Centro de Investigación y de Estudios Avanzados
del Instituto Politécnico Nacional

Unidad Cinvestav Tamaulipas

Nuevos Planteamientos Multi-Objetivo Para Resolver el Problema de Máxima Parsimonia.

Tesis que presenta:

Daniel Rafael Torres Avalos

Para obtener el grado de:

**Maestro en Ciencias
en Ingeniería y Tecnologías
Computacionales**

Dr. Eduardo Arturo Rodríguez Tello, Co-Director
Dr. Gregorio Toscano Pulido, Co-Director

Cd. Victoria, Tamaulipas, México. Diciembre, 2017

Center for Research and Advanced Studies
of the National Polytechnic Institute

Cinvestav Tamaulipas

New Multi-Objective Optimization Reformulations for Solving the Maximum Parsimony Problem.

Thesis by:

Daniel Rafael Torres Avalos

as the fulfillment of the
requirement for the degree of:

**Master of Science
in Engineering and Computing
Technologies**

Dr. Eduardo Arturo Rodríguez Tello, Co-Director
Dr. Gregorio Toscano Pulido, Co-Director

Cd. Victoria, Tamaulipas, México. December, 2017

The thesis of Daniel Rafael Torres Avalos is approved by:

____

Dr. Ricardo Landa Becerra

Dr. José Gabriel Ramírez Torres

Dr. Eduardo Arturo Rodríguez Tello, Committee Co-chair

Dr. Gregorio Toscano Pulido, Committee Co-chair

Cd. Victoria, Tamaulipas, México., December 8 2017

A mi familia y amigos.

Acknowledgements

- I thank CONACyT for the provided economic support which allowed me to concentrate in my studies and CINVESTAV - Unidad Tamaulipas for the opportunity to pursue graduate studies
- I thank my advisors, for their patience, their support , and their help during this research work.
- I thank the administrative personnel at CINVESTAV Unidad Tamaulipas for their help during my studies.
- I thank the administrative personnel at CINVESTAV-Tamaulipas for their help during my stay
- Thanks to my friends for their cheering and support.

Contents

List of Figures

List of Tables

List of Algorithms

Resumen

Nuevos Planteamientos Multi-Objetivo Para Resolver el Problema de Máxima Parsimonia.

por

Daniel Rafael Torres Avalos
Unidad Cinvestav Tamaulipas
Centro de Investigación y de Estudios Avanzados del Instituto Politécnico Nacional, 2017
Dr. Gregorio Toscano Pulido, Co-Director
Dr. Eduardo Arturo Rodríguez Tello, Co-Director

En bioinformática, el problema de construcción filogenética implica construir hipótesis evolutivas mediante árboles, usualmente binarios, en los que las hojas representan un set de n especies conocidas y los nodos internos son ancestros hipotéticos derivados de las mismas. El problema de *Máxima Parsimonia* (MP) consiste en la búsqueda o construcción de la topología de árbol para la cual los cambios evolutivos sean mínimos.

El problema MP está clasificado como un problema $\mathcal{NP}$-Completo, es altamente combinatorio, y el tamaño de su espacio de búsqueda crece factorialmente respecto al número de especies estudiadas. Existen estudios que indican que la complejidad del espacio de búsqueda también incrementa con el número de especies, añadiendo planicies y óptimos locales conforme las especies estudiadas aumentan. La multi-objetivización del problema presenta una alternativa para permitir que un algoritmo evolutivo encuentre soluciones competitivas con respecto a aquellas reportadas en el estado del arte.

En este trabajo de investigación se analizaron dos paradigmas de multi-objetivización para el problema de MP. Se presentaron seis re-formulaciones del problema basadas en la descomposición de la función objetivo original y doce basadas en la adición de funciones objetivo suplementarias.

Después de comparación experimental extensa se seleccionaron las tres propuestas mas prometedoras y se implementaron en el algoritmo NSGA-II, utilizando un algoritmo de cruza topológica y cinco funciones de vecindad.

Se realizó una comparación de las tres propuestas prometedoras contra un algoritmo mono-objetivo (evaluando solo MP), y contra algoritmos del estado del arte. Se utilizaron ocho instancias binarias reales, 14 instancias binarias sintéticas, y 19 instancias reales con caracteres multi-estado.

Los resultados obtenidos de la experimentación muestran que las propuestas de multi-objetivización presentadas son competitivas. Contra un algoritmo mono-objetivo que evalúa MP, la primer propuesta obtuvo resultados competitivos para el 71.4% de las instancias, la segunda para 64.2% de las instancias, y la tercera para 57.1% de las instancias. La comparación contra dos métodos representativos del estado del arte permite observar que la segunda propuesta iguala la calidad de los resultados para 52.6% de las instancias de prueba, y las otras dos igualan las soluciones de 47.3% de las instancias de prueba utilizando una fracción del tiempo requerido por los algoritmos del estado del arte.

Abstract

New Multi-Objective Optimization Reformulations for Solving the Maximum Parsimony Problem.

by

Daniel Rafael Torres Avalos
Cinvestav Tamaulipas
Center for Research and Advanced Studies of the National Polytechnic Institute, 2017
Dr. Gregorio Toscano Pulido, Co-advisor
Dr. Eduardo Arturo Rodríguez Tello, Co-advisor

In bioinformatics, the filogenetic construction problem consists in construting evolutionary hypothesis trough , usually binary, tree topologies, where the leaves represent a set of n known species and the inner nodes represent hypothetical ancestry derived from them. The Maximum Parsimony problem (MP) consists in the search or construction of the tree topologies for which the amount of evolutionary changes are minimal.

The MP problem is classified as $\mathcal{NP}$-Complete, it is highly combinatorial and its search space grows at a factorial rate as the number of known species increases. Published studies indicate that the complexity of the search space also increases with respect of the number of studied species, presenting more locally optimal solutions and regions with low gradient. A suitable alternative to approach this problem is the multi-objectivization.

In this research work we explored two multi-objectivization paradigms for the MP problem. First, we proposed six reformulations of the problem by means of decomposing the original objective function. Then, we presented twelve reformulations of the problem consisting in the adition of new *helper* objectives to the problem.

After an extensive experimental comparison, we selected three reformulations of the problem and implemented them using the NSGA-II algorithm. The implemented algorithm includes a topological based crossover function, and five different neighborhood functions.

A comparison was conducted between the proposed reformulations of the problem, a single-objective variation of the implemented algorithm (evaluating only MP), and two representative algorithms from the state of the art. For this comparison we used eight real-life binary-encoded instances, 14 synthetic binary-encoded instances, and 19 real-life multi-state character encoded instances.

The results from the experimentation show the competitiveness of th presented proposals. Against the single-objective variation, the first reformulation presented competitive solutions for 71.4% of the instances, the second was competitive for 64.2% of the instances, and the third for 57.1% of them. The comparison against the state of the art shows that the second proposal equals the results of 52.6% of the used instances, and the remaining two for 47.3% of them, employing only a fraction of the time required by the state of the art algorithms.

Nomenclature

DNA Deoxyribonucleic Acid.

MOEA Multi-Objective Evolutionary Algorithm.

MP Maximum Parsimony.

Multi-objectivization Re-statement of a single-objective problem in an alternative multi-objective formto facilitate the process of finding a solution.

Newick Format Representation of a tree topology as plain text using nested parentheses, each pair of parenthesis represents an inner node of the tree, for which its branches are separated by commas and terminal nodes are represented as an identifier. The terminal symbol for a Newick formatted tree is a semi-colon.

NSGA-II Non-dominated Sorting Genetic Algorithm II.

Phylogeny Study of evolutionary relationships between organisms by means of tree-like representations.

RNA Ribonucleic Acid.

Taxon Each of the species or sequences at the tip of a branch in a phylogenetic tree. It is a group of one or more populations of an organism or organisms seen by taxonomists to form a unit, *Operational Taxonomic Unit* (OTU). Plural:*Taxa*

Transition Substitution of a purine or a pyrimidine by another purine or pyrimidine, respectively.

Transversion Substitution of a purine by a pyrimidine or vice-versa.

1

Introduction

This chapter presents a brief introduction to bioinformatics and the MP problem, the hypothesis about the multi-objectivization of the problem that motivates this research, and the main objective pursued. At the end of the chapter, an outline of this document with a brief description of the remaining chapters is presented.

1.1 Background

Bioinformatics is an interdisciplinary field that develops and applies computational and statistical tools to solve practical and theoretical problems derived from working with information related to biological macromolecules such as deoxyribonucleic acid (DNA), ribonucleic acid (RNA), and proteins (Xiong, 2006). Some of the main areas of research in bioinformatics are:

- Molecular structural analysis, which includes protein and nucleic structure analysis, comparison, classification, and prediction.

- Molecular functional analysis, which includes gene expression profiling, protein interaction prediction, metabolic pathway reconstruction, and simulation.

- Sequence analysis, which includes sequence alignment, sequence database searching, motif and pattern discovery, and reconstruction of evolutionary relationships.

The reconstruction of evolutionary relationships, under the sequence analysis research, is the main focus of this thesis work. Haeckel (1866), was the first to propose an evolutionary theory to explain the origin of the first living cell, and to propose a tree-like classification to reflect the evolutionary ancestry of a set of known organisms (also called *taxa*), for which he coined the term *phylogeny* (Dayrat and Linder, 2003).

Phylogenetic reconstruction focuses on finding similarities among a set of known organisms, which are assumed to have a common ancestor, in order to infer their evolutionary relationships. For this kind of analysis there are two main approaches: *Morphological comparison*, which is based on physical and measurable characteristics (e.g., anatomy, physiology, behavior) of the studied set of species, and *biochemical comparison*, that studies similarities in the sequences of nucleic acids and proteins of the known species.

Several inference methods for phylogenetic reconstruction problem have been proposed in the literature. One of the most widely known, and used for its simplicity of evaluation, is the Maximum Parsimony (MP) method. This method assumes that the hypothesis that presents the simplest explanation for the evolutionary relationships of the known species has a higher probability of being a true explanation (*Occam's razor principle*) (Xiong, 2006).

The MP method attempts to maximize the evolutionary similitude among the inferred individuals inside the proposed tree topology, which means finding the topology that requires the least number of evolutionary changes (i.e., character state transformations) to explain the differences among the sequences that represent the known species (Kluge and Farris, 1969).

1.2 Motivation

There are several reasons for the study of phylogenetic trees construction. On the side of biological sciences, they are used in the process of developing vaccines, antibacterial agents, herbicides, and the study of microbial communities (Pace, 1997). In pharmaceutics, phylogenetic trees are used in the smart development of new drugs. Likewise, in the field of molecular biology, phylogenetic trees are used in the prediction and classification of protein sequences, determining the homology of a sequence, classification of proteins, among others. (Murakami and Jones, 2006).

On the other hand, the main motivation of this research work lies within the field of computational science. The MP problem has been proven to be $\mathcal{NP}$-Complete, being equivalent to the Steiner tree problem in the hypercube (Gusfield, 1997). It is a highly combinatorial problem whose search space's growth rate (the number of possible binary rooted trees constructed with n known species) is factorial according to the expression $\mathscr{T} = (2n-3)!/(2^{n-2}(n-2)!)$, where n represents the number of known species (taxa) (Xiong, 2006), this is illustrated graphically in Figure 1.1. It was observed by Kirkup and Kim (2000), that as the taxa number increases, so does the number of attraction basins that lead to different local optimum solutions, and that different locally optimal solutions might diverge from one another's topology from 40% to 50%, and still only have from 0.5% to 1.0% discrepancy in their parsimony scores. Furthermore, the existence of tree islands (collections of trees with similar topologies and with parsimony scores that fall under an upper bound) (Maddison, 1991), and terraces of trees (groups of trees that are at distance one and have the same parsimony score) (Sanderson and McMahon, 2011), give a hint of high neutrality in the search space of the problem. These characteristics make it difficult, for search algorithms, to obtain near optimal solutions in small time frames.

A known approach to problems with difficult search spaces (i.e., multiple local optima, rugged or neutral landscapes) is the re-formulation of the problem to incorporate more than one objective funcion, this is the multi-objectivization.

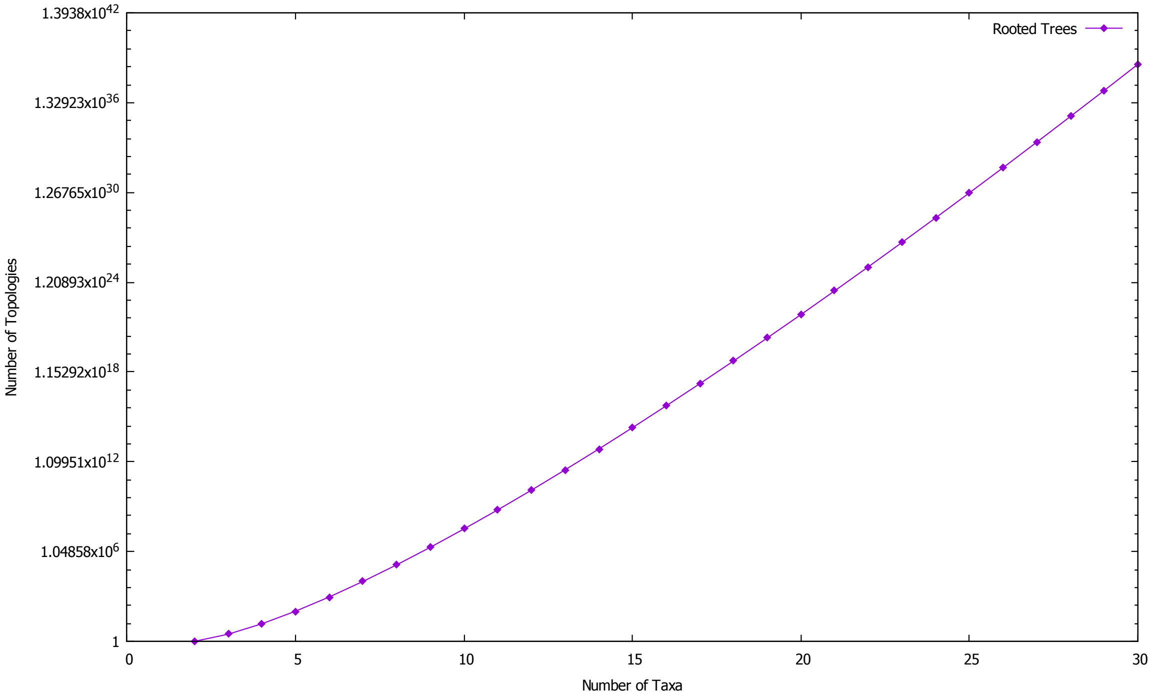


Figure 1.1: Number of rooted tree topologies as the number of studied taxa grows. Values on the y-axis are in log scale.

Knowles et al. (2001) and Jensen (2003) have studied different approaches to multi-objectivizate a single-objective problem. Knowles et al. propose the decomposition of an original objective function in order to enhance the results of searches conducted in spaces with multiple local optima. On the other hand, Jensen uses *helper objectives* to approach problems whose original function was hard to decompose.

Because of the characteristics of its search space it is evident that finding good quality solutions in a small time frame is a challenge as the number of taxa grows. In the state of the art there are few multi-objective formulations of the problem, and most of them include computationally expensive evaluations. Therefore, proposing new objective functions to re-formulate the MP problem, that are more cost efficient is an area of opportunity.

1.3 Hypothesis

This research attempts to verify the following hypothesis:

There is at least one multi-objective formulation of the Maximum Parsimony problem, whether by decomposing the original evaluation function or adding helper objectives, that allows a multi-objective evolutionary algorithm to have an easier navigation trough the search space of the problem by modifying its fitness landscape; allowing it to find solutions that are competitive to those reported in the state of the art of the problem, either by improving the quality of the found solutions or by reducing the computational time required to obtain them.

1.4 Objectives

Below, we present the main objective of this research and those specific objectives that will help with its fulfillment.

1.4.1 General objective

The main goal of this research work is to contribute to the state of the art of the MP problem, with the proposal and validation of at least one new multi-objective reformulation of the problem. This reformulation should allow a multi-objective evolutionary algorithm (MOEA) to find solutions better than those found by a single-objective approach, that compete favorably with those achieved by the existing reference methods.

1.4.2 Specific objectives

In order to fulfill the general objective of this research work the following specific objectives are defined:

1. To propose at least one new multi-objective reformulation for the MP problem, whether by decomposing of the original objective function or by adding helper objective functions, that allows to discern between solutions with the same parsimony score.

2. To efficiently implement the proposed reformulation of the problem in a MOEA, that finds competitive solutions to those achieved by a single-objective algorithm in at least 50% of the test cases.

3. To make a comparative analysis against the reference methods existing in the state of the art (both single and multi-objective). To assess the practical usefulnes of the proposed reformulation of the problem, it must find competitive solutions for at least 50% of the used instances.

1.5 Thesis organization

The remainder of this thesis work is organized as follows:

- Chapter 2 - Background. This chapter provides the formal definition of the Maximum Parsimony problem, its complexity and the representation used for the analyzed taxa. It also provides an overview to the application of multi-objectivization.

- Chapter 3 - State of the art. This chapter reviews the literature of the MP problem, the works of other authors are described and compared. Furthermore, the usual instance benchmarks for the problem are described.

- Chapter 4 - Multi-objectivization of the problem. This chapter presents the proposal, evaluation, and selection of new multi-objective formulations for the MP problem. It describes the proposed objective functions and the exhaustive evaluation applied to select the most promising ones.

- Chapter 5 - Implementation of athe selected MOEA. This chapter describes the algorithms implemented in order to assess the performance of the selected multi-objective formulations. It includes the description of the genetic operators (crossover and mutation) and the components of the NSGA-II algorithm, which was chosen to assess the performance of the proposed multi-objectivizations of the MP problem.

- Chapter 6 - Experimentation and results. This chapter presents the results of the experimentation conducted with the implemented MOEA and its analysis. The results are divided in two parts: first we contrast its performance against a single-objective evolutionary algorithm in terms of the quality of the best solutions found, then the comparison against the state of the art algorithms in the literature is presented, Finally, the results are analyzed.

- Chapter 7 - Conclusions and future work. Contains the conclusions reached during this research and the future work derived from it.

2

Background

This chapter presents the formal definition of the MP problem, including its computational complexity and the growth rate of its search space; the codification of the parsimony sequences used to represent known species; and an overview of the multi-objectivization technique.

2.1 Problem statement

A rooted phylogenetic tree for a set of n operational taxonomic units (OTUs); represented as n aligned sequences $\vartheta = \{S_1, S_2, \ldots, S_n\}$ of size k, in which each of the k_i sites is a set of possible states over an alphabet α; is defined as a binary tree $T = (V, E)$. Here, $V = \{I, L\}$ are the nodes of the tree; where L represents the leafs that contain the information of the known OTUs (ϑ), and I are the inner nodes that represent the hypothetical ancestry that is inferred with the information of its descendants; and E represent the evolutionary relationships between them.

Using a Maximum Parsimony (MP) criterion to construct phylogenetic trees aims to build hypothesis with the least number of evolutionary changes, based on the principle of Occam's razor

(Xiong, 2006): "The simplest hypothesis as an explanation is more likely to be the true one", which suggests that the simplest tree (i.e., with the least number of mutations or changes) has a higher probability of being right.

The evaluation of a phylogenetic tree using MP requires that every sequence in an inner node of the tree has an assignment of state sets inferred from the nodes connected to them. This assignment is applied using Fitch's algorithm (Fitch, 1971). Fitch's algorithm is implemented as a two step process: first, a bottom-up propagation of the known values that computes the character state sets for every position $1 \leq i \leq k$ of the sequences in the inner nodes of the tree, which we denote as *parsimony sequences* (p). The parsimony sequence $p_w = \{z_1, z_2, \ldots, z_k\}$ for each inner node $w \in I$ whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, is calculated as:

$$\forall i,\ 1 \leq i \leq k,\ z_i = \begin{cases} x_i \cup y_i, & \text{if } x_i \cap y_i = \emptyset, \\ x_i \cap y_i, & \text{otherwise.} \end{cases} \tag{2.1}$$

The second step consists in assigning final character states to every parsimony sequence of the tree. This is a top-down process that starts at the root of the topology, for every position $1 \leq i \leq k$ of the sequence at the root of the tree a state is selected, then, for every inner node from the root of the tree to the leaves' parents, similar states are selected if they exist within the state sets of the child node, else a random state within the child's set is selected.

Once the character state sets for the entire topology have been selected the tree can be evaluated under MP. For every inner node $w \in I$ of the tree, whose children are represented by the sequences $S_u = \{x_1, x_2, \ldots, x_k\}$ and $S_v = \{y_1, y_2, \ldots, y_k\}$, the parsimony cost is calculated as:

$$\phi(p_w) = \sum_{i=1}^{k} c_i, \text{ where } \begin{cases} c_i = 1, & \text{if } x_i \cap y_i = \emptyset, \\ c_i = 0, & \text{otherwise.} \end{cases} \tag{2.2}$$

Where c_i represents the cost at position i of the parsimony sequence p_w. And thus, the parsimony cost for the entire tree is given by:

$$\Phi(T) = \sum_{w \in I} \phi(p_w). \tag{2.3}$$

Solving the MP problem implies finding the tree T^* where $\Phi(T)$ is minimal (Vazquez Ortiz and Rodriguez Tello, 2011), i.e.,

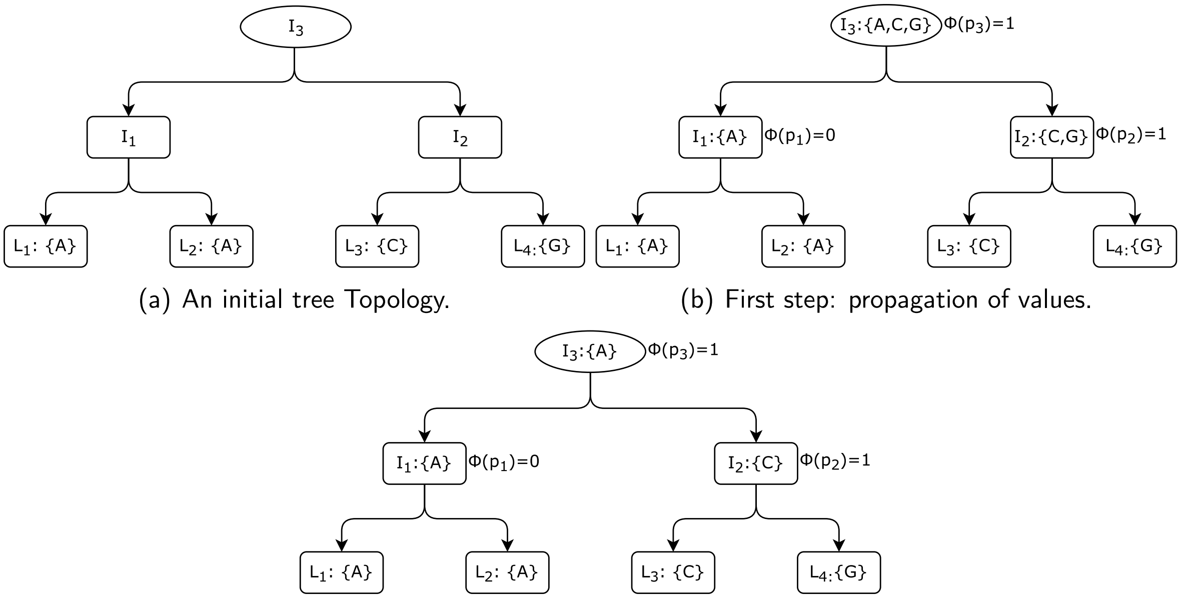
$$T^* = \arg\min_{T \in \mathscr{T}} \{\phi(T)\}.$$

where $\mathscr{T}$ represents the search-space of all the possible roted trees constructed with the analyzed taxa.

An example of the selection of character states and evaluation of the topology,for a set of four taxa $n = 4$ of size $k = 1$, can be seen in Figure 2.1. The tree topology in Figure 2.1(a) has assigned only the leaf nodes as the known taxa of size k, represented by a set of one possible state each. In Figure 2.1(b) the parsimony sequences for each internal node have been calculated: for I_1 the set of states remains A because $L_1 \cap L_2 \neq \emptyset$ and its parsimony cost ($\phi(p_1)$) remains zero. On the other side, for I_2 the intersection of the state sets is nonexistent, therefore the states assigned to I_2 are C and G and the parsimony score of that sequence is $\phi(p_2) = 1$, same applies to the root of the tree. In Figure 2.1(c) the selection of the final states takes place, the parsimony costs for each internal node remain the same. In this example, the parsimony score of the entire tree is $\Phi(T) = 2$.

2.2 Codification of the parsimony sequences

We use the term *Parsimony Sequence* to refer to a sequence assigned to an inner node in a phylogenetic tree. The length of this sequence is the same as the length of those assigned to the node's children, and the values at each i-th position of this sequence depend solely on the i-th position of the children assigned to the node. A parsimony sequence p represents the information of an inferred ancestor that depends on the known information of its successors.



(a) An initial tree Topology.

(b) First step: propagation of values.

(c) Second step: selection of final states.

Figure 2.1: Graphic example of the Fitch's algorithm for the selection of character state sets and the evaluation of a tree topology using MP.

The codification used for these sequences depends on the data stored in the input aligned sequences that represent the studied taxa, usually being of two types: using binary data (for morphological information) or using multi-state character data (for molecular data).

If the aligned sequences represent morphological information, each site of the sequence represents a characteristic, and the value assigned to it indicates whether the studied OTU presents such characteristic (1) or not (0).

Otherwise, if the aligned sequences represent molecular data, the possible symbols for each site have a wider alphabet. If the aligned sequences represent proteins, each of the symbols represent a codon (a triplet of nucleotides) as shown in Table 2.2, otherwise, the assigned values represent the nucleic acids present in each position, so the numeric values assigned to each symbol in the alphabet α are as shown in Table 2.1. The values for the four nucleic base symbols are the first four power of two values (1, 2, 4, and 8), the remaining symbols are assigned the sum of the values of the combined bases they represent, a gap (-) represents the deletion or insertion of a base, *N* represents the possible presence of all four bases, and the $?$ symbol represents either the presence or the absence of any base, the parsimony cost of a site with the $?$ symbol is always 0. This particular assignment of values enables the use of bitwise operators to calculate parsimony sequences and their costs.

Symbol	Nucleotide	Integer Value
A	Adenine	1
C	Cytosine	2
G	Guanine	4
T (or U)	Thymine (or Uracil)	8
M	A or C	3
R	A or G	5
S	G or C	6
W	A or T	9
Y	C or T	10
K	G or T	12
V	A or C or G	7
H	A or C or T	11
D	A or G or T	13
B	C o G o T	14
N	Any base	15
-	Gap	16
?	Any symbol	255

Table 2.1: Numerical values assigned to every symbol in the alphabet of an aligned sequence or parsimony sequence for nucleic acids (Liébecq, 1992; Metanomski, 1991).

The adopted encoding represents the symbols as integer values. The characteristic that each subsequent symbol is the sum of the value of the contained base nucleotides, enables to treat each position either as an integer value or as a set of nucleotides, which might yield relevant information for further analysis.

2.3 Multi-objectivization

The term multi-objectivization was originally coined by Knowles et al. (2001) to name a reformulation of a single objective problem using two or more evaluation functions. This might be achieved by adding complementary *helper* objectives (Jensen, 2005), or decomposing the original evaluation function (Handl et al., 2008). Multi-objectivization, regardless of the reformulation method, modifies the fitness landscape, affecting the performance of the algorithms at hand. If a reformulation is achieved by adding complementary objectives it is said to be a multi-objective problem of the form:

$$\mathbf{f}(x) = [f(x), g_1(x), \ldots, g_h(x)]^T,$$

where f is the original evaluation function of the problem, and g_i is the i-th complementary objective, $1 \leq i \leq h$ (Garza Fabre et al., 2015).

When the multi-objectivization is achieved by decomposition, the original function is divided into separate components, each of them treated as a different objective function, creating a problem of the form:

$$\mathbf{d}(x) = [f_1(x), f_2(x), \ldots, f_d(x)]^T,$$

where the sum of the $d \geq 2$ objectives equals the original function (Garza Fabre et al., 2015).

In multi-objective optimization, the most used approach to determine the quality of a solution is based on *dominance*. According to the Pareto approach, a solution a dominates a solution b if a is not worse than y in all objectives and is better in at least one of them. Solving a multi objective problem implies finding the Pareto optimal solutions that represent a trade-off among objective functions. In a front of Pareto optimal solutions all of them have the same importance (Cancino and Delbem, 2007).

Handl et al. (2007), proposed five different contexts in which a problem might be multi-objectivized:

1. **Standard.** This category refers to problems where multiple objectives are clearly defined and are optimizable.

2. **Counterbalance for bias.** An aditional objective is introduced to counterbalance an existing

bias in the first objective.

3. **Multiple source integration.** This category is used to integrate multiple data-sources that might generate noise when combined.

4. **Performance approximation by proxies.** This category is used when some of the variables (needed to estimate the quality of a solution) are not available during the optimization process. In this case, some proxy objectives are used to capture some good aspects of the obtained solutions.

5. **Multi-objectivization** This category refers to the use of Multi-Objective Optimization (MOO) solely to guide the search in a single-objective problem, this is used when:

   - The problem presents great amounts of local optima in the fitness landscape, in which case the decomposition of the original objective function might help reducing the amount of local optima (Knowles et al., 2001).
   - The search landscape presents flat regions (with no gradient), in which case the incorporation of supplementary objectives might guide a search algorithm trough low gradient regions (Jensen, 2003; Knowles et al., 2001)

2.4 Chapter summary

This chapter presented the formal definition for the MP problem, the codification of solutions, and the definition and classification of multi-objectivization. According to the classification proposed by Handl et al., the approach used in this thesis work falls within the fifth category, the MP problem presents complex fitness landscapes and the decomposition of the original objective function, or the addition of new supplementary objectives could be used to guide the search toward better solutions.

The following chapter reviews the relevant literature of the problem, including the state of the art algorithms and the current multi-objective formulations of the MP problem.

Symbol	Three letter code	Amino acid	Possible codons
A	Ala	Alanine	GCA, GCC, GCG, GCT
B	Asx	Aspartic acid or Asparagine	AAC, AAT, GAC, GAT
C	Cys	Cysteine	TGC, TGT
D	Asp	Aspartic acid	GAC, GAT
E	Glu	Glutamic acid	GAA, GAG
F	Phe	Phenylalanine	TTC, TTT
G	Gly	Glycine	GGA, GGC, GGG, GGT
H	His	Histidine	CAC, CAT
I	Ile	Isoleucine	ATA, ATC, ATT
K	Lys	Lysine	AAA, AAG
L	Leu	Leucine	CTA, CTC, CTG, CTT, TTA, TTG
M	Met	Methionine	ATG
N	Asn	Asparagine	AAC, AAT
P	Pro	Proline	CCA, CCC, CCG, CCT
Q	Gln	Glutamine	CAA, CAG
R	Arg	Arginine	AGA, AGG, CGA, CGC, CGG, CGT
S	Ser	Serine	AGC, AGT, TCA, TCC, TCG, TCT
T	Thr	Threonine	ACA, ACC, ACG, ACT
U	Sec	Selenocysteine	
V	Val	Valine	GTA, GTC, GTG, GTT
W	Trp	Tryptophan	TGG
X	Xaa	unknown or 'other'	NNN
Y	Tyr	Tyrosine	TAC, TAT
Z	Glx	Glutamic acid or Glutamine	
*	*(Ter)	Termination	TAA, TAG, TGA

Table 2.2: Codons assigned to every symbol in the alphabet of an aligned sequence or parsimony sequence for proteins (Liébecq, 1992; Metanomski, 1991).

3

State of the art

This chapter reviews the most relevant works related to this research: First, it describes the construction methods for phylogenetic trees, then it presents algorithms based on maximum parsimony, and finally, it reviews the current multi-objective works in the literature referent to the MP problem as a main objective.

3.1 Methodologies for phylogenetic trees construction

Construction of phylogenetic trees can be divided in two groups: distance-based methods and character-based methods.

3.1.1 Distance-based methods

Distance-based methodologies measure the differences between the sequences that represent the known species as an evolution criterion. These methodologies can be classified as group methods or optimality criterion methods.

Group methods have their main exponent in the unweighted pair group method using arithmetic average (UPGMA) proposed by Sokal and Michener (1958). UPGMA begins with a distance matrix that includes all the known taxa, then, it applies an iterative process, where it groups the closest pair of nodes in the matrix; and once the nodes have been grouped, an inferred ancestor is created and replaces the grouped nodes in the distance matrix. The process is repeated until the last two nodes are grouped in a root node, similar to the process followed by a hierarchical agglomerative clustering algorithm (Kaufman and Rousseeuw, 1990). UPGMA is a simple methodology that assumes that the evolution rate of the species is constant; therefore the resulting tree might not be a true hypothesis.

On the other hand, the main exponent of the optimality methods is the Fitch-Margoliash method (Fitch and Margoliash, 1967). This method aims to reduce the standard deviation of the distances between species present on a constructed tree, from those in a x| matrix of the known species, obtaining a statistically optimal tree. Even if the method is able to find an optimal tree, its calculation is inefficient and not viable for large datasets.

3.1.2 Character-based methods

The most widely used character-based methods are maximum likelihood (ML) and maximum parsimony (MP).

The maximum likelihood method uses evolutionary models to assess the probability that a tree represents the true evolutionary history of the set of known species. An example of an evolutionary model is the Jukes-Cantor model (Jukes and Cantor, 1969), in which the probability P that a nucleotide in a DNA sequence does not mutate after a time t is $P(t) = \frac{1}{4} + \frac{3}{4e^{-\sigma t}}$ where σ is the substitution rate of a nucleotide, which might be empirically inferred or assigned based on a previous analysis of the species. With this model the likelihood of a phylogenetic tree is measured as the sum of the likelihood at every position of the sequences in every level of the tree. When every tree is evaluated, the tree with the biggest likelihood value is the accepted one (Xiong, 2006).

Phylogenetic tree construction using MP aims to build trees with a minimum number of evolutionary changes (tree length) (Xiong, 2006). Among the biological community there are specialized software tools for phylogenetic tree construction based on parsimony, among these tools the one with the best known performance is Tree analysis using New Technology (TNT[1]) developed by Goloboff et al. (2008).

Knowing diverse approaches to the problem should be useful when proposing complementary objectives, therefore we present the most relevant information we have found in the literature of the MP problem.

