UNIVERSITY OF GRANADA

Departament of Computer Architecture and
Computer Technology

PhD Thesis Dissertation:

Development of Advanced Computational Systems for Multiple Sequence Alignments by using Heterogeneous Biological Information

by

Francisco Manuel Ortuño Guzmán

Advisor:

Ignacio Rojas Ruiz

Granada, June 2014

Editor: Editorial de la Universidad de Granada
Autor: Francisco Manuel Ortuño Guzmán
D.L.: GR 1978-2014
ISBN: 978-84-9083-178-6

UNIVERSITY OF GRANADA

Development of Advanced Computational Systems for Multiple Sequence Alignments by using Heterogeneous Biological Information

Desarrollo de Sistemas Computacionales Avanzados para
Alineamientos Múltiples de Secuencia mediante el uso de
Información Biológica Heterogenea

Dissertation presented by:
Francisco Manuel Ortuño Guzmán

To apply for the:

International PhD degree in Science

Signed. Francisco Manuel Ortuño Guzmán

El doctorando **Francisco Manuel Ortuño Guzmán** y el director de la tesis **Ignacio Rojas Ruiz** GARANTIZAMOS, al firmar esta tesis doctoral, que el trabajo ha sido realizado por el doctorando bajo la dirección del director de la tesis y hasta donde nuestro conocimiento alcanza, en la realización del trabajo, se han respetado los derechos de otros autores a ser citados, cuando se han utilizado sus resultados o publicaciones.

Granada, a 23 de Junio de 2014

Director de la Tesis	Doctorando
Fdo. *Ignacio Rojas Ruiz*	Fdo. *Francisco M. Ortuño Guzmán*

Para Carolina, para Carlota...

Agradecimientos

Me gustaría aprovechar estas páginas para agradecer a todas las personas que me han apoyado y han hecho realidad el desarrollo de este trabajo.

En primer lugar a Ignacio Rojas, mi director de tesis, por enseñarme y guiarme con su buena labor durante estos cuatro años. Gracias por la confianza depositada en mí, por orientarme dentro del amplio mundo de la investigación, por servir como ejemplo y solucionarme todo y cuanto siempre he necesitado. A Héctor Pomares Cintas y Olga Valenzuela Cansino porque, aunque cuestiones administrativas han impedido que aparezcáis en la portada de este trabajo como se hubiese merecido, yo siempre os he considerado directores y supervisores de este trabajo. Gracias a ambos por aportarme cada uno vuestra experiencia y saber hacer sin los cuales no hubiese conseguido alcanzar esta meta.

Extiendo estos agradecimientos a todo el departamento de Arquitectura y Tecnología de Computadores, por acogerme y facilitarme cualquier trámite. Gracias por el fenomenal ambiente de trabajo que se respira y por el apoyo constante durante mis cuatro años de formación investigadora.

A todos mis compañeros y amigos que durante estos cuatro años han pasado por el CITIC-UGR y han compartido conmigo muchos y buenos momentos: Fergu, Paloma, A. Mora, Antares, Javi P. Florido, Javi, Oresti, Nuria, Gonzalo, Nolo, Fran, Leo, Quique, Juanlu, Cristina, Victor, Antonio, Urquiza, Ana Belén (disculpad, no sé si me dejo alguien). Gracias por los desestresantes desayunos, las palabras de ánimo, los spoilers de series y algún que otro Simulink III Arena.

A mis padres, gracias por enseñarme a aprender, a observar, a pensar por mí mismo y a buscar soluciones a cada reto que me he ido encontrando, porque todo ello es también una buena base para la carrera investigadora. Gracias también por vuestro constante apoyo, preocupación y buenos consejos. A mi hermana, porque siendo más joven que yo me ha enseñado que el esfuerzo y la dedicación a lo que haces siempre tiene sus frutos. Gracias además por hacerme sentir importante y haberte picado también el gusanillo de la investigación.

Muy especialmente, a Caro. Gracias por sufrir conmigo, levantarme cuando me caigo y animarme cuando desisto. Gracias de corazón por todo tu amor. Nadie mejor que tú sabe lo que esta tesis significa para mí y nadie más que tú ha hecho este sueño posible. Además de todo, eres un ejemplo de investigadora, un espejo en el que cada día me miro. Gracias también por darme lo más bonito que hay en el mundo, a Carlota. Aunque el momento de su llegada no ha sido fácil, gracias a las dos porque me habéis aportado todos los ánimos que necesitaba para este último empujón.

Finally, I would like to thank all the members of the Computational Biology and Data Mining Group of the Max Delbruck Center for Molecular Medicine, specially Miguel, Jean-Fred and Enrique. I am so grateful for your welcome in this lab and for giving me the opportunity to learn and develop my earliest researches. I found this stay a very enriching experience. Also, thank you very much for be available to help me every time I have needed it.

Siento si me he dejado a alguien sin nombrar, seguro que las prisas me están jugando una mala pasada. Gracias de todo corazón a todos!!!!

