1.0 (- (LF X62) (LF (+ (+ C0.75 (- (cos() (- X95 X19))
    X80)) X24))))) (LF (sin() (- (/ X65 (^(1/2) (cos() C-0.75))) X3)))) (*
    (- C1.0 (LF (sin() (- (/ X65 (^(1/2) (cos() C-0.75))) X3)))) (+ (- X95
    (^(1/2) (^(1/2) (- (/ (sin() (+ X74 X68)) (/ (sin() (+ (+ (* X64 X87)
    X77) (^(1/2) X64))) (cos() X62))) X36)))) (* C-1.0 (- (LF X78) (LF X26))
    )))) (* C-1.0 (- (LF (/ X90 (+ (* X5 (/ X96 (* X42 X93))) (* (+ (sin()
    X89) (cos() X36)) C-0.25)))) (LF (/ (+ X71 (/ X11 (cos() (cos() X16))))
    X83))))) (* C-1.0 (- (LF (cos() X12)) (LF (* (^(1/2) (sin() (sin() X74))
    ) X68))))) (* C-1.0 (- (LF X41) (LF (* (cos() (+ (- (^(1/2) X8) X96)
    (^(1/2) X24))) X42)))))
```

Table 4.9 summarizes general assessment measures of the individual presented in above:

MAD	s	depth
1.64	1.16	16

Table 4.9: Characteristics of individual for prediction of objective-area E

Figure 4.17 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It is clearly seen that GMFM features with highest importance in prediction of global score for objective-area E are A1, D62 and global score for objective-area E. A1 stands for "turns head without moving shoulders and arms". D62 stands for "performs controlled sitting from standing position". Obviously, total score for objective-area E, measured before the therapy, also exhibits on of the highest worth measures. This happens because the variable exhibits high correlation with the target.

Global Score: at the end of 41 runs representation of the final individual represented by means of Java code looks as follows:

```
(+ (+ (* (+ (+ (+ (+ X96 (- X42 (- (^(1/2) (* (+ (cos() X63) (- X7 (cos()
    X66))) X42)) (* (/ (* X5 X59) (* X86 X51)) X55)))) (* C-1.0 (- (LF (+ (
    sin() X34) X70)) (LF (/ (/ (* X79 (/ (* X63 X32) X56)) X79) X58))))) (*
    C-1.0 (- (LF (+ X21 (sin() X68))) (LF (sin() (- X62 X0)))))) (* C-1.0 (-
     (LF (/ (- X45 (/ X93 (sin() X23))) X19)) (LF X34)))) (LF (* (cos() (-
    X22 X49)) (cos() X80)))) (* (- C1.0 (LF (* (cos() (- X22 X49)) (cos()
    X80)))) (+ (+ (* (+ (+ X96 (+ (/ X55 (* X86 (/ X53 X69))) X59)) (* C-1.0
     (- (LF X52) (LF X4)))) (LF (cos() X59))) (* (- C1.0 (LF (cos() X59)))
    (+ (+ X96 (- X42 (- (^(1/2) (* (+ (cos() X63) (- X7 (cos() X66))) X42))
    (* (/ (* X5 X59) (* X86 X51)) X55)))) (* C-1.0 (- (LF X19) (LF C0.25))))
    )) (* C-1.0 (- (LF (* X36 (sin() (/ X45 X15)))) (LF C-0.25)))))) (* C
    -1.0 (- (LF (+ (sin() (/ (+ X5 (cos() X13)) (* X31 (cos() X73)))) X40))
    (LF X24))))
```

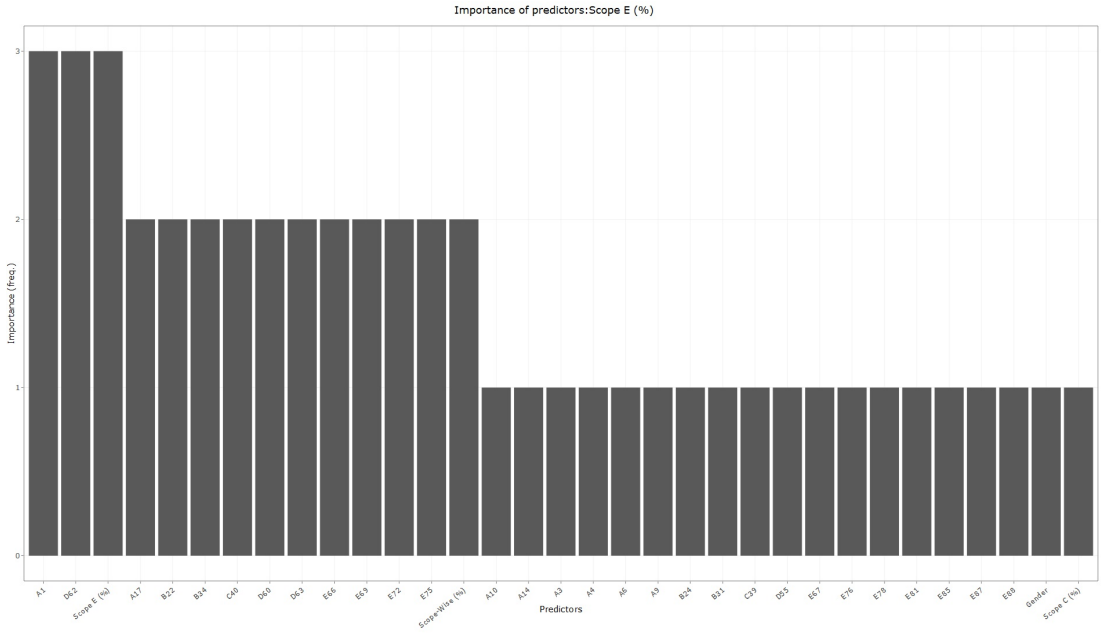


Figure 4.17: Worth plot of input variables for prediction of objective-area E

Table 4.10 summarizes general assessment measures of the individual presented in above:

MAD	s	depth
2.08	2.28	15

Table 4.10: Characteristics of individual for prediction of global score

Figure 4.18 exhibits worth of all input features in regard to prediction of the target, function of their frequency in the final individual. It is clearly seen that GMFM features with highest importance in prediction of global score are D57 and C40. D57 means "lifts left foot for 10 seconds, arms free". C40 means "starting from crawling position, can sit without hands support". Additionally, features A3, D53, D61 and E87 also exhibit relevant importance.

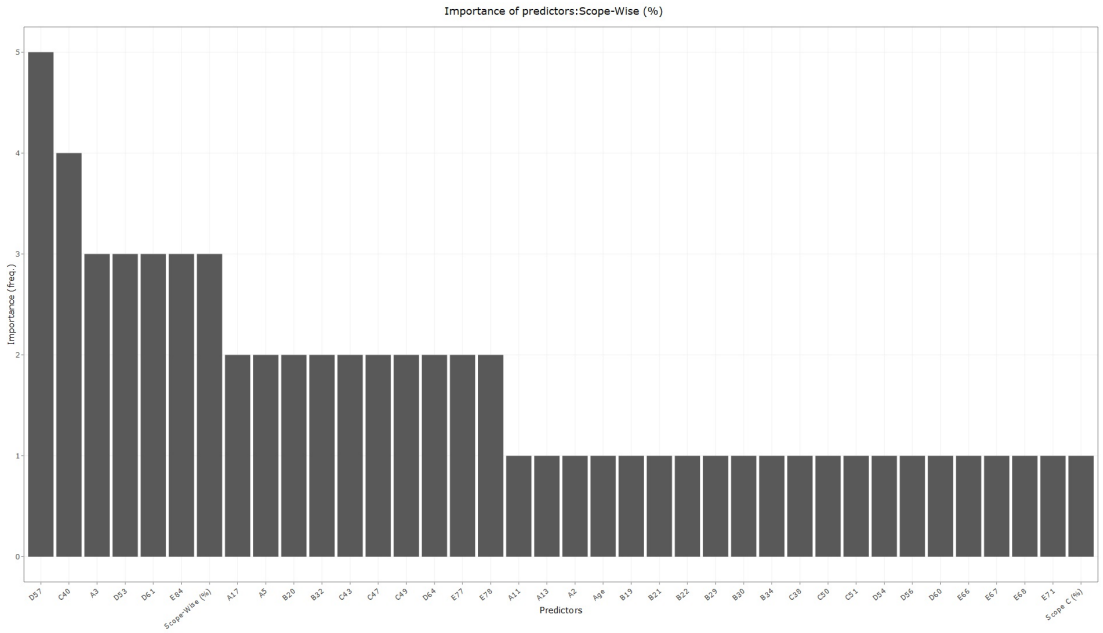


Figure 4.18: Worth plot of input variables for prediction of global score

Chapter 5

R Shiny Web-Application for *Rare Diseases*

In this chapter screen-shots and brief description of the web application, specially designed to fulfill entity's needs, are presented. It was developed as an easy and intuitive tool to enhance knowledge-discovery and support decision-making processes in broad range of topics. Its main features are multivariate visualization, characterization of groups of patients and prediction of therapy's effects.