3.1.3 Exact methods

Exact methods explore the search space to find the optimal solution. However, they are only plausible for small instances of the problem.The most common exact methods are *exhaustive search* and *branch and bound* algorithms (B&B).

Exhaustive algorithms explore every existent tree topology in the search space and returns the one with the lowest objective function score. In practice, they are not widely used because their utility is limited due to the factorial growth rate of the search-space of the problem.

B&B algorithms present a considerable reduction in computing time with respect to an exhaustive search by setting upper bounds on the cost of constructing a phylogenetic tree and automatically discarding those that break the limit. In 2006, Bader et al. proposed a parallelized B&B algorithm that solves datasets up to 27 taxa (Bader et al., 2006).

[1] http://www.lillo.org.ar/phylogeny/tnt/

Later, White and Holland (2011) published XMP (*Exact Maximum Parsimony*). By optimizing the parsimony score and upper bound calculations they can solve instances of up to 35 taxa in less than 20 minutes (or less than 1 minute running a parallelized version of the algorithm in a 256 cores machine).

3.1.4 Approximation algorithms

When using a tree representation for the solutions of the MP problem, the main neighborhood functions used in local search and mutation algorithms are:

- *Nearest Neighbor Interchange* (NNI) (Andreatta and Ribeiro, 2002; Moore et al., 1973; Vazquez Ortiz and Rodriguez Tello, 2011; Waterman and Smith, 1978): Proposed by Moore et al. (1973). NNI swaps inner branches of the tree that have a distance of up to 1 between them, generating a relatively small neighborhood ($2n - 6$ neighbors for n taxa) (Allen and Steel, 2001).

- *Subtree Pruning and Regraft* (SPR) (Andreatta and Ribeiro, 2002; Vazquez Ortiz and Rodriguez Tello, 2011): It prunes an inner node of the tree and inserts it back in a random position, generating up to $2(n - 3)(2n - 7)$ neighbors (Allen and Steel, 2001).

- *Tree Bisection and Reconnection* (TBR) (Swofford et al., 1996; Vazquez Ortiz and Rodriguez Tello, 2011): It divides the tree into two subtrees and reconnects them in any of their branches, generating up to $(2n - 3)(n - 3)^2$ neighbors (Allen and Steel, 2001).

- *Leaf Swap* (LSwap) (Cotta and Moscato, 2002; Sonco Alvarez and Ayala Rincon, 2017): It selects and swaps two random leaves in the tree, generating up to $n(2n - 4)$ neighbors.

- *Single Step* (STEP) (Andreatta and Ribeiro, 2002; Sonco Alvarez and Ayala Rincon, 2017; Waterman and Smith, 1978): A leaf is pruned from the tree and inserted in any other edge. It generates up to $n(n - 1)$ neighbors.

Congdon (2001) published a genetic algorithm for the MP problem called GAPhyl, this algorithm mixes the source code of PHYLIP[2], a free phylogenetic analysis package from Washington University, and Genesis[3], a genetic algorithm support library. GAPhyl uses a crossover method that selects a subtree from the first parent, removes the leafs of such a subtree from the second parent and then inserts the subtree in a random inner node. This crossover operation is aimed to partially maintain the information of the parent's inferred ancestry. GAPhyl also uses a group of isolated populations that "migrate" after a given number of iterations in order to avoid convergence problems. This algorithm improved solutions reported in the literature. However, its temporal cost is relatively high.

Cotta (2006) proposed a scatter search algorithm that uses Prune-Delete-Graph (GAPhyl's crossover function) as a diversification and resetting operation. It also applies a Path-Relinking algorithm to unify solutions inside the population. This algorithm provides near optimal solutions. Temporal cost might be reduced by adjusting the maximum iterations but it has a direct impact in the performance and the quality of the final solution.

Goëffon et al. (2006) published a memetic algorithm called Hybrid Distance Recombination Algorithm (HYDRA). The crossover mechanism used by this algorithm adds up the distance matrices that represent both parent solutions and then constructs the child solution with a stochastic variant of the UPGMA algorithm. The mutation mechanism applies a descent algorithm with progressive neighborhoods (a modification of a SPR neighborhood with variable distance that ranges from exploring all the possible changes to a NNI neighborhood) (Goëffon et al., 2008). Hydra's performance was measured by comparing the obtained solutions on 28 instances (8 real instances and 20 randomly generated ones) with the results given by the analysis tool TNT. Both methods produced the same parsimony scores in the real instances, byt HYDRA improved the score on 19 artificially generated instances.

[2] http://evolution.genetics.washington.edu/phylip.html
[3] https://www.bioconductor.org/packages/devel/bioc/html/GENESIS.html

Richer et al. (2013), introduced an alternative to Hydra by using a simulated annealing algorithm called Simulated Annealing for Maximum Parsimony (SAMPARS). This algorithm uses SPR and TBR neighborhoods with a stochastic descent algorithm. SAMPARS uses a geometrical cooling strategy that determines the amount of visited solutions in each neighborhood. SAMPARS improved 11 of the solutions reported by Hydra while reaching the same parsimony scores for the rest of the remaining instances with lower computational time.

3.2 Multi-objective methodologies

In the specialized literature of the problem there are some multi-objectivization proposals for the MP problem, most of them have adapted supplementary objectives trough different existent construction criteria (usually maximum likelihood).

Cancino and Delbem (2007) proposed PhyloMOEA, a multi-objective algorithm based on NSGA-II that uses maximum likelihood and maximum parsimony as objectives. This method produces a Pareto front of non-dominated solutions with the best parsimony scores found, eliminating repeated parsimony costs using likelihood as a differentiation criteria. The approach was tested on instances: *rbcL_55*, *mtDNA_186*, *RDPII_218* and *ZILLA_500*, with numbers of taxa in the range of $55 \leq n \leq 500$ and informative sites in the range of $1314 \leq k \leq 4128$. The long execution time of this algorithm is a disadvantage because for some given initial solutions the convergence of the algorithm could take several hours.

Coelho et al. (2010) implemented an adaptation of a multi-objective artificial immune system. Such an implementation uses distance matrices to calculate the standard deviations from a current tree to the theoretical minimum distance tree and minimizes this assessment as a secondary objective.

Santander and Vega proposed three multi-objective adaptations for bio inspired algorithms that improved the results obtained by PhyloMOEA

- An implementation of a multi-objective firefly algorithm (MOFA) (Santander Jiménez and Vega Rodríguez, 2013a) that uses maximum parsimony and maximum likelihood as objectives. This adaptation uses the progressive neighborhood proposed by Goëffon et al. (2006).

- A multi-objective artificial bee colony algorithm (MOABC) (Santander Jiménez and Vega Rodríguez, 2013b) that uses maximum parsimony and maximum likelihood.

- An indicator based multi-objective bat algorithm (IMOBA) (Santander Jiménez, 2016) using maximum parsimony and maximum likelihood.

Santander and Vega used the same dataset used by Cancino and Delbem (2007), additionally there are results reported for instances *HIV2_72*, *membracidae_81*, *HIV1_192*, and *S1482* with $72 \leq n \leq 192$ species, and $817 \leq k \leq 3321$ informative sites.

The most common approach for multi-objectivizing the MP problem in the literature consists in evaluating the likelihood of a topology under different substitution models. Barry and Hartigan (1987) proposed a method that attempted to maximize likelihood and parsimony in tree estimations called "Most parsimonious likelihood". They noted that this combination may produce inconsistent estimations. Furthermore, experimentation conducted by Yang (1996) showed that, for certain cases, using a "wrong" substitution model for a group of taxa can yield topologies closer to the true tree than applying the "correct" model, proving inconsistency in the use of likelihood.

3.3 Chapter summary

The reviewed multi-objective algorithms of the state of the art are mainly focused in the first category proposed by Handl et al. (2007). They use defined and measurable objectives in order to pursue a trade-off in the Pareto front of the solutions.Table 3.1 presents a comparison between the algorithms enumerated in this chapter, mentioning the main innovations presented and the weaknesses observed for each of them. For the multi-objective approaches used to multi-objectivize the MP problem, most of them (Cancino and Delbem, 2007; Santander Jiménez, 2016; Santander Jiménez and Vega Rodríguez, 2013a,b) use variations of the Likelyhood score of a tree as a secondary objective, and one of them (Coelho et al., 2010) uses minimization of the mean squared error, both of them being expensive evaluations to conduct over a tree.

Year/Author	Algorithm	Innovations	Weaknesses
2006, Bader et al. [3]	Branch & Bound	Shared memory parallelization	Limited to 27 taxa
2011, White and Holland [70]	Exact maximum parsimony (XMP)	Sequential and parallel implementation, 36 taxa in 20 minutes less than a minute running in 256 cores	Limited to 36 taxa
2001, Congdon [10]	GAPHYL (Genesis + PHYLIP)	Isolated populations and migration movements, Prune-Delete-Graph	Long computational time
2006, Cotta [11]	Scatter search	Prune-Delete-Graph + Path Relinking	Long computational time
2006, Goëffon et al. [21]	Memetic algorithm (HYDRA)	Distance matrices crossover and progressive neighborhoods	Lack of real instances in the experimentation
2013, Richer et al. [52]	Simulated annealing (SAMPARS)	SPR and TBR Neighborhoods and a descent algorithm applied with a low probability	Lack of real instances in the experimentation
2007, Cancino and Delbem [8]	PhyloMOEA, NSGA-II based	Multi-objectivization with MP and maximum likelihood	Long computational time, No comparison with known datasets
2010, Coelho et al. [9]	Multi-objective artificial immune system	Multi-objectivization with MP and minimization of mean squared error	No comparison with known datasets
2013, Santander and Vega [54; 55; 56]	Multi-objective firefly algorithm	Multi-objective bio inspired algorithm MP and maximum likelihood	Expensive evaluation of secondary objective function
	Multi-objective ant colony	Multi-objective bio inspired algorithm Uses MP and maximum likelihood	Expensive evaluation of secondary objective function
	Multi-objective bat algorithm	Multi-objective bio inspired algorithm Uses MP and maximum likelihood Takes into account information of the Pareto front Publishes pareto front of known datasets	Expensive evaluation of secondary objective function Uses the same objectives and does not explore other possibilities

Table 3.1: Relevant algorithms in the literature of the problem.

4

Multi-objectivization of the MP problem

This chapter describes the methodology followed to multi-objectivize the MP problem. It introduces the proposal, evaluation, and selection of promising new multi-objective formulations for the problem.

4.1 Proposal of supplementary objective functions

The first step for the multi-objectivization of the MP problem is to propose new reformulations of the problem. These reformulations must be applicable to the encoded candidate solutions in the algorithm, and the resulting values must have information to discern good from bad solutions.

In order to multi-objectivize the MP problem two approaches are tested: the first one is the decomposition of the original function, and the second is the proposal of *helper* or *supplementary* objective functions to aid the evolutionary algorithm to navigate trough the search space.

4.1.1 Decomposition of the original function

The decomposition of the original function can be managed by using a series of partial evaluations over a complete phylogenetic tree. The resulting objective functions can be used either to replace the original function, or to add a partial function as helper objective in order to differentiate similar trees. The proposed evaluation functions resulting from breaking down the parsimony score of a tree are the following:

1. Evaluation of the parsimony score of the inner nodes in even levels of the tree, i,e.

$$\mathit{\Phi}_1(T) = \sum_{i=1}^{k} \begin{cases} \phi(w_i), & \text{if } H(w_i) \equiv 0 \pmod 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.1}$$

   where $H(I_i)$ is the depth H of the i-th inner node $w \in I$ of the tree.

2. Evaluation of the parsimony score of only the nodes in odd levels of the tree, i,e.

$$\mathit{\Phi}_2(T) = \sum_{i=1}^{k} \begin{cases} \phi(w_i), & \text{if } H(w_i) \equiv 1 \pmod 2, \\ 0, & \text{otherwise.} \end{cases} \tag{4.2}$$

3. Evaluation of the accumulated parsimony at the internal nodes connected directly at the root of the tree, such as $\mathit{\Phi}_3(T) = F(v_* \to left)$ or $\mathit{\Phi}_4(T) = F(v_* \to right)$, where $v_* \in V$ represents the root node of the tree; $v_* \to left$ and $v_* \to right$ represent the left and right nodes connected to v_*, respectively; and $F(v)$ is calculated as:

$$F(v) = \begin{cases} \phi(p_v) + F(v \to right) + F(v \to left), & \text{if } v \in I \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

   where p_v is the parsimony sequence of node v.

4. Score of the least parsimonious inner node in the tree, i,e.

$$\mathit{\Phi}_5(T) = \arg\max_{w \in I}\{\phi(p_w)\} \tag{4.4}$$

5. Score of the most parsimonious inner node in the tree, i,e.

$$\mathit{\Phi}_6(T) = \arg\min_{w \in I}\{\phi(p_w)\} \tag{4.5}$$

An initial experimentation was conducted in order to test the usefulness of these proposed decompositions, whether as stand-alone multi-objective formulations, or as *helper* objectives derived from a decomposition of the original function. The evaluation was conducted by comparing the values obtained by each of the objective functions over a set of randomly generated topologies in a test instance of 7 taxa.

It was observed that for $\mathit{\Phi}_1(T)$ and $\mathit{\Phi}_2(T)$ the favored tree topologies were those in which most of the leaf nodes of the tree were placed in an even inner level, or odd inner level, respectively, regardless of the overall parsimony score of the tree.

A simmilar behaviour occured with the application of $\mathit{\Phi}_3(T)$ and $\mathit{\Phi}_4(T)$, where tree topologies were deemed better if a leaf node was directly connected at the root of the tree.

As for $\mathit{\Phi}_5(T)$ and $\mathit{\Phi}_6(T)$, the presence of a specific subtree of minimal, or maximal cost in a tree topology, would make it indistinguishable from another containing the same subtree, regardless of their overall parsimony score.

Under the enumerated conditions, these formulations failed to discern between topologies that had the same overall parsimony score, and, in some cases, even failed to discern between topologies with different parsimony scores. Therefore, further experimentation with these proposals was dismissed.

4.1.2 Supplementary objective functions

Taking advantage of the information generated within a phylogenetic tree, in the inferred ancestors, we propose seven helper objectives that are inspired in distance functions used for numeric vectors, two objective functions that apply weighted versions of the parsimony score, and tree functions that use evolution models applied for the evaluation of the likelihood of a tree.

In order to maximize the information used by these supplementary *helper* objectives, we propose an evaluation of the tree topology in which Fitch's algorithm for the propagation of state sets has been stopped after the first step (without selecting final states for the inner nodes).

Each of the proposed objectives is presented in the form $\varphi(p_w)$, which calculates the cost of the parsimony sequence p in an inner node w.

Therefore, the complete cost of a topology evaluated under these objectives is given by

$$\Phi(T) = \sum_{w \in I} \varphi(p_w). \tag{4.6}$$

First we define the helper objectives based on distance functions:

1. D_1. Inspired by the *Jaccard index* of dissimilarity between sample sets (Jaccard, 1901). It is defined as the cardinality of the intersection divided by the cardinality of the union of two sample sets. Taking advantage of the integer values encoded in the sequences of the inner nodes, we consider that every site z of the sequence p_w is a set of bits, and the cardinality $\#(z)$ of that site is denoted by the amount of bits set to 1. The evaluation of this objective function for an inner node w, whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$ is defined as:

$$\varphi(p_w) = \sum_{1}^{k} \frac{\#(x_i \cap y_i)}{\#(x_i \cup y_i)} \tag{4.7}$$

2. D_2. Inspired by the *BrayCurtis dissimilarity* (Bray and Curtis, 1957). It estimates a similitude between geographic sites based on the species of trees found in each site. This dissimilarity index is calculated for two unidimensional arrays a and b as $diss(a,b) = \sum_i \frac{|a_i - b_i|}{|a_i + b_i|}$. For this evaluation we use the integer values encoded in each site of the parsimony sequence p_w of an inner node w, whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, and is evaluated as:

$$\varphi(p_w) = \frac{\sum_1^k |x_i - y_i|}{\sum_1^k |x_i + y_i|} \tag{4.8}$$

It must be noted that even if the values at the denominator of the equation will never reach zero or below values, the absolute value symbols have been mantained for respect to the original formula.

3. D_3. Inspired by the *Canberra distance* (Lance and Williams, 1966), applied to measure distance between points in a vectorial space, the distance between the points a and b in an m dimensional space is measured as $dist(a,b) = \sum_1^m \frac{|a_i - b_i|}{|a_i| + |b_i|}$. Adapted to a parsimony sequence p_w of an inner node w whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, we use the integer values at each site to evaluate the cost φ of this objective as:

$$\varphi(p_w) = \sum_1^k \frac{|x_i - y_i|}{|x_i| + |y_i|} \tag{4.9}$$

In the denominator of the equation, the absolute value symbols have been mantained to respect the original formula.

4. D_4. Inspired by the *Chebyshev distance* (Bienaymé, 1867), in which the distance of two vectors is the greatest difference along any of their dimensions. We apply this metric as the evaluation of the greatest difference along the integer values that encode the sites of two

sequences. For an inner node w whose whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, the evaluation is presented as:

$$\varphi(p_w) = \arg\max_{i \in k}(|x_i - y_i|) \tag{4.10}$$

5. D_5 Inspired by the *Euclidean distance*, in which the distance between two points in the Euclidean space is the length of the straight line between them, calculated in an m dimensional space as $dist(a, b) = \sqrt{(b_1 - a_1)^2 + (b_2 - a_2)^2 + \ldots + (b_m - a_m)^2}$ for the vectors a and b (Deza and Deza, 2009). It is applied for an inner node w, whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, by using the integer values at each site of its parsimony sequence as follows:

$$\varphi(p_w) = \sqrt{\sum_{1}^{k}(x_i - y_i)^2} \tag{4.11}$$

6. D_6. Inspired by the *Manhattan distance* for unidimensional vectors (Krause, 1987), which is the distance of the vertical and horizontal components of two vectors, defined as $dist(a, b) = \sum_i |a_i - b_i|$. Which we apply to the parsimony sequence of an inner node w, whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$ as:

$$\varphi(p_w) = \sum_{1}^{k} |x_i - y_i| \tag{4.12}$$

7. D_7. This distance function attempts to count similarities sim between sequences, for the descendants $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$ of the inner node w, and divides them by the length of the sequence.

$$\varphi(p_w) = \frac{\sum_1^k sim(x_i, y_i)}{k}, \text{ where } sim(x,y) = \begin{cases} 1, \text{ if } x \cap y \neq \emptyset \\ 0, \text{ otherwise} \end{cases} \tag{4.13}$$

The proposed *helper* objectives based on weighted variations of the parsimony score are:

1. **Generalized weighted parsimony** A weighting schema that gives a different value to the transitions (changes from purines to purines and pyrimidines to pyrimidines) and transversions (changes from purine to pyrimidine and vice versa) (Xiong, 2006). Two different weightings have been taken into account, for an inner node w whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$:

   (a) D_8. Uses the weighting schema proposed by Swofford and Olsen (1990), that is evaluated as:

   $$\varphi(p_w) = \sum_1^k c_i, \text{ where } c_i =, \begin{cases} 1, \text{ if } Transition \\ 5, \text{ if } Transversion \\ 0, \text{ otherwise} \end{cases} \tag{4.14}$$

   (b) D_9. Uses the weighting schema proposed by Brown et al. (2006), that is evaluated as:

   $$\varphi(p_w) = \sum_1^k c_i, \text{ where } c_i =, \begin{cases} 1, \text{ if } Transition \\ 2, \text{ if } Transversion \\ 0, \text{ otherwise} \end{cases} \tag{4.15}$$

The last supplementary objective functions, that take into account evolutionary models from the literature to evaluate an individual, use generalizations that enable the evaluation of individual nodes of the tree. The first one is a generalized model of the Jukes-Cantor model, and the remaining two are generalizations of the Kimura two parameter model (K2P) as presented in two evolutionary analysis software tools: TreeCon (Van de Peer and De Wachter, 1994) and MEGA (Tamura et al., 2013).

1. D_{10}. Jukes-Cantor model (Jukes and Cantor, 1969), assumes all substitutions are independent and equally subject to change. For an inner node w whose descendants are represented by the sequences $S_t = \{x_1, x_2, \ldots, x_k\}$ and $S_u = \{y_1, y_2, \ldots, y_k\}$, it is calculated as:

$$\varphi(p_w) = -\frac{3}{4}\ln(1 - \frac{3}{4}f_{tu}) \tag{4.16}$$

   where f_{tu} is the estimated evolutionary distance between species t and u.

2. D_{11}. Kimura two parameter model (K2P) (Kimura, 1980) as generalized in TreeCon (Van de Peer and De Wachter, 1994).

$$\varphi(p_w) = -\frac{1}{2}\ln[(1 - 2R - Q) * \sqrt{1 - 2Q}], \tag{4.17}$$

   where R is the proportion of transitions and Q the proportion of transversions in the sequences S_t and S_u that represent w's descendants.

3. D_{12}. K2P as generalized in MEGA (Tamura et al., 2013).

$$\varphi(p_w) = -\frac{1}{2}\ln(1 - 2R - Q) - \frac{1}{4}\ln(1 - 2Q) \tag{4.18}$$

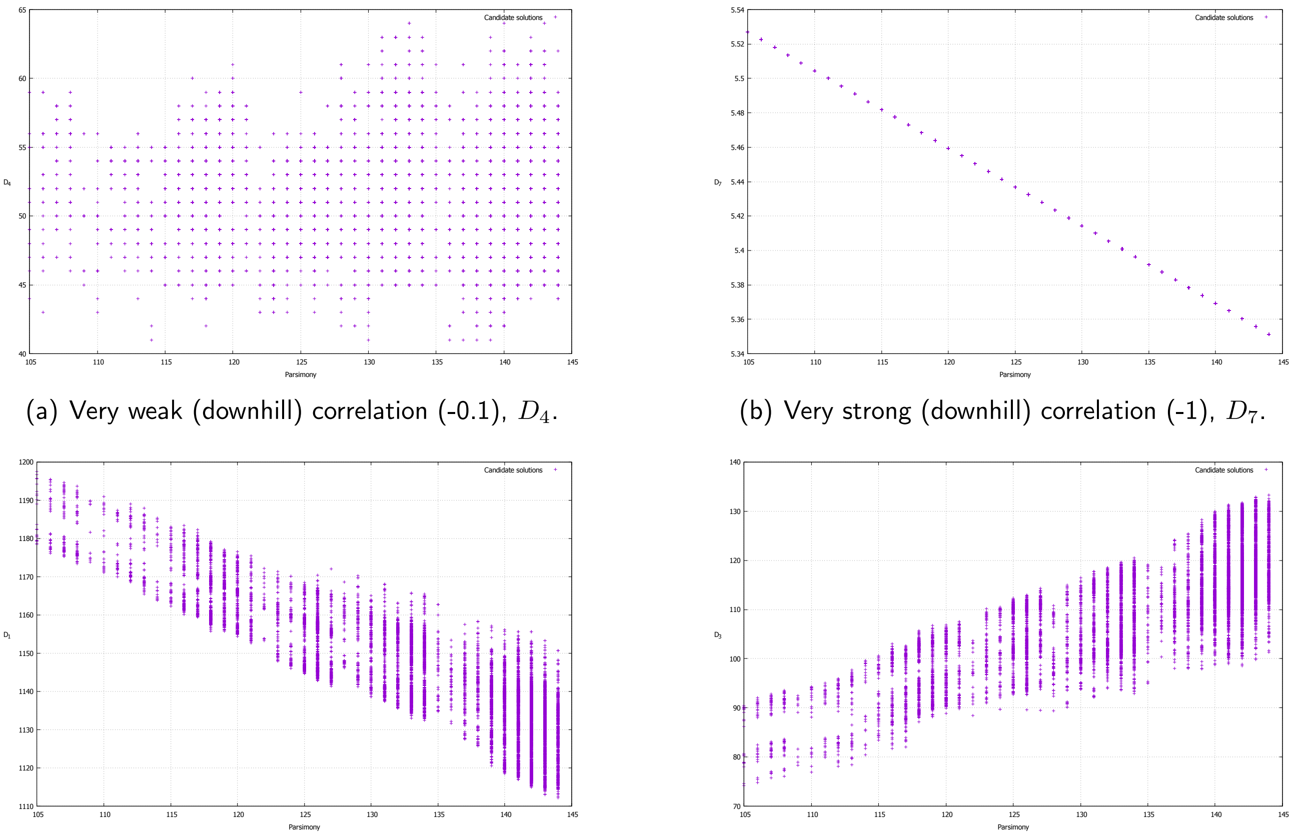
   where R is the proportion of transitions and Q the proportion of transversions in the descendant sequences of w.

4.2 Evaluation and selection

To evaluate the usefulness of the proposed functions as supplementary objectives in a multi-objective formulation, their correlation against the parsimony score was measured. According to Evans (1996) the strength of a correlation can be described as: very weak ($0.00 \rightarrow 0.19$), weak ($0.20 \rightarrow 0.39$), moderate ($0.40 \rightarrow 0.59$), strong ($0.60 \rightarrow 0.79$), and very strong ($0.80 \rightarrow 1.00$). A very strong correlation to the parsimony score of a tree would not be useful for differentiating trees of the same quality, conversely, an objective function with very weak correlation would fail to introduce a trade-off between objectives for a solution in the Pareto front obtained by an algorithm. An ideal function would have a moderate to strong correlation, whether uphill (positive) or downhill (negative), in order to guide the search towards better solutions and being able to differentiate solutions with similar evaluations of the first objective function, these characteristics can be observed in Figure 4.1. To correctly evaluate the correlation between the proposed objective functions and the parsimony score of a candidate solution, an exhaustive evaluation of the search space was conducted for 30 instances of 7 taxa (10395 trees each). These instances were generated using aligned sequences from larger datasets found in the literature:

- 3 instances generated from *drosophyl2*, with 17 taxa of size $k = 222$.
- 5 instances generated from *RDPII_218*, with 218 taxa of size $k = 4182$.
- 7 instances generated from *rbcL_55*, with 55 taxa of size $k = 1314$.
- 7 instances generated from *ZILLA_500* with 500 taxa of size $k = 759$.
- 8 instances generated from *mtDNA_186*, with 186 taxa of size $k = 16608$.

In order to evaluate the entire search space, all possible rooted tree topologies for 7 taxa were generated in Newick format using the phylogenetic analysis tool PAUP* (Swofford, 2001). These



(a) Very weak (downhill) correlation (-0.1), D_4.

(b) Very strong (downhill) correlation (-1), D_7.

(c) Strong (downhill) correlation (-.7), D_1 index.

(d) Strong (uphill) correlation (+.7), D_3.

Figure 4.1: Graphs showing the correlation between two objectives. Each mark represents a solution in the search space of a synthetic test instance evaluated with Parsimony as first objective(x axis) and a secondary objective (y axis).

trees were reconstructed using the information of the generated instances and evaluated with each proposed objective function.

For every evaluated instance an individual analysis of the correlation between objective functions was conducted. The resulting data allows us to graphically depict the interaction between evaluations.

Figure 4.2 shows a square matrix with the data obtained from the evaluations of every proposed *helper* objective over the entire search-space of instance $drosophyl_3$. The main diagonal displays the name of the objective function alongside a histogram showing the distribution of costs among the evaluated search-space. The lower triangular shows scatter plots of the interaction between every pair of evaluated objective functions. The upper triangular shows the correlation between every pair of objective functions. The number ranges, shown at the start and end of every row and column, display the range of the values obtained for every objective function in the entire search-space. Similar graphs for all the test instances are available in Appendix A.

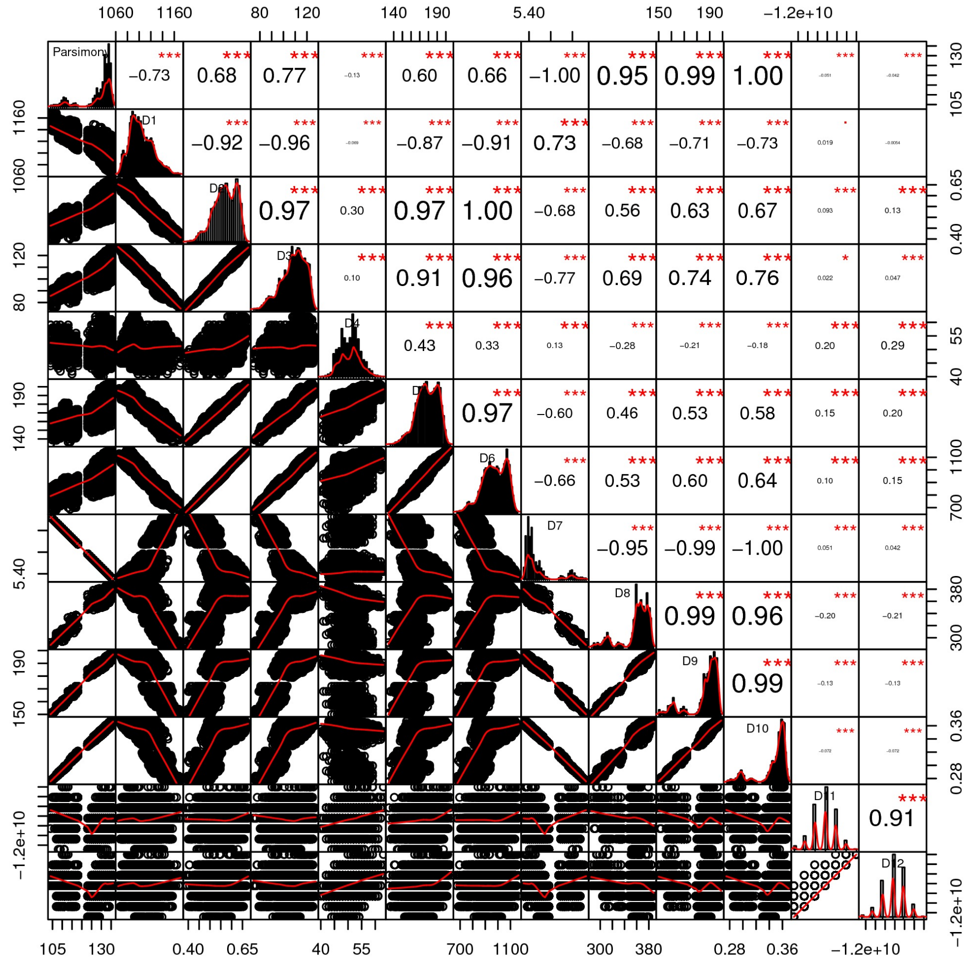


Figure 4.2: Example of the plot obtained from evaluating instance $drosophyl_3$. The lower triangle of the matrix shows scatter plots with the obtained sores from the evaluation, the upper triangle shows the correlation between two functions, and the diagonal shows the name and the histogram of the obtained values for every function.

Avg. COR	ϕ	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}	D_{12}
ϕ	1												
D_1	**-0.6396**	1											
D_2	**0.5859**	-0.8740	1										
D_3	**0.6041**	-0.8259	0.8596	1									
D_4	0.0303	-0.0985	0.2483	0.2378	1								
D_5	0.4643	-0.7039	0.9337	0.7465	0.3605	1							
D_6	**0.5801**	-0.8783	0.9939	0.8524	0.2462	0.9292	1						
D_7	-0.9999	0.6396	-0.5859	-0.6041	-0.0304	-0.4643	-0.5801	1					
D_8	0.8959	-0.5489	0.4944	0.5145	0.0171	0.3984	0.4889	-0.8959	1				
D_9	0.9761	-0.6079	0.5501	0.5715	0.0198	0.4363	0.5441	-0.9761	0.9682	1			
D_{10}	0.9936	-0.6387	0.5729	0.5946	0.0132	0.4481	0.5681	-0.9936	0.9009	0.9756	1		
D_{11}	-0.0316	0.0100	-0.0050	-0.0177	0.0018	0.0065	-0.0041	0.0316	-0.0353	-0.0385	-0.0351	1	
D_{12}	-0.0284	-0.0114	0.0297	0.0067	0.0150	0.0393	0.0289	0.0284	-0.0445	-0.0407	-0.0344	0.7232	1

Table 4.1: Average correlation between objective functions over 30 test instances of 7 taxa.

As shown in Table 4.1, on average there are four objective functions that fulfill the previously stated requirement of having a strong correlation: D_1, D_2, D_3, and D_6. However, D_2 and D_6 have a really strong correlation between them (0.9939), meaning that the results provided by them might be too similar during experimentation.

Therefore, according to the average results and the behavior of the proposed objective functions during testing, the selection of promising supplementary objectives can be narrowed to three: D_1, D_2, and D_3.

4.3 Effect of the selected multi-objective reformulations in the fitness landscape of the problem

According to Verel et al. (2006), the geometry of a neutral fitness landscape derives from the neutral neighborhoods.

For a solution $T \in \mathscr{T}$, the neutral neighborhood of T is the set of solutions obtained by the neighborhood function $B(T)$ such that $neut(T) = \{T' \in B(T) | cost(T) = cost(T')\}$, and the neutral *degree* of T is the cardinality of its neutral neighborhood.

In order to assess the reduction of neutrality in the fitness landscape of the problem, through the selected multi-objective reformulations, we analyzed different neighborhoods generated from random initial solutions using SPR as neighborhood function. The analysis was conducted using the instance $mtDNA_2$, with seven taxa and sequences of size $k = 16608$.

Figure 4.3 shows a histogram that represents the distribution of the different parsimony scores present in the 56 neighbors of a random solution T. For the 57 different trees in the neighborhood (including T, with a normalized parsimony score of $norm(\Phi(T)) = 0.8$), there are only eight different parsimony scores. For this particular case, the neutral degree of the initial solution T is $neut_{pars}(T) = 21$, which represents a plateau that includes 37% of the neighborhood. Furthermore, a second plateau can be observed, with 16 tree topologies that share the normalized parsimony cost of $norm(\Phi(T)) = 0.9$.