Contents

Agradecimientos ix

List of Figures xvii

List of Tables xxvii

Abbreviations xxxi

Abstract 1

Resumen 5

1 From Biology to Bioinformatics 9

1.1 Introduction 11

1.2 Biological background 12

1.2.1 The cell 12

1.2.2 The Deoxyribonucleic Acid (DNA) 13

1.2.3 The Central Dogma: *from genes to proteins* 16

1.2.4 Gene mutations 18

1.3 Advances in massive DNA sequencing 22

1.3.1 Human Genome Project (HGP) 22

1.3.2 Next generation sequencing (NGS) 23

1.4 Bioinformatics 25

1.5 Bioinformatics Resources and Databases 27

1.5.1 GenBank 28

1.5.2 Gene Expression Omnibus (GEO) 32

1.5.3 Uniprot 36

1.5.4 Pfam 39
1.5.5 IntAct 43
1.5.6 Protein Data Bank (PDB) 46
1.5.7 DSSP 48
1.5.8 Gene Ontology (GO) 50
1.6 Conclusions 53

2 Introduction to Multiple Sequence Alignments 55
2.1 Background 57
2.2 Sequence alignment methodologies 61
2.2.1 Pairwise alignments 62
2.2.1.1 Classical methods with dynamic programming . 62
2.2.1.2 Heuristic methods 63
2.2.2 Multiple sequence alignments 65
2.2.2.1 Progressive algorithms 65
2.2.2.2 Consistency-based methodologies 70
2.2.2.3 Algorithms with additional features 75
2.3 Multiple sequence alignment benchmarks 79
2.3.1 Oxbench 79
2.3.2 Prefab 80
2.3.3 BAliBASE 80
2.4 Comparison of MSA methodologies 81
2.5 Principal conclusions of the MSA comparison 87
2.6 Objectives of the Dissertation 88
2.7 Structure of the Dissertation 89

3 PAcAlCI: A priori Prediction of Accuracy Alignment based on Computational Intelligence 91
3.1 Motivation and goals 93
3.2 Creation of a dataset of multiple sequence alignments 96
3.2.1 BAliBASE benchmark 96
3.2.2 Multiple sequence alignment tools 98
3.3 Databases and feature extraction 99
3.3.1 Features from BAliBASE sequences 101

3.3.2 Features from Uniprot 102
3.3.3 Features from Pfam 104
3.3.4 Features from Gene Ontology 105
3.3.5 Features from the Protein Data Bank 106
3.3.6 Additional features 108
3.4 Selection of features based on mutual information 110
3.4.1 Feature selection procedures 110
3.4.2 Mutual Information definition 111
3.4.3 Minimal-Redundancy-Maximal-Relevance (mRMR) algorithm 112
3.5 Prediction algorithm based on Least-Square Support Vector Machine 114
3.5.1 Least-Square Support Vector Machines (LS-SVMs) 114
3.5.2 LS-SVM assessment 117
3.5.3 Selection of the most suitable methodologies 118
3.6 Experiments and Results 118
3.6.1 Determination of significance for biological features 119
3.6.2 Estimation of accuracies for MSA tools 121
3.6.3 Filtering alignments with low accuracies 122
3.6.4 Selection of the most suitable MSA tools 125
3.6.5 Qualitative comparison against AlexSys 128
3.7 Conclusions and Recommendations 130

4 Regression models for an intelligent estimation of the MSA quality 131
4.1 Introduction and motivation 133
4.2 Construction of the alignment dataset 137
4.3 Dataset with the extraction of alignment features 138
4.3.1 Features directly extracted from alignments 139
4.3.2 Alignment features extracted from Uniprot 142
4.3.3 Alignment features extracted from Pfam 144
4.3.4 Alignment features extracted from PDB 145
4.3.5 Alignment features extracted from chemical properties . . 146
4.3.6 Alignment features from additional scores 148

4.4 Feature selection with normalized mutual information 149
4.5 Regression models 151
    4.5.1 Regression Trees 152
    4.5.2 Bagging Regression Trees 154
    4.5.3 Gaussian Processes 155
4.6 Implementation and assessment of the scoring schemes 157
4.7 Assessment by comparisons of pairwise alignments 160
4.8 Experimental Results 162
    4.8.1 Tuning of parameters in the regression models 162
    4.8.2 Determination of the optimal subsets of features 164
    4.8.3 Correlation between different score schemes and BAliscore 167
    4.8.4 Assessment of scores by comparing pairwise alignment . 168
4.9 Final conclusions 171

5 MO-SAStrE: Multiobjective Optimizer for Sequence Alignment based on Structural Evaluation 173
5.1 Background, motivations and goals 175
5.2 Bases of evolutionary and genetic algorithms 177
    5.2.1 Multiobjective evolutionary optimization 180
    5.2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II) . 182
    5.2.3 Performance assessment in multiobjective optimization . . 183
5.3 MO-SAStrE Implementation 188
    5.3.1 MSA Representation 190
    5.3.2 Initialization of the population 192
    5.3.3 Genetic algorithm operators 193
    5.3.4 Multiobjective fitness function 195
    5.3.5 Termination condition 199
5.4 MO-SAStrE Assessment 199
    5.4.1 Dataset of sequences 199
    5.4.2 Multiobjective and statistical assessment 200
    5.4.3 Other similar optimizers 201
5.5 Experimental results and discussion 204
    5.5.1 Selection of parameters in the genetic algorithm 204

- 5.5.2 Optimization results 206
- 5.5.3 Hypervolume analysis 209
- 5.5.4 Statistical assessment of MO-SAStrE 210
- 5.5.5 Comparison with other MSA methodologies 213
  - 5.5.5.1 MO-SAStrE vs. MSA-GA and VDGA 213
  - 5.5.5.2 MO-SAStrE vs. SAGA, RBT-GA and VDGA . . . 215
  - 5.5.5.3 Statistical evaluation for both comparisons 217
- 5.6 Conclusions and final statements 218

6 Conclusions and Future Work **221**

- 6.1 Conclusions 223
  - 6.1.1 Conclusions regarding multiple sequence alignments . . . 223
  - 6.1.2 Conclusions about PAcAlCI 224
  - 6.1.3 Conclusions about intelligent scores for the estimation of alignment qualities 226
  - 6.1.4 Conclusions about MO-SAStrE 228
- 6.2 Future Work 231