In annex of this thesis you can find the user-manual, based on this chapter, that was provided to the entity in order to empower adoption of this technological solution.

5.1 GMFM88-Changes

This section reports the first tab of the web-application, entitled *GMFM88 changes*. Figure I.1 presents a bar-chart for one of GMFM-88. Additionally, its corresponding ARC is presented in text output (computed as in sub-sub-section 4.5.1.1).

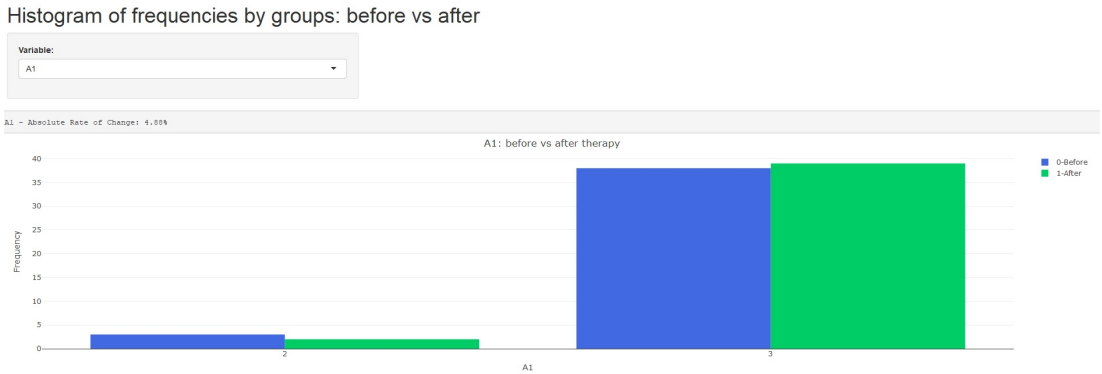


Figure 5.1: GMFM-88 bar-chart to visualize change after the therapy (measure A1)

By changing the feature from drop-down list in sidebar panel, its is possible to

update the bar-chart and corresponding text output. In figure I.2, measure A4 was selected and the output was automatically updated from A1.

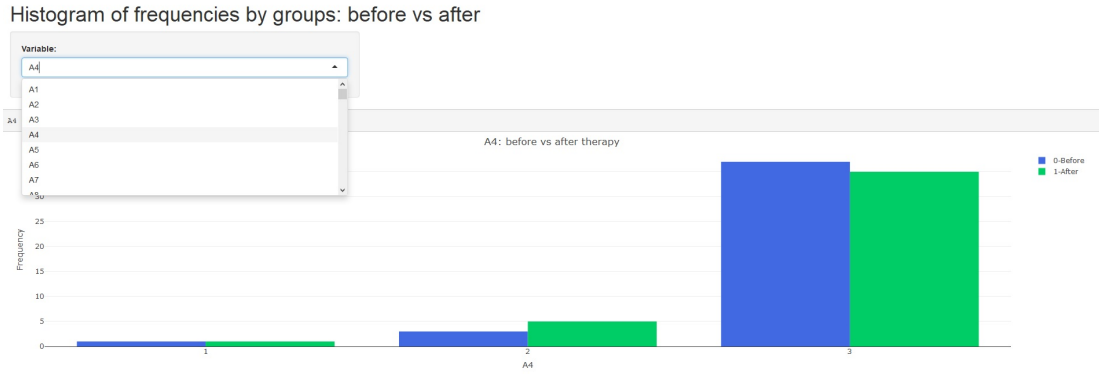


Figure 5.2: GMFM-88 bar-chart to visualize change after the therapy (measure B29)

Bar-chart in figure I.3 represents all GMFM-88 ranked according to ARC (in descending order). Here, we can see that the feature with highest ARC is B29. Features A3, E77, E82, E83 and E88 did not suffer any change after the therapy.

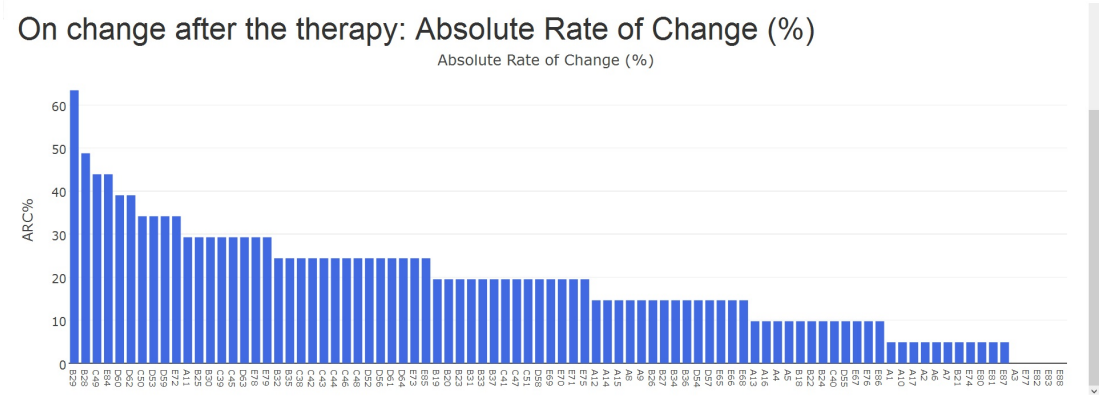


Figure 5.3: Bar-chart of GMFM-88 ranked by ARC (in descending order)

5.2 Analysis of Change in Main Objective-Areas: Summary

This section reports the second tab of the web-application, entitled *Analysis of change in main objective-areas*. Figure I.4 presents a dumbbell plot for, concretely in this example, global score. By looking at it, we can track patients improvement after the therapy. As we can see, for the majority of patients (note that each patient is represented in one single y-axis coordinate), therapy had positive impact: green points tend to be on the right while red on the left. Moreover, the magnitude of improvement is more noticeable when points are dots (a dot means that patient attended therapy for the first time, while triangle and square mean second and third therapy, respectively). Line color also has a meaning: blue line represents male patients while pink line

represent female patients. By paying attention to the gender, it is hard to say that there is significant difference between male and female patients.

The plot, as any other plot in the application, is extremely dynamic: zoom in/out operation is allowed, any plot can be downloaded as *.png*. Each point has a personalized pop-up description. In this example, you can see that it provides information regarding patients ID, age, protocol (therapy) order and corresponding score (x-axis).

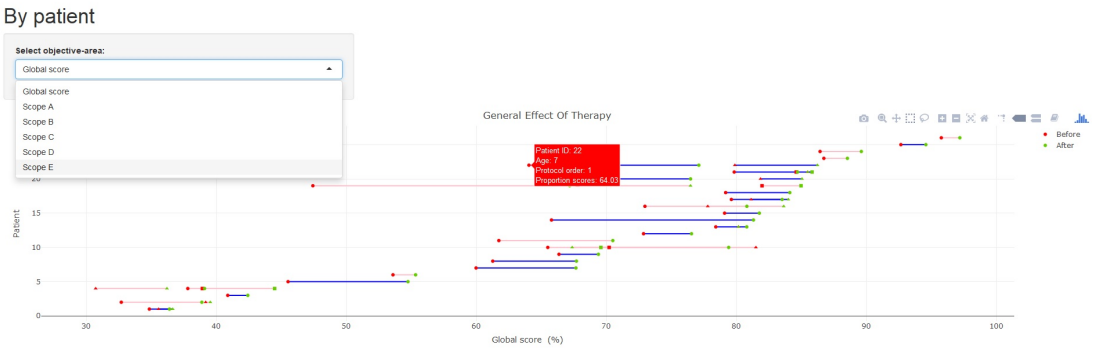


Figure 5.4: Dumbbell plot for six GMFM-88 summary measures

In order to support interpretation and avoid unbiasedness that may be introduced by high amount of information concentrated in one single plot I.4, in figure I.5 you can find summary tables which present average improvements by number of performed therapies, age and gender groups:

Average improvement by number of therapies

Therapy order	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
1	5.96	1.47	7.02	8.10	8.65	4.57
2	1.79	1.07	0.57	2.60	2.27	2.46
3	2.29	3.68	2.81	2.68	0.96	1.30

Average improvement by age groups

Age group	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
[3,7]	6.17	1.55	5.59	9.40	9.01	5.32
(7,8]	2.15	2.06	6.50	1.79	0.38	0.00
(8,12]	3.51	2.73	5.00	3.32	3.85	2.68
(12,20]	3.13	0.59	2.62	3.93	5.38	3.13

Average improvement by gender

Gender	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
Female	4.45	1.96	5.28	5.56	5.34	4.11
Male	4.51	1.28	4.57	6.52	6.86	3.35

Figure 5.5: Average improvement for of patients by number of performed therapies, age and gender groups, measured through GMFM-88 summary measures

5.3 Analysis of Change in Main Objective-Areas and GMFM-88: Bivariate

This section reports the third tab of the web-application, entitled *Analysis of change in main objective-areas and GMFM-88: bivariate*. Figure I.6 represents two bubble-plots, one regarding before another after the therapy. In this plots, x and y axes can be any of global measures. Each ball stands for one patients and the color represents either gender or number of therapies performed. The size of the ball represents global score.