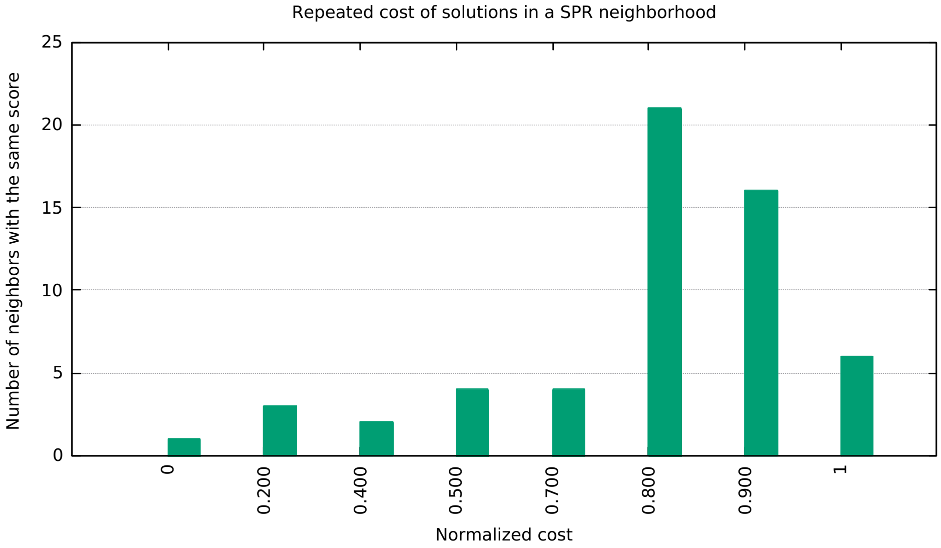


Figure 4.3: Distribution of the parsimony scores found in the SPR neighborhood of a randomly generated topology for instance $mtDNA_2$.

Figure 4.4 represents the distribution of different solutions for the linear combination of the parsimony score and D_1, we can observe that the number of possible solutions increases to 34, and the most repeated one only includes 5 solutions in the entire neighborhood. The normalized cost for the initial solution T is $norm(\mathit{\Phi}(T)) = 1.520$, giving it a neutral degree of $neut_{D_1}(T) = 3$.

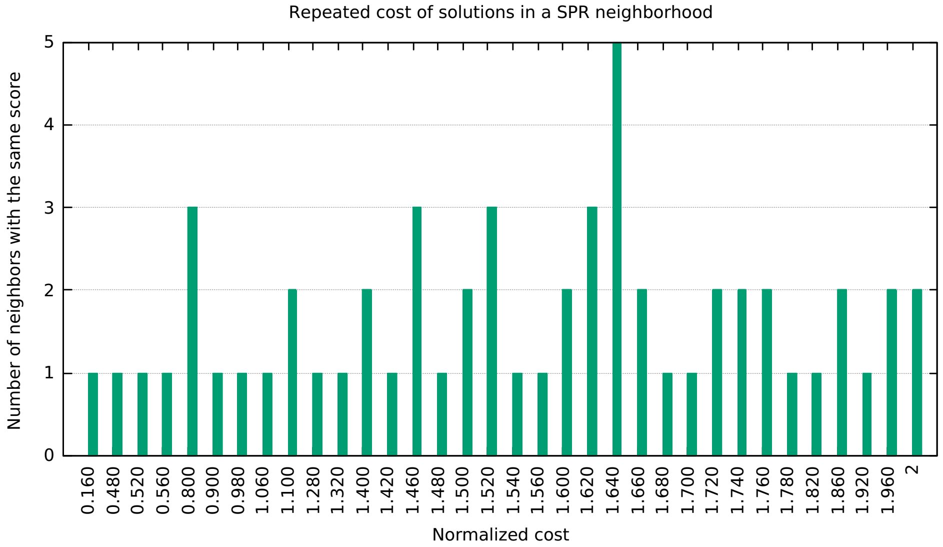


Figure 4.4: Distribution of the parsimony scores found by the linear combination of parsimony and D_1 in the SPR neighborhood of a randomly generated topology for instance $mtDNA_2$.

The application of the linear combination between the parsimony score and D_2 is shown in Figure 4.5. It depicts the existence of 56 possible evaluations for the 57 solutions in the neighborhood of T. The neutral degree of T is $neut_{D_2}(T) = 1$, being the only solution in the neighborhood with a normalized cost of $norm(\Phi(T)) = 1.542$.

The use of D_3 combined with the parsimony score is shown in Figure 4.6. It shows that the biggest plateaus in the neighborhood are composed of two solutions. Since the normalized cost of T is $norm(\Phi(T)) = 1.518$, the neutrality degree of this neighborhood is also $neut_{D_3}(T) = 1$.

The behavior reported by this experiment under the SPR neighborhood is also observed under the

Table 4.2 shows a general view of the neutrality of the fitness landscape over this particular instance. It shows the number of different evaluations that exist for the 10395 different tree topologies in the search space, the maximum (Max) and minimum (Min) number of topologies that share the same score, and the average number of solutions under each different evaluation (Avg). It can be observed that the neutrality of the search space is greatly reduced with the application of

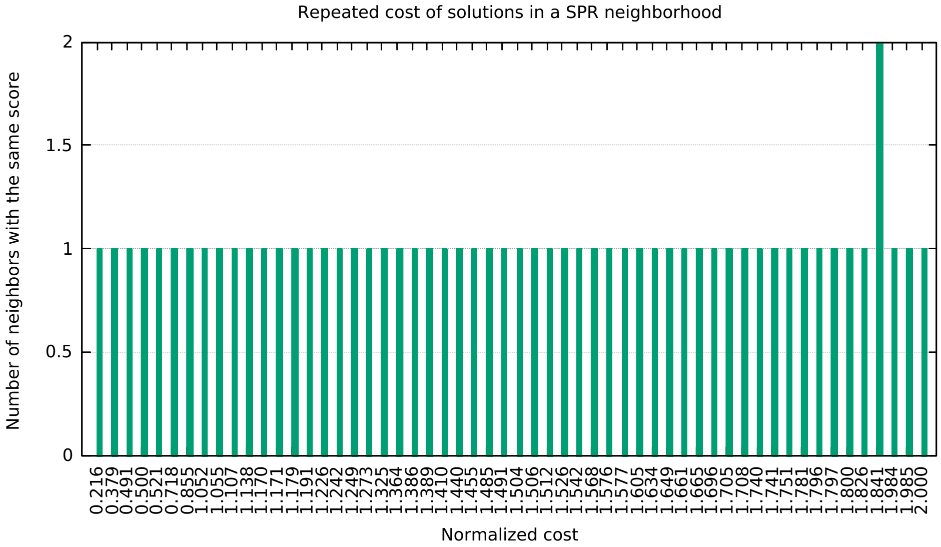


Figure 4.5: Distribution of the parsimony scores found by the linear combination of parsimony and D_2 in the SPR neighborhood of a randomly generated topology for instance $mtDNA_2$.

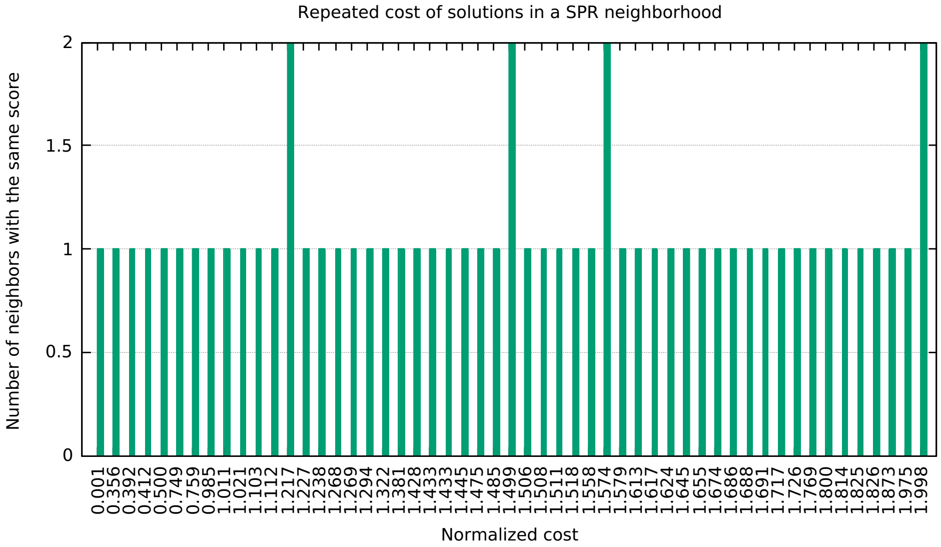


Figure 4.6: Distribution of the parsimony scores found by the linear combination of parsimony and D_3 in the SPR neighborhood of a randomly generated topology for instance $mtDNA_2$.

the *helper* objectives selected. The increase of different evaluation scores obtained with the multi-objectivization helps discern similar solution and break the plateaus that exist in the fitness landscape of the problem.

Evaluation	$Different$	Max	Min	Avg
Parsimony Score	12	2156	11	866.25
D_1	345	171	1	30.13
D_2	10282	3	1	1.01
D_3	10136	4	1	1.02

Table 4.2: Cost of solutions over the search space of instance $mtDNA_2$.

4.4 Chapter summary

In this chapter, we proposed six functions based on the decomposition of the parsimony score, and a set of 12 additional *helper* objective functions. Initial experimentation allowed us to discover that the proposed decompositions were unfit for a multi-objective formulation, and an extensive exploration of the search space for instances of seven taxa narrowed the proposed *helper* objectives to only tree, that will be tested within a MOEA implementation.

The effect of the additional *helper* objective functions over the fitness landscape of the problem was tested by using the evalution of tree topologies belonging to the same SPR neighborhood, assessing the behavior predicted by the correlation of the functions with the parsimony score.

Next chapter contains the description of the algorithms used to assess the performance of the proposed multi-objective reformulations, including the selected MOEA, and the genetic operators implemented.

5

Design and implementation of algorithms

This chapter describes the implementation of the selected MOEA: the Nondominated Sorting Genetic Algorithm II (NSGA-II), and the selected genetic operators (crossover and mutation). It also describes a parallelization approach used in order to speed up the evaluation of tree topologies.

5.1 Non-dominated sorting genetic algorithm - II

In order to test the selected *helper* objectives, the chosen MOEA algorithm is the NSGA-II, proposed by Deb et al. (2002). This algorithm was selected for its availability, its adaptability, and its excelent performance over an extensive range of problems in the multi-objectivization literature.

NSGA-II is a fast, elitist algorithm that has been found to be effective in nummerous applications. This algorithm can be easily adapted to different problems by modifying the genetic operators applied to the problems at hand. This algorithm was implemented and modified to work with a binary tree representation of the solutions. The complexity of the algorithm is governed by the non-dominated sorting that has a complexity of $O(\varrho(2M)^2)$, where ϱ is the number of objective functions and M refers to the size of the population.

The main loop of the NSGA-II algorithm involves the creation a new generation of solutions and the selection among these and the existing solutions (elitism). This approach prefers those solutions that are not dominated and those that could add diversity to the population (crowding distance) in case the main Pareto front is not enough to fill the next population. Algorithm 1 shows the pseudocode of the NSGA-II algorithm.

Algorithm 1 NSGA-II algorithm.

```
Require: Set of aligned sequences.
Ensure: Pareto front of non-dominated phylogenetic trees.
  Initialize ParentPop with $M$ random solutions
  Evaluate(ParentPop)
  FastNonDominatedSort(ParentPop)
  AssignCrowdingDistance(ParentPop)
  for $i \leftarrow 2$ to $maxIter$ do
    ChildPop←getNewPopulationFrom(ParentPop)
    Mutate(ChildPop)
    Evaluate(ChildPop)
    MixedPop←Merge(ParentPop,ChildPop)
    FastNonDominatedSort(MixedPop)
    AssignCrowdingDistance(MixedPop)
    ParentPop←FillWithNonDominatedFronts(MixedPop)
  end for
  Report pareto front in ParentPop
```

5.1.1 Fast non-dominated sorting

The non-dominated sorting algorithm is separates a population into layers. Every layer includes only solutions that are dominated by the previous layer but no other solution. This is useful for selecting the individuals that will remain to the next generation. Once the mixed population has been sorted, the parent population is filled by adding up layers of solutions until it is complete, the complexity of the non-dominated sorting is $O(\varrho M^2)$ where ϱ is the number of objectives and M the size of the population.

Algorithm 2 FastNonDominatedSort algorithm.

```
Require: Population.
Ensure: Sorted population according to rank.
  for all $\mathbf{T} \in Population$ do
    $D_{\mathbf{T}} \leftarrow \emptyset$ {Solutions dominated by $\mathbf{T}$.}
    $N_{\mathbf{T}} \leftarrow 0$ {Number if solutions that dominate $\mathbf{T}$.}
    for all $T \in Population$ do
      if $\mathbf{T} \prec T$ then {if $\mathbf{T}$ dominates $T$}
        $D_{\mathbf{T}} = D_{\mathbf{T}} \cup \{T\}$
      else if $T \prec \mathbf{T}$ then {if $T$ dominates $\mathbf{T}$}
        $N_{\mathbf{T}} = n_{\mathbf{T}} + 1$
      end if
    end for
    if $N_{\mathbf{T}} = 0$ then
      $\mathbf{T}_{rank} \leftarrow 1$
      $\mathcal{F}_1 \leftarrow \mathcal{F}_1 \cup \{\mathbf{T}\}$ {Front of rank 1}
    end if
    $i \leftarrow 1$
  while $\mathcal{F}_i \neq \emptyset$ do
    $\mathcal{Q} \leftarrow \emptyset$ {Members of next front}
    for all T $\in \mathcal{F}_i$ do
      for all $T \in D_{\mathsf{T}}$ do
        $N_solq \leftarrow N_solq - 1$
        if $N_T = 0$ then {$T$ belongs to the next front}
          $T_{rank} \leftarrow i + 1$
          $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{T\}$
        end if
      end for
    end for|
    $i \leftarrow i + 1$
    $\mathcal{F}_i \leftarrow \mathcal{Q}$
  end while
end for
```

5.1.2 Crowding distance

The crowding distance of a solution is an estimate of the distance between itself and the rest of the solutions in the population. It is assigned to each element in every layer from the non-dominated sorting, and is calculated from the distances between the ϱ objective functions. The complexity of the crowding distance calculation is $O(M(2\varrho)log(2\varrho))$.

Algorithm 3 AssignCrowdingDistance algorithm

```
Require: $\mathcal{F}$ a non-dominated set of solutions
Ensure: Crowding distance of every individual in the set
  $N \leftarrow |\mathcal{F}|$
```

```
for $i \leftarrow 1$ to $N$ do
    $\mathcal{F}[i]_{dist} \leftarrow 0$
end for
for $m \leftarrow 1$ to $\varrho$ do
    $SORT(\mathcal{F}, m)$ {sort solutions according to the $m$-th evaluation function}
    $\mathcal{F}[i]_{dist} \leftarrow \mathcal{F}[N]_{dist} \leftarrow \infty$
    for $i \leftarrow 2$ to N do
        $\mathcal{F}[i]_{dist} = \mathcal{F}[i]_{dist} + \frac{(\mathcal{F}[i+1] \cdot m - \mathcal{F}[i-1] \cdot m)}{f_m^{max} - f_m^{min}}$
    end for
end for
```

5.2 Initial Population

The individuals in the initial population are created randomly from the analyzed taxa. The initialization employs a randomized variation of the UPGMA algorithm. Each individual starts with a set of n nodes that include the sequences of the studied taxa. Then, at every iteration, a pair of nodes is randomly selected and grouped into new node that then replaces them in the set. The selection and grouping of pairs is repeated until only one node remains. The last node is selected as root and a fully formed tree is obtained. This process can be observed in Figure 5.1.

5.3 Genetic operators

The genetic operators in evolutionary algorithms are meant to guide the search toward promising sections of the search space. There are three kind of operators: selection, crossover, and mutation. The adopted selection operator is the one already defined as part of the NSGA-II algorithm. The crossover and mutation operators depend on the adopted representation, their design is discussed below.

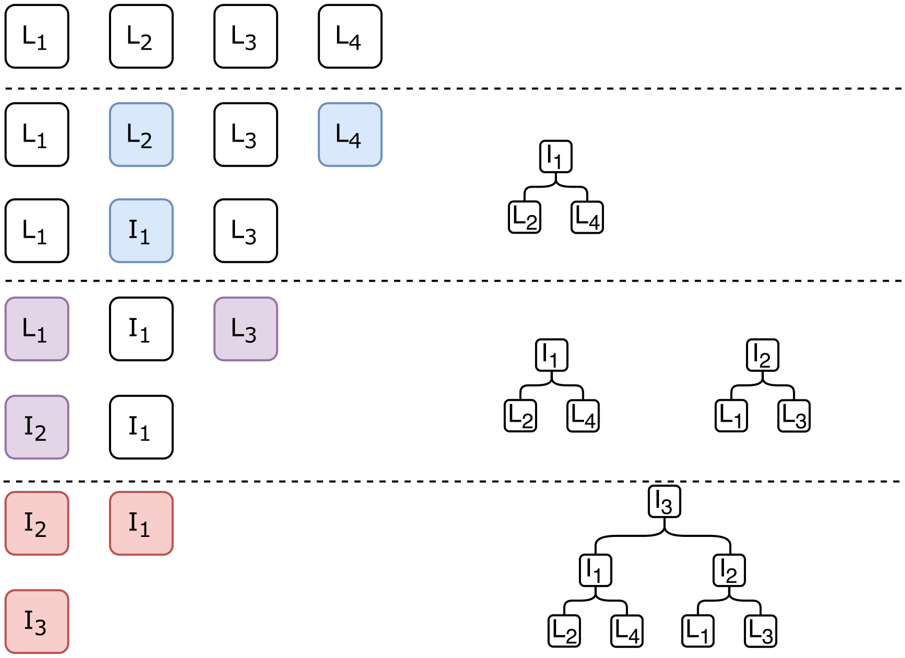


Figure 5.1: Process of creation of a new random tree starting with a set of $n = 4$ nodes that represent the studied taxa.

5.3.1 Crossover operator

The crossover operator is designed to combine the information of two solutions in the population (parents) to generate new solutions (children) by mixing their information.

The selected crossover operator is based on the TBR neighborhood. It has been used in (Congdon, 2001; Lewis, 1998; Matsuda, 1995), and has a relatively low computational cost. This operator copies the information of the first parent into a new child. Then, it selects a random sub-tree from the second parent as a crossover point. All the leaves present in the selected sub-tree are removed from the new child. After the elimination, any inner node with only one child node is replaced by their direct descendant. Finally, a copy of the selected subtree is inserted in a random node of the child. The complexity of this operator is $O(nk)$ where n is the number of taxa and k the size of the parsimony sequences to be copied to the new individual. A graphical example of the crossover operator can be seen in Figures 5.2, 5.3, and 5.4.

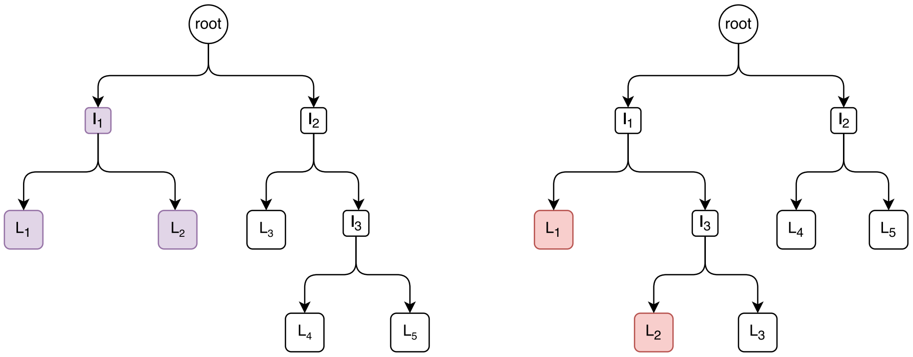


Figure 5.2: A parent solution (left) and a child with a copy of the other parent (right). A subtree is selected from the parent (I_3) and the leaves in said subtree are located in the child solution.

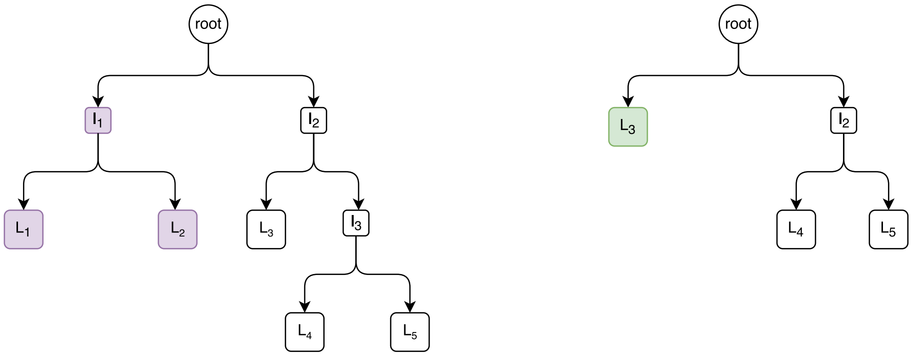


Figure 5.3: Leaves from the selected subtree are pruned from the child solution, any inner node that would have only one child is removed.

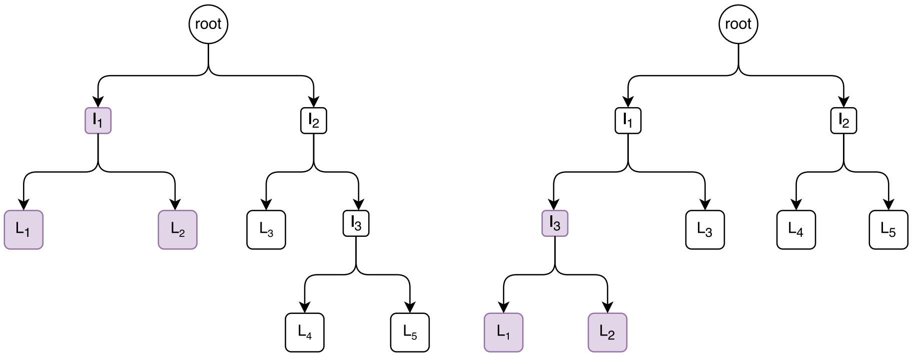


Figure 5.4: A copy of the sub-tree is inserted in the child in order to form a valid tree.

5.3.2 Mutation operators

In order to mutate the solutions in the children population five neighborhood functions have been implemented as mutation operator. The high number of mutation operators helps us to prevent getting trapped in the local optima that are inherent to each neighborhood.

- *Nearest Neighbor Interchange* (NNI), used by Andreatta and Ribeiro (2002); Moore et al. (1973); Vazquez Ortiz and Rodriguez Tello (2011); Waterman and Smith (1978). NNI switches any node from the topology (except the root node) with one of its closest neighbors (except sibling nodes). It is exemplified in Figure 5.5.

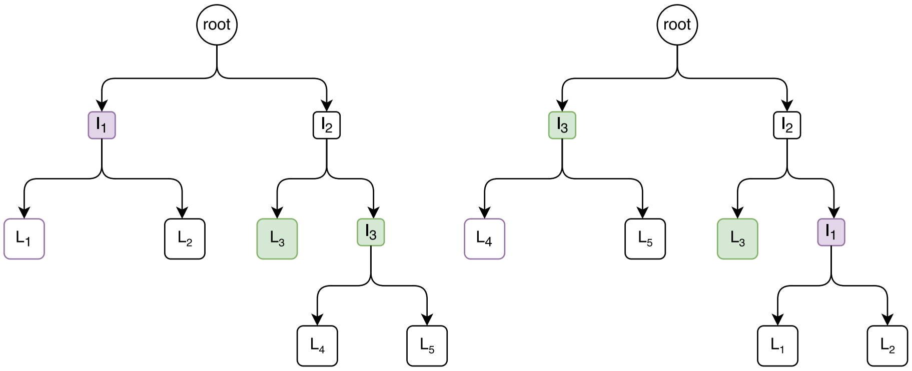


Figure 5.5: NNI operation. Selection of the inner node I_1 and swap with one of its nearest neighbors (I_2 and I_3).

- *Subtree Pruning and Regraft* (SPR), used in Andreatta and Ribeiro (2002); Vazquez Ortiz and Rodriguez Tello (2011). It prunes an inner node of the tree, and removes any node that would end up with only one child. Then it inserts the pruned node in a random position of the tree. A graphical description of this operation can be seen in Figure 5.6.

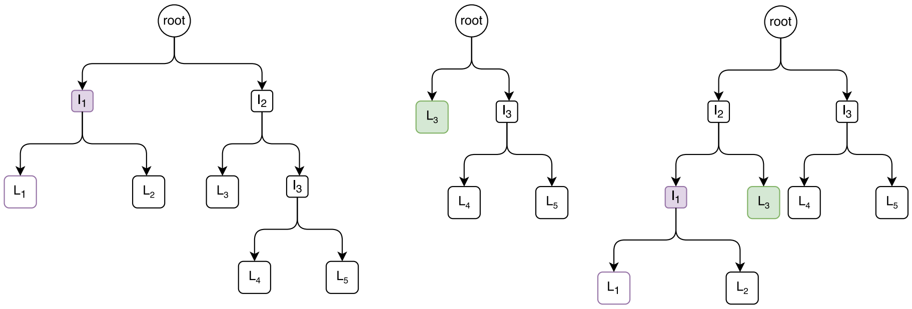


Figure 5.6: SPR operation. Pruning of inner node I_1, and insertion in L_3.

- *Tree Bisection and Reconnection* (TBR), used in Swofford et al. (1996); Vazquez Ortiz and Rodriguez Tello (2011). It divides the tree into two subtrees and reconnects them in a randomly selected node. Figure 5.7 shows how TBR works.

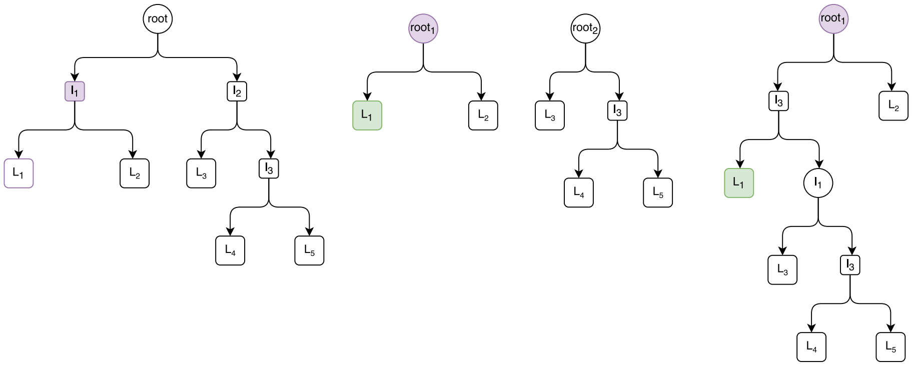


Figure 5.7: TBR operation. Division of the tree in I_1 and insertion of $root_2$ in L_1.

- *Leaf Swap* (LSwap), used in (Cotta and Moscato, 2002; Sonco Alvarez and Ayala Rincon, 2017). This operator selects a random leaf and swaps it with another. This approach is shown in Figure 5.8.

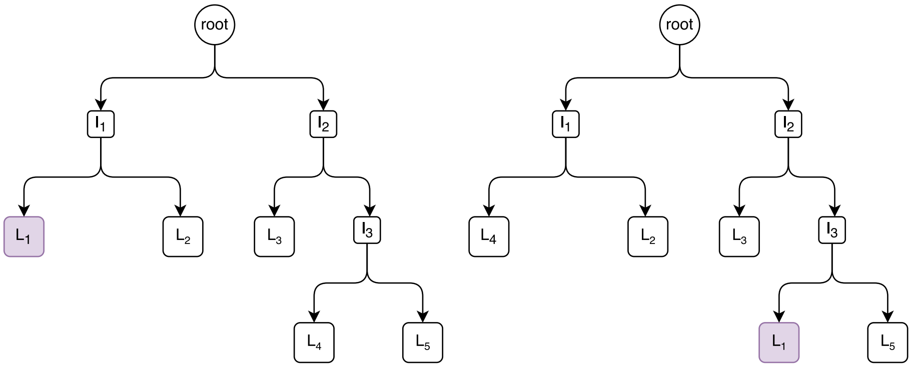


Figure 5.8: LSwap operation. Selection of L_1 and swap with L_4.

- *Single Step* (STEP), used in (Andreatta and Ribeiro, 2002; Sonco Alvarez and Ayala Rincon, 2017; Waterman and Smith, 1978): A leaf is pruned from the tree and inserted in any other edge. This operator is depicted in Figure 5.9.

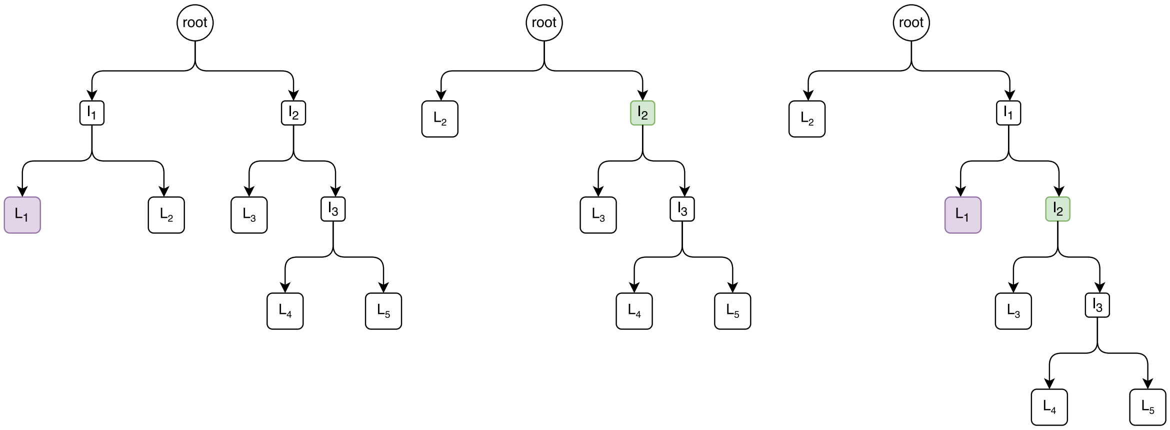


Figure 5.9: STEP operation. Pruning of L_1 and insertion in I_2.

5.3.3 Local search

Local search within the NSGA-II algorithm can be applied in order to enhance the solutions in the population. This procedure is applied with a low probability during the mutation process of the individual solutions to prevent a premature convergence of the algorithm.

Local search can be achieved by removing the cost of a pruned or moved node and evaluating the added cost of inserting it in each available position of the tree topology, selecting the best possible scenario.

In order to evaluate this added cost, an auxiliary parsimony sequence is used to propagate the values as they escalate in the topology of the tree and adding the generated cost of using that sequence in the tree. In the worst case scenario, the complexity of an iteration of the local search algorithm is $O(nk)$, the overall complexity depends on the size of the used neighborhood function and the imposed bound on the number of iterations, we apply a limit of $10,000$ evaluated topologies per search to mantain a low computational cost.

5.3.4 Parallelization of the algorithm

In order to speed up the execution time of the algorithm, we apply parallelization using OpenMP (Dagum and Menon, 1998).

Parallelization can be achieved by distributing the overall length of the parsimony sequence at nodes between different cores and executing a reduction to obtain its total parsimony cost. This parallelization approach can be applied to any of the implemented objective functions, and can be applied in two main phases of the algorithm: the evaluation of individuals, andt he propagation of the auxiliary values during a local search procedure.

5.4 Chapter summary

This chapter introduced the implemented MOEA and the genetic operators used to guide the search process. The proposed local search, and paralellization of the evaluation are described. The next chapter describes the experimentation conducted to assess the performance of the proposed multi-objective reformulations of the MP problem, including the description of the instances used, the experimental conditions, and the obtained results.

6

Experimentation and results

This chapter presents the experimental phase of this thesis work, including the instance sets, the parameters of the algorithms, and the experimental conditions during their execution.

This chapter also includes the comparison of the obtained results. First the results obtained are compared with respect to solutions produced by a mono-objective version of the implemented algorithm. Then they are compared against solutions obtained by methods in the state of the art of the problem.

6.1 Comparison criteria

In order to assess the performance of the multi-objective formulations for the MP problem proposed during this thesis work, we compared the obtained results to those obtained by a mono-objective algorithm under the same circumstances. The standard metrics used in these comparisons are:

- *Best:* Quality of the best solution found by each formulation, using parsimony score as comparison criteria.
- *Mean:* Average value of the quality of the solutions found by each formulation.
- *σ:* Standard deviation of the quality of the solutions found.
- *τ:* Average time (in seconds) of the execution of the algorithm.

6.2 Test instances

Two major groups of instances were used: instances that are encoded as binary characters, and instances that are encoded as multi-state characters. The characteristics of every test instance are described below.

6.2.1 Instances encoded with binary characters

For these instances the binary digits represent the presence or absence of a morphological trait for each known specie, the special symbol "**?**" is used for uncertainty. An example of a sequence encoded with binary characters extracted from the instance *ANGI* is as follows:

```
austroba    0000?10010010000000000000001000000000000000000?0000????????00
```

This group of testing instances is composed by eight real-life instances and 14 synthetic instances. The real instances have been previously used in (Andreatta and Ribeiro, 2002; Goëffon et al., 2006; Ribeiro and Vianna, 2005; Vazquez Ortiz and Rodriguez Tello, 2011), and are shown in Table 6.1. The ten synthetic instances, created by Ribeiro and Vianna (2003), have been used in (Goëffon et al., 2006; Ribeiro and Vianna, 2005; Vazquez Ortiz and Rodriguez Tello, 2011), and are shown in Table 6.2. These tables display the name of the instances, their number of taxa n, the size of their sequences k, the quality of the most parsimonious tree reported for each instance, and the algorithm that found that tree. For these instances, the best results have been reported by the Hybrid Distance Recombination Algorithm (HYDRA) (Goëffon et al., 2006), Simulated Annealing for Maximum Parsimony (SAMPARS) (Richer et al., 2013), and Simulated Annealing-Maximum Parsimony (SA-MP) (Vazquez Ortiz and Rodriguez Tello, 2011).

Instances	Taxa (n)	Size (k)	*Best*	Algorithm
ANGI	49	59	216	HYDRA
CARP	117	110	548	HYDRA
ETHE	58	86	372	HYDRA
GOLO	77	97	496	HYDRA
GRIS	47	93	172	HYDRA
ROPA	75	82	325	HYDRA
SCHU	113	146	759	HYDRA
TENU	56	179	682	HYDRA

Table 6.1: Real-life instances with binary characters, with best parsimony scores found by algorithms in the state of the art.

Instances	Taxa (n)	Size (k)	*Best*	Algorithm
tst01	45	61	545	HYDRA
tst02	47	151	1354	SA-MP
tst03	49	111	833	HYDRA
tst04	50	97	587	SAMPARS
tst05	52	75	789	HYDRA
tst06	54	65	596	HYDRA
tst07	56	143	1269	SA-MP
tst08	57	119	852	HYDRA
tst09	59	93	1141	SAMPARS
tst10	60	71	720	SA-MP
tst17	71	159	2450	SAMPARS
tst18	73	117	1521	SAMPARS
tst19	74	95	1012	SAMPARS
tst20	75	79	659	SAMPARS

Table 6.2: Synthetic instances with binary encoding, with best parsimony scores found by algorithms in the state of the art.