Conclusiones y Trabajo Futuro **233**

- Conclusiones 235
  - Conclusiones respecto los alineamiento múltiple de secuencias . . 235
  - Conclusiones acerca de PAcAlCI 236
  - Conclusiones respecto a los métodos avanzados para la evaluación de alineamientos 239
  - Conclusiones acerca de MO-SAStrE 241
- Trabajo Futuro 245

Bibliography **247**

Curriculum Vitae **266**

List of Figures

1.1 Structure and main components of a eukaryotic cell [NLM, 2014] 12

1.2 Diagram of the double helical strand in DNA formed by the A-T and G-C base pairs interaction by hydrogen bonds (extracted from [NLM, 2014]) 14

1.3 Chromosome structure formed by packaged DNA and histone proteins [NLM, 2014]. 15

1.4 Schematic processes of transcription and translation inside a cell (Central Dogma) [NLM, 2014]. 17

1.5 Levels of protein structures. There are four different levels in the protein conformation from the lineal sequence (primary structure) to the most complex association of structures (quaternary structure). This diagram was extracted from the National Human Genome Research Institute [NHGRI, 2014]. 20

1.6 Type of mutations and alterations in DNA sequences (the mutated bases are in shading). (A) Synonymous mutations do not vary the coded amino acid because the resulting codon codes the same amino acid; (B) Missense mutations produce the modification of one amino acid ; (C) Nonsense mutations convert a codon into the stop codon and, therefore, the corresponding protein is shortened; (D) One DNA base pair is inserted in the sequence (insertion); (E) One base pair is removed from DNA sequence (deletion). 21

1.7 Cost of sequencing for one whole genome. The sequencing cost has been substantially reduced from early 2008 [National Human Genome Research Institute (NHGRI), 2014]. 23

1.8 Example of a standard GEO Profile output. This profile has been obtained for the 79091962 accession by searching the *ABL2* human gene related to the pancreatic adenocarcicoma in the database of GEO Profiles. Six samples have been analyzed for this gene (3 control + 3 treated patients). 36

2.1 Standard representation of a sequence alignment. Five sequences (I-V) are being aligned in this case. 58

2.2 (A) Identity matrix for DNA alphabet (A, C, G and T). (B) Dayhoff similarity matrix with PAM 250 (20% of identities). This matrix is built for the 20 amino acids in proteins, grouping them according to their physico-chemical properties. 59

2.3 Alignment construction using dynamic programming. The scoring matrix is calculated according to the proposed score scheme ($m = 1$, $mm = -1$ and $gp = -2$). Next, the accumulated matrix is performed by adding the maximum score from the three previous neighboring cells according to the Equation 2.1 ($+a$ if maximum score comes from position (i,j) and $+gp$, otherwise). The arrows show from which cell each score has been obtained. Finally, the best alignment is determined by identifying the back path with the best total scores. When the path moves diagonally a pair of residues are aligned whereas horizontal or vertical movements introduce gaps. This figure has been interpreted from [Xiong, 2006]. 64

2.4 Typical alignment procedure with the progressive algorithms. Four sequences (S1-S4) are aligned in this case. All possible pairwise alignments are first performed with a dynamic programming procedure (for clarity, matches in alignments are shadowed). A distance matrix between each two sequences is then calculated to build the guide tree (the Equation 2.4 is applied in this case). The guide tree to progressively align pairwise sequences, pairwise partial alignments (profiles) or sequence/profile is then designed. The distance matrix is updated in each node considering the previously created profiles (partial alignments). The final multiple sequence alignment is generated following the guide tree until including all the initial sequences. 67

2.5 Comparison between a suboptimal multiple alignment and its corresponding optimal solution. The second alignment is more accurate (see red vs. green residues) than the progressive solution. 68

2.6 Typical alignment procedure with consistency-based algorithms (in this case, T-Coffee). (A) Four input sequences to be aligned. (B) All possible pairs of sequences are aligned by using a progressive algorithm (e.g. ClustalW) (identities are shadowed). (C) The primary library is performed by weighting each pair of aligned residues according to the number of identities in the pairwise alignment (see Equation 2.5). (D) The initial library is extended considering the consistency of the previous pairwise alignments through the remaining sequences. For simplicity, only the case of $S1 - S2$ pairwise alignment is shown. Here, it is checked if each pair of aligned residues is also aligned through $S3$ and $S4$. In these cases, the weights in the library for these pairs of residues are increased (see Equation 2.6) (see explanation in text for detail). Lines in the alignments indicate the degree of consistency that each pair of residues achieves (the more consistency, the wider lines). (E) The pairwise alignments are rebuilt with dynamic programming using the extended library weights for the scoring matrix. Thus, the possible mistakes in the previous progressive method are avoided by introducing the consistency with other sequences. (F) The final multiple alignment is performed with a standard progressive algorithm but including the consistent pairwise alignment previously rebuilt. 72

2.7 Pairwise alignment based on structural information with SAP algorithm (Scheme adapted from Taylor and Orengo [1989]). (A) Scoring matrices based on the distance among residues are created, centring each sequence in a specific residue. In this case, 56 scoring matrices should be calculated according to the 56 possible pairs of residues (for simplicity, only matrices centring in the residue pairs F-C and V-C are shown). (B) The optimal pathways from scoring matrices (colored cells) are accumulated in a substitution matrix. (C) The final alignment is rebuilt from the accumulated matrix according to the best final pathway. 77