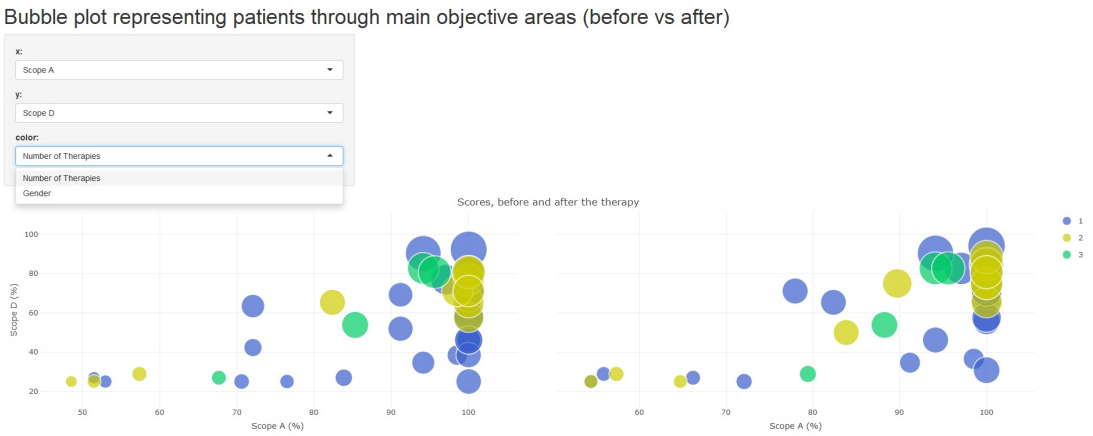


Figure 5.6: Bubble-plot of patients in regard to GMFM-88 summary measures

Figure I.8 represents Spearman Correlation Matrix between all pairs of GMFM-88 and its summary indicators:

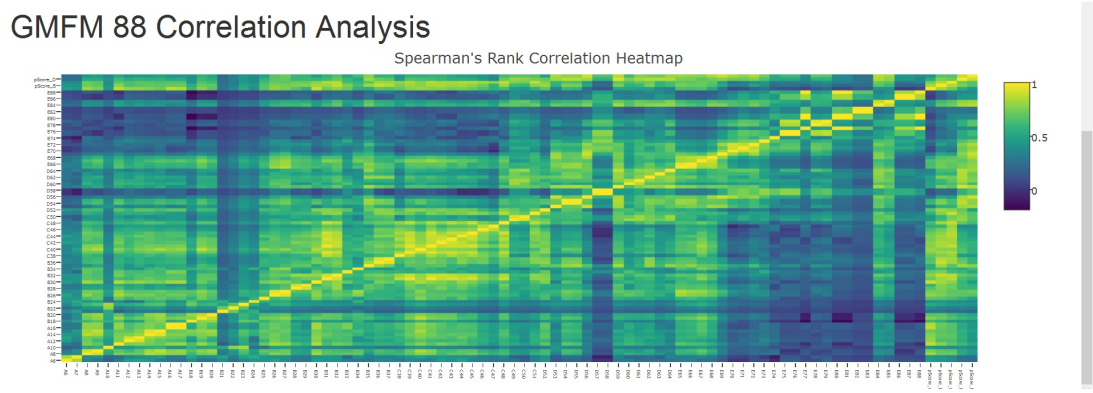


Figure 5.7: Spearman Correlation Matrix

5.4 Patients Clustering

This section reports the fourth tab of the web-application, entitled *Patients Clustering*. Figure I.9 represents the heat-map features selected to perform cluster analysis

(see section 4.5), where each row stands for one therapy performed by a given patient (justified in 4.1.3). On the right-hand-side of this plot you can find clustering *dendrogram* colored by the number of clusters selected in sidebar panel, which groups rows according to their similarity.

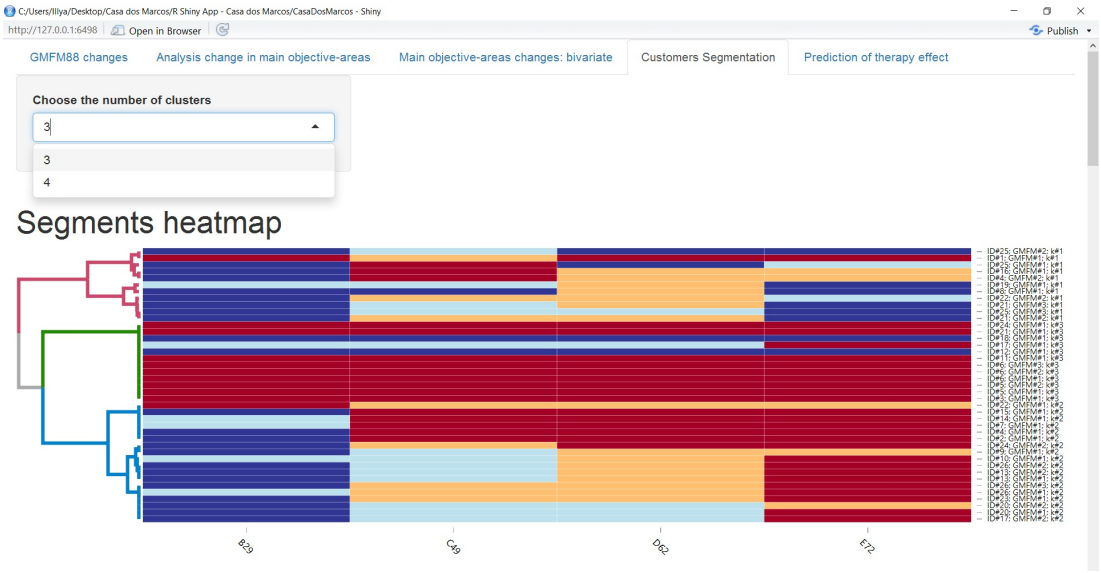


Figure 5.8: Heat map for clustering of patients

Figure I.10 exhibits two bubble-plot (similar to the ones in figure I.6), with one difference: color of each bubbles represents the cluster where each patient belongs.

Figure 5.9: Bubble-plot of patients in regard to GMFM-88 summary measures, colored by clusters

In figure I.11, user can find summary statistics for pre-defined number of clusters. Also, a detailed table is provided.

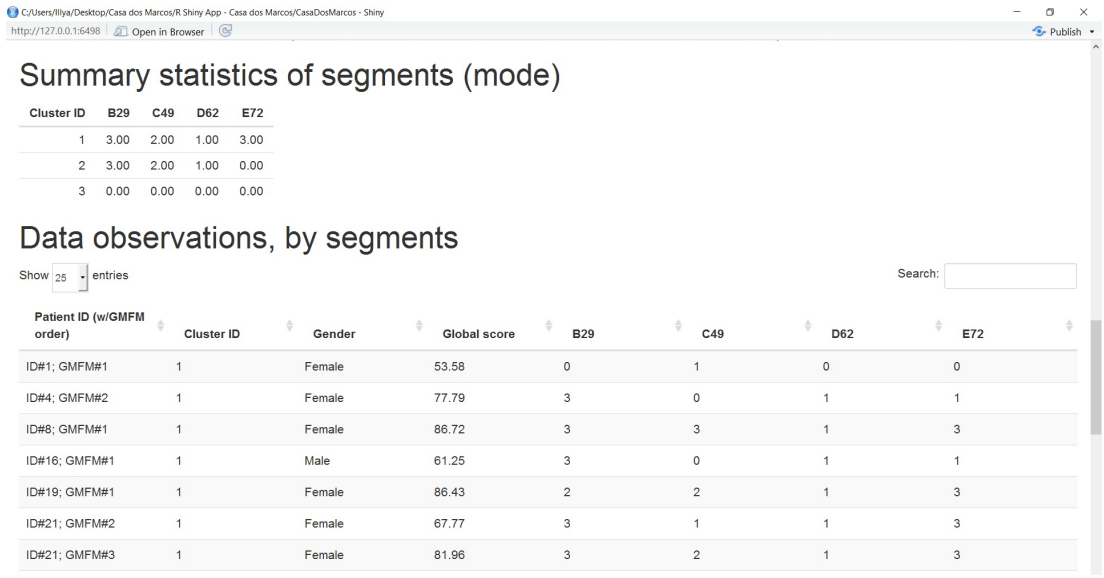


Figure 5.10: Summary table of clusters

5.5 Prediction of Therapy Effect

This section reports the fifth and last tab of the web-application, entitled *Prediction of therapy effect*.

Figures I.12 represent the form that must be submitted to the application in order to perform prediction. Input values will form input vector for predictive models that run on the server side.