6.2.2 Instances encoded with multi-state characters

This group of testing instances is composed by 16 real-life instances encoded with multi-state character data, as shown in Table 2.1.

Table 6.3 shows the first four instances, these have been used by Santander Jiménez (2016); Santander Jiménez and Vega Rodríguez (2013a,b) to evaluate multi-objective approaches to the MP problem. This table shows the name of the instances, their number of taxa n, the size of their sequences k, the quality of the most parsimonious tree reported for each of them, and the algorithm that found that tree. For these instances, the best results have been found using the Multi-objective Artificial Bee Colony algorithm (MOABC) (Santander Jiménez and Vega Rodríguez, 2013b).

Instances	Taxa (n)	Size (k)	*Best*	Algorithm
mtDNA_186	186	16608	2431	MOABC
rbcL_55	55	1314	4874	MOABC
RDPII_218	218	4182	41488	MOABC
ZILLA_500	500	759	16218	MOABC

Table 6.3: Real-life instances with multi-state character encoding, with best parsimony scores found by multi-objective algorithms in the state of the art.

The remaining 12 instances are shown in Table 6.4. These instances were used by Strobl and Barker (2016) in a Simulated Annealing study over phylogeny reconstruction. The best obtained results for these instances are not reported. For these instances a *Tag* name has been added, this name will be used to reference these instances from now on, in order to maintain order in the comparison tables.

Instance	Taxa (n)	Size (k)	Tag
1_1399893393_Molecular_noct.phy	85	9584	Molecular
Adams_etal_Syst_Biol_unique_cytb_haplotypes.phy	277	605	Adams
Alignment_4genes.phy	52	2364	Alignment4
alldata_noout.phy	232	4703	Alldata
Alstrom5.smh.phy	41	3426	Alstrom5
angiosperm-rps11.phy	5	402	Angiosperm
Bahl.phy	525	987	Bahl
COI_CAD_SystBiol.phy	435	1434	COI_CAD
Pasach_run1.phy	27	3062	Pasach
rabosky_6genes_concatenated.phy	238	5373	Rabosky
S4.phy	31	31674	S4
sphaero-4gene.phy	28	5489	Sphaero
THOMOMYS1.phy	26	4385	Thomomys
VATI_6genes_concatenated.phy	36	3039	Vati6
VATI_ND2_Align_Final.phy	85	1041	VatiN

Table 6.4: Real-life instances with multi-state encoding with no parsimony score known.

6.3 Experimental conditions

All the necessary algorithms were coded in C language and compiled with gcc.

The experimentation was conducted in CINVESTAV-Tamaulipas's computer cluster *Minerva*, using the processing nodes *Medusa1-2* and *Medusa1-3*, both of them with 12 Intel Xeon(R) X5660 2.80GHz with 12 GB of RAM; and *Neptuno*, with ten processing nodes (*Hydra1-1*, *Hydra1-2*, $\ldots$, *Hydra1-10*), each of them with two Intel Xeon(R) X5550 2.67GHz with four cores and 16 GB of RAM.

Due to the stochastic nature of the genetic algorithm 31 executions of the algorithm were run for each instance and each supplementary objective. In order to exploit the search using every supplementary objective function, the stop criteria is set as the number of consecutive generations without any improvement, for the experimentation process we set this limit as 50 generations.

For this experimentation, 10% of the individuals of the population of each algorithm are initialized using a set tree topologies obtained by a greedy approach contained in the *biosbl* package by Jean-Michel Richer[1], and 90% of the individuals is initialized with randomized trees to mantain diversity in the population.

6.4 Fine-Tuning of the algorithm's parameters

The tuning process for our implementation was executed using an iterated racing procedure through the *irace* package (López Ibañez et al., 2016). This package samples configurations and applies them to an execution of the algorithm. It iterates trough new configurations and selects those with better results, guiding the new parameter samples towards better configurations

The size of the population was set to 100 and the number of generations to run the experiment was set to 500, this in order to prevent the tuning of the number of individuals and stop criteria to

[1] https://sourceforge.net/projects/biosbl

cause a bias in the algorithm. The parameters studied as well as the ranges defined and the resulting values are shown in Table 6.5.

The selection of the neighborhood function used in the mutation process employs a roulette wheel. Therefore, the probability of use of the neighborhood functions are tuned under the restriction that the sum of the resulting values should always be less or equal than one. For this tunning process we restricted the application of each mutation function up to 40%, to prevent that the dominance of a single operator would lead the algorithm towards local optima.

Parameter	Range of values	Final Value
Crossover Rate	[0.60,0.95]	0.81
Mutation Rate	[0.05,0.30]	0.27
NNI	[0,0.4]	0.25
SPR	[0,0.4]	0.33
TBR	[0,0.4]	0.07
LSWAP	[0,0.4]	0.29
STEP	[0.0.4]	0.11
Local Search	[0,0.2]	0.13

Table 6.5: Parameters of the implemented NSGA-II algorithm and their corresponding values after the fine-tuning process.

6.5 Experimental results

This section shows a comparative analysis of the results found during our experimentation. First we present a comparison between the most promising multi-objective formulations implemented in the NSGA-II algorithm versus the same algorithm in its single-objective for. All experiments are executed using the parameters obtained during the fine-tuning of the algorithm and the same stop criteria is applied in order to conduct a fair comparison.

Secondly a comparison of the best results obtained by our formulations for the experimental multi-state data instances against the results published by Sergio Santander with the MOABC multi-objective algorithm (Santander Jiménez, 2016), and for the results obtained for the binary instances against Jean-Michel Richer's SAMPARS algorithm (Richer et al., 2013). For those instances with no published results we compare against the best execution obtained using SAMPARS.

6.5.1 Performance of the proposed reformulations against a single-objective algorithm

In order to assess the performance of the proposed multi-objective reformulations, a comparison against the same search algorithm using only one objective, the original MP function, is executed. A summary of the results found during this experimentation is shown below. As previously stated, for every instance used we present the most parsimonious tree found (*Best*), the average quality of the solutions found (*Mean*), the standard deviation of the results found (σ), and the average time employed by each one of the 31 executions of the algorithm (τ).

6.5.1.1 Results for binary character instances

Table 6.6 shows the performance of D_1, D_2, and D_3 as *helper* objectives on real instances with binary encoding. All results are from using the NSGA-II algorithm, in its single-objective and multi-objective form, with the same parameters. For this set of instances, all but one of the results found by the single-objective version of the algorithm were matched or improved by at least one of the proposed multi-objective formulations of the problem.

For this experimentation the only instance that was not improved by any of the proposed multi-objective approaches was *CARP*. The best performance, for this set of instances, is shown by applying D_1, that improves five of the results found by the single-objective algorithm and matches one of them. In second place, D_2 outperforms the single-objective algorithm in four of the instances and equals two of them. Lastly, applying D_3 improves three instances and matches two of them.

Table 6.7 shows the performance of D_1, D_2, and D_3 applied to synthetic instances. Out of the 14 instances used in the experimentation, the single-objective variation of the algorithm outperforms the proposed reformulations in only four: *tst02*, *tst05*, *tst10*, and *tst17*.

For these instances, the best performance is displayed by the use of D_1, outperforming the single-objective version of the algorithm in eight instances and matching one of them. The use of D_2 allows the algorithm to find better solutions in six of the instances, and one aditional equal solution. Finally, using D_3 the algorithm improved five of the tested instances.

Overall, for the 22 binary instances used in this experimentation, D_1 presented competitive solutions for 68% of them. It outperforms the single objective variation in 13, and produced similar results in two of them. D_2 behaved well for 59% of the instances, outperforming the single-objective algorithm in 10 instances and matching three. finally, D_3 found competitive solutions for 45% of the instances, with eight improvements and two matching solutions.

Even though the proposed *helper* objectives were formulated to take advantage of the information of the multi-state encoded sequences, its application to binary encoded sequences seems helpful when compared to a single-objective search.

6.5.1.2 Results for multi-state character instances

The comparison of multi-state character instances has been divided in two tables. First we present the instances reported by Santander Jiménez (2016), and then those published by Strobl and Barker (2016).

The results found for the instance set used by Sandander are shown in Table 6.8. For this dataset all of the proposed formulations matched the single-objective algorithm for the quality of instance rbcL_55, and improved the quality of the solution found for instance mtDNA_186 by 10 (D_1 and D_2) and 11 (D_3) points.

Table 6.9 shows the performance of the proposed multi-objective formulations against a single-objective algorithm on the set of real instances published by Strobl and Barker (2016). For these instances, only three were unmatched by the multi-objective formulations proposed. For this dataset of 15 instances, the use of D_1, D_2, and D_3 allows the multi-objective algorithm to outperform its single-objective version in five, four, and five of the tested instances, respectively. And allows to even the quality of seven instances with all of the proposed reformulations.

In summary, for 19 multi-state character encoded instances D_1 provided competitive solutions for 73% of the instances, while D_2 and D_3 found competitive solutions for 68% of them. Five of the instances used were not improved or matched by any of the proposed multi-objective formulations, two of them belonging to the dataset presented by Santander Jiménez (2016), and three of them to the dataset of Strobl and Barker (2016).

6.5.2 Performance of the proposed multi-objective reformulations against algorithms in the state of the art

We evaluated the performance of the proposed formulations by comparing the obtained results against those reported in the literature. The set of binary instances is compared against the results obtained by SAMPARS (Richer et al., 2013). The first subset of multi-state character encoded instanes is compared against MOABC (Santander Jiménez, 2016). To obtain a comparison basis for the instances with no published results, the algorithm SAMPARS was obtined from Jean-Michel Richer's personal page[2], and used to evaluate all test instances.

The criteria used for the following comparison involves the *Best* parsimony score (either reported in the literature or obtained by the algorithm SAMPARS), the average time (τ) required by the algorithm to reach that solution, and the difference between the best scores found Δ. A possitive value of Δ means the quality of the best tree found by the algorithm of the state of the art was not matched, a zero or negative value means the best tree quality was matched or improved.

[2] http://www.info.univ-angers.fr/ richer/recbio_phylo_sampars.php

6.5.2.1 Results for binary encoded instances

Table 6.10 shows the results obtained, for real-life binary coded instances, by the proposed multi-objective formulations, compared against those obtained by using SAMPARS (Richer et al., 2013).

Table 6.11 shows a comparison for synthetic instances in the literature, the results used for comparison were obtained with the SAMPARS algorithm and are published in (Richer et al., 2013).

Results for binary encoded instances show that multi-objective proposals only obtained competitive solutions for two (D_1, D_3) and three (D_2) real instances, and none of the synthetic instances was matched. Function D_3 shows the worst overall performance by having a higher distance to the best solution reported in most instances.

6.5.2.2 Results for real multi-state encoded instances

Comparison for multi-state character data varies between two subsets. The first one includes results published by Santander Jiménez (2016) using a parallelized variation of his MOABC algorithm (AN-MOABC). We take the times reported for the algorithm using 8 cores as a base for a fair comparison, being the same number of cores used by each processing node in our experimentation . The second subset of instances are those used by Strobl and Barker (2016) for which there are no published results, therefore results used in the comparison were obtained by running SAMPARS, with a time limit of 1 day of execution if the stop criteria is not met automatically.

Table 6.12 shows the results of the comparison against *AN-IMOBA* and *SAMPARS*, for these datasets, the use of any of the proposed reformulations outperforms both of the state of the art algorithms for one of the instances (rbcL_55), reaching results of the same quality using only one fourth of the time required by (AN-IMOBA) and only 3% of the time required by SAMPARS. Regarding the remaining instances, the quality of the best solution reported by the state of the art algorithms was not matched, but the discrepancy between the best results reported and those found by the proposed re-formulations only goes up to 1% in the worst case scenario, and uses a fraction of their required time.

Table 6.13 shows the comparison of the results obtained by SAMARS and thos eobtained by the proposed approaches over the instances used by Strobl and Barker (2016). For this dataset, the best performance is presented by the use of D_2, that found competitive solutions for nine out of 14 instances. Not far behind, D_1 and D_3 were able to find competitive solutions for 8 of the used instances, all of them in a fraction of the time required by SAMPARS. For the remaining instances, the discrepancy between the best found results and those found by the proposed approaches goes up to 2% with the instance *COI_ CAD* using D_3, followed by a 1% discrepancy for *Adams* using D_1 and *Raboski* using D_2, and less than 0.5% difference for the rest of them.

6.6 Discussion of results

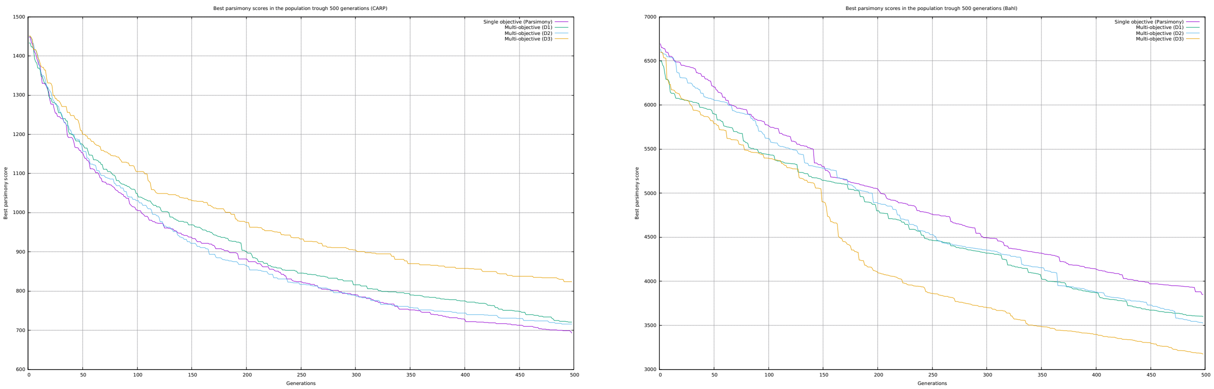
The obtained results with the comparison against a single-objective version of NSGA-II show that the multi-objectivization of the problem helps navigating the search-space. The increase in time when the proposed *helper* objectives are evaluated, indicate that it took longer for the algorithm to be trapped in a local optima. Furthermore, the improvement of the best found solutions, indicate that the gradient added to the search-space when additional information is used leads the MOEA towards better solutions.

Comparing the obtained results over binary-encoded instances against the state of the art algorithms it is noticed that, even if the multi-objectivization with the proposed objectives helps ease the search when implemented in a MOEA, the information obtained from the proposed objectives is not enough to actually help the algorithm find competitive solutions. However, for multi-state character instances, the results were good enough to be competitive against more robust algorithms in the state of the art, finding solutions of matching quality in a fraction of the time.

To grasp the different behavior of the proposed objective functions. We analyze the initial, intermediate, and final population over a 500 generation period for two instances:

- **CARP:** A binary encoded instance, that was not improved by the multi-objectivization. Neither by the comparison against a single-objective NSGA-II, nor against the state of the art.
- **Bahl:** A multi-state character encoded instance, that was competitive using multi-objectivization. Both against a single-objective search and in comparison with the state of the art.

The quality of the best solution in the population over the 500 evaluated generations can be seen for both instances in Figure 6.1.



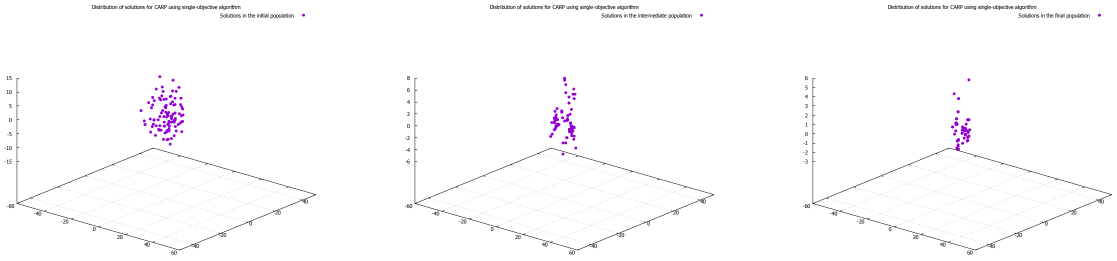
(a) Best individual per generation for *CARP* instance.

(b) Best individual per generation for *Bahl* instance.

Figure 6.1: Comparison between the best individual found per generation using the four variations of NSGA-II used in experimentation.

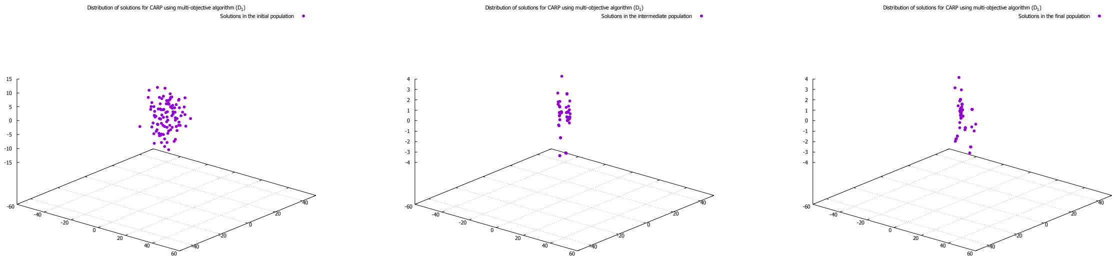
To have an insight in the population during the search process, we examined the spatial distribution of the entire population at the first, middle and last generations of the search for each of the approaches: using the original function (MP), and applying the proposed *helper* objectives (D_1, D_2, D_3). This analysis, inspired by Lai and Hao (2016); Porumbel et al. (2010), obtains an image of the distribution of the solutions, at a given time, in Euclidean R^3 space. This analysis consists in two steps: First, it obtains a $Z_{n\times n}$ distance matrix from the population. Then, it maps the elements in the distance matrix to coordinate points in the Euclidean space R^3.

To obtain the $Z_{n\times n}$ distance matrix between solutions we employ the *SPR distance function* implemented in the *Phangorn* R package (Schliep, 2010; Schliep et al., 2017). Then, p coordinate points in the Euclidean space R^3 can be generated by employing the *cmdscale* algorithm implemented in the R language. Finally, we plot a scatter using the generated coordinate points. Figures 6.2, 6.3, 6.4, and 6.5 show the obtained graphs for binary instance CARP, and Figures 6.6, 6.7, 6.8, and 6.9 for multi-state encoded instance Bahl.



(a) Initial population. (b) Intermediate population. (c) Final population.

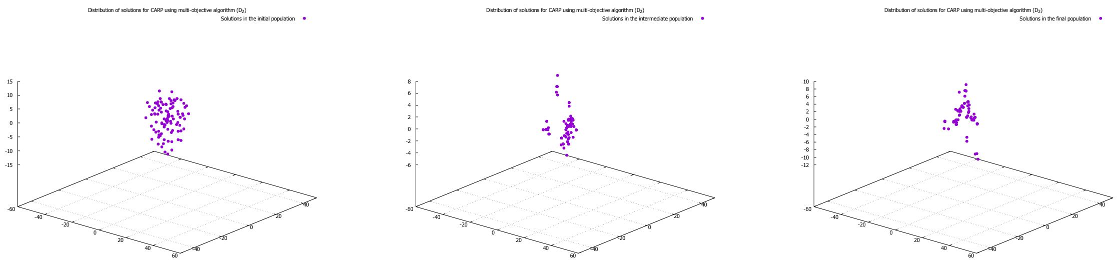
Figure 6.2: Spatial distributions of the solutions in the initial, intermediate and final population of a single-objective search on the binary encoded instance *CARP*.



(a) Initial population. (b) Intermediate population. (c) Final population.

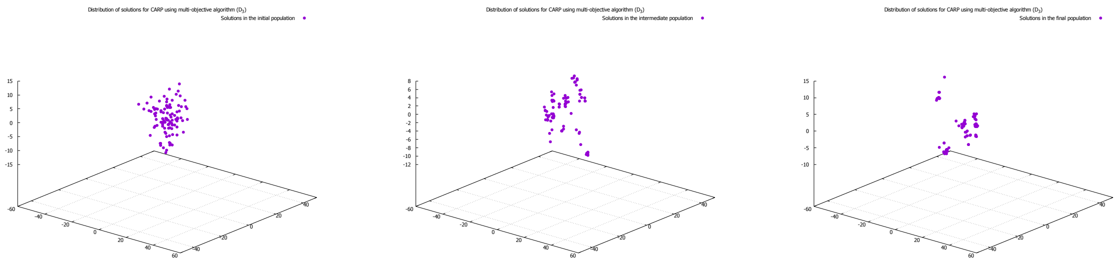
Figure 6.3: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_1) on the binary encoded instance *CARP*.

For the CARP instance, we can observe that the resulting graphs are quite similar, and the populations in the final iteration of the algorithm are really close. This might indicate that the



(a) Initial population. (b) Intermediate population. (c) Final population.

Figure 6.4: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_2) on the binary encoded instance *CARP*.

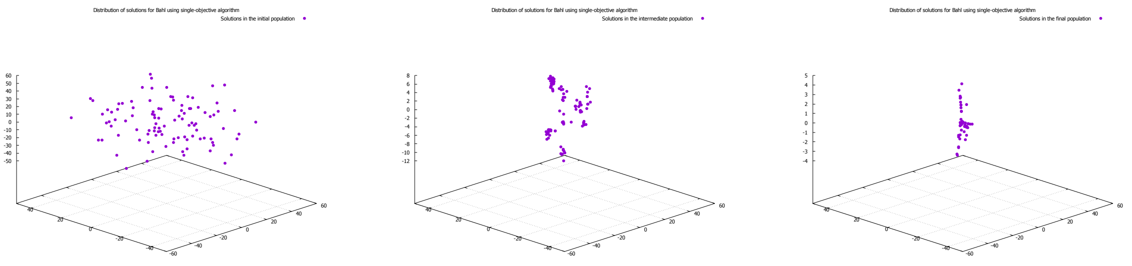


(a) Initial population. (b) Intermediate population. (c) Final population.

Figure 6.5: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_3) on the binary encoded instance *CARP*.

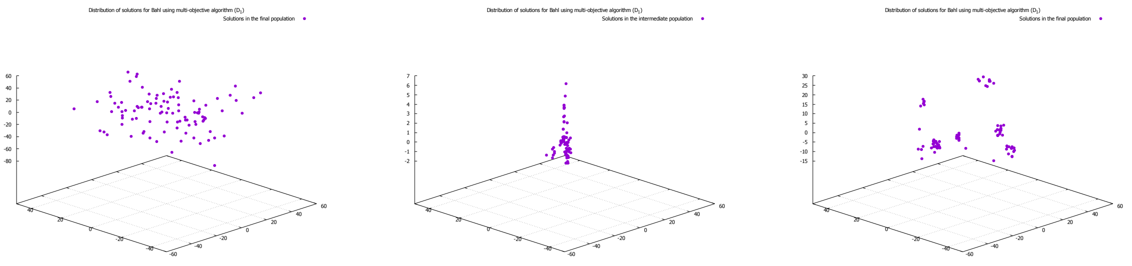
information obtained from the binary encoding is not enough to help the algorithm escape from the basin attraction of the current local optima.

In contrast with the CARP instance, the results for the multi-objective variations of the NSGA-II algorithm over the Bahl instance seem to diversify the solutions in the population as the algorithm proceeds. It is evident that the solutions obtained in the single-objective variation of the algorithm seem too similar among each other. In contrast, the final population of the multi-objective variations seem to draw away from each other, allowing an exploration of a greater portion of the search space.



(a) Initial population. (b) Intermediate population. (c) Final population.

Figure 6.6: Spatial distributions of the solutions in the initial, intermediate and final population of a single-objective search on the multi-state character encoded instance *Bahl*.



(a) Initial population. (b) Intermediate population. (c) Final population.

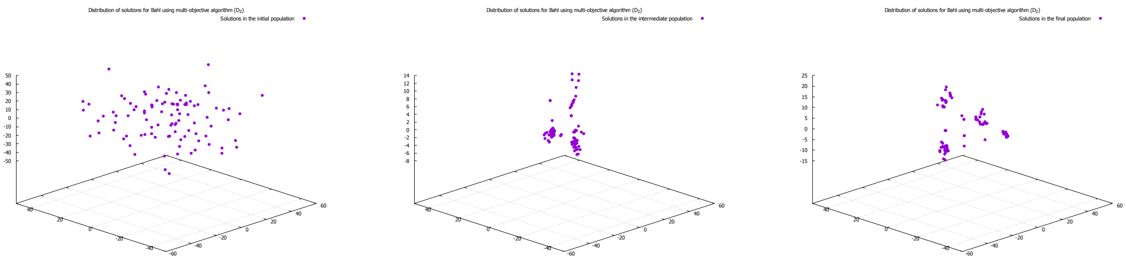
Figure 6.7: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_1) on the multi-state character encoded instance *Bahl*.

6.7 Chapter summary

This chapter described the experimentation conducted with the implemented algorithms and the proposed *helper* objective functions.

The obtained results show that the reduction of neutrality in the fitness landscape of the problem, caused by the use of the selected multi-objective reformulations and the gradient they provide, allow the NSGA-II algorithm outperform a single-objective algorithm that operates under the same circumstances. It can be observed that even if the proposed objective functions are designed for multi-state character datasets, binary datasets can also benefit from them to a certain degree.

By comparing the performance obtained by our proposal against algorithms in the state of the

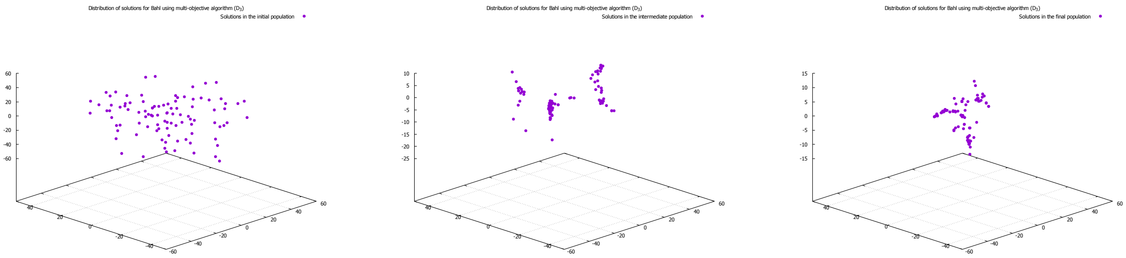


(a) Initial population.

(b) Intermediate population.

(c) Final population.

Figure 6.8: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_2) on the multi-state character encoded instance *Bahl*.



(a) Initial population.

(b) Intermediate population.

(c) Final population.

Figure 6.9: Spatial distributions of the solutions in the initial, intermediate and final population of a multi-objective search (D_3) on the multi-state character encoded instance *Bahl*.

art we see that, for a considerable percentage of the used test instances, we can obtain competitive solutions in a fraction of the time employed by the state of the art algorithms.

The following chapter presents the conclusions obtained from this research, and a duscussion of the future work that could derive from it.

	Single objective				D_1				D_2				D_3			
Instance	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ
ANGI	221	222.933	1.596	27.610	**220**	223.595	1.633	48.975	**221**	225.442	1.028	69.424	**219**	226.101	1.273	94.940
CARP	553	563.533	5.818	121.120	556	562.341	3.151	183.683	557	566.627	3.763	237.627	566	620.038	2.818	307.264
ETHE	375	377.133	0.730	32.659	**373**	378.447	1.242	65.016	**372**	380.975	1.893	94.865	**375**	378.855	0.887	88.713
GOLO	512	517.033	3.011	63.980	**509**	515.432	2.725	129.164	**510**	520.119	3.168	134.064	513	538.000	2.545	163.409
GRIS	173	173.633	0.490	11.929	**172**	174.143	0.525	35.166	**172**	176.776	0.662	62.555	**172**	186.230	0.629	101.209
ROPA	328	331.300	1.822	53.654	**328**	331.841	1.769	104.124	329	335.467	1.432	154.113	332	359.554	2.033	157.534
SCHU	779	786.300	3.725	60.039	**773**	782.527	4.209	187.093	**773**	786.004	4.059	273.130	**772**	785.232	4.350	259.105
TENU	686	694.267	4.362	30.408	687	693.138	3.693	85.548	**686**	694.861	1.845	105.548	**686**	699.871	1.956	103.003

Table 6.6: Comparison between the single-objective version of the NSGA-II algorithm against the multi-objective version using D_1, D_2, and D_3 as secondary objective on real binary encoded instances.

	Single objective				D_1				D_2				D_3			
Instance	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ
tst01	562	566.633	2.810	39.925	**552**	569.540	3.984	70.538	**558**	575.463	3.411	75.638	564	578.692	3.044	77.941
tst02	1380	1392.700	7.130	66.485	1386	1405.543	4.475	86.710	1393	1418.973	5.602	80.966	1393	1428.579	5.356	90.670
tst03	863	871.500	4.289	49.583	**859**	874.645	4.686	77.666	**856**	881.704	4.713	87.376	864	892.395	4.687	90.904
tst04	627	634.700	2.037	38.519	**622**	635.025	5.090	88.464	**618**	636.642	3.742	103.088	**621**	652.354	5.003	99.602
tst05	807	817.400	5.685	57.972	810	820.988	3.612	99.894	813	836.846	3.854	103.077	813	840.802	5.699	97.968
tst06	623	627.733	2.924	41.067	**621**	631.195	2.176	90.922	624	642.918	2.480	85.971	**621**	645.712	3.291	97.425
tst07	1324	1333.233	2.897	56.216	**1310**	1336.147	7.304	117.706	**1323**	1353.703	5.699	105.703	**1321**	1358.093	5.898	110.280
tst08	895	901.533	4.100	70.465	**895**	906.997	2.741	98.665	899	920.640	4.518	95.593	905	933.897	2.484	99.328
tst09	1184	1191.400	3.738	70.164	1186	1206.334	5.361	103.035	**1184**	1212.471	5.775	121.118	**1182**	1207.846	5.849	130.211
tst10	754	767.967	4.612	73.583	760	778.261	5.599	97.329	767	782.707	3.241	94.077	766	784.316	3.518	115.975
tst17	2507	2519.600	5.685	93.611	2514	2534.884	4.752	148.276	2523	2566.851	5.359	134.328	2511	2556.794	7.565	151.975
tst18	1589	1596.267	4.884	101.623	**1586**	1605.888	4.587	151.467	1592	1622.849	4.398	149.926	1593	1627.470	5.078	155.631
tst19	1078	1086.533	3.159	64.170	**1075**	1089.736	3.552	140.719	**1074**	1101.167	5.902	165.984	1086	1120.237	2.683	132.608
tst20	733	733.800	0.664	27.354	**714**	729.087	5.453	114.098	**705**	732.856	6.736	170.083	**722**	752.483	3.652	140.416

Table 6.7: Comparison between the single-objective version of the NSGA-II algorithm against the multi-objective version using D_1, D_2, and D_3 as secondary objective on synthetic binary encoded instances.

	Single objective				D_1				D_2				D_3			
Instance	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ
mtDNA_186	2442	2443.33333	0.84418225	296.129354	**2437**	2441.270	1.668	2061.326	**2437**	2446.757	1.562	4219.646	**2436**	2442.423	2.063	3493.332
rbcL_55	4874	4877.13333	1.61316424	63.954453	**4874**	4891.102	2.599	119.922	**4874**	4895.254	2.426	161.200	**4874**	4901.163	2.738	146.534
RPDII_218	41652	41830.0333	59.7762975	1826.40784	41876	42031.254	45.988	1749.471	41973	42118.937	41.357	2150.482	41928	42053.996	45.977	2240.465
ZILLA_500	16324	16371.1333	23.2315679	795.581476	16344	16386.916	14.358	1093.073	16374	16425.446	9.745	868.819	16382	16482.974	8.699	984.651

Table 6.8: Comparison between the single-objective version of the NSGA-II algorithm against the multi-objective version using D_1, D_2, and D_3 as secondary objective on real multi-state character encoded instances.

	Single objective				D_1				D_2				D_3			
Instance	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ	Best	Mean	σ	τ
Molecular	70148	70430.9667	214.510105	821.433946	70228	70622.616	92.148	1351.179	70318	71859.213	232.959	1378.489	70227	71074.902	174.037	1470.099
Adams	1074	1076.43333	0.81720015	63.3714542	**1073**	1080.299	1.106	309.792	**1067**	1078.972	2.369	1637.385	**1069**	1079.694	1.938	598.256
Alignment4	4115	4116.7	1.98529074	66.563969	4119	4217.745	2.771	203.939	4117	4264.022	3.457	265.668	4116	4224.539	2.350	229.283
Alldata	2925	2931.06667	2.70291456	188.328614	**2920**	2937.867	4.160	537.753	**2919**	2933.933	3.542	1184.089	**2920**	2931.771	4.166	761.967
Alstrom5	2866	2866.7	1.08754707	40.5012911	**2866**	2898.774	0.699	113.934	**2866**	2933.195	0.819	181.111	**2866**	2905.859	0.761	138.334
Angiosperm	93	93	0	0.47277197	**93**	93.000	0.000	0.484	**93**	93.000	0.000	0.456	**93**	93.000	0.000	0.443
Bahl	983	983.833333	0.37904902	101.135798	**982**	983.936	0.679	222.982	**980**	985.166	1.125	7511.874	**982**	986.826	0.868	618.852
COI_CAD	10448	10501.1	34.3615162	646.616729	**10418**	10478.966	21.372	1909.576	10464	10576.856	17.178	906.904	10471	10537.807	16.295	1405.345
Pasach	363	363	0	13.5440144	**363**	363.259	0.000	19.109	**363**	365.219	0.000	46.416	**363**	365.182	0.000	26.655
Rabosky	49516	49593.7333	52.5815841	2347.86279	49725	49947.602	43.620	1561.920	49768	50090.717	39.076	1779.178	49758	50079.967	41.501	1514.962
S4	44960	44960	0	159.224795	**44960**	45070.267	3.776	421.311	**44960**	45123.940	9.957	574.669	**44960**	45072.044	8.097	517.189
Sphaero	15629	15632.2333	7.45415105	41.4434911	**15629**	15684.032	3.793	74.174	**15629**	15770.990	7.613	115.843	**15629**	15724.516	3.847	128.995
Thomomys	1500	1500	0	19.3389116	**1500**	1520.512	0.346	82.552	**1500**	1525.379	0.379	116.864	**1500**	1520.477	0.254	96.552
Vati6	1072	1072	0	16.7775354	**1071**	1071.287	0.407	67.033	**1071**	1071.589	0.490	109.543	**1071**	1072.047	0.407	67.624
VatiN	278	278	0	26.0752983	**278**	278.000	0.000	41.081	**278**	278.471	0.000	290.350	**278**	282.127	0.000	72.956

Table 6.9: Comparison between the single-objective version of the NSGA-II algorithm against the multi-objective version using D_1, D_2, and D_3 as secondary objective on real multi-state character encoded instances.