2.8 Promals algorithm is graphically represented. Sequences are firstly aligned by groups (>60% of identities), selecting one representative sequence from each group. Homological information and secondary structure predictions are retrieved respectively from PSI-BLAST and PSIPRED for each representative sequence. This information is then incorporated to profiles for each sequence. Each pair of profiles are compared by designing a profile-profile HMM to determine the probability of aligning these profiles according to their homologies and secondary structures. This probabilities is incorporated to the consistency library. Finally, representative sequences and their groups are aligned. This diagram has been interpreted from Pei and Grishin [2007]. 78

2.9 Average BAliscore values together with standard errors obtained for each MSA tool in every BAliBASE subset. Progressive approaches, consistency-based algorithms and algorithms with additional data are highlighted in blue, green and red color schemes, respectively. 83

2.10 Average time per alignment used by the different alignment approaches (progressive methods, consistency-based algorithms and approaches with additional data) to align the BAliBASE dataset (218 sets of sequences). Note that time is represented in logarithmic scale. 84

3.1 PAcAlCI scheme. The architecture is developed into four modules: *(i)* the *Input Dataset* module composes the alignments from 10 different tools; *(ii)* the *Feature Extraction* module consults and retrieves 23 features from well-known databases, *(iii)* the *Feature Selection* module determines the most relevant features for the prediction of accuracy; and *(iv)* the *LS-SVM Prediction* module performs the final accuracy estimation and selects the most suitable alignment tools according to such estimations. 95

3.2 Evolution of the mean relative error. The number of features progressively increases in ascendant relevance order. The training and validation errors are shown. 122

3.3 Distribution of relative errors for training and validation sets. The corresponding LS-SVM prediction was performed using 10 features. The *relative frequency* is represented in the Y-axis (number of samples with their errors in each range normalized by the total number of samples in the dataset). In the case of the validation error The highest errors with less frequency are enlarged to be appreciated in detail. 124

3.4 Distribution of relative errors for training and validation processes by filtering out low accurate alignments. The *relative frequency* is represented in the Y-axis (number of samples with their errors in each range normalized by the total number of samples in the dataset). The highest errors with less frequency have now been almost removed. The LS-SVM prediction is then improved, avoiding predictions with high errors ($\alpha = 0.5$). 125

3.5 Intersection of real and predicted suitable methodologies (Venn diagrams) corresponding to the four alignments whose accuracies are shown in Table 3.4. 127

4.1 Flowchart of the full proposed implementation. The regression procedures were implemented including several stages: *(i)* The *Alignment Dataset* is constructed by aligning the BAliBASE and OxBench sets of sequences with 10 different aligners, *(ii)* the *Feature Extraction* retrieves features associated with several biological properties from different sources and databases, *(iii)* the *Feature Selection* applies the NMIFS algorithm to determine the subset of the most relevant features and *(iv)* four different regression models are performed to estimate the quality of alignments in the *Regression* stage. 136

4.2 Error evolution and feature selection criterion according to the number of features included in the regression approaches. Both errors and AIC_{error} are calculated for the validation subsets in the 10-fold cross-validation process applied to the dataset of 3700 alignments. These errors are calculated as the mean absolute error (MAE) (see Equation 4.39). The AIC_{error} criterion determines the optimal number of features (minimal AIC_{error} value) for each model. This criterion based on Akaike Information Criterion (AIC) is calculated according to the Equation 4.41 167
4.3 Graphical correlation between BAliscore and the remaining considered scores (PAM250, RBLOSUM62, GONNET and STRIKE) as well as the proposed advanced scores based on regression methodologies (LS-SVM, Regression Tree, Bagging trees and Gaussian Processes). For simplicity, BLOSUM62 is omitted as it returned a correlation similar to RBLOSUM62. 169
4.4 Sensitivity, specificity and accuracy obtained for the 8280 pairwise comparison of alignments in each score scheme. The values of each score are compared against the BAliscore value to determine these measurements. 170

5.1 General scheme of an evolutionary algorithm as a flow-chart (Introduction to Evolutionary Computing, [Eiben and Smith, 2008]) 179
5.2 Graphical definition of Pareto front. For simplicity, only two objectives to be maximized (f_1 and f_2) are shown. As appreciated, solutions in the Pareto front (red circles) have not any other solution with better values in all the objectives (these solutions are non-dominated), whereas the remaining solutions (blue circles) have at least another solution with better values in all the objectives (these solutions are dominated). 181

5.3 Pareto ranking and selection schemes in NSGA-II. The extended population in each generation is composed by the previous population (P) and offspring (O) created by operators. The population is classified in different Pareto fronts ($F_1, \ldots, F_N$) in order to obtain a non-dominated sorting. The first ranked solutions are included in the next population. The last selected solutions are determined by the crowding distance in the last considered front F_t (see Equation 5.3). This figure has been adapted from Deb et al. [2002]. 184

5.4 Two stochastic multiobjective optimizers are compared with regard to two objectives (f_1 and f_2) by using dominance ranking: MO_1 (red) and MO_2 (blue). In this case, both objectives must be maximized. According to the definition in Equation 5.4, none of the Pareto fronts is *better* than Runs 1 in MO_1 and MO_2 and, therefore, both front are assigned the lowest rank 1. However, all Pareto fronts are *better* than Run 3 in MO_1 and accordingly its rank is 5 (the worst one). Overall, the resulting ranks are (1, 3, 5) for MO_1 and $(1, 2)$ for MO_2. 185

5.5 Graphical definition of the hypervolume for two objective (f_1 and f_2). The hypevolume value is calculated as the area (orange region) which is covered from the Pareto front (red line) to the bounding point (BP) (orange square). Since objectives are maximized in this case, the BP is located under the minimal objective range. . . 186