Figure I.13 represents calculated prediction for each objective area and global score. Each table outputs the real score before the therapy, expected score after the therapy and expected improvement computed as difference between previous two.

Figure I.14 represents worth of all input features in regard to prediction of the target, function of their frequency in the final individual.

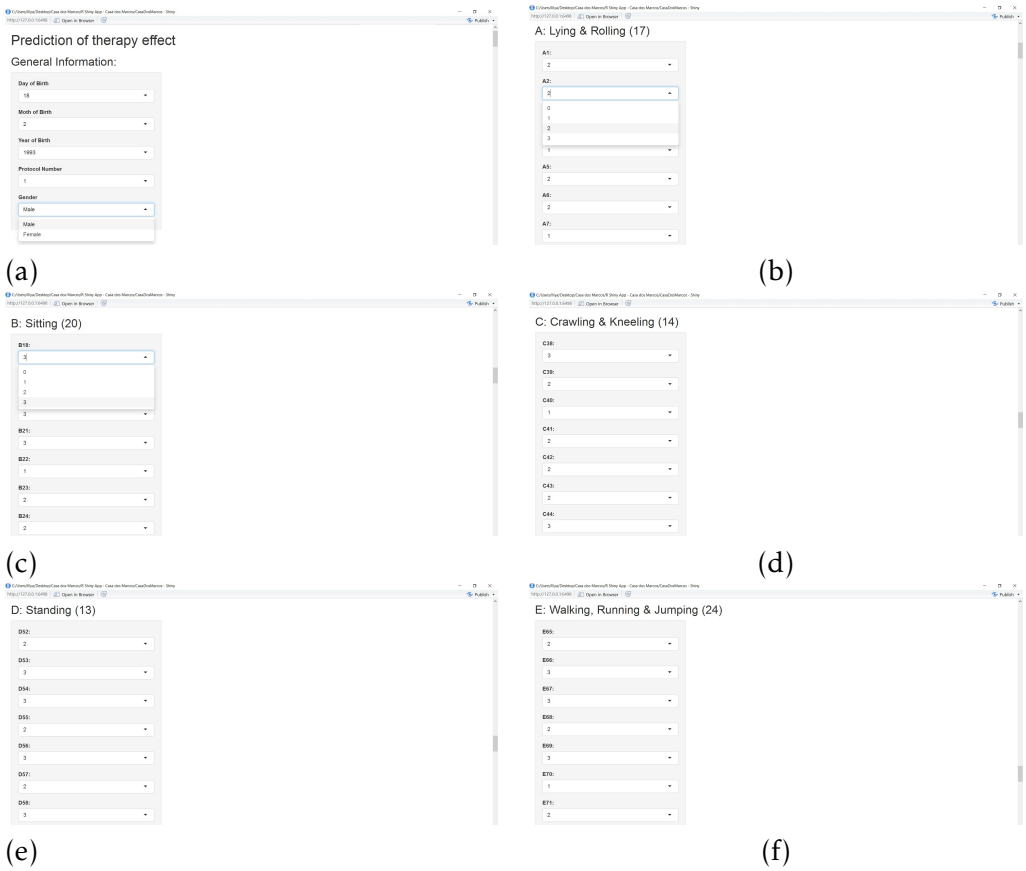


Figure 5.11: Plot (a): screen-shot of socio-demographic form. Plot (b): screen-shot of GMFM-88 regarding objective-area A. Plot (c): screen-shot of GMFM-88 regarding objective-area B. Plot (d): screen-shot of GMFM-88 regarding objective-area C. Plot (e): screen-shot of GMFM-88 regarding objective-area D. Plot (f): screen-shot of GMFM-88 regarding objective-area E.

Calculate Expected Effect

Predicted improvement for objective-area A

Current scope A (%)	Expected scope A (%)	Expected improvement (%)
25.00	26.07	1.07

Predicted improvement for objective-area B

Current scope B (%)	Expected scope B (%)	Expected improvement (%)
25.00	26.91	1.91

Predicted improvement for objective-area C

Current scope C (%)	Expected scope C (%)	Expected improvement (%)
25.00	100.00	75.00

Predicted improvement for objective-area D

Current scope D (%)	Expected scope D (%)	Expected improvement (%)
25.00	24.29	-0.71

Predicted improvement for objective-area E

Current scope E (%)	Expected scope E (%)	Expected improvement (%)
25.00	23.96	-1.04

Predicted global improvement

Current global score (%)	Expected global score (%)	Expected improvement (%)
25.00	23.76	-1.24

Figure 5.12: Prediction tables for each one of six targets

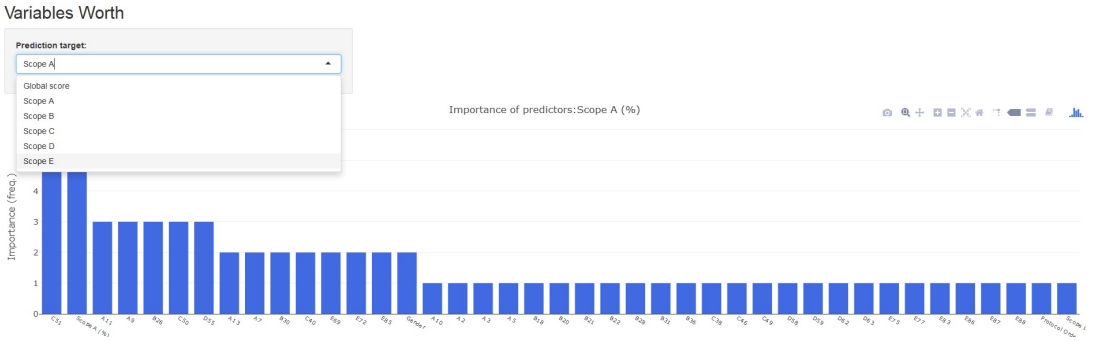


Figure 5.13: Worth plot of input features for prediction of a given objective-area (features are presented in descending order)

Chapter 6

Conclusions and Future Work

This final chapter provides general conclusions in regard EDDA system and institutional collaboration with *Casa dos Marcos*. Additionally ideas for future research and collaboration are proposed.

6.1 Conclusions and Future Work

A new population initialization method for Genetic Programming (GP), based on demes evolution and despeciation, was presented in this paper and applied to Geometric Semantic GP (GSGP). In Biology, demes are local populations, or subpopulations, of individuals that actively interbreed with one another, and the term despeciation indicates the combination of demes of previously distinct species into a new population. The method we presented, called Evolutionary Demes Despeciation Algorithm (EDDA), seeds a population of N individuals with the best solutions obtained by the independent evolution of N different populations, or demes. The presented experimental results have shown the effectiveness of the method on six complex real-life symbolic regression problems. More specifically, for each one of the studied problems, EDDA has been able to find solutions that are better, or at least comparable, on unseen data, but significantly smaller, than the ones that were found by GSGP using the RHH algorithm to initialize the population.

In the future, we plan to extend the idea in several possible directions. First of all, the use of several different GP versions for evolving the demes, and not only standard GP and GSGP, deserves investigation. Also, we plan to implement EDDA on a parallel and distributed framework, in order to make it faster and to be able to increment the number of demes that can be used. The possibility of letting several versions of GP, including standard GP and GSGP, interact and exchange genetic material in a

parallel and distributed system not only in the initialization phase, but also during all the other evolution steps, is also an active track of our current research. Last but not least, we plan to develop a system in which the diversity among the different demes is even more remarkable than in the present work, fostering the parallel evolution of demes with different representations of the solutions, characterized by different parameters and running different algorithms.

In order to fully understand the practical advantage of EDDA, please check again visual representations of the final individuals in sub-section 4.6.2. Only the fact that each individual fits in one A4 page (most of them need one third of the space), is fascinating. Compared to classical initialization algorithm (like RHH), to solve the same problem with comparable generalization, very probably, one individual would not fit in all the pages of this document. Probably, it would need ten or even one hundred times more pages. The issue is extremely important since, for example, it allowed us easily settle the individuals (predictive models) running in a web-application.

By the end of collaboration, all requirements were achieved. The final product was settle on a R Shiny Web-Application (delivered to the entity), where both descriptive and predictive models were deployed.

The importance of supporting decision-making process with objective and tangible facts will strongly empower the entity in achieving more elaborated and personalized therapeutic approach (taking in consideration existence of several natural groups of patients), helping patients to achieve higher scores and manage internal resources with more caution. For our knowledge, this is the first time ever advanced analytical tools were applied in context of *Rare Diseases*. This fact highlight broad applicability of such tools, although knowledge in the filed remains an indispensable component for their successful application.

Bibliography

[1] O. Akbilgic, H. Bozdogan, and M. E. Balaban. "A novel Hybrid RBF Neural Networks model as a forecaster." In: *Statistics and Computing* 24.3 (2014), pp. 365–375.