	SAMPARS		D_1			D_2			D_3		
Instances	Best	τ	Best	τ	Δ	Best	τ	Δ	Best	τ	Δ
ANGI	216	260.550	220	48.975	4	221	69.424	5	219	94.940	3
CARP	548	5534.510	556	183.683	8	557	237.627	9	566	307.264	18
ETHE	372	419.280	373	65.016	1	**372**	94.865	**0**	375	88.713	3
GOLO	496	693.220	509	129.164	13	510	134.064	14	513	163.409	17
GRIS	172	329.620	**172**	35.166	**0**	**172**	62.555	**0**	**172**	101.209	**0**
ROPA	325	675.210	328	104.124	3	329	154.113	4	332	157.534	7
TENU	682	789.880	687	85.548	5	686	105.548	4	686	103.003	4
SCHU	759	4003.370	773	187.093	14	773	273.130	14	772	259.105	13

Table 6.10: Performance comparison among SAMPARS (Richer et al., 2013) and NSGA-II using the proposed *helper* objectives over nine real-life binary instances.

	SAMPARS		D_1			D_2			D_3		
Instances	Best	τ	Best	τ	Δ	Best	τ	Δ	Best	τ	Δ
tst01	545	1407.570	552	70.538	7	558	75.638	13	564	77.941	19
tst02	1354	1938.230	1386	86.710	32	1393	80.966	39	1393	90.670	39
tst03	833	2506.300	859	77.666	26	856	87.376	23	864	90.904	31
tst04	587	1341.170	622	88.464	35	618	103.088	31	621	99.602	34
tst05	789	2007.900	810	99.894	21	813	103.077	24	813	97.968	24
tst06	596	1164.270	621	90.922	25	624	85.971	28	621	97.425	25
tst07	1269	4063.800	1310	117.706	41	1323	105.703	54	1321	110.280	52
tst08	852	2884.730	895	98.665	43	899	95.593	47	905	99.328	53
tst09	1141	3237.530	1186	103.035	45	1184	121.118	43	1182	130.211	41
tst10	720	2288.000	760	97.329	40	767	94.077	47	766	115.975	46
tst17	2450	8020.230	2514	148.276	64	2523	134.328	73	2511	151.975	61
tst18	1521	4451.370	1586	151.467	65	1592	149.926	71	1593	155.631	72
tst19	1012	6875.300	1075	140.719	63	1074	165.984	62	1086	132.608	74
tst20	654	7149.430	714	114.098	60	705	170.083	51	722	140.416	68

Table 6.11: Performance comparison among SAMPARS (Richer et al., 2013) and NSGA-II using the proposed *helper* objectives over 14 synthetic binary instances reported by Santander.

	AN-MOABC		SAMPARS		D_1			D_2			D_3		
Instances	Best	τ	Best	τ	Best	τ	Δ	Best	τ	Δ	Best	τ	Δ
MtDNA_186	2431	6265.240	2431	86456.160	2437	2061.326	6	2437	4219.646	6	2436	3493.332	5
rbcL_55	4874	775.470	4874	4827.480	**4874**	119.922	**0**	**4874**	161.200	**0**	**4874**	146.534	**0**
RPDII_218	41488	6642.010	41488	86469.170	41876	1749.471	388	41973	2150.482	485	41928	2240.465	440
ZILLA_500	16218	8690.350	16218	86481.110	16344	1093.073	126	16374	868.819	156	16382	984.651	164

Table 6.12: Performance comparison among AN-MOABC (Santander Jiménez, 2016), SAMPARS (Richer et al., 2013), NSGA-II using the proposed *helper* objectives over four real multi-state character instances reported by Santander.

	SAMPARS		D_1			D_2			D_3		
Instances	Best	τ	Best	τ	Δ	Best	τ	Δ	Best	τ	Δ
Molecular	70036	86483.190	70228	1351.179	192	70318	1378.489	282	70227	1470.099	191
Adams	1062	86480.220	1073	309.792	11	1067	1637.385	5	1069	598.256	7
Alignment4	4115	7423.490	4119	203.939	4	4117	265.668	2	4116	229.283	1
Alldata	2914	86481.020	2920	537.753	6	2919	1184.089	5	2920	761.967	6
Alstrom5	2866	4885.020	**2866**	113.934	**0**	**2866**	181.111	**0**	**2866**	138.334	**0**
Angiosperm	93	3.580	**93**	0.484	**0**	**93**	0.456	**0**	**93**	0.443	**0**
Bahl	980	86485.700	982	222.982	2	**980**	7511.874	**0**	982	618.852	2
COI_CAD	10233	86480.000	10418	1909.576	185	10464	906.904	231	10471	1405.345	238
Pasach	363	1744.710	**363**	19.109	**0**	**363**	46.416	**0**	**363**	26.655	**0**
Rabosky	49234	86465.680	49725	1561.920	491	49768	1779.178	534	49758	1514.962	524
S4	44960	23894.950	**44960**	421.311	**0**	**44960**	574.669	**0**	**44960**	517.189	**0**
Sphaero	15629	3017.300	**15629**	74.174	**0**	**15629**	115.843	**0**	**15629**	128.995	**0**
Thomomys	1500	3441.950	**1500**	82.552	**0**	**1500**	116.864	**0**	**1500**	96.552	**0**
Vati6	1071	4041.420	**1071**	67.033	**0**	**1071**	109.543	**0**	**1071**	67.624	**0**
VatiN	278	19422.180	**278**	41.081	**0**	**278**	290.350	**0**	**278**	72.956	**0**

Table 6.13: Performance comparison among SAMPARS (Richer et al., 2013) and NSGA-II using the proposed *helper* objectives over four real multi-state character instances published by Strobl and Barker (2016).

7

Conclusions and future work

In this thesis work we explored two multi-objectivization paradigms applied to the MP problem. We proposed six reformulations based on the decomposition of the original objective function, and 12 *helper* objective functions for the problem, for a total of 18 proposals of multi-objectivization.

After extensive evaluation and analysis, the three most promising of them were selected to be implemented in a multi-objective evolutionary algorithm. These reformulations were selected according to their correlation with the parsimony score, a strong correlation between them increases the capacity to discern between similar solutions, while mantaining the most parsimonious ones as high quality solutions.

The NSGA-II algorithm (Deb et al., 2002) was implemented using a topological crossover operator based on Tree Bisection and Reconnection (Congdon, 2001; Lewis, 1998; Matsuda, 1995), and five mutation operators: Nearest Neighbor Interchange (Andreatta and Ribeiro, 2002; Moore et al., 1973; Vazquez Ortiz and Rodriguez Tello, 2011; Waterman and Smith, 1978), Subtree Pruning and Regraft (Andreatta and Ribeiro, 2002; Vazquez Ortiz and Rodriguez Tello, 2011), Tree Bisection and Reconnection (Swofford et al., 1996; Vazquez Ortiz and Rodriguez Tello, 2011), Leaf Swap (Cotta and Moscato, 2002; Sonco Alvarez and Ayala Rincon, 2017), and Single Step (Andreatta and Ribeiro, 2002; Sonco Alvarez and Ayala Rincon, 2017; Waterman and Smith, 1978).

A comparison was conducted, using the implemented NSGA-II algorithm applying the three selected multi-objective reformulations, against a single-objective version of the NSGA-II using MP, and against algorithms of the state of the art. This comparisson was conducted in terms of quality of the solution (MP score), and time required by each algorithm, using real-life multi-state encoded instances, synthetic, and real binary encoded instances taken from the literature.

This chapter presents the main conclusions obtained from this research, the verification of the hypothesis presented in Section 1.3, and the future work that can be derived from this study.

7.1 Conclusions

The evaluation conducted over the proposed multi-objective reformulations of the MP problem allowed to select three reformulations that help discern between solutions of similar quality, and that, as shown in Section 4.3, reduce the neutrality in the fitness landscape of the problem. Therefore, the first specific objective of this research is fullfilled.

Based on the results of the comparison against a single-objective search presented in Section 6.5, we conclude that the proposed *helper* objectives are a competitive approach for the multi-objectivization of the MP problem.

For the comparison against a single-objective search in Section 6.5.1, the performance of the NSGA-II algorithm using the selected helper objectives is as follows:

- D_1 found competitive solutions for 71.43% of the test instances, improving the results found by the single-objective algorithm for 45.2% of them.

- D_2 found competitive solutions for 64.29% of the test instances, improving the results found by the single-objective algorithm for 35.7% of them.

- D_3 found competitive solutions for 57.1% of the test instances, improving the results found by the single-objective algorithm for 30.9% of them.

The comparison against the state of the art, using real multi-state encoded instances, in Section 6.5.2 shows that the proposed formulations match the algorithms in the state of the art in 47.3% (D_1 and D_3) and 52.6% (D_2) of the test instances, doing so in only a fraction of the time required by the state of the art algorithms. Thus, completing the second and third specific objective of this research work.

The results obtained during this research agree with the initial hypothesis. We found that the three proposed reformulations of the MP problem successfully modify the fitness landscape of the problem by reducing its neutrality, increasing the capacity of the NSGA-II algorithm to discern between similar solutions and allowing it to find competitive solutions for the tested instances, doing so in less time than the required by the algorithms in the state of the art.

7.2 Future work

Performance shown by the proposed multi-objective formulations might be improved by implementing them in specialized algorithms, such as those found in the state of the art of the problem. The low evaluation time required by the proposed objective functions and the competitiveness of the found solutions hint that a specialized algorithm might be able to find matching or improving solutions in less computational time than the reported.

The results obtained by the evaluation of binary instanes suggest that there might exist a different formulation applicable to binary encoded instances that yield competitive solutions to the state of the art algorithms. Therefore, the analysis of new functions designed for this kind of instance remains an open problem.

The evaluation of individual solutions might be accelerated by evaluating both criteria in a single step, doing so while applying paralelization to the algorithm could reduce the computational time for large instances.

An improvement in the crossover function used could be achieved by means of local search, a method to evaluate the subtree bisected or the selection of the reconnection site could be evaluated, in order to obtain a crossover that improves the search but not up to a point that rushes toward local optima solutions.

A

Correlation Graphs

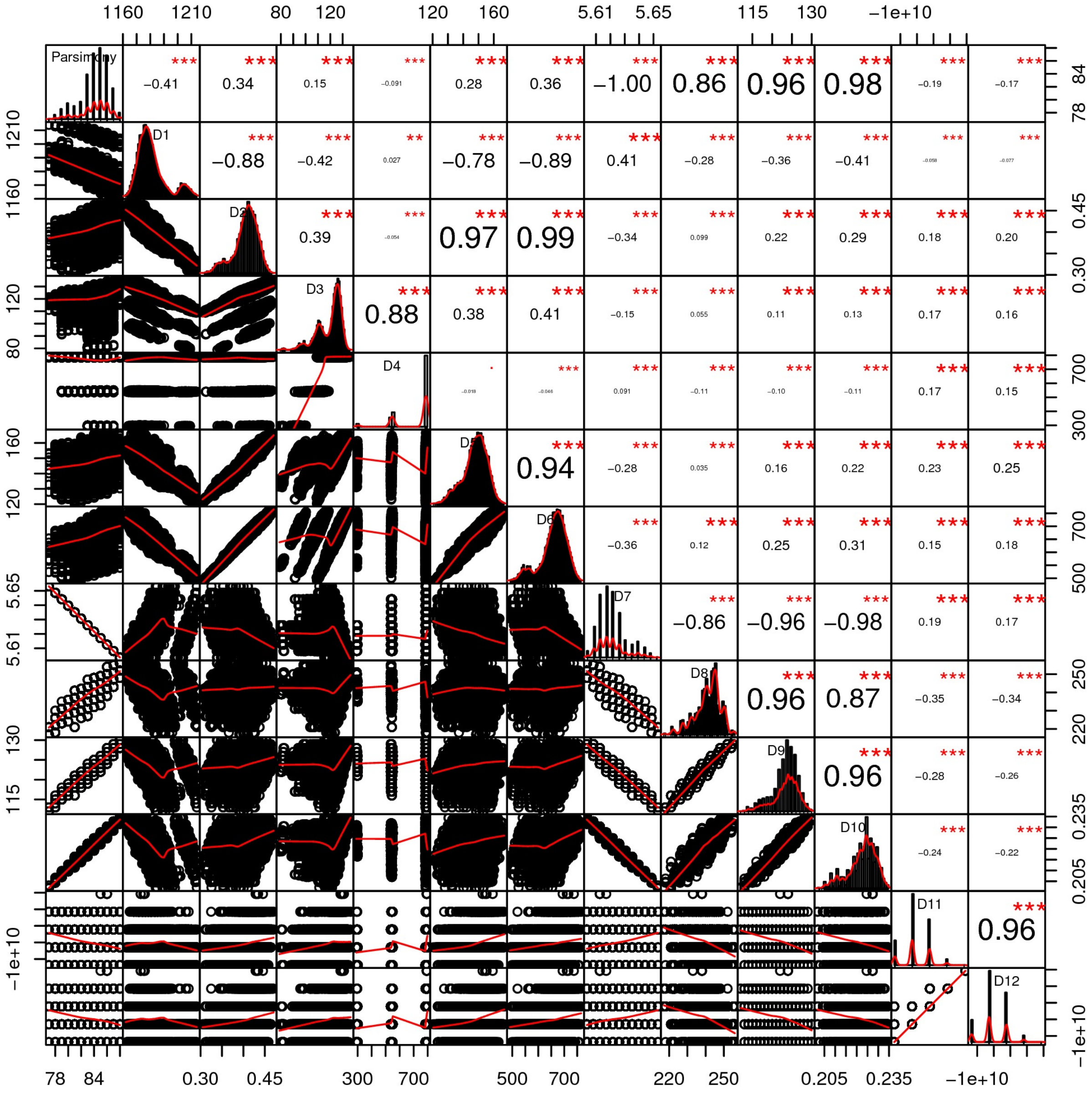


Figure A.1: Correlation between objective function evaluations for instance $drosophyl_1$

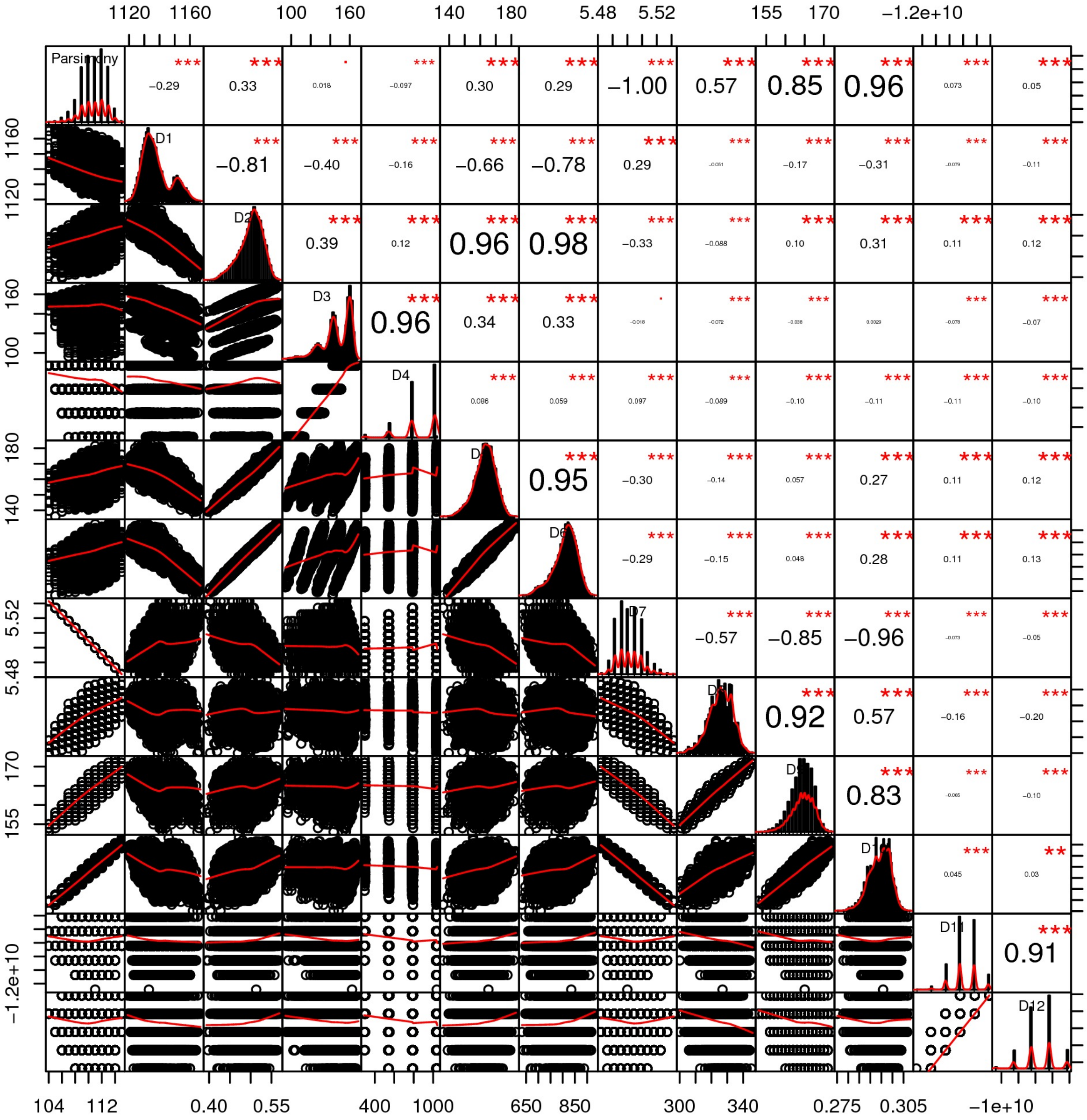


Figure A.2: Correlation between objective function evaluations for instance $drosophyl_2$

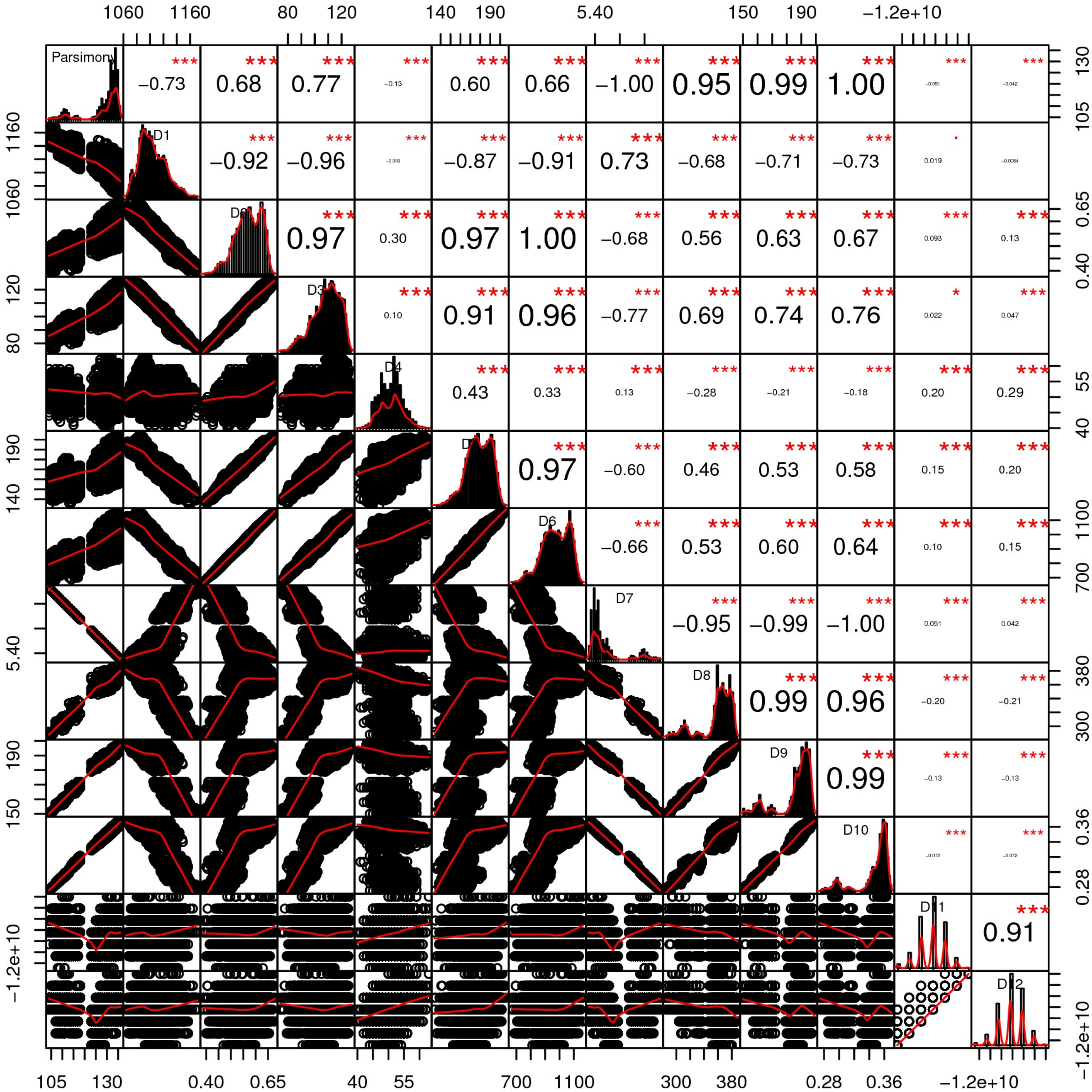


Figure A.3: Correlation between objective function evaluations for instance $drosophyl_3$

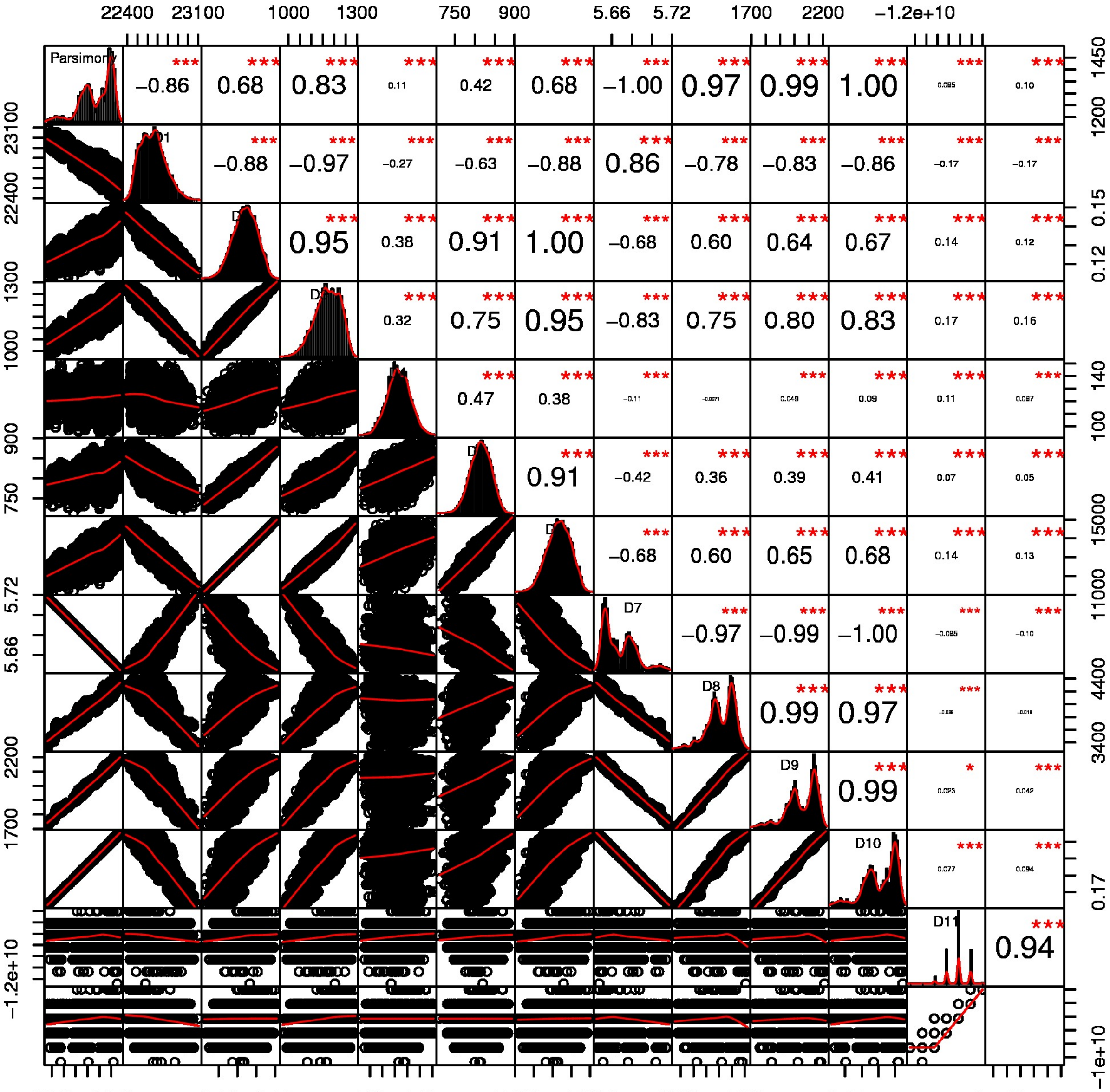


Figure A.4: Correlation between objective function evaluations for instance $RPDII_1$

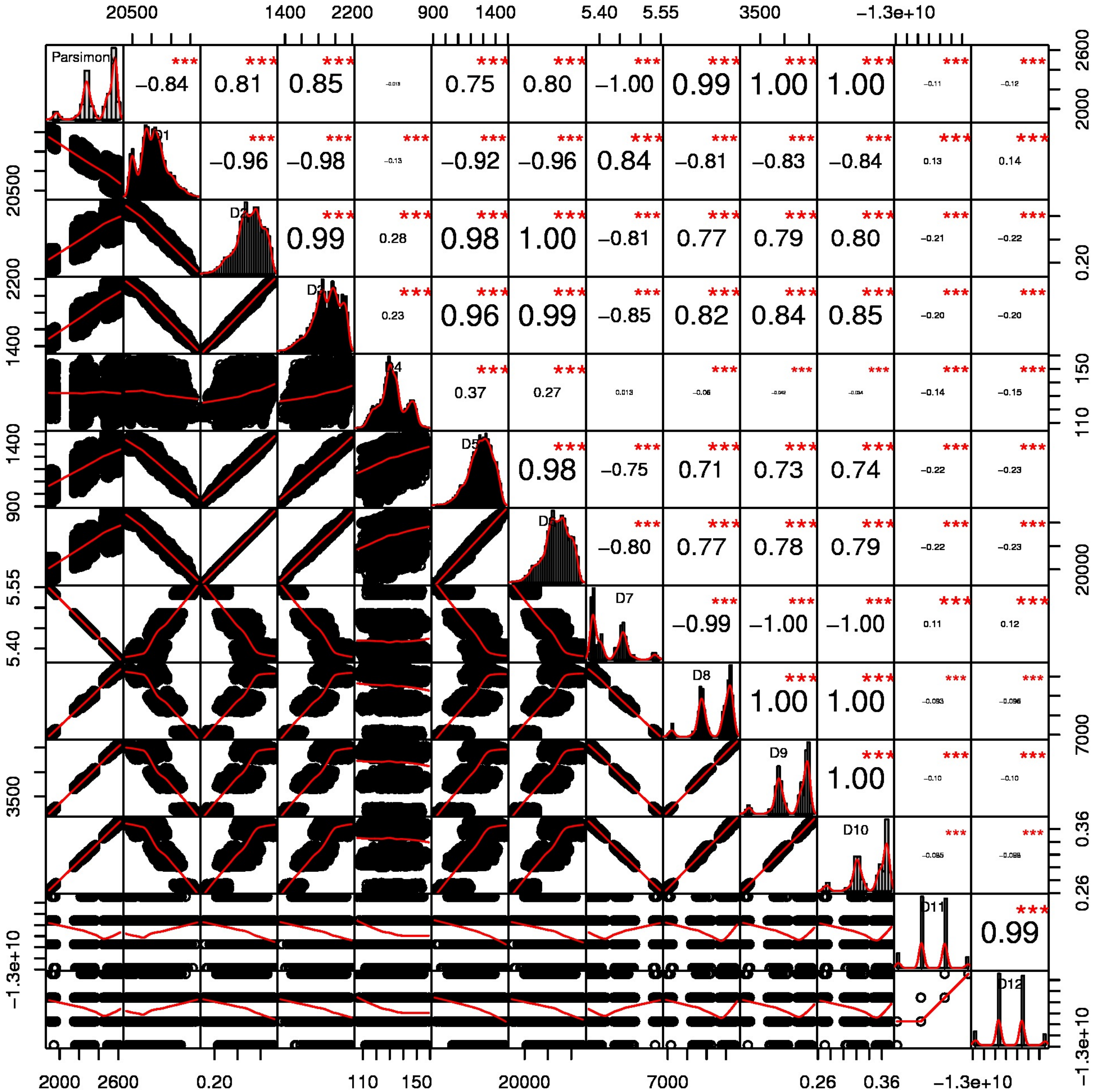


Figure A.5: Correlation between objective function evaluations for instance $RPDII_2$

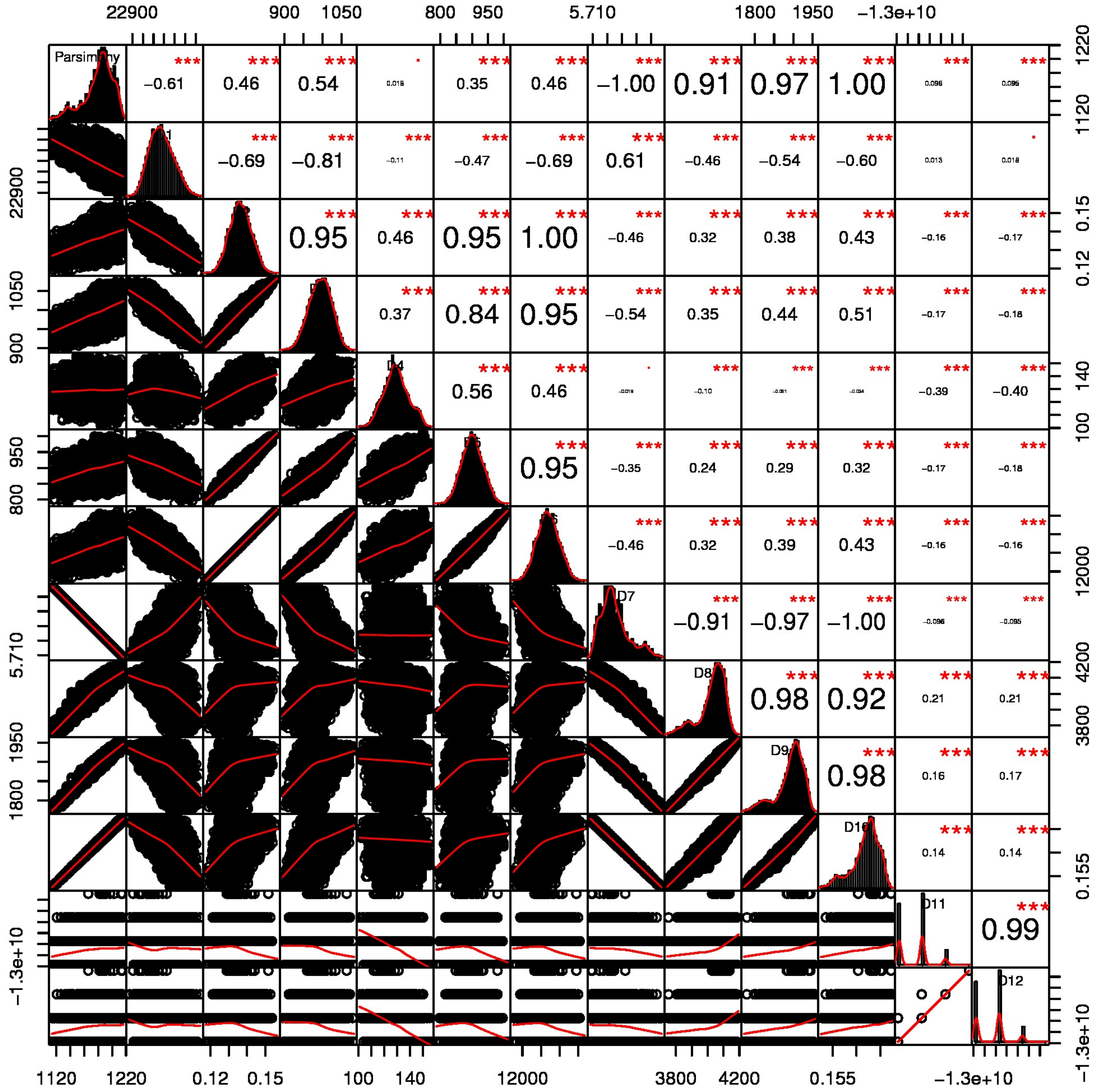


Figure A.6: Correlation between objective function evaluations for instance $RPDII_3$