5.6 Graphical representation of the empirical attainment function (EAF). The EAF divides the objective space in regions according to the probability of being weakly dominated by the multiobjective optimizer. Four regions can be differentiated in this example: the orange region is weakly dominated by the three fronts and therefore it is assigned the probability 1 (3/3); the red region is not assigned probability (0/3) since it is not dominated by any front; the remaining blue and green regions are assigned probabilities 1/3 and 2/3 because they are weakly dominated by one and two fronts, respectively. 187

5.7 MO-SAStrE flowchart. The initial population is performed by eight previous suboptimal alignments and filled with the crossover operator until N individuals $^{(1)}$ (*Initial population* stage). Each population is then extended by using the mutation and crossover operators, with p_c and p_m representing the crossover and mutation probabilities $^{(2)}$ (*Operators* stage). Finally, the selection procedure is done by using the NSGA-II selection approach (*Selection* stage). F_t defines the last included Pareto front, where individuals must be selected according to the crowding distance $^{(3)}$. 189

5.8 Pseudo-codification of alignments in MO-SAStrE. (A) Standard representation of a multiple sequence alignment. (B) Alignment encoded by a matrix of integer values: positions in their corresponding sequences (amino acids) and positions of the last amino acid in such sequences denoted with a negative sign (gaps). . . . 191

5.9 Mutation procedure. A region of closed gaps (shading area) is randomly chosen and shifted to another random position. Full columns of gaps are then removed if they are found. 194

5.10 Crossover operator. Both standard and novel codification are shown (see the codification procedure in Figure 5.8). The first parent is divided into two blocks (P1.A and P1.B) according to a randomly selected column. Two blocks are also obtained from the second parent according to the same positions of the selected column (P2.A and P2.B). The blocks are crossed by filling with gaps in order to assemble them. 195

5.11 Optimized solution in MO-SAStrE for the first problem in BAliBASE (1^{st} alignment in RV11). In this case, the optimizer builds an alignment joining aligned blocks from ClustalW (green), RetAlign (blue), and TCoffee (red). Gaps are also shifted by the mutation operator (yellow). 206

5.12 3D surfaces for the input and optimized Pareto fronts in the 1^{st} problem of the BAliBASE RV11 subset. 208

5.13 MO-SAStrE performance in terms of hypervolume (HV) progression (4 different problems). The increase of the HV indicator is represented with respect to the number of generations. For simplicity, only three runs per problem are shown. 211
5.14 Average hypervolume values for each BAliBASE subset. HV values are shown for the input dataset and the optimized Pareto front of alignments. 212

List of Tables

1.1 Correspondence between codons and their coded amino acids (AAs). The 20 amino acids and the stop codons are shown. 19

1.2 Summary of several highly used biological databases. Database names, content and URLs to be accessed are shown. 28

1.3 Summary of the main fields included in a GB file from GenBank. Examples for the ABL2 gene are shown. 31

1.4 Several alternative and formats in Uniprot to access the information associated to the human protein called *Tyrosine-protein kinase ARG*. Depending on the link, different information can be retrieved from this protein. 37

1.5 Examples of principal headers in Uniprot tab-separated text format for the *ABL2_HUMAN* protein. 38

1.6 Principal tags in XML files downloaded from Pfam for a specific protein. Examples for the *ABL2_HUMAN* protein are shown. . . . 42

1.7 Summary of fields in a PSI-MITAB file provided from the IntAct database. The example column is extracted from the first interaction with the *P42684* protein query. 45

1.8 Examples of principal headers in the PDB file for the structure of the *Ubuquitin* protein (1UBI). 48

1.9 Column headers provided by DSSP for the secondary structure description. These columns are part of the standard DSSP file, together with information retrieved from PDB. 50

1.10 Principal column headers in the tab-separated text file provided by QuickGO for the GO annotation of a protein. Some examples included in the *Tyrosine-protein kinase ARG* protein annotation are presented. 52

2.1 Subsets of sequences provided by BAliBASE benchmark 81
2.2 Wilcoxon test for each pairwise comparison between MSA aligners 86

3.1 Summary of the applied methodologies. Ten different methodologies were run to align multiple sequences. Their versions and the applied strategies are also shown. 99
3.2 Summary of the extracted features. 23 features are retrieved from different databases. $^{(1)}$ These features are calculated as the percentage of amino acids (AA) with that specific feature. $^{(2)}$ These features are calculated as the number of occurrences per sequence. 100
3.3 Feature ranking obtained by the mRMR procedure taking into account their dependence against the accuracies in the alignments by means of the maximal relevance and minimal redundancy criteria. 120
3.4 Example of predicted accuracies for four different sets of sequences. Predicted accuracies are compared to those real ones (BAliscore values) obtained by each methodology. Values in bold represent those tools that are selected for each example according to their accuracies and the confidence interval (see section 3.5.3 for details). Therefore, these tools are considered suitable for aligning these sequences. The prediction relative error is also shown (see $E_R(s)$ definition in Equation 3.33). 123
3.5 Qualitative comparison between two similar tools (PAcAlCI and AlexSys) to select adequate sequence aligners. The performance and main attributes of both procedures are shown. 129

4.1 Summary of the 27 features associated with alignments. The three first features ($f_1 - f_3$) are incorporated from the PAcAlCI dataset (see Table 3.2). *Matches* refer to the percentage of paired amino acids with a particular property in common that have been aligned (Equation 4.1). 140
4.2 Parameters optimized for each regression methodology. These optimal values are determined for the final implementation of each approach, also including the optimal number of features. . . 163
4.3 Feature ranking obtained by the NMIFS procedure taking into account their dependence against the BAliscore values (real accuracies). 165