[2] F. Archetti, S. Lanzeni, E. Messina, and L. Vanneschi. "Genetic programming for computational pharmacokinetics in drug discovery and development." In: *Genetic Programming and Evolvable Machines* 8.4 (2007), pp. 413–432.

[3] L. C. J. Beadle. "Semantic and Structural Analysis of Genetic Programming." Doctoral dissertation. Canterbury: University of Kent, July 2009. URL: http://www.beadle.me/Me/LBeadle_PhD_Thesis.pdf.

[4] L. C. J. Beadle and C. G. Johnson. "Semantic Analysis of Program Initialisation in Genetic Programming." In: *Genetic Programming and Evolvable Machines* 10.3 (Sept. 2009), pp. 307–337.

[5] I. Brown. *Developing Credit Risk Models Using SAS® Enterprise MinerTM and SAS/STAT®: Theory and Applications*. New York, NY, USA: SAS Institute Inc, 2014.

[6] M. Castelli, L. Vanneschi, and S. Silva. "Prediction of high performance concrete strength using Genetic Programming with geometric semantic genetic operators." In: *Expert Systems with Applications* 40.17 (2013), pp. 6856 –6862.

[7] M. Castelli, S. Silva, and L. Vanneschi. "A C++ framework for geometric semantic genetic programming." English. In: *Genetic Programming and Evolvable Machines* 16.1 (2015), pp. 73–81.

[8] M. Castelli, L. Trujillo, L. Vanneschi, and A. Popovic. "Prediction of energy performance of residential buildings: A genetic programming approach." In: *Energy and Buildings* 102 (2015), pp. 67 –74.

[9] M. Castelli, L. Manzoni, I. Gonçalves, L. Vanneschi, L. Trujillo, and S. Silva. "An Analysis of Geometric Semantic Crossover: A Computational Geometry Approach." In: *Proc. of the 8th Int. Joint Conf. on Computational Intelligence - Vol. 3: ECTA*, 2016, pp. 201–208.

[10] *Comparison of Hierarchical Cluster Analysis Methods by Cophenetic Correlation*. July 13, 2017. URL: https://link.springer.com/article/10.1186/1029-242X-2013-203.

[11] J. M. Daida, H. Li, R. Tang, and A. M. Hilss. "What Makes a Problem GP-Hard? Validating a Hypothesis of Structural Causes." In: *Genetic and Evolutionary Computation – GECCO-2003*. Ed. by E. Cantú-Paz textitet al. Vol. 2724. LNCS. Springer-Verlag, 2003, pp. 1665–1677. ISBN: 3-540-40603-4.

[12] C. Darwin. *On the Origin of Species by Means of Natural Selection*. London, England: John Murray, 1859.

[13] *European Commission Public Health - Neurodegenerative Disorders*. July 12, 2017. URL: http://ec.europa.eu/health/major_chronic_diseases/diseases/brain_neurological_en.

[14] *European Patients' Forum - Data Protection*. July 14, 2017. URL: http://www.eu-patient.eu/whatwedo/Policy/Data-Protection.

[15] *European Union Joint Programme – Neurodegenerative Disease Research (JPND)*. July 12, 2017. URL: http://www.neurodegenerationresearch.eu/about/what/.

[16] *Gross Motor Function Measure*. July 12, 2017. URL: https://canchild.ca/en/resources/44-gross-motor-function-measure-gmfm.

[17] S. Gustafson, E. K. Burke, and N. Krasnogor. "The Tree-String Problem: An Artificial Domain for Structure and Content Search." In: *Proceedings of the 8th European Conference on Genetic Programming*. Ed. by M. Keijzer textitet al. Vol. 3447. Lecture Notes in Computer Science. Lausanne, Switzerland: Springer, 2005, pp. 215–226.

[18] H. M.S. P. Hand David. *Principles of Data Mining*. Cambridge, Massachusetts, London, England: The MIT Press, 2001.

[19] *Hierarchical Clustering implementation in R*. July 12, 2017. URL: https://stat.ethz.ch/R-manual/R-devel/library/stats/html/hclust.html.

[20] J. R. Koza. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge, MA, USA: MIT Press, 1992.

[21] W. Langdon and R. Poli. *Foundations of Genetic Programming*. Springer, 2002.

[22] J. Y. Lin, J. Y. Yeh, and C.-C. Liu. "Applying layered multi-population genetic programming on learning to rank for information retrieval." In: *2012 International Conference on Machine Learning and Cybernetics*. Vol. 5. 2012, pp. 1754–1759.

[23] J.-Y. Lin, H.-R. Ke, B.-C. Chien, and W.-P. Yang. "Designing a classifier by a layered multi-population genetic programming approach." In: *Pattern Recognition* 40.8 (2007), pp. 2211–2225.

[24] In: *Genetic Programming*. Ed. by p. a. p. M. Keijzer textitet al. title=Sampling of Unique Structures and Behaviours in Genetic Programming. Vol. 3003. Lecture Notes in Computer Science. 2004.

[25] U. of Minho, ed. *Introduction to Genetic Programming*. 6ELBCE - University of Minho. University of Minho, 2016.

[26] M. Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, Massachusetts; London, England: MIT Press, 1999.

[27] A. Moraglio, K. Krawiec, and C. Johnson. "Geometric Semantic Genetic Programming." English. In: *Parallel Problem Solving from Nature - PPSN XII*. Ed. by C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia, and M. Pavone. Vol. 7491. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 21–31.

[28] L. O. V. Oliveira, F. E. Otero, and G. L. Pappa. "A Dispersion Operator for Geometric Semantic Genetic Programming." In: *Proceedings of the Genetic and Evolutionary Computation Conference 2016*. GECCO '16. Denver, Colorado, USA: ACM, 2016, pp. 773–780. ISBN: 978-1-4503-4206-3.

[29] O. Parr-Rud. *Business Analytics Using SAS® Enterprise Guide® and SAS® Enterprise Miner ™ A Beginner's Guide*. New York, NY, USA: SAS Institute Inc, 2014.

[30] T. P. Pawlak and K. Krawiec. "Semantic Geometric Initialization." In: *Genetic Programming: 19th European Conference, EuroGP 2016, Porto, Portugal, March 30 - April 1, 2016, Proceedings*. Ed. by M. I. Heywood, J. McDermott, M. Castelli, E. Costa, and K. Sim. Springer International Publishing, 2016, pp. 261–277.

[31] W. F. Punch, D. Zongker, and E. D. Goodman. "The Royal Tree Problem, a Benchmark for Single and Multiple Population Genetic Programming." In: *Advances in Genetic Programming 2*. Ed. by P. J. Angeline and K. E. Kinnear, Jr. Cambridge, MA, USA: MIT Press, 1996. Chap. 15, pp. 299–316.

[32] *Spearman Correlational Distance*. July 12, 2017. URL: https://artax.karlin.mff.cuni.cz/r-help/library/bioDist/html/spearman.dist.html.

[33] E. B. Taylor, J. W. Boughman, M. Groenenboom, M. Sniatynski, D. Schluter, and J. L. Gow. "Speciation in reverse: morphological and genetic evidence of the collapse of a three-spined stickleback (Gasterosteus aculeatus) species pair." In: *Molecular Ecology* 15.2 (2006), pp. 343–355. ISSN: 1365-294X.

[34] M. Tomassini, L. Vanneschi, P. Collard, and M. Clergue. "A Study of Fitness Distance Correlation as a Difficulty Measure in Genetic Programming." In: *Evolutionary Computation* 13.2 (2005), pp. 213–239.

[35] L. Vanneschi. "An Introduction to Geometric Semantic Genetic Programming." In: *Results of the Numerical and Evolutionary Optimization Workshop (NEO 2015)*. Ed. by O. Schutze, L. Trujillo, P. Legrand, and Y. Maldonato. Springer, 2017, pp. 3–42.

[36] L. Vanneschi, M. Castelli, and L. Manzoni. "The K Landscapes: A Tunably Difficult Benchmark for Genetic Programming." In: *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*. GECCO 11. Dublin, Ireland: ACM, 2011, pp. 1467–1474.

[37] L. Vanneschi, M. Castelli, and S. Silva. "A survey of semantic methods in genetic programming." English. In: *Genetic Programming and Evolvable Machines* 15.2 (2014), pp. 195–214. ISSN: 1389-2576.

[38] L. Vanneschi, S. Silva, M. Castelli, and L. Manzoni. "Geometric Semantic Genetic Programming for Real Life Applications." In: *Genetic Programming Theory and Practice XI*. Ed. by R. Riolo, J. H. Moore, and M. Kotanchek. Springer New York, 2014, pp. 191–209.

[39] D. S. Wilson. "Structured Demes and the Evolution of Group-Advantageous Traits." In: *The American Naturalist* 111.977 (1977), pp. 157–185. DOI: 10.1086/283146.

ANNEX I

Annex 1: User Manual for R-Shiny Web Application

This document presents the user-guide for specially designed web application in context of institutional collaboration between *NOVA-IMS* and *Casa dos Marcos*. The application was developed as an easy and intuitive tool to enhance knowledge-discovery and support decision-making processes in broad range of topics. Its main features are multivariate visualization, characterization of groups of patients and prediction of effects for *Pedia Suit Protocol*.