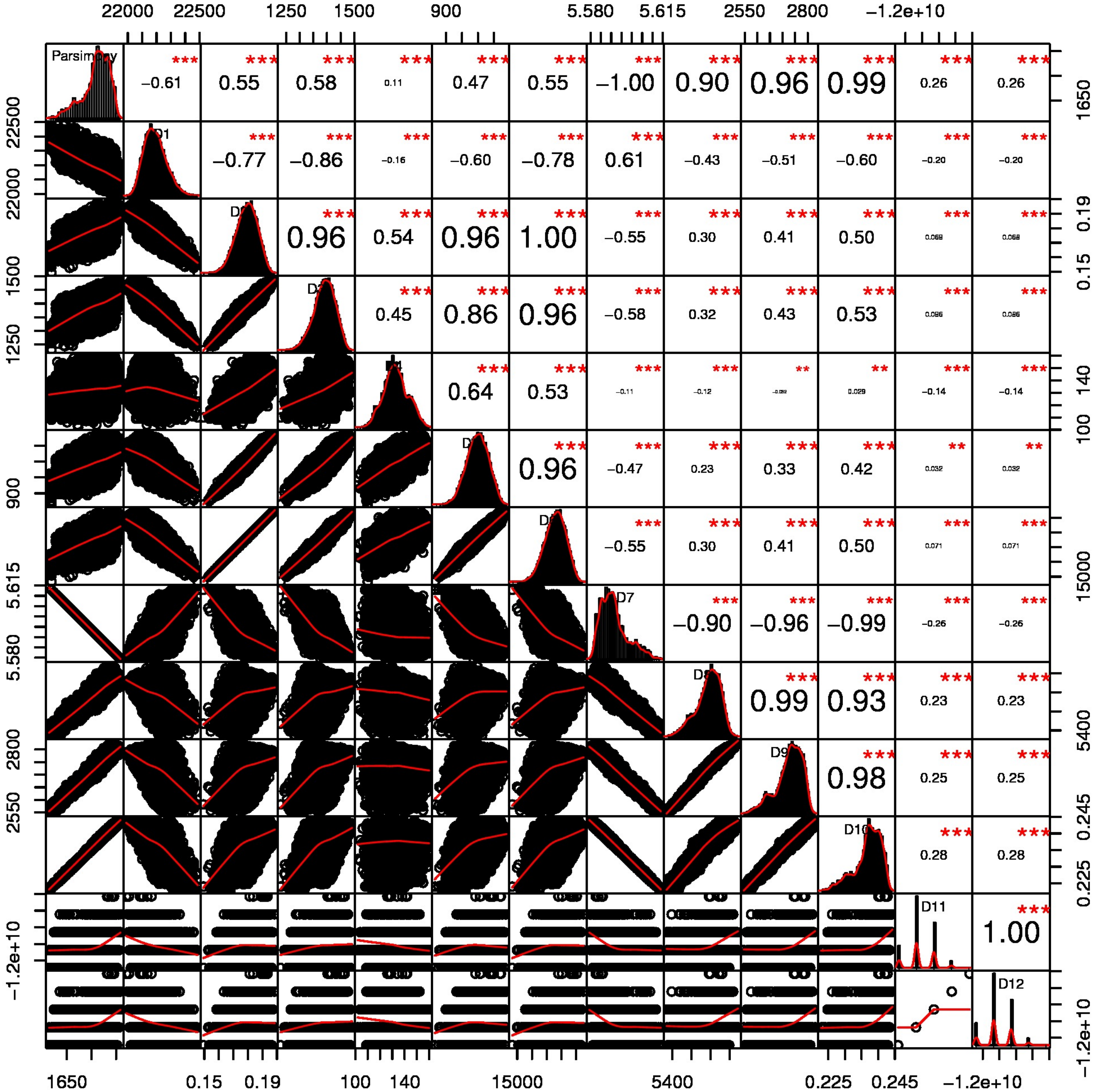


Figure A.7: Correlation between objective function evaluations for instance $RPDII_4$

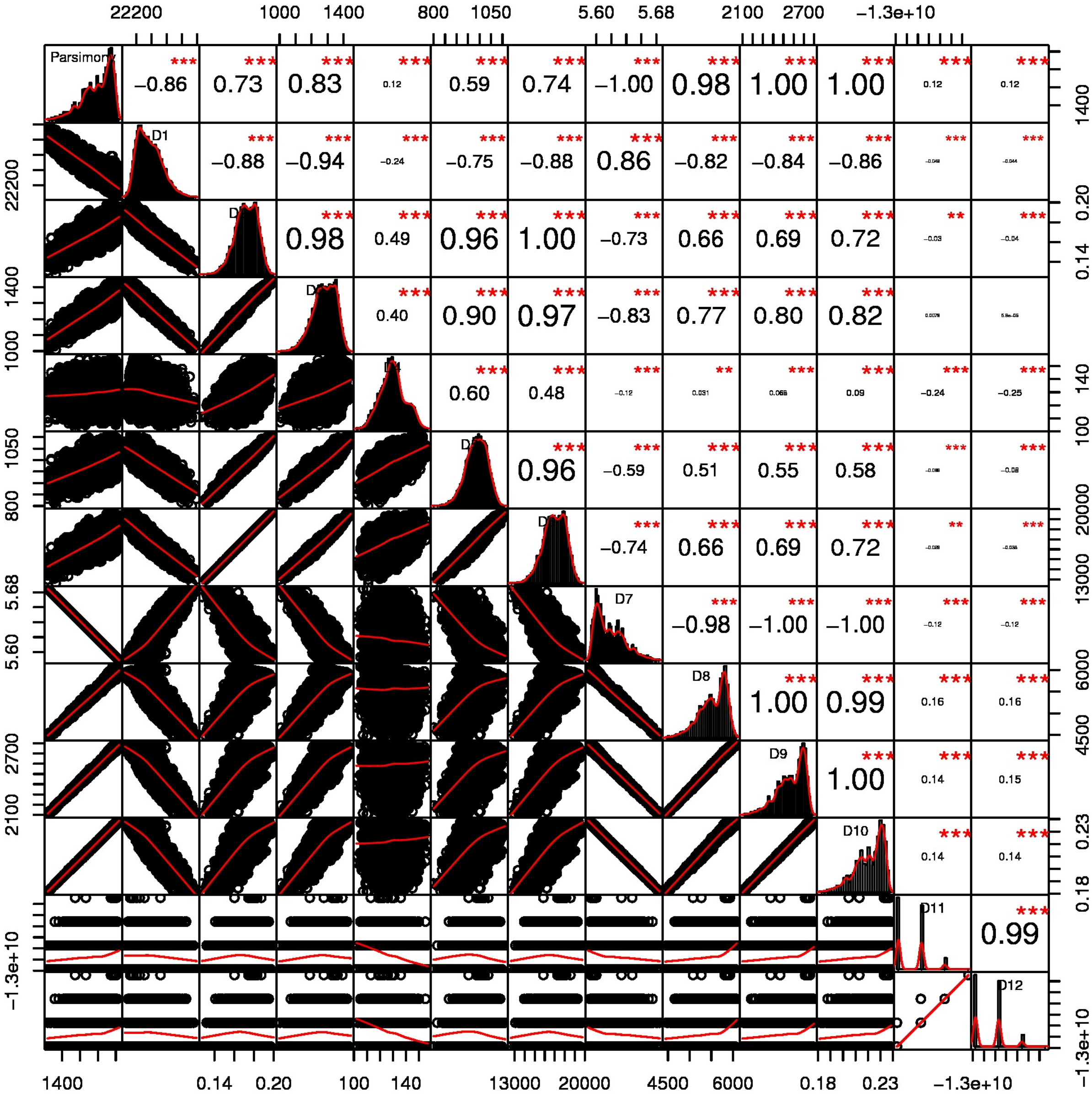


Figure A.8: Correlation between objective function evaluations for instance $RPDII_5$

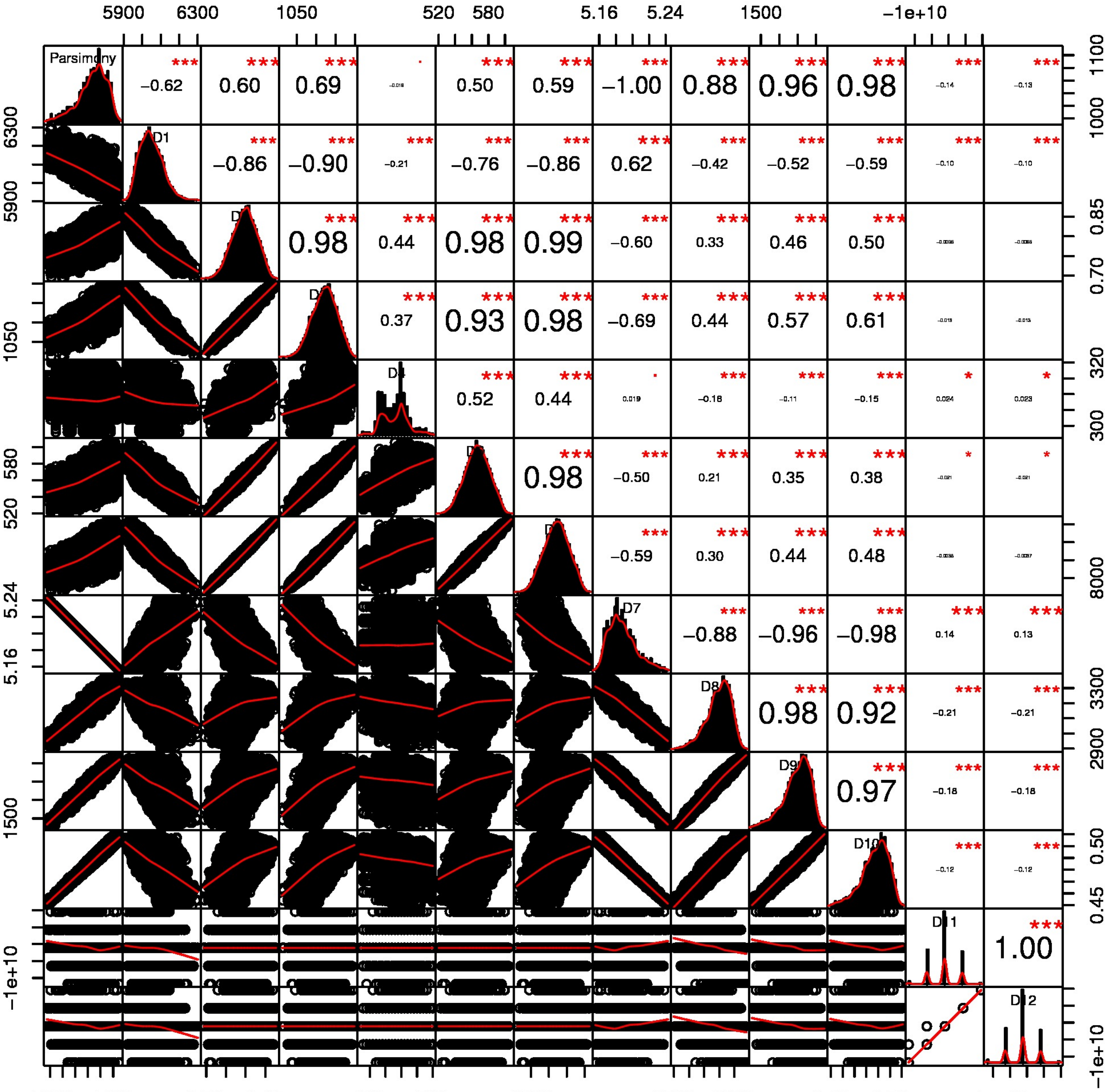


Figure A.9: Correlation between objective function evaluations for instance $rbcL_1$

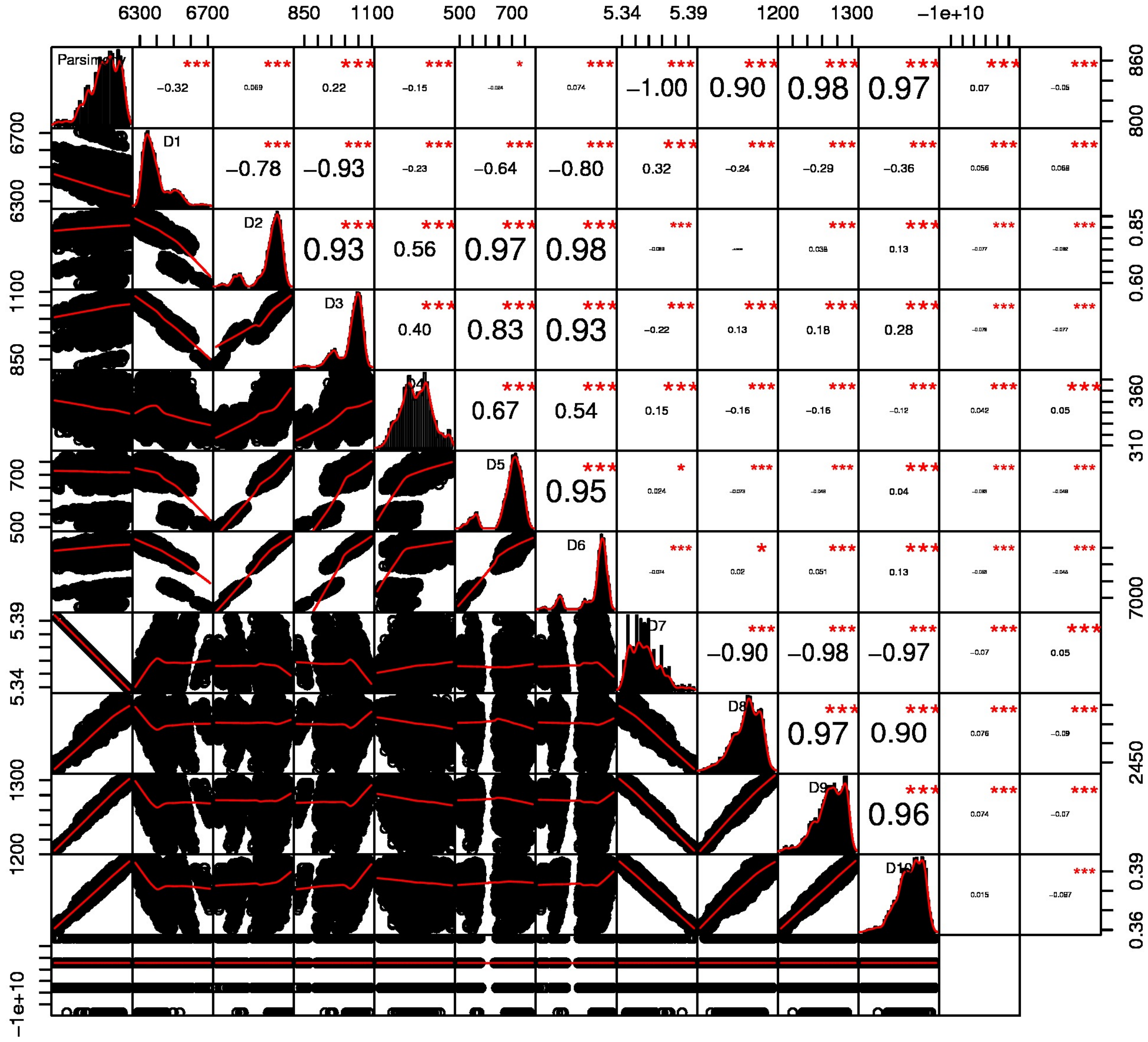


Figure A.10: Correlation between objective function evaluations for instance $rbcL_2$

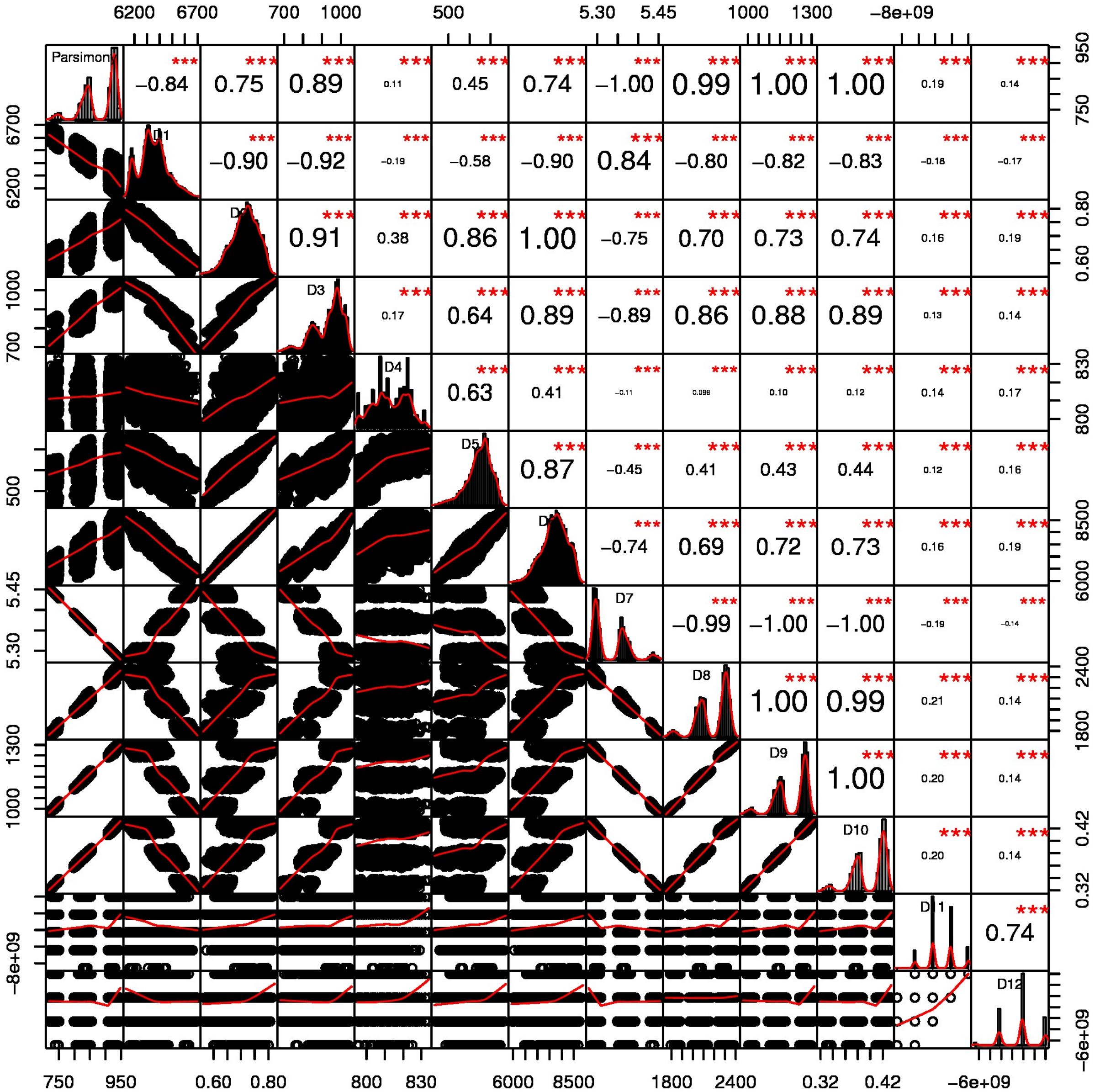


Figure A.11: Correlation between objective function evaluations for instance $rbcL_3$

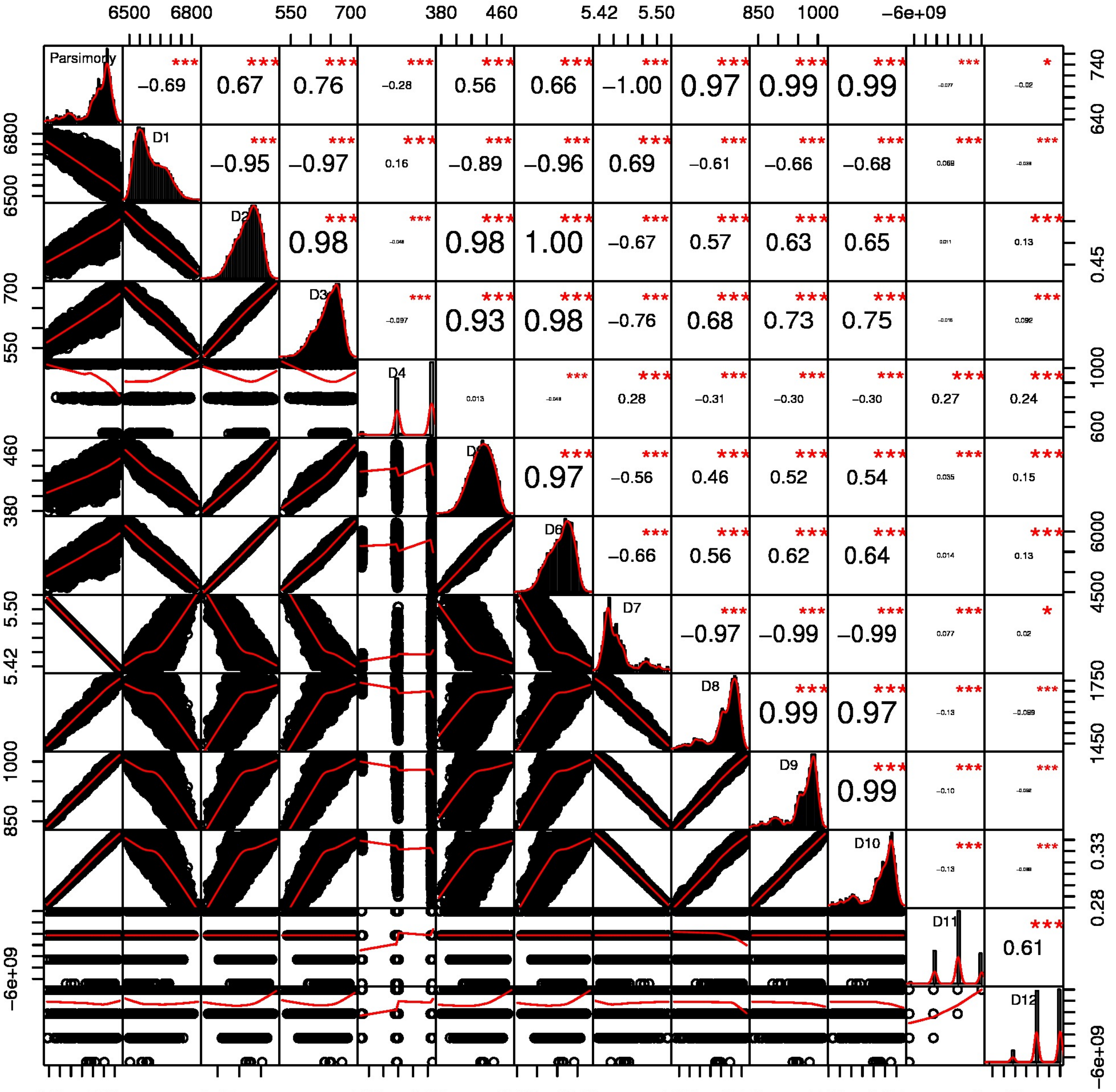


Figure A.12: Correlation between objective function evaluations for instance $rbcL_4$

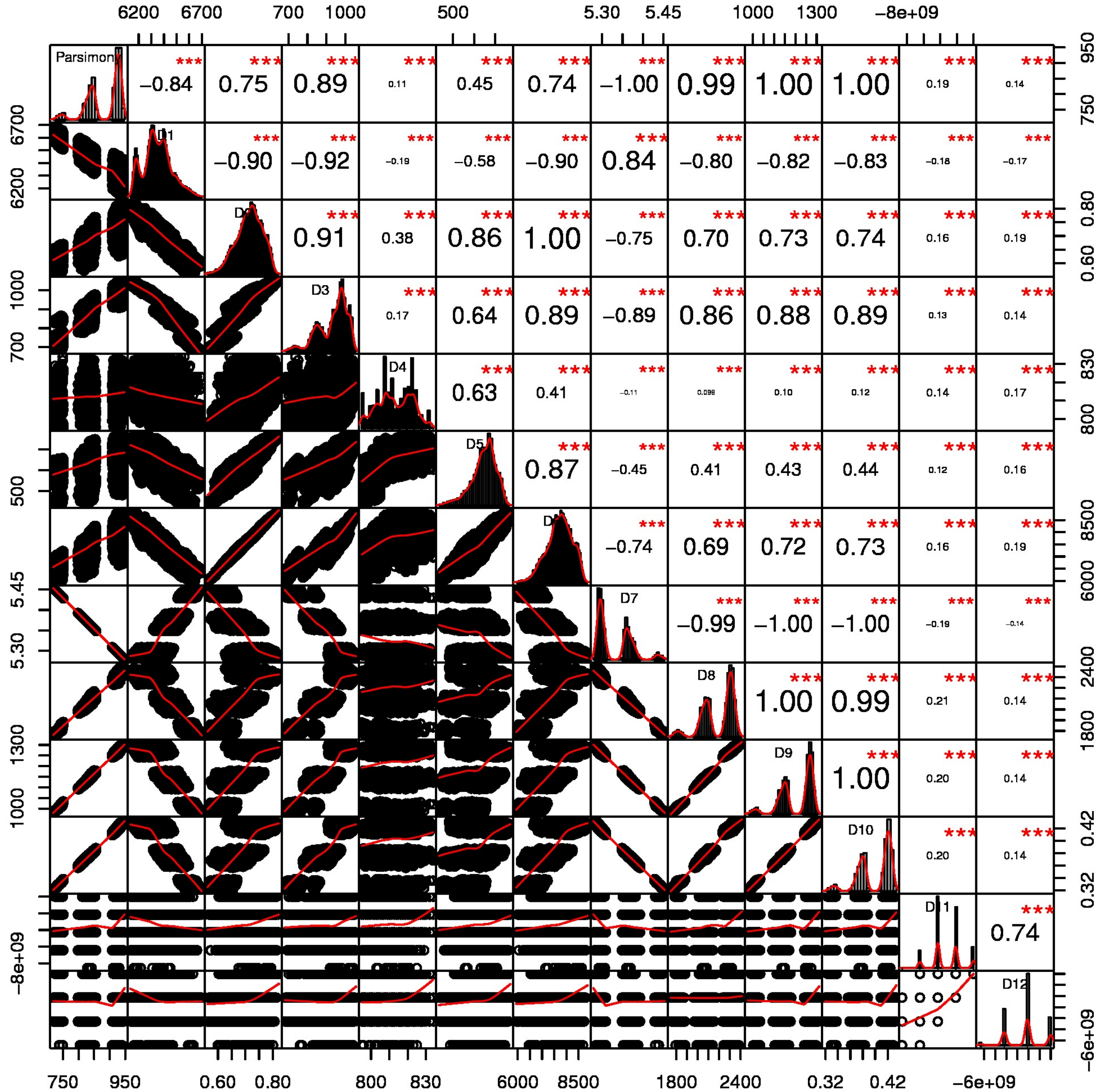


Figure A.13: Correlation between objective function evaluations for instance $rbcL_5$

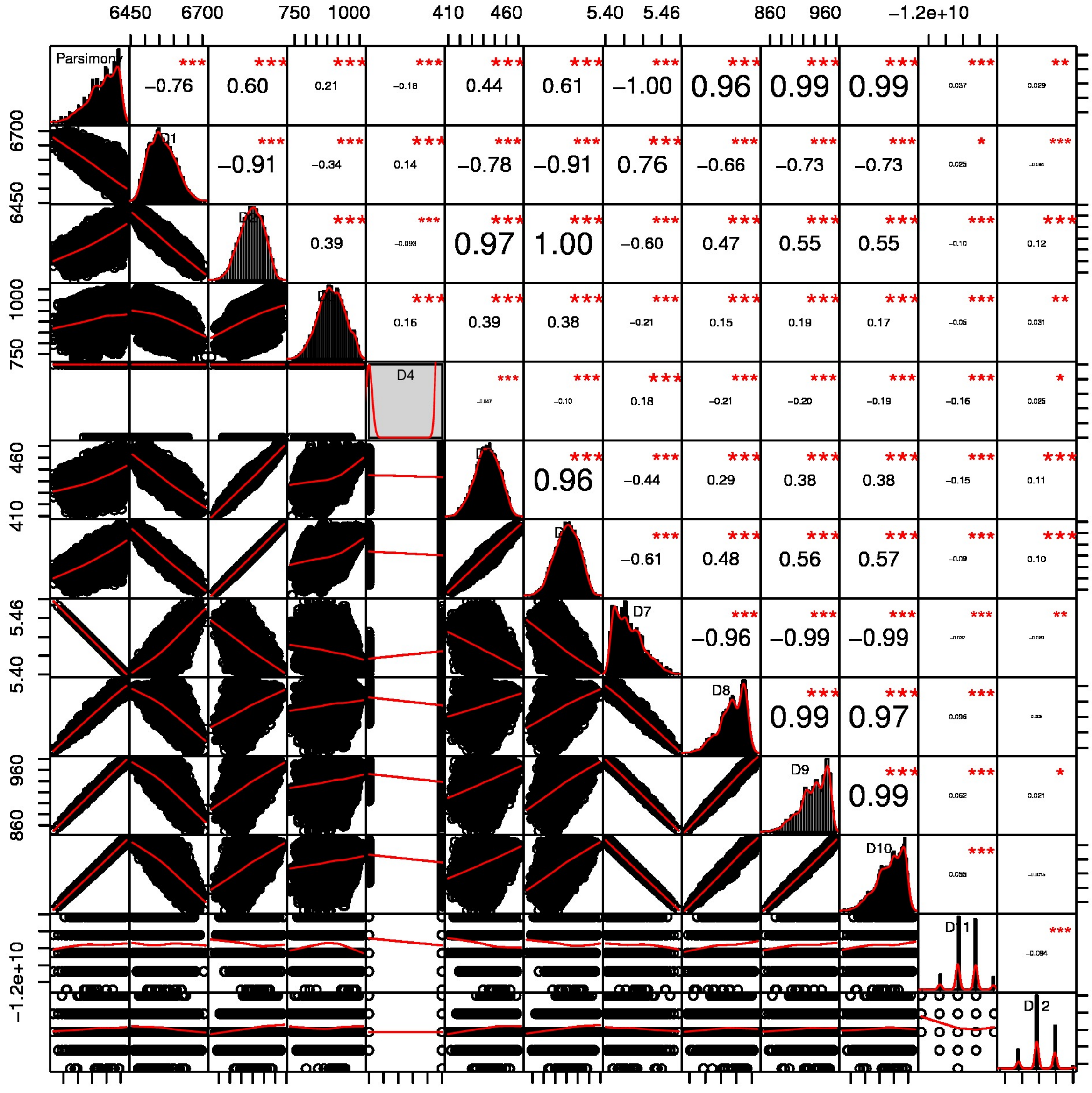


Figure A.14: Correlation between objective function evaluations for instance $rbcL_6$

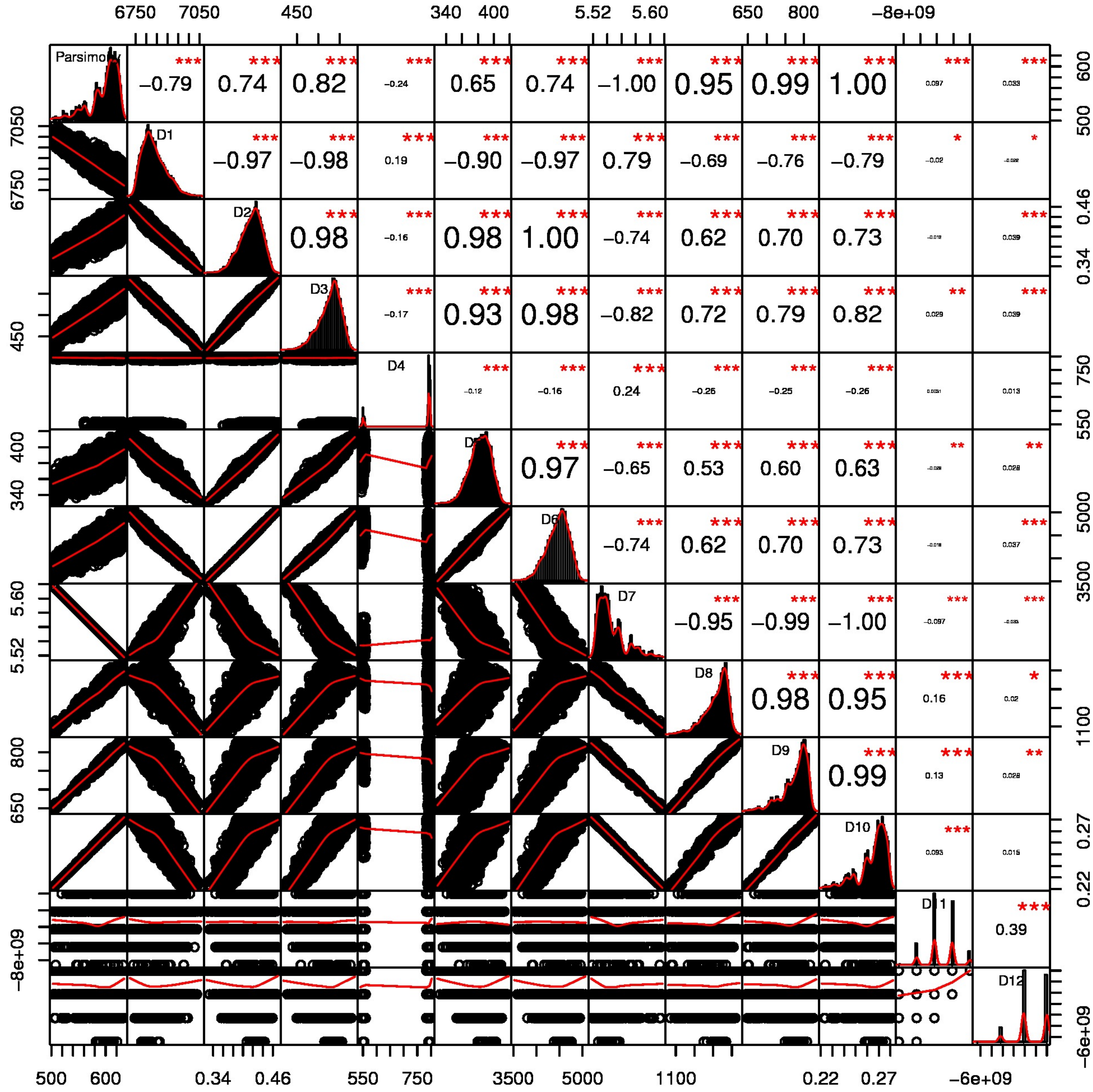


Figure A.15: Correlation between objective function evaluations for instance $rbcL_7$