5.1 MO-SAStrE parameter configuration. Six parameters were determined according to common values proposed by Eiben and Smith [2008] for genetic algorithms. 205
5.2 Multiobjective scores for the 1^{st} alignment in the RV11 dataset. Evaluations for input alignments and for optimized MO-SAStrE alignments are represented. Although MO-SAStrE returned 30 alignments from the optimal Pareto front, five representative ones are shown to simplify. 207
5.3 Average scores and standard deviation for the 218 BAliBASE problems optimized by MO-SAStrE. 209
5.4 Comparison with MSA-GA, VDGA, ClustalW and 3D-COFFEE. The BAliscore values are shown for 8 different BAliBASE sets. The two best scores are highlighted in bold. 214
5.5 BAliscore comparison with SAGA, RBT-GA, VDGA and other known methodologies. The BAliscore values are shown for 12 BAliBASE datasets. The two best scores are highlighted in bold. 216

5.6 Wilcoxon non-parametric test. Pairwise comparisons between MO-SAStrE and each other method. ′Sign+′/′Sign-′ identifies the number of problems that MO-SAStrE respectively wins/loses other method . The ′Z-score′ represents the distance between the two compared methods according to the obtained BAliscore (see Equation 2.7 in Chapter 2). 218

Abbreviations

AA	Amino Acid
ADDA	Automatic Domain Decomposition Algorithm
AIC	Akaike Information Criterion
ANOVA	ANalysis Of VAriance
BLAST	Basic Local Alignment Search Tool
BLOSUM	BLOcks SUbstitution Matrix
bp	base pairs
CAO	Contact Accepted mutatiOn
CART	Classification And Regression Trees
CG	Conjugate Gradient
CLUSTAL	CLUSTering ALignment
CPU	Central Processing Unit
DAG	Directed Acyclic Graph
DAG	Directed Acyclic Graphs
DDBJ	DNA Data Bank of Japan
DELTA-BLAST	Domain Enhanced Lookup Time Accelerated BLAST
DIP	Database of Interacting Proteins
DNA	DeoxyriboNucleic Acid
DSSP	Define Secondary Structure of Proteins
DSSP	Define Secondary Structure of Proteins
EA	Evolutionary Algorithm
EAF	Empirical Attainment Function
EBI	European Bioinformatics Institute
EMBL	European Molecular Biology Laboratory
ENA	European Nucleotide Archive
EP	Evolutionary Programming
ES	Evolutionary Strategy
FFT	Fourier Fast Transform
FN	False Negative
FP	False Positive
FSA	Fast Statistical Alignment
GA	Genetic Algorithm
GEO	Gene Expression Omnibus

GO	Gene Ontology
GP	Genetic Programming
GPs	Gaussian Processes
HGP	Human Genome Project
HMM	Hidden Markov Model
HMMT	Hidden Markov Model Training
HOMSTRAD	HOMologous STRucture Alignment Database
HV	Hypervolume
KEGG	Kyoto Encyclopedia of Genes and Genomes
LSSVM	Least-Squares Support Vector Machine
MAE	Mean Absolute Error
MAFFT	Multiple Alignment based on FFT
MI	Mutual Information
MO	Multi-Objective
MO	Multiobjective Optimization
MOEA	Multi-Objective Evolutionary Algorithm
MOGA	Multi-Objective Genetic Algorithm
MO-SAStrE	Multiobjective Optimizer for Sequence Alignments based on Structural Evaluation
MRE	Mean Relative Error
mRMR	minimum-Redundancy Maximum-Relevance
MSA	Multiple Sequence Alignment
MSA-GA	Multiple Sequence Alignment Genetic Algorithm
NCBI	National Center of Biotechnology Information
NGS	Next-Generation Sequencing
NHGRI	National Human Genome Research Institute
NHI	National Institute of Health
NJ	Neighbor Joining
NLM	National Library of Medicine
NMIFS	Normalized Mutual Information Feature Selection
NMR	Nuclear Magnetic Resonance
NSGA-II	Non-Dominated Sorting Genetic Algorithm II
OMIM	Online Mendelian Inheritance in Man
ORF	Open Reading Frame
PAcAlCI	Prediction of Accuracy Alignments based on Computational Intelligence
PAM	Point Accepted Mutation
PDB	Protein Data Bank
PIMA	Pattern-Induced Multi-sequence Alignment
PPI	Protein-Protein Interactions
PREFAB	Protein REFerence Alignment Benchmark
QP	Quadratic Programming
RBF	Radial Basis Function

RBFN	Radial Basis Function Network
RBT	Rubber Band Technique
RBT-GA	Rubber Band Technique Genetic Algorithm
RNA	RiboNucleic Acid
SAGA	Sequence Alignment Genetic Algorithm
SAP	Structure Alignment Protein
SNP	Single-Nucleotide Polymorphism
SOFT	Simple Omnibus Format in Text
SP	Sum of Pairs
SVM	Support Vector Machine
TC	Totally Conserved columns
TN	True Negative
TP	True Positive
UPGMA	Unweighted Pair Group Method with Arithmetic Mean
VDGA	Vertical Decomposition Genetic Algorithm
WSP	Weighted Sum of Pairs

Abstract

The Bioinformatics field is increasingly providing new challenges for bioinformaticians and computer scientists due to the need of analyzing the great amount of exiting biological information, mainly extracted from new massive sequencing techniques. Some challenging tasks in bioinformatics are currently focused on the analysis of protein sequences, retrieving and predicting other related features like structures, functionality or homologies. One of the more powerful tools to compare proteins inferring their associated biological data are the multiple sequences alignments (MSAs).