Any plot in this application was designed to be extremely dynamic: zoom in and out operations are allowed, any plot can be downloaded as *.png*. Each data-point has a personalized pop-up description.

I.1 Change in GMFM-88

This section reports the first tab of the web-application, entitled *Change in GMFM-88*. Figure I.1 presents a bar plot for one of GMFM-88. On the *x* axis you can find the values taken by a given GMFM-88. The height of the bars correspond to frequency of data observations (patients) at each value. The bars are colored according to measure moment - before (green) or after (blue) the therapy.

Additionally, text output above the plot presents corresponding *Average Rate of Change* (ARC) defined as the sum of absolute differences between values taken before and after the therapy, divided by number of data instances.

Consider **X={0, 1, 2, 3,}** as one of GMFM-88 measures. Let **Xb** denote **X** before the therapy and **Xa** after. The first step is to create a frequency table (table I.1) for **X**. Then absolute differences are computed between values of **Xb** and **Xa**, for the same level (table I.2).

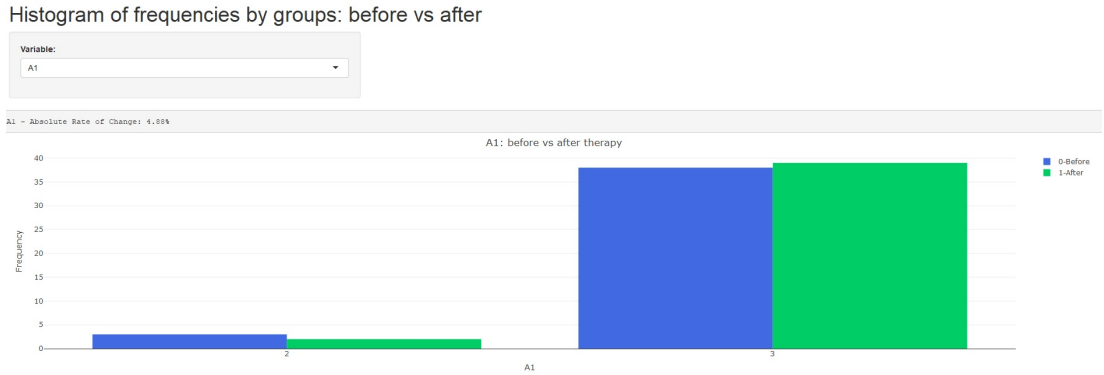


Figure I.1: GMFM-88 bar-chart to visualize change after the therapy (measure A1)

level	Xb	Xa
0	5	6
1	0	2
2	2	1
3	34	32

Table I.1: Frequency table for **X={0, 1, 2, 3,}**.

level	Xb	Xa	$\lvert Xb - Xa \rvert$
0	5	6	1
1	0	2	2
2	2	1	1
3	34	32	2

Table I.2: Frequency table with absolute differences

Finally, differences are summed, divided by number of data instances and multiplied by 100. Considering the example in tables I.1and I.2:

$$(\frac{1+2+1+2}{41}) * 100 \approx 14.63$$

As such, ARC for feature **X** is approximately 14.63%.

By changing the feature from drop-down list in sidebar panel (upper left corner), its is possible to update the bar plot and corresponding ARC on the text output. In figure I.2, measure B29 was selected and the output was automatically updated.

Bar plot in figure I.3 represents all GMFM-88 ranked according to ARC (in descending order). Here, we can see that the feature with highest ARC is B29. Features A3, E77, E82, E83 and E88 did not suffer any change after the therapy.

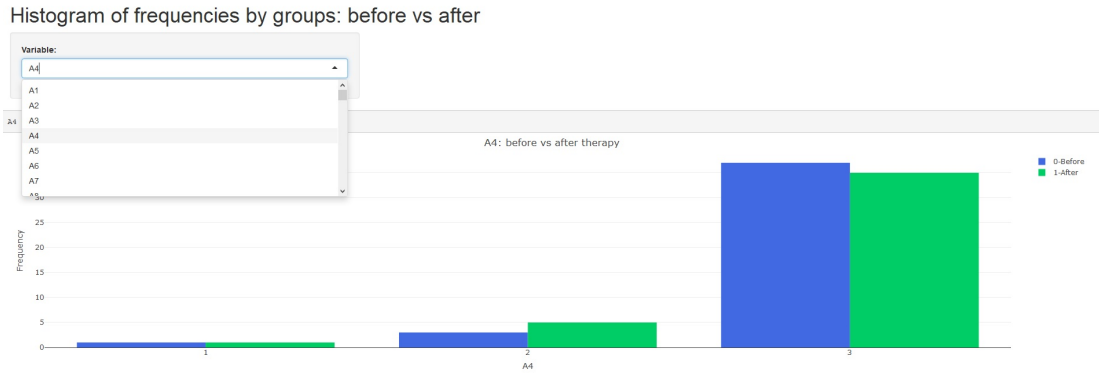


Figure I.2: GMFM-88 bar-chart to visualize change after the therapy (measure B29).

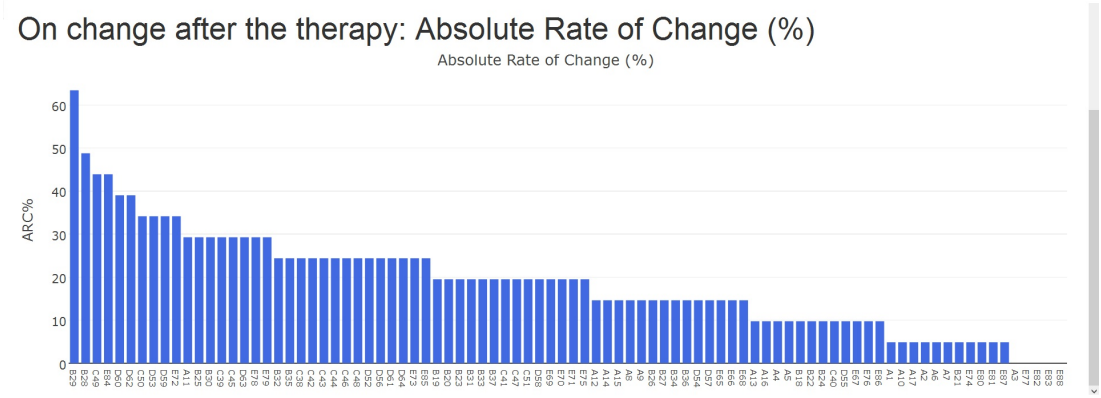


Figure I.3: Bar-chart of GMFM-88 ranked by ARC (in descending order).

I.2 Change in main objective-areas (I)

This section reports the second tab of the web-application, entitled *Change in main objective-areas (I)*. Figure I.4 presents a dumbbell plot for, concretely in this example, global score. By looking at it, we can track patients improvement after the therapy. As we can see, for the majority of patients (note that each patient is represented in one single line of y axis), therapy had positive impact: green points tend to be on the right while red on the left. Moreover, the magnitude of improvement is more noticeable when points are dots (**a dot means that patient attended therapy for the first time, while triangle and square mean second and third therapy, respectively**). Line color also has a meaning: **ablue line represents male patients while pink line represent female patients**. By paying attention to the gender, it is hard to say that there is significant difference between male and female patients.

The plot, as any other plot in the application, is extremely dynamic. In this example, you can see that by pointing the cursor to a given point, pop-up window provides information regarding patients ID, age, protocol (therapy) order and corresponding score (x-axis).

In order to support interpretation and avoid visual bias that can be introduced

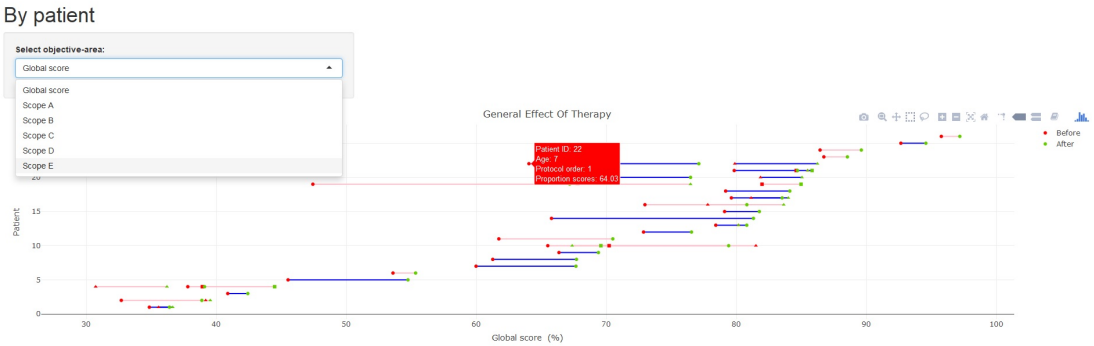


Figure I.4: Dumbbell plot for six GMFM-88 summary measures.