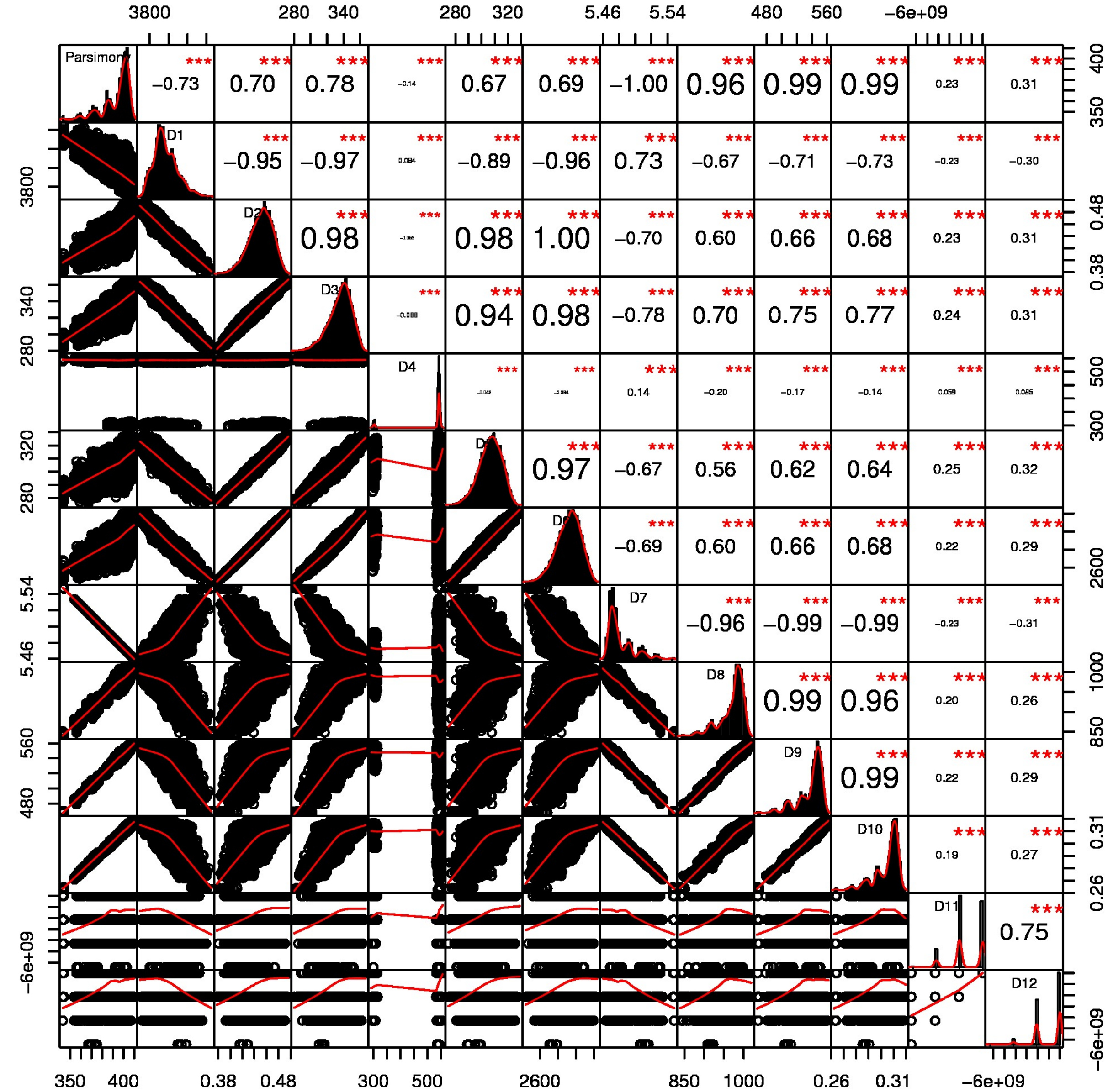


Figure A.16: Correlation between objective function evaluations for instance $ZILLA_1$

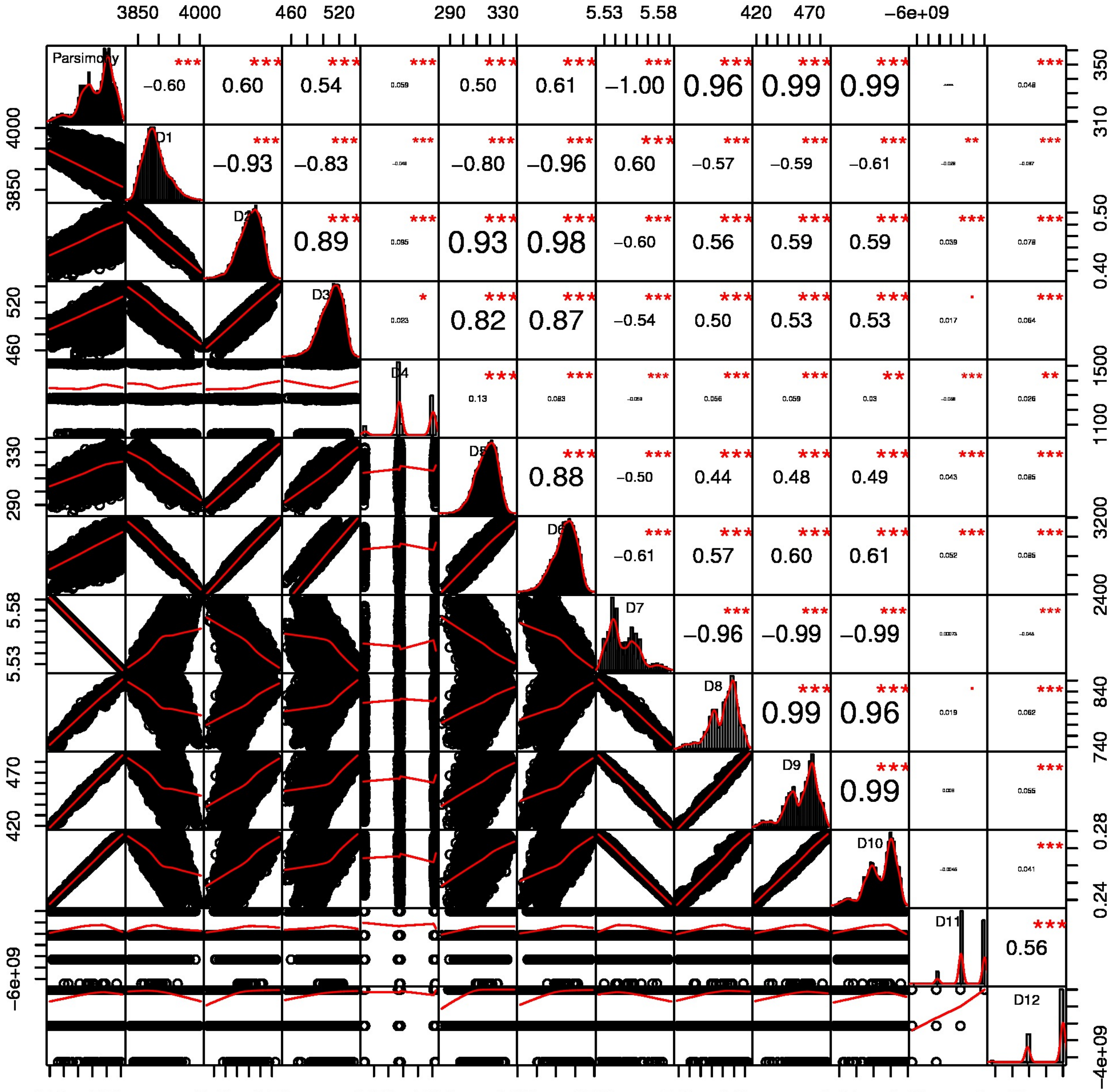


Figure A.17: Correlation between objective function evaluations for instance $ZILLA_2$

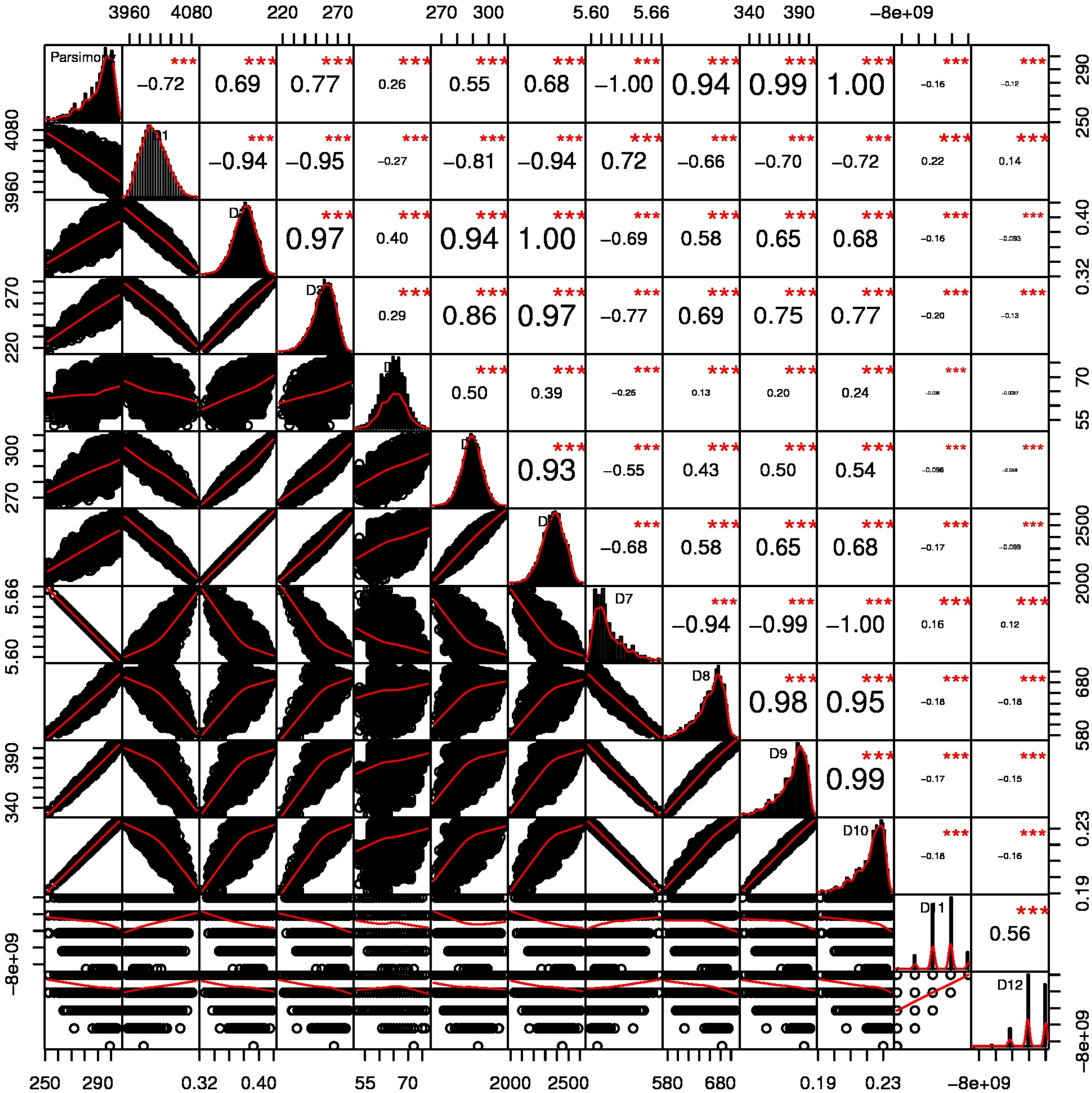


Figure A.18: Correlation between objective function evaluations for instance $ZILLA_3$

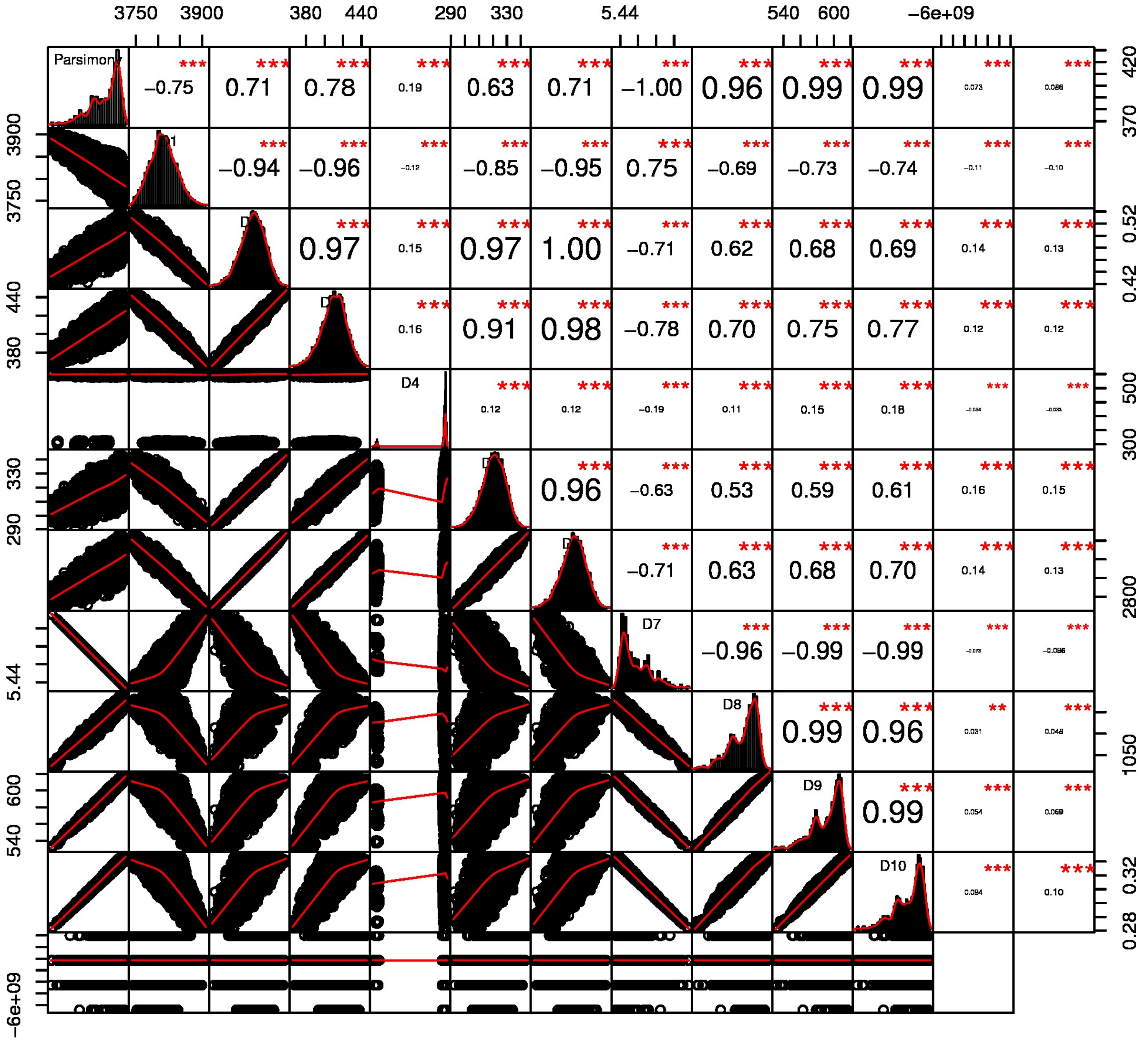


Figure A.19: Correlation between objective function evaluations for instance $ZILLA_4$

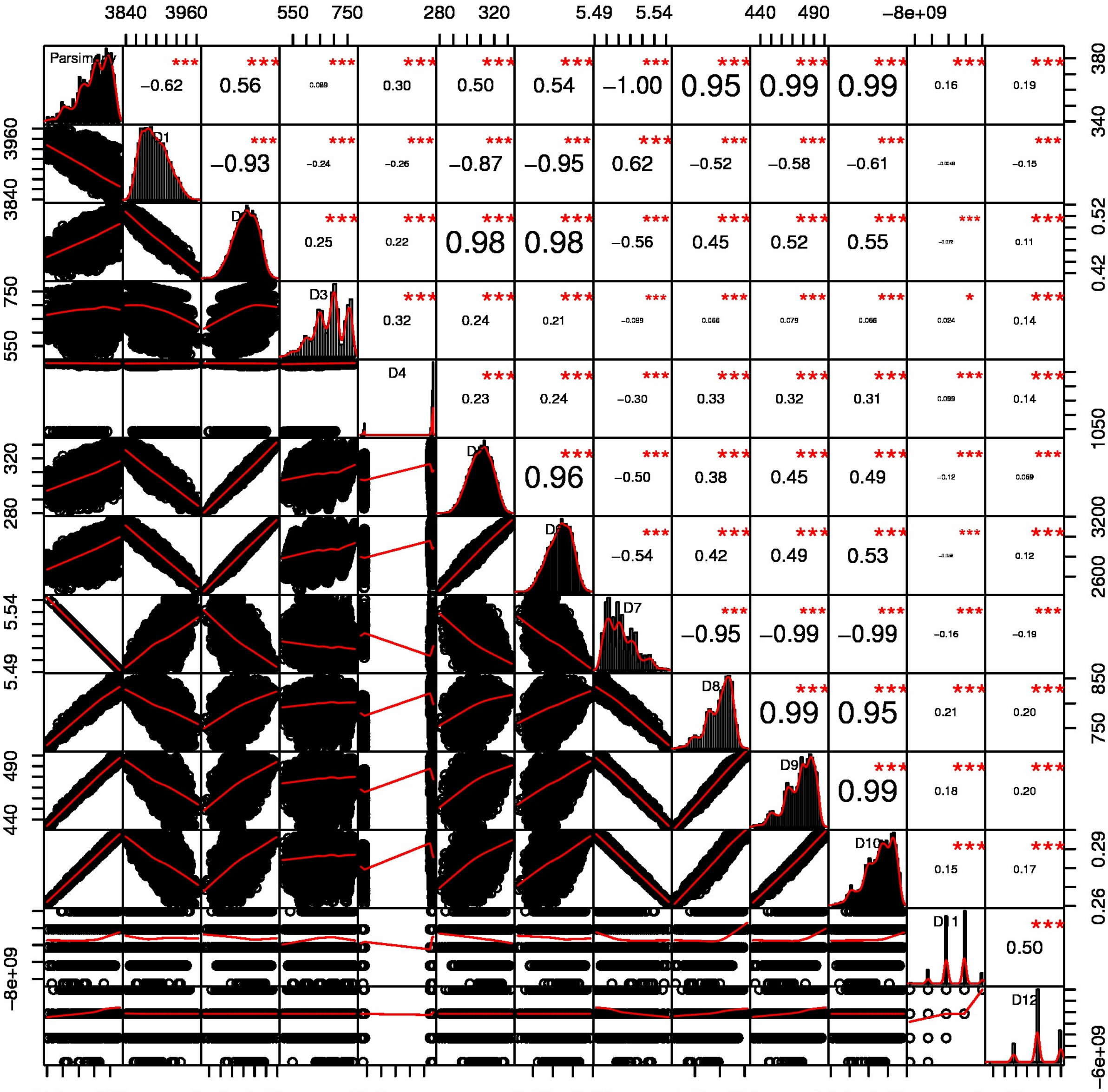


Figure A.20: Correlation between objective function evaluations for instance $ZILLA_5$

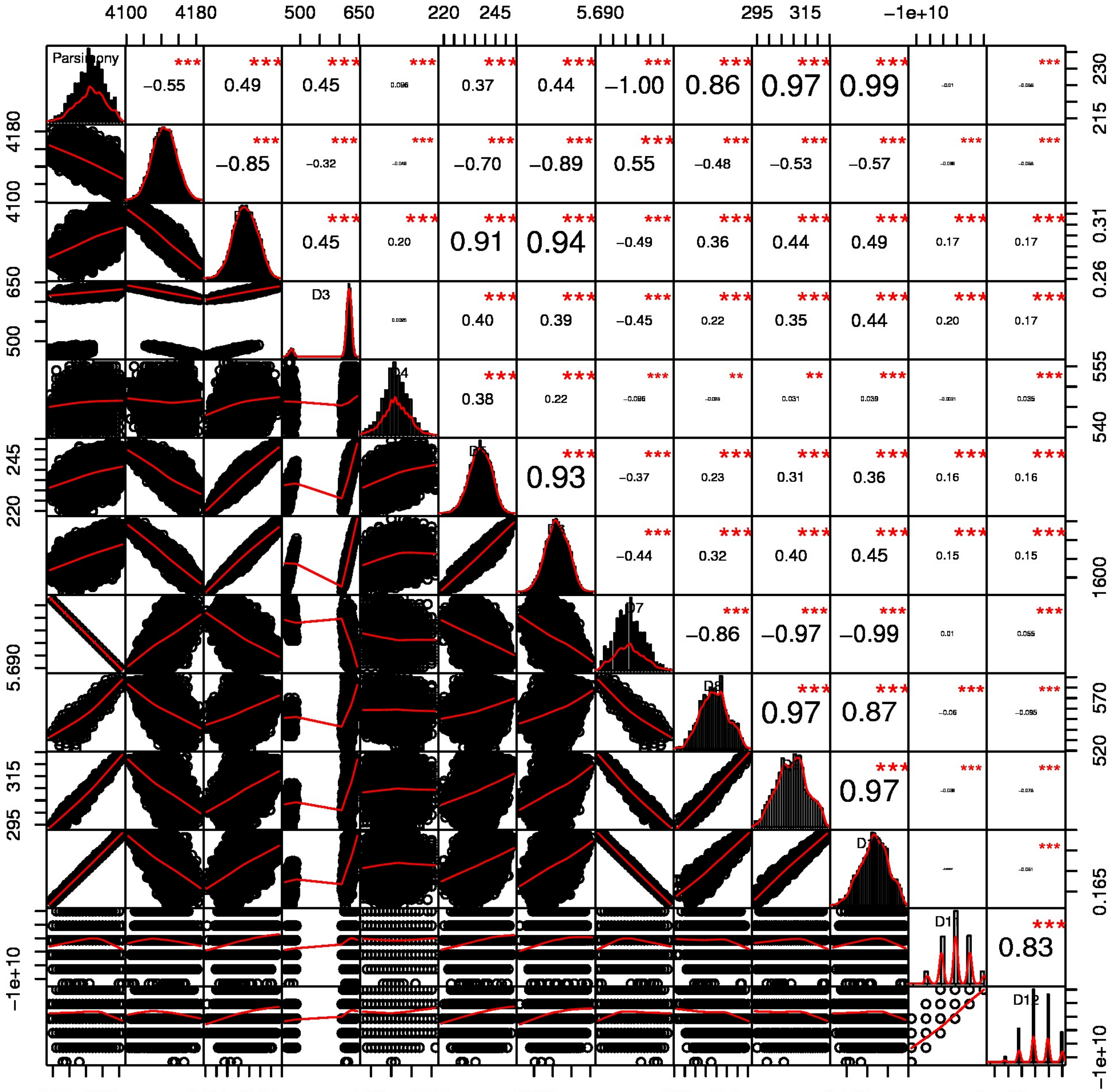


Figure A.21: Correlation between objective function evaluations for instance $ZILLA_6$

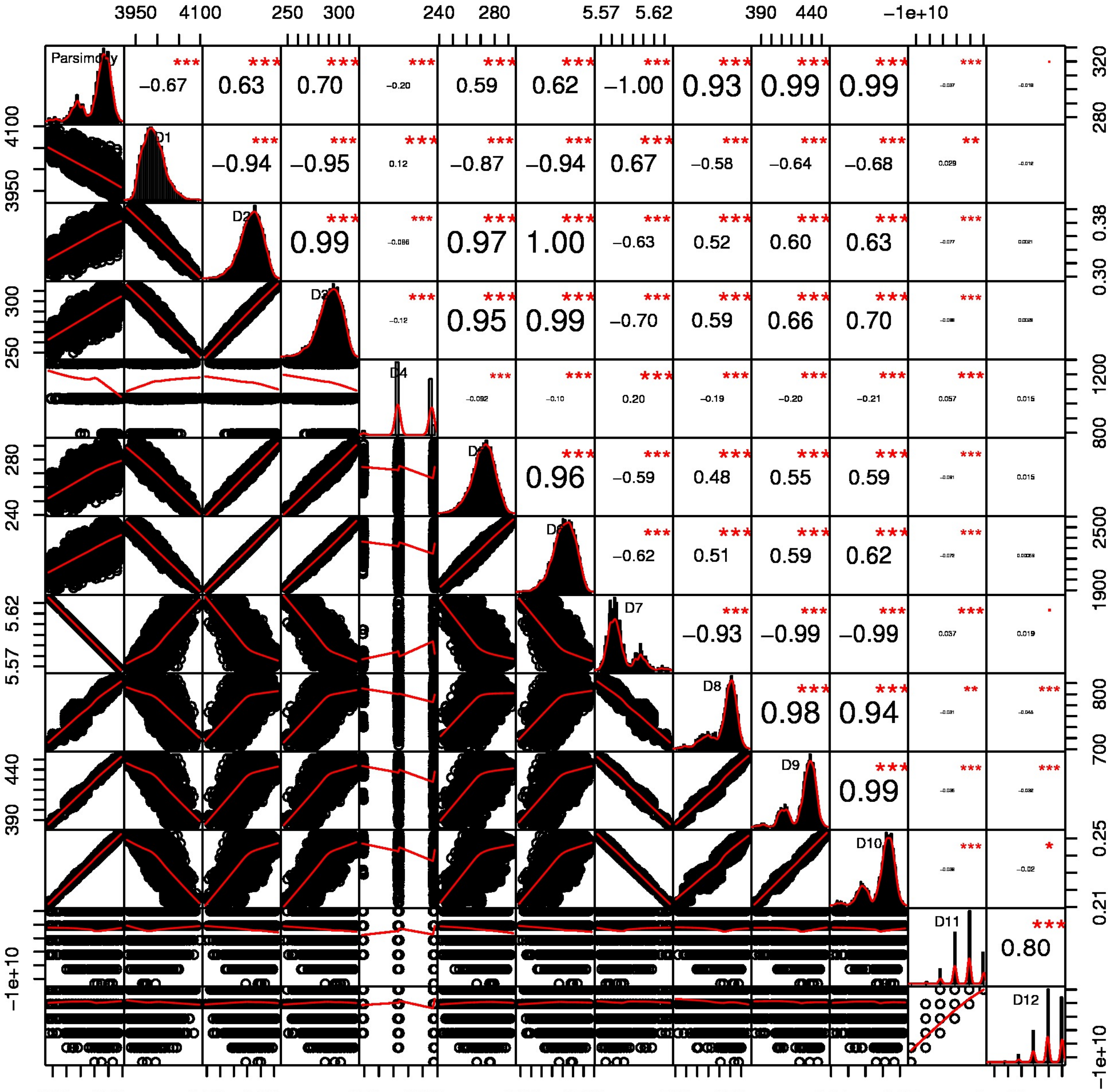


Figure A.22: Correlation between objective function evaluations for instance $ZILLA_7$

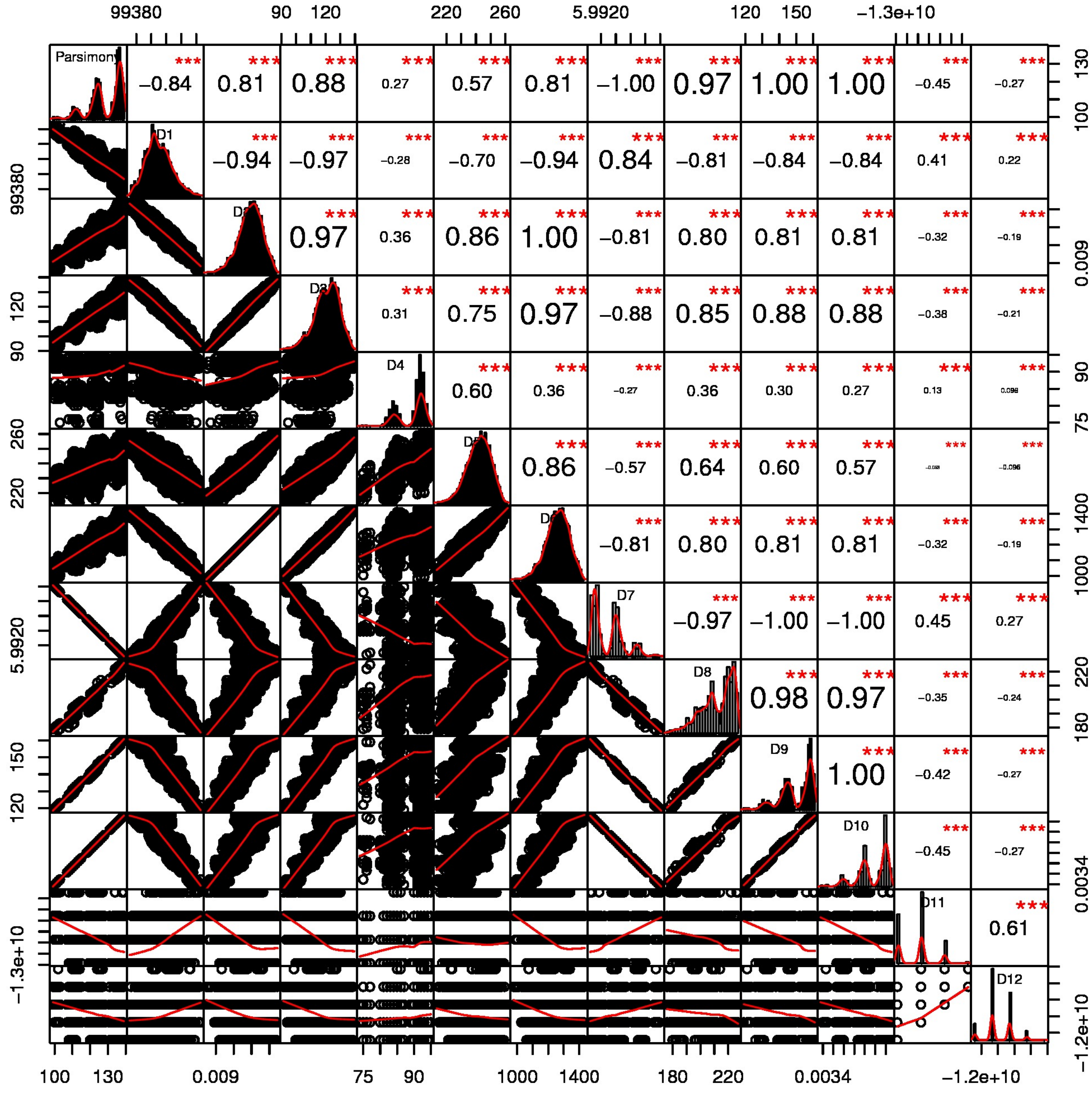


Figure A.23: Correlation between objective function evaluations for instance $mtDNA_1$

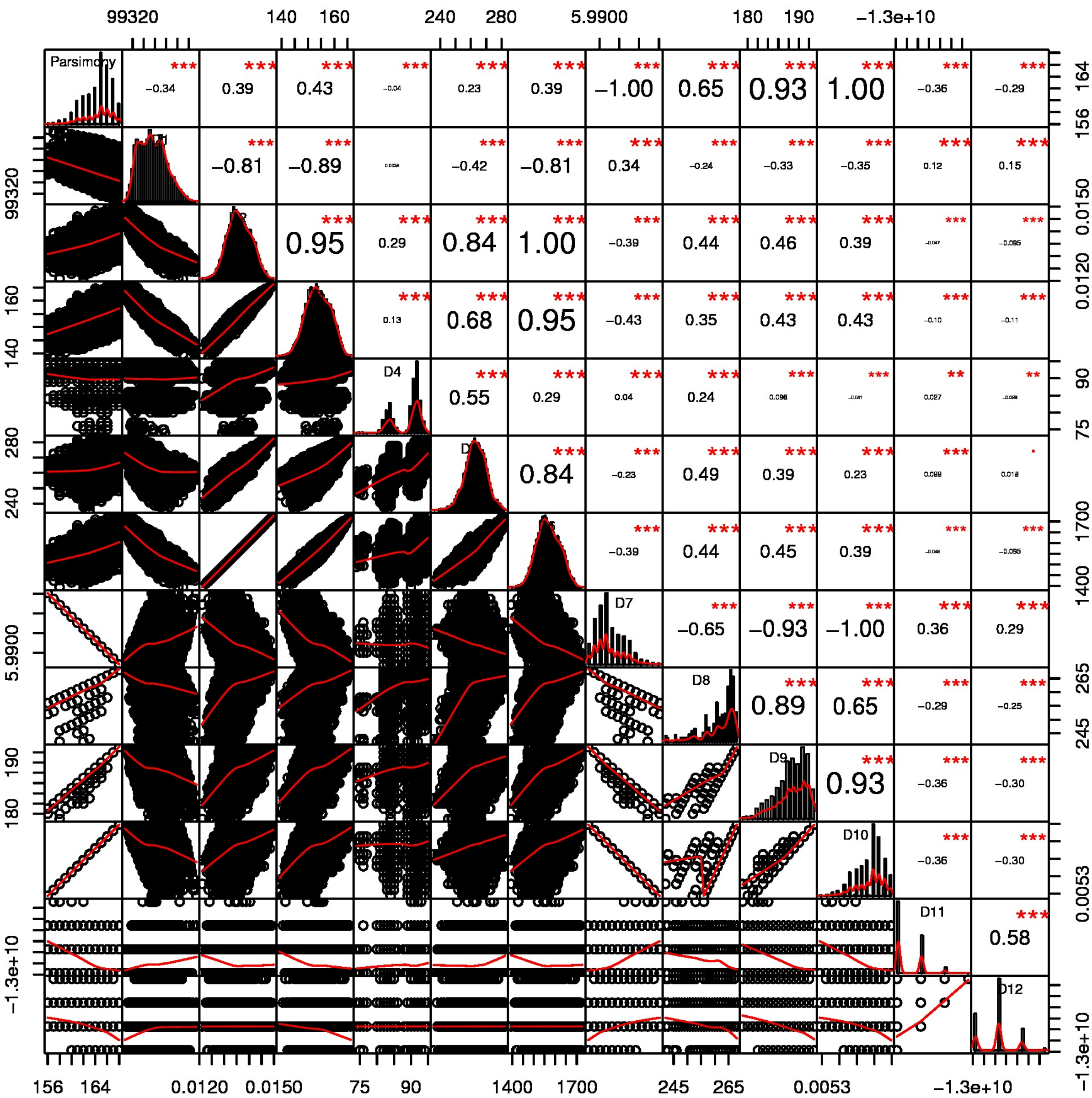


Figure A.24: Correlation between objective function evaluations for instance $mtDNA_2$