MSAs are currently widely used strategies in molecular biology. The main objective of these alignments is the comparison of molecular chains (nucleotides or amino-acids, mainly) to extract their relevant similarities and differences. These approaches were originally designed for homology transfer, where poorly characterized protein sequences could be compared to well-studied homologs from typical model organisms. More recently, the usage of MSA strategies has been spread to other researches in phylogenetic analyses, protein structure modeling or functional predictions.

In the last years, the development of novel experimental techniques, such as next-generation sequencing and high-throughput experiments, has prompted a great demand of this kind of analysis. MSA strategies usually help to retrieve biological meanings from the coincidences and matches in nucleotide and amino acid sequences. Therefore, it is essential that MSA tools are able to deal with the massive amount of information generated by these former techniques. With this purpose, advanced computational approaches based on well-known artificial intelligence and machine-learning algorithms like hidden Markov models (HMMs), support vector machines (SVMs) or genetic algorithms (GAs) are being applied. Thus, MSAs are today considered one of the most powerful and necessary procedures in bioinformatics. However, some difficult issues in MSA technologies must still be deeply addressed.

Firstly, although there is a huge amount of existing tools for MSAs, a fair standard to build alignments has not been found yet. As a consequence, each MSA tool usually provides quite different alignments according to their own criterion. In addition to that, the evaluation of alignments is also a controversial issue. Since there is no consensus about which is the best way to evaluate alignments, MSA tools tend to use classical evaluation schemes like PAM or BLOSUM which could result in inaccurate alignments. In this sense, improving these kind of scoring schemes with the incorporation of complementary information may lead to a more realistic idea of the quality of the alignments and more efficient MSA tools. Finally, it is also widely recognized that classical MSA tools cannot often provide high quality alignments when sequences are evolutionarily more distant. In these cases, the knowledge retrieved from the sequences could become insufficient to reach the optimal alignment.

This dissertation is then focused on the application of efficient solutions for the previously commented problems in MSAs. Such solutions are mainly based on advanced/intelligent systems and applied to regression, prediction, classification and optimization problems associated with MSAs. More specifically, three different contributions are proposed considering both the need for obtaining more efficient and accurate alignments and the need for improving novel alignment evaluation schemes.

The first contribution in this thesis takes into account several well-known MSA tools to predict which one/s may provide an accurate alignment for a set of sequences. Specifically, this contribution estimates the quality of each MSA tool for this particular set of sequences before they are aligned. This algorithm is based on a Least-Squares Support Vector Machine (LS-SVM) model and it takes advantage of the integration of relevant biological features from several sources and databases.

The second tool presents a similar approach to the previous one but with a quite different purpose. In this case, several regression models (Gaussian Processes, Regression trees, Bagging trees and LS-SVM) are proposed to design diverse scoring schemes for MSAs. Such schemes seek to integrate not only information directly provided by the alignments but also some other features related to the aligned proteins. Consequently, a more sophisticated evaluation score is presented which is capable of detecting more distant relationships between sequences and, therefore, more realistic estimations of the alignment accuracy.

Finally, the third contribution is focused on the optimization of MSAs. The principal goal is to improve the quality of alignments previously aligned by other tools. This optimizer is based on a genetic algorithm with a multi-objective fitness function. The presented algorithm takes advantage of our own-designed crossover and mutation operators as well as three different objectives, one of them based on the structural conservation of the sequences. Thus, the addition of structural data allows us to achieve more accurate alignments when sequences are evolutionarily less related.

Resumen

El campo de la Bioinformática está continuamente generando nuevos retos debido a la necesidad de analizar la gran cantidad de información biológica de la que se dispone en la actualidad, principalmente extraída de las nuevas técnicas de secuenciación masiva (NGS). Algunos de estos desafíos están enfocados al análisis de secuencias de proteínas, para obtener o predecir otras características relacionadas tales como estructuras, funcionalidades u homologías. Una de las herramientas más potentes en este sentido son los alineamientos múltiples de secuencias (MSAs).

Los MSAs constituyen una de las estrategias más ampliamente utilizadas en la actualidad en la Biología Molecular. Su principal cometido es la comparación de cadenas moleculares (principalmente nucleótidos o aminoácidos) en la búsqueda de las semejanzas y diferencias más relevantes. Estas técnicas fueron inicialmente diseadas para la transferencia de homología gracias a lo cual secuencias de proteínas pobremente caracterizadas podían compararse con otras homólogas, profundamente conocidas, pertenecientes a organismos modelos. En la actualidad, el uso de las estrategias de MSAs se ha extendido a otros numerosos campos como los análisis filogenéticos, modelado estructural de proteínas o predicciones de funcionalidad.

El desarrollo en los últimos aos de novedosas técnicas experimentales tales como la secuenciación masiva o de nueva generación (NGS) y experimentos de alto rendimiento, han conllevado una gran demanda de este tipo de análisis. Las estrategias de MSAs contribuyen a la obtención de información biológica a partir

de las coincidencias entre las secuencias de nucleótidos o de aminoácidos. Así, es esencial que las herramientas de MSAs sean capaces de procesar la enorme cantidad de información generada a través de las anteriormente citadas técnicas. Con este fin se están aplicando numerosas estrategias computacionales avanzadas basadas en algoritmos de inteligencia artificial y de aprendizaje supervisado (*machine learning*) tales como modelos ocultos de Markov (*hidden Markov models*, HMMs), máquinas de vector soporte (*support vector machines*, SVMs) o algoritmos genéticos (GAs). Así, las técnicas de alineamiento múltiple de secuencias están considerados en la actualidad uno de los procedimientos más potentes y necesarios en la Bioinformática. Sin embargo, es todavía necesario abordar ciertas carencias que presentan estas técnicas.