by high amount of information concentrated in one single plot, in figure I.5 you can find summary tables which present average improvements by number of performed therapies, age and gender groups.

Average improvement by number of therapies

Therapy order	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
1	5.96	1.47	7.02	8.10	8.65	4.57
2	1.79	1.07	0.57	2.60	2.27	2.46
3	2.29	3.68	2.81	2.68	0.96	1.30

Average improvement by age groups

Age group	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
[3,7]	6.17	1.55	5.59	9.40	9.01	5.32
(7,8]	2.15	2.06	6.50	1.79	0.38	0.00
(8,12]	3.51	2.73	5.00	3.32	3.85	2.68
(12,20]	3.13	0.59	2.62	3.93	5.38	3.13

Average improvement by gender

Gender	Global score	Scope A	Scope B	Scope C	Scope D	Scope E
Female	4.45	1.96	5.28	5.56	5.34	4.11
Male	4.51	1.28	4.57	6.52	6.86	3.35

Figure I.5: Average improvement for of patients by number of performed therapies, age and gender groups, measured through GMFM-88.

I.3 Change in main objective-areas (II)

This section reports the third tab of the web-application, entitled *Change in main objective-areas (II)*. Figure I.6 shows two bubble plots, one regarding before another after the therapy. In this plots, x and y axes can be any of six global measures. Each ball represents one patient and the color represents either gender or number of therapies performed. The diameter expresses global score. As you can see, in the left corner, the user can easily change any of global measures on the axes and the color of balls (either by gender or number of therapies performed).

By pressing any given color on the legend, user can remove corresponding group from the plot, if needed (see figure I.7).

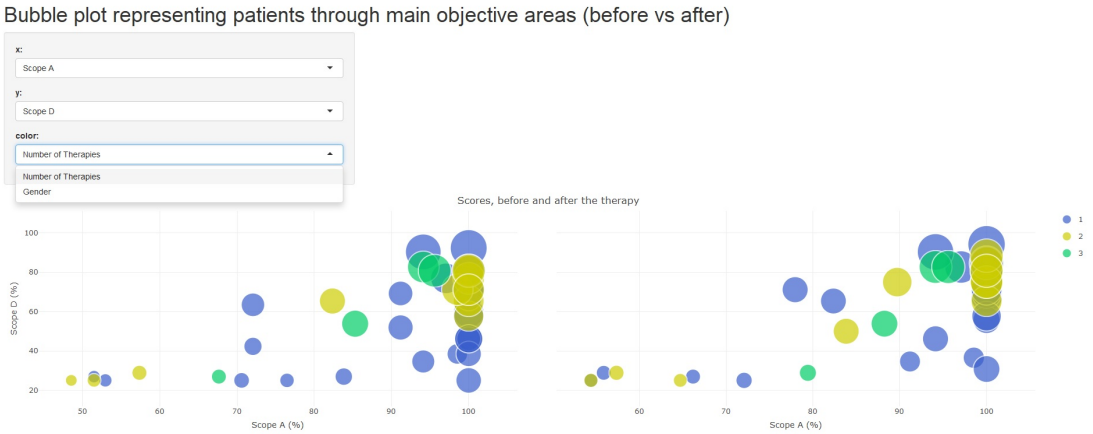


Figure I.6: Bubble-plot of patients in regard to GMFM-88 summary measures (1).

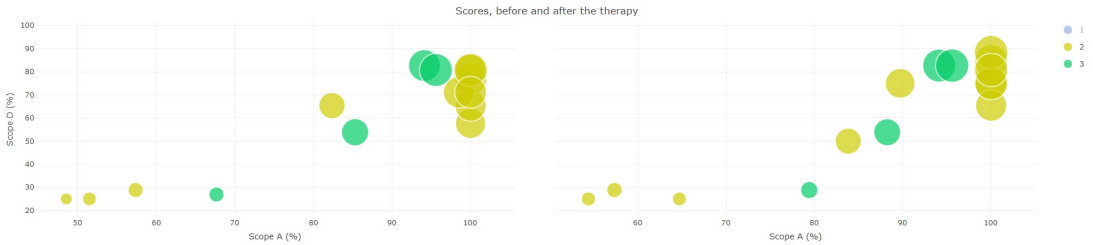


Figure I.7: Bubble-plot of patients in regard to GMFM-88 summary measures (2).

Figure I.8 represents Spearman Correlation Matrix between all pairs of GMFM-88 and its summary indicators.

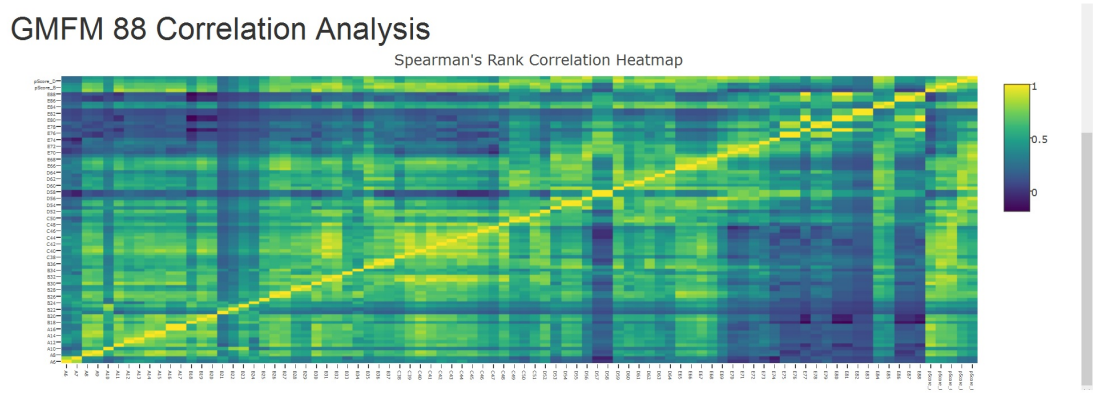


Figure I.8: Spearman Correlation Matrix.

I.4 Patients clustering

This section reports the fourth tab of the web-application, entitled *Patients clustering*. Figure I.9 represents the heat-map with those features that were selected to perform cluster analysis. In the map, every row stands for one therapy performed by a given

patient - **it is assumed that n successive therapies performed by the same patient are seen as n independent records of patients**. On the right-hand-side of this plot you can find clustering *dendrogram*, colored by the number of clusters selected in sidebar panel, which groups rows according to their similarity.

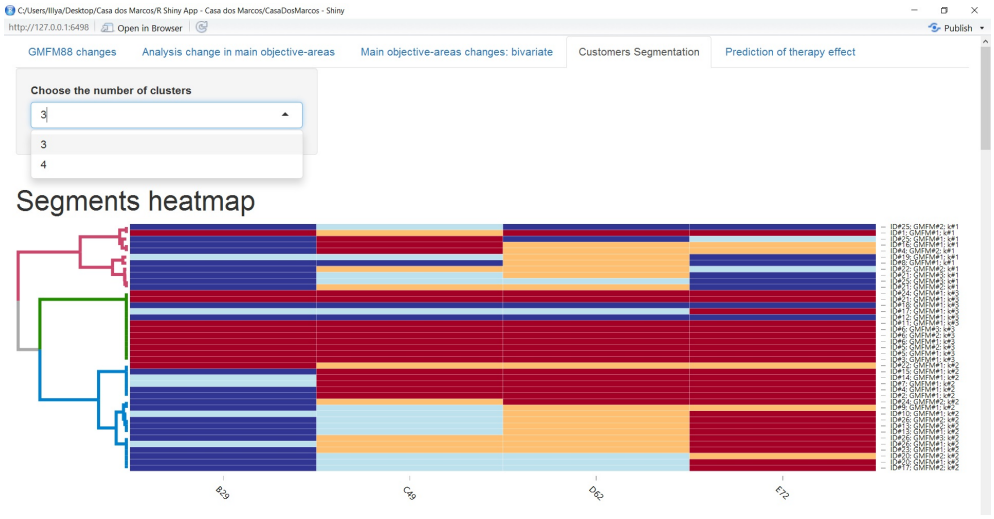


Figure I.9: Heat map for clustering of patients

Figure I.10 exhibits two bubble plots (similar to the ones in figure I.6), with one difference: color of each bubble represents the cluster where each patient belongs.

Figure I.10: Bubble-plot of patients in regard to GMFM-88, colored by clusters

In figure I.11, user can find summary statistics for pre-defined number of clusters. Also, a detailed table with records is provided.

I.5 Prediction of therapy effects

This section reports the fifth and last tab of the web-application, entitled *Prediction of therapy effects*. Figure I.12 represents the form that must be submitted to the application in order to perform prediction. Input values will form input vector for predictive models that run on the server side.