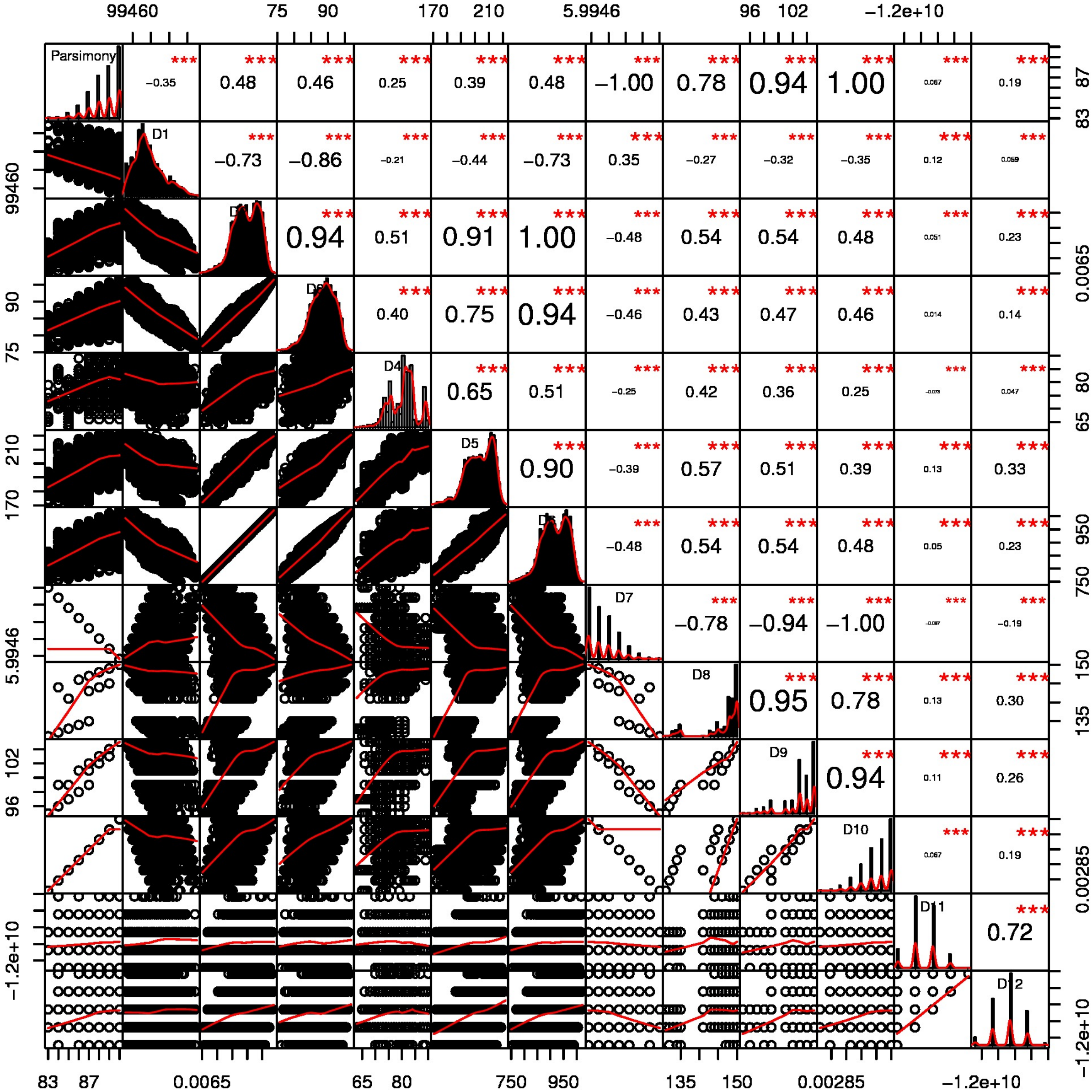


Figure A.25: Correlation between objective function evaluations for instance $mtDNA_3$

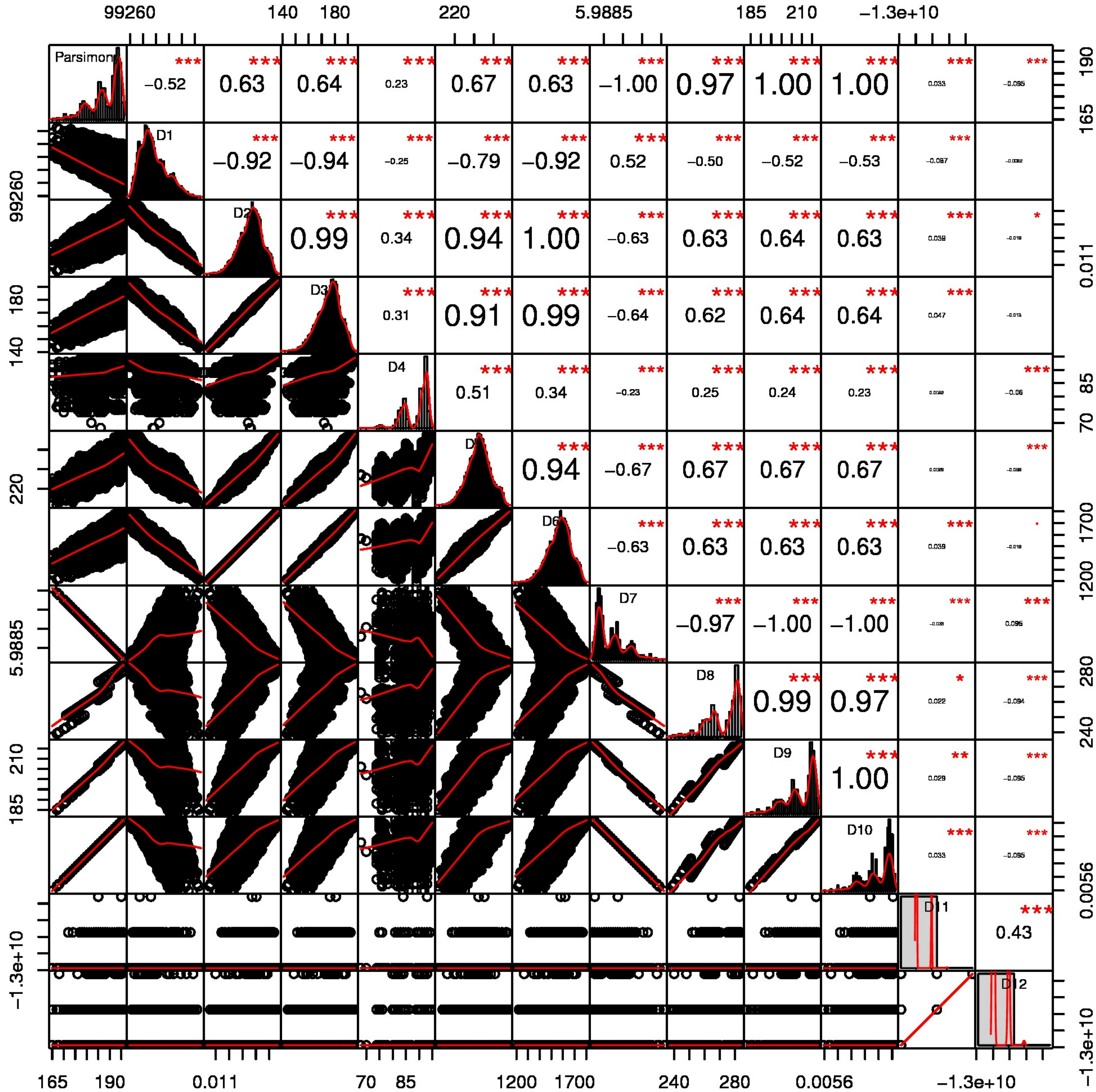


Figure A.26: Correlation between objective function evaluations for instance $mtDNA_4$

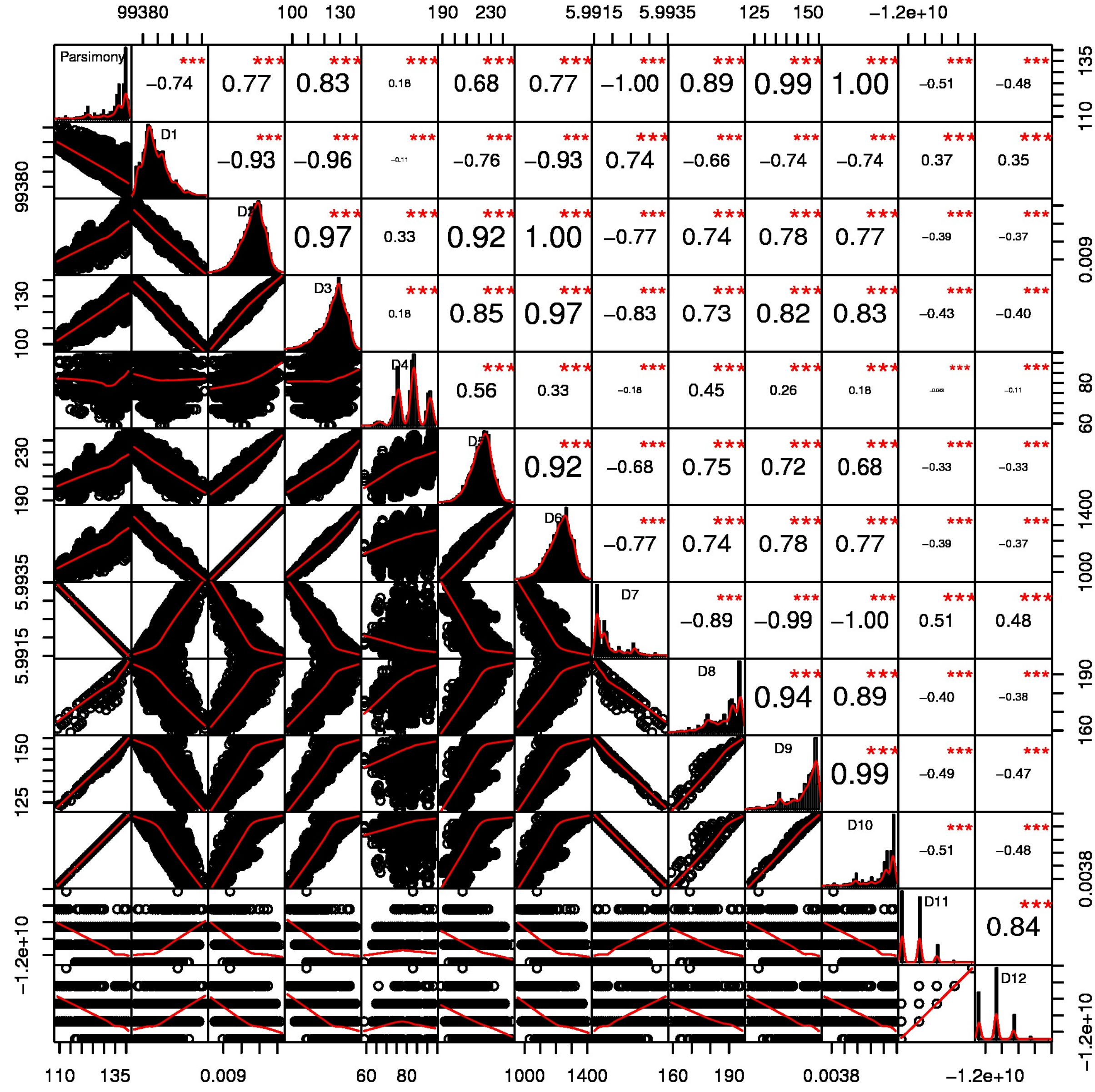


Figure A.27: Correlation between objective function evaluations for instance $mtDNA_5$

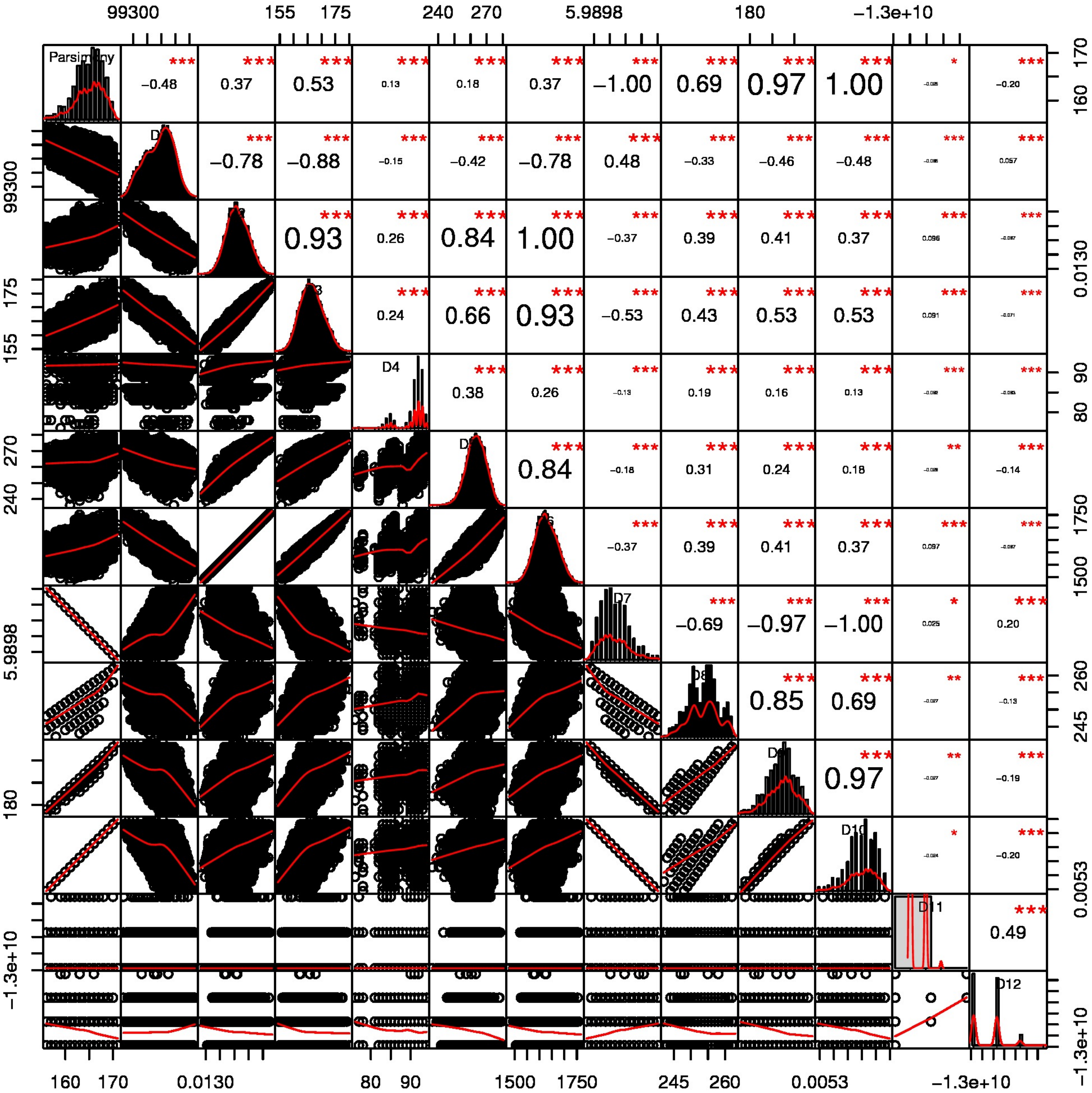


Figure A.28: Correlation between objective function evaluations for instance $mtDNA_6$

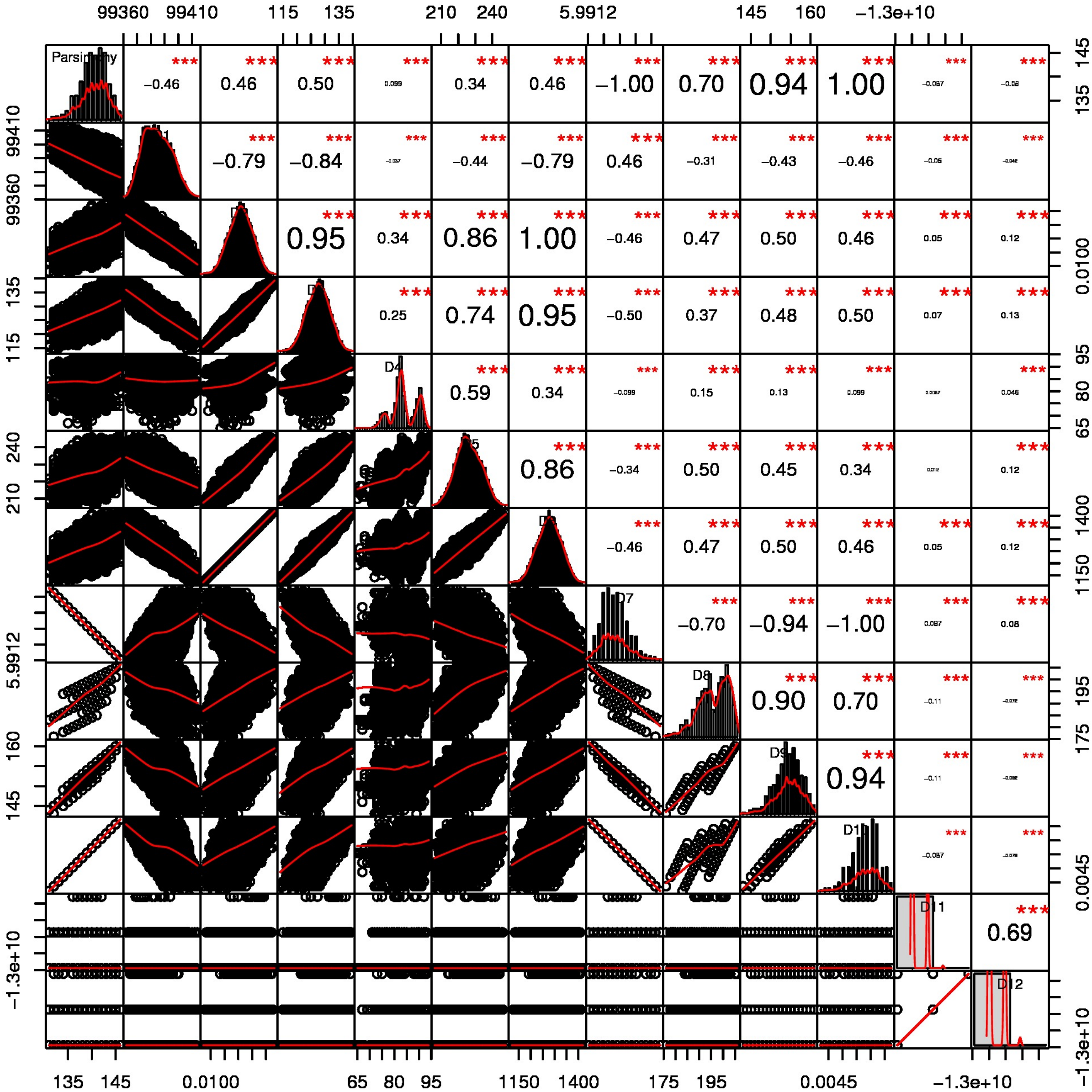


Figure A.29: Correlation between objective function evaluations for instance $mtDNA_7$

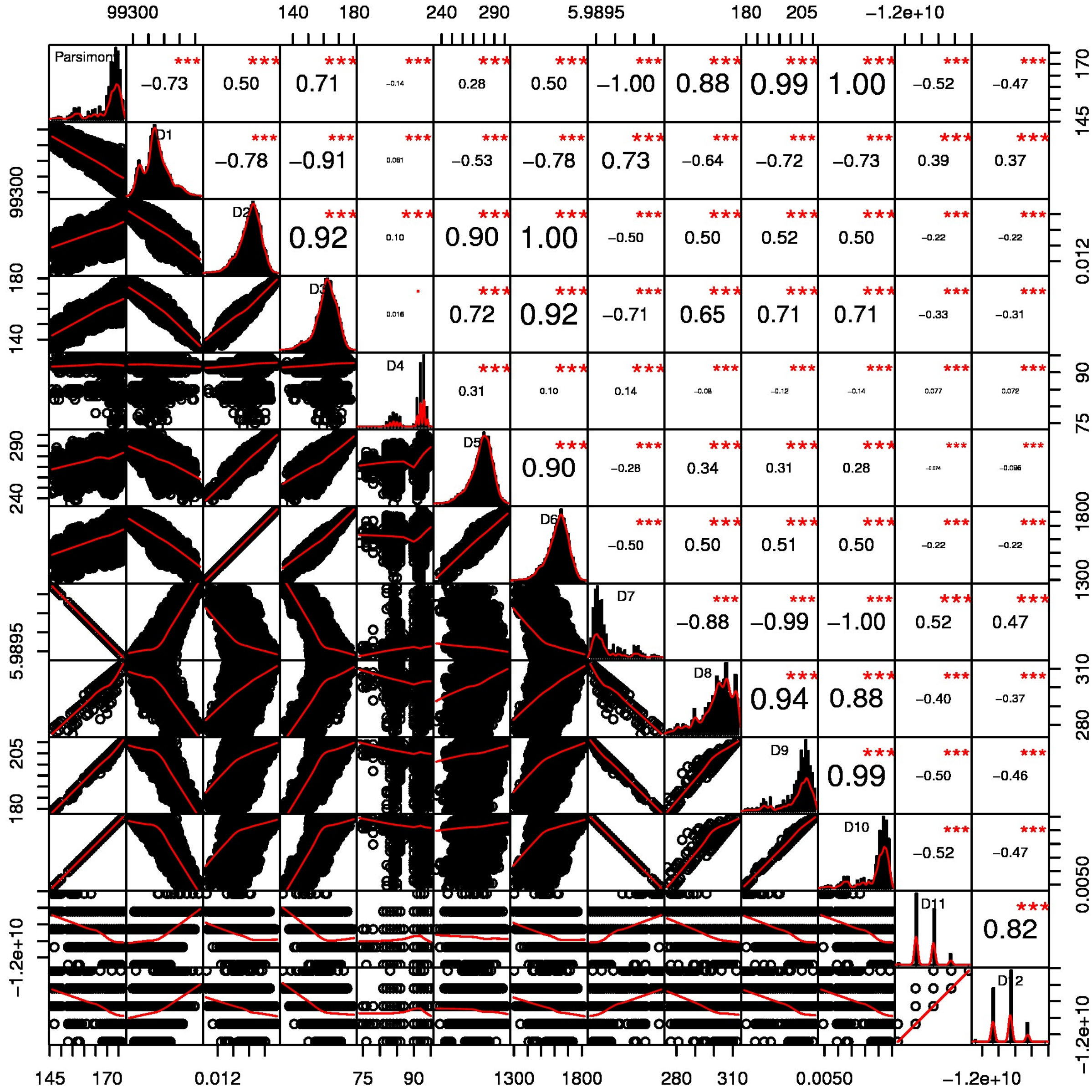


Figure A.30: Correlation between objective function evaluations for instance $mtDNA_8$

Bibliography

Allen, B. L. and Steel, M. (2001). Subtree Transfer Operations and Their Induced Metrics on Evolutionary Trees. *Annals of Combinatorics*, 5:1–15.

Andreatta, A. A. and Ribeiro, C. C. (2002). Heuristics for the Phylogeny Problem. *Journal of Heuristics*.

Bader, D. A., Chandu, V. P., and Yan, M. (2006). ExactMP: An Efficient Parallel Exact Solver for Phylogenetic Tree Reconstruction Using Maximum Parsimony. In *International Conference on Parallel Processing*.

Barry, D. and Hartigan, J. A. (1987). Statistical analysis of hominoid molecular evolution. *Statistical Science*, 2:209–210.

Bienaymé, I. J. (1867). Considérations à l'appui de la découverte de Laplace sur la loi de probabilité dan la méthode des moindres carrés. *Journal de Mathématiques Pures et Appliquées*, pages 158–176.

Bray, J. R. and Curtis, J. T. (1957). An ordination of the upland forest comunities of southern Wisconsin. *Ecological monographs*, 27:325–349.

Brown, J. H., Lomolino, M. V., Riddle, B. R., and Renner, S. S. (2006). *Biogeography*, volume 55. Sinnauer Associates Sunderland.

Cancino, W. and Delbem, A. C. B. (2007). A Multi-objective Evolutionary Approach for Phylogenetic Inference. In *Evolutionary Multi-Criterion Optimizarion: 4th International Conference*, pages 428–442.

Coelho, G. P., da Silva, A. E. A., and Von Zuben, F. J. (2010). An Immune-inspired Multi-objective Approach to the Reconstruction of Phylogenetic Trees. *Neural Computing and Applications*, 19:1103–1132.

Congdon, C. B. (2001). Gaphyl: A Genetic Algorithms Approach to Cladistics. In *Principles of Data Mining and Knowledge Discovery*, pages 67–78.

Cotta, C. (2006). Scatter Search With Path Relinking for Phylogenetic Inference. *European Journal of Operational Research*, 169:520–532.

Cotta, C. and Moscato, P. (2002). Inferring Phylogenetic Trees Using Evolutionary Algorithms. In *Parallel Problem Solving from Nature — PPSN VII*, pages 720–729. Springer Berlin Heidelberg.

Dagum, L. and Menon, R. (1998). OpenMP: an industry standard API for shared-memory programming. *IEEE Computational Science and Engineering Magazine*, 5(1):46–55.

Dayrat, B. and Linder, P. (2003). The Roots of Phylogeny: How Did Haeckel Build His Trees? *Systematic Biology*, 52:515–527.

Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE transactions on evolutionary computation*, 6:182–197.

Deza, M. M. and Deza, n.-E. (2009). *Encyclopedia of Distances*. Springer Berlin Heidelberg.

Evans, J. D. (1996). *Straightforward Statistics for the Behavioral Sciences*. Brooks/Cole Publishing Company.

Fitch, W. M. (1971). Toward Defining the Course of Evolution: Minimum Change for a Specific Tree Topologt. *Systematic Zoology*, 20(4):406–416.

Fitch, W. M. and Margoliash, E. (1967). Construction of Phylogenetic Trees. *Science*, 155:279–284.

Garza Fabre, M., Toscano Pulido, G., and Rodriguez Tello, E. A. (2015). Multi-objectivization, fitness landscape transformation and search performance: A case of study on the hp model for protein structure prediction. *European Journal of Operational Research*, 243(2):405–422.

Goëffon, A., Richer, J. M., and Hao, J. K. (2006). A Distance-Based Information Preservation Tree Crossover for the Maximum Parsimony Problem. In *Parallel Problem Solving from Nature - PPSN IX*, pages 761–770. Springer Berlin Heidelberg.

Goëffon, A., Richer, J. M., and Hao, J. K. (2008). Progressive Tree Neighborhood applied to the Maximum Parsimony Problem. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 5:136–145.

Goloboff, P. A., Farris, J. S., and Nixon, K. C. (2008). TNT, a free program for phylogenetic analysis. *Cladistics*, 24:774–786.

Gusfield, D. (1997). *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*. Cambridge University Press.

Haeckel, E. (1866). *Generelle Morphologie der Organismen*.

Handl, J., Kell, D. B., and Knowles, J. D. (2007). Multiobjective Optimization in Bioinformatics and Computational Biology. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(2):279–292.

Handl, J., Lovell, S. C., and Knowles, J. D. (2008). Multi-objectivization by Decomposition of Scalar Cost Functions. In *Parallel Problem Solving from Nature: 10th International Conference*, pages 31–40.

Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547–579.

Jensen, M. T. (2003). Guiding Single-Objective Optimization Using Multi-objective Methods. In *Applications of Evolutionary Computing: EvoWorkshops 2003*, pages 268–279.

Jensen, M. T. (2005). Helper-objectives: Using multi-objective evolutionary algorithms for single-objective optimization. *Journal of Mathematical Modelling and Algorithms*, 3:323–347.

Jukes, T. H. and Cantor, C. R. (1969). Evolution of Protein Molecules. *Mammalian Protein Metabolism*, pages 21–132.

Kaufman, L. and Rousseeuw, P. (1990). *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley & Sons, Inc.

Kimura, M. (1980). A simple method for estimating evolutionary rates of base substitutions through comparative studies of nucleotide sequences. *Journal of Molecular Evolution*, 16(2):1432–1432.

Kirkup, B. and Kim, J. (2000). From rolling hills to jagged mountains: Scaling of heuristic searches for phylogenetic estimation. Mol. Biol. Evol. (In Revision).

Kluge, A. G. and Farris, J. S. (1969). Quantitative Phyletics and the Evolution of Anurans. *Systematic Biology*, 18:1–32.

Knowles, J. D., Watson, R. A., and Corne, D. (2001). Reducing Local Optima in Single-Objective Problems by Multi-objectivization. In *Proceedings of the First International Conference on Evolutionary Multi-Criterion Optimization*, pages 269–283.

Krause, E. F. (1987). *Taxicab geometry: an adventure in non-Euclidean geometry*. Dover Publ.

Lai, X. and Hao, J. K. (2016). Iterated maxima search for the maximally diverse grouping problem. *European Journal of Operational Research*, 254(3):780–800.

Lance, G. N. and Williams, W. T. (1966). Computer Programs for Hierarchical Polythetic Classification ("Similarity Analyses"). *The Computer Journal*, 9(1):60–64.

Lewis, P. (1998). A genetic algorithm for maximum-likelihood phylogeny inference using nucleotide sequence data. *Molecular Biology and Evolution*, 15:277–283.

Liébecq, C. (1992). *Biochemical nomenclature and related documents: a compendium*. Portland Press.

López Ibañez, M., Dubois Lacoste, J., Pérez Cáceres, L., Stützle, T., and Birattari, M. (2016). The irace package: Iterated Racing for Automatic Algorithm Configuration. *Operations Research Perspectives*, 3:43–58.

Maddison, D. R. (1991). The Discovery and Importance of Multiple Islands of Most-Parsimonious Trees. *Systematic Zoology*, 40(3):315–328.

Matsuda, H. (1995). Construction of phylogenetic trees from amino acid sequences using a genetic algorithm. *Genome Informatics*, 6:19–28.

Metanomski, W. V. (1991). *Compendium of Macromolecular Nomenclature*. Blackwell Science.

Moore, G. W., Goodman, M., and Barnabas, J. (1973). An iterative approach from the standpoint of the additive hypothesis to the dendrogram problem posed by molecular data sets. *Journal of Theoretical Biology*, 38:423–457.

Murakami, Y. and Jones, S. (2006). SHARP2: protein-protein interaction predictions using patch analysis. *Bioinformatics*, 22:1794–1795.

Pace, N. R. (1997). A Molecular View of Microbial Diversity and the Biosphere. *Science*, 276:734–740.

Porumbel, D. C., Hao, J. K., and Kuntz, P. (2010). A search space "cartography" for guiding graph coloring heuristics. *Computers & Operations Research*, 37:769–778.

Ribeiro, C. C. and Vianna, D. S. (2003). A genetic algorithm for the phylogeny problem using an optimized crossover strategy based on path-relinking. In *Anais do II Workshop Brasileiro de Bioinformática*, pages 97–102.

Ribeiro, C. C. and Vianna, D. S. (2005). A GRASP/VND heuristic for the phylogeny problem using a new neighborhood structure. *International Transactions in Operational Research*, 12(3):325–338.

Richer, J. M., Rodriguez Tello, E. A., and Vazquez Ortiz, K. E. (2013). *EVOLVE - A Bridge between Probability, Set Oriented Numerics, and Evolutionary Computation II*, chapter Maximum Parsimony Phylogenetic Inference Using Simulated Annealing, pages 189–203. Springer Berlin Heidelberg.

Sanderson, M. J. and McMahon, M. M. abd Steel, M. (2011). Terraces in Phylogenetic Tree Space. *Science*, 333:448–450.

Santander Jiménez, S. (2016). *Análisis e Inferencia Multiobjetivo de Hipótesis Filogenéticas Mediante Computación Paralela y Bioinspirada*. PhD thesis, Universidad de Extremadura.

Santander Jiménez, S. and Vega Rodríguez, M. A. (2013a). A Multi-objective Proposal Based on the Firefly Algorithm for Inferring Phylogenies. In *Evolutionary Computation, Machine Learning and Data Mining in Bioinformatics: 11th European Congerence*, pages 141–152.

Santander Jiménez, S. and Vega Rodríguez, M. A. (2013b). Applying a Multi-objective Metaheuristic Inspired by Honey Bees to Phylogenetic Inference. *Biosystems*, 114:39–55.

Schliep, K. P. (2010). Phangorn: phylogenetic analysis in R. *Bioinformatics*, 27(4):592–593.

Schliep, K. P., Potts, A. J., Morrison, D. A., and Grimm, G. W. (2017). Intertwining phylogenetic trees and networks. *Methods in Ecology and Evolution*, 8(10):1212–1220.

Sokal, R. R. and Michener, C. D. (1958). A Statistical Method for Evaluating Systematic Relationships. *University of Kansas Science Bulletin*, pages 1409–1438.

Sonco Alvarez, J. L. and Ayala Rincon, M. (2017). Variable neighborhood search for the large phylogeny problem using gene order data. In *2017 IEEE Congress on Evolutionary Computation (CEC)*. IEEE.

Strobl, M. A. R. and Barker, D. (2016). On simulated annealing phase transitions in phylogeny reconstruction. *Molecular Phylogenetics and Evolution*, 101:46–55.

Swofford, D. L. (2001). PAUP*: Phylogenetic Analysis Using Parsimony (and other methods) 4.0.b5.

Swofford, D. L. and Olsen, G. J. (1990). Phylogenetic Reconstruction. *Molecular Systematic*.

Swofford, D. L., Olsen, G. J., Waddell, P. J., and Hillis, D. M. (1996). Phylogenetic Inference. In *Molecular Systematics*. Sinauer Associates, Inc.

Tamura, K., Stecher, G., Peterson, D., Filipski, A., and Kumar, S. (2013). MEGA6: Molecular Evolutionary Genetics Analysis Version 6.0. *Molecular Biology and Evolution*.

Van de Peer, Y. and De Wachter, R. (1994). TREECON for Windows: a software package for the construction and drawing of evolutionary trees for the Microsoft Windows enviroment. *Bioinformatics*.

Vazquez Ortiz, K. E. and Rodriguez Tello, E. A. (2011). Metaheuristics for the Maximum Parsimony Problem. In *Proceedings of the Sixt IASTED International Conference on Computational Intelligence and Bioinformatics*.

Verel, S., Collard, P., and Clergue, M. (2006). Measuring the Evolvability Landscape to study Neutrality. In Keijzer, M. and et al., editors, *Genetic and Evolutionary Computation – GECCO-2006*, pages 613–614. ACM Press.

Waterman, M. S. and Smith, T. F. (1978). On the similarity of dendrograms. *Journal of Theoretical Biology*, 73:789–800.

White, T. J. and Holland, B. R. (2011). Faster Exact Maximum Parsimony Search With XMP. *Bioinformatics*, 27:1359–1367.

Xiong, J. (2006). *Essential Bioinformatics*. Cambridge University Press.

Yang, Z. (1996). Phylogenetic analysis using parsimony and likelyhood methods. *Journal of Molecular Evolution*, 42:294–307.