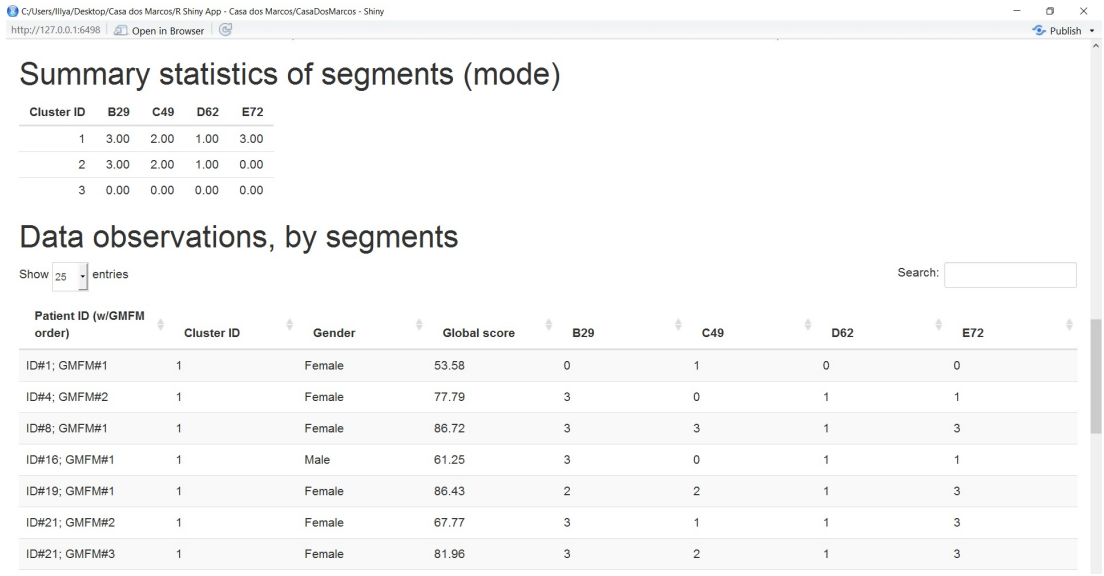


Figure I.11: Summary table of clusters

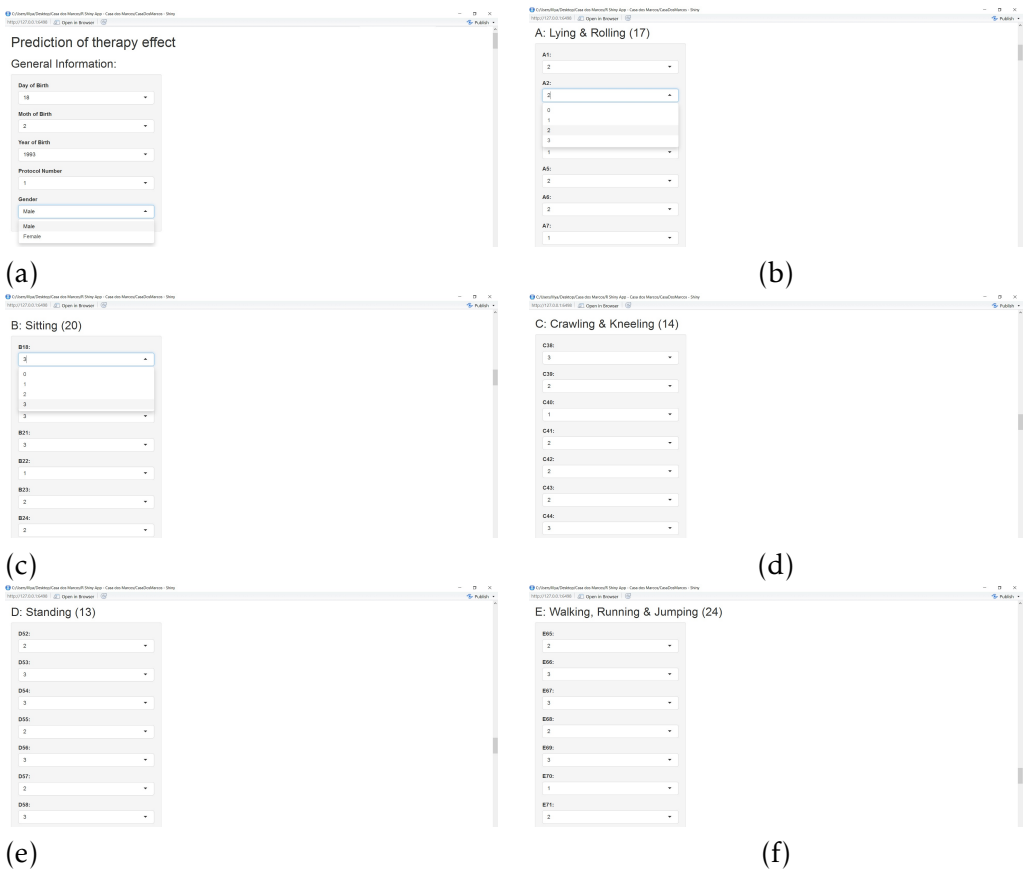


Figure I.12: Plot (a): screen-shot of socio-demographic form. Plot (b): screen-shot of GMFM-88 regarding objective-area A. Plot (c): screen-shot of GMFM-88 regarding objective-area B. Plot (d): screen-shot of GMFM-88 regarding objective-area C. Plot (e): screen-shot of GMFM-88 regarding objective-area D. Plot (f): screen-shot of GMFM-88 regarding objective-area E.

Figure I.13 represents calculated prediction for each objective area and global score. Each table outputs the real score before the therapy, expected score after the therapy and expected improvement computed as difference between previous two.

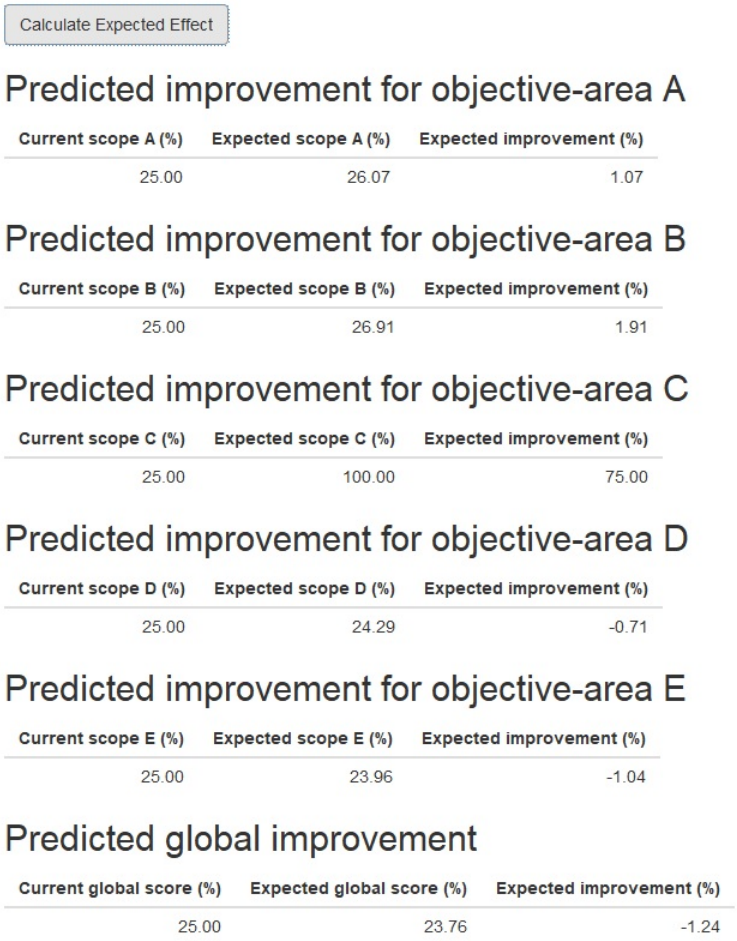


Figure I.13: Prediction tables for each one of six targets

Figure I.14 represents worth of all input features in regard to prediction of the target, function of their frequency in the final individual.

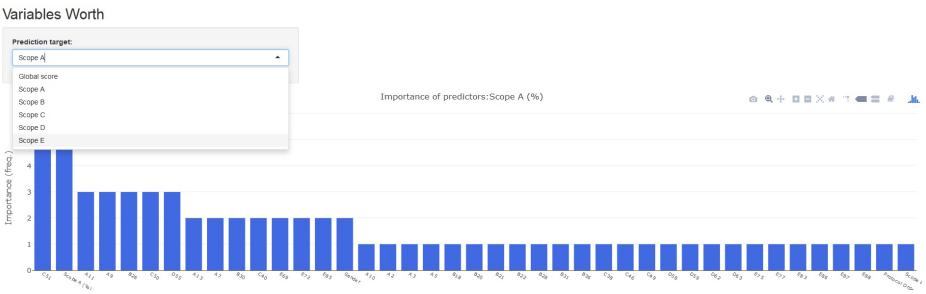


Figure I.14: Worth plot of input features for prediction of a given objective-area (features are presented in descending order)