En primer lugar, a pesar de la existencia de numerosas herramientas para el alineamiento múltiple de secuencias, todavía no se dispone de un estándar apropiado para construir los alineamientos. Como consecuencia, cada herramienta genera un alineamiento que puede diferir notablemente del generado por otra, debido a la aplicación de sus propios criterios. La evaluación de los alineamientos también genera un problema adicional. Dado que no existe un consenso acerca de qué metodología es la más adecuada para evaluar el alineamiento, se tiende a evaluarlo aplicando los sistemas clásicos de evaluación tales como PAM o BLOSUM lo que puede conllevar a alineamientos no suficientemente precisos. Así, la mejora de estos sistemas de evaluación mediante la incorporación de información complementaria podría contribuir a la mejora del análisis de calidad y a la obtención de herramientas de alineamiento más eficientes. Finalmente, también es ampliamente conocido que las técnicas clásicas de alineamientos de secuencias no proporcionan una calidad aceptable en el alineamiento cuando se trata de secuencias evolutivamente distantes. En estos casos la información obtenida de las secuencias podría ser insuficiente para alcanzar el alineamiento más óptimo.

Por tanto, esta tesis está orientada a tratar de dar solución a los problemas previamente expuestos sobre las técnicas de alineamiento múltiple de secuencias. Estas soluciones están basadas principalmente en sistemas inteligentes y avanzados que se han aplicado a los problemas de regresión, predicción, clasificación y optimización subyacentes a los MSAs. En concreto, hemos propuesto tres aportaciones a este campo considerando la necesidad de obtener alineamientos eficientes y precisos así como la necesidad de mejora de los sistemas de evaluación.

La primera contribución de esta tesis considera numerosas herramientas ampliamente conocidas de MSA para predecir cuál de ellas proporcionaría un alineamiento más preciso para un conjunto de secuencias que se quieren alinear. Específicamente, en esta sección de la tesis, se estimará la calidad de cada herramienta de MSA analizada para alinear un conjunto determinado de secuencias, antes de que el alineamiento sea realizado. Este algoritmo está basado en un modelo de *Least-Squares Support Vector Machine* (LS-SVM) e integra características biológicas relevantes obtenidas de varias fuentes y bases de datos.

El segundo aporte de esta tesis doctoral es una herramienta similar a la anterior pero con un propósito considerablemente diferente. En este caso se propondrán una serie de modelos de regresión (procesos Gausianos, árboles de regresión, Bagging trees y LS-SVM) para disear diversos sistemas de evaluación de alineamientos. Estos sistemas de evaluación tienen como objetivo integrar no sólo la información que se extrae de los alineamientos sino también otras características de las proteínas que han sido alineadas. De esta manera presentamos un sofisticado sistema de evaluación capaz de detectar relaciones más distantes entre secuencias y, por tanto, capaz de estimar de forma más realista la precisión de los alineamientos. Por último, la tercera propuesta de esta tesis doctoral es una optimización de las técnicas de alineamiento múltiple de secuencia. Con esta aportación se pretende mejorar la calidad de los alineamientos llevados a cabo por otras técnicas. Este optimizador está basado en un algoritmo genético con una función de fitness multiobjetivo. Dicho algoritmo aplica operadores de crossover y de mutación diseados por nuestro grupo de investigación así como tres objetivos diferentes, uno de los cuales está basado en la conservación de la estructura de las secuencias. Así, la adicción de información estructural nos permite obtener alineamientos más precisos en los casos de secuencias menos relacionadas evolutivamente.

Chapter 1

From Biology to Bioinformatics

1.1 Introduction

Biology is considered a fundamental science, especially dedicated to the study of life. It describes the mechanisms, classifications, processes and behaviors of living organisms and their interactions with the environment. As other sciences, biology can seem to be extremely complex and not easy to completely understand. However, its study has benefited from the advance of technology and the discovery of novel experiments that have helped to retrieve more biological information and, therefore, to increase the generated knowledge. These advances are specially producing a high impact in the molecular biology.

As noted by the National Center for Biotechnology Information (NCBI) [2013], new trends in molecular biology are generating such an amount of data that are dramatically altering the understanding of the processes which underlie all living things. This new knowledge is specially affecting important fields like medicine and biotechnology. The information is being stored in thousand of repositories and databases all over the world. Such information must be handled by powerful personal computer but also supercomputers and specialized systems with large storage capacity (i.e. computer clusters). It is therefore essential to focus the efforts on the development of specialized tools and environments which could take advantage and optimize the management of the generated data. These data are still far from being fully understood and there are still millions of properties and relationships that should be more widely studied.

Therefore, some basic biological concepts are outlined in this chapter in order to clarify why molecular biology is increasingly requiring more and more complex computational resources (section 1.2). Subsequently, the Human Genome Project (HGP) and Next Generation Sequencing (NGS) technologies are briefly explained since they are two of the main technologies responsible for the increase of data in molecular biology (section 1.3). Next, the field of bioinformatics and its importance to handle these existing biological data is introduced (section 1.4). Finally, an overview of how this information is currently organized in databases or how it can be consulted is provided (section 1.5). Since these issues will be repeatedly used throughout this dissertation, it is essential to defined them in detail within this introductory chapter.

1.2 Biological background

1.2.1 The cell

To understand the complexity of life, it is first necessary to focus our understanding on the most basic unit, the cell, and its functionality. Organisms could be composed from one single cell (unicellular) to trillions of cells like the human beings. Cells carry out all the diverse functions which synchronously perform the complex life's functions. Cells also contain the genetic material necessary to the cell replication [NLM, 2014]. Two kinds of cells can be defined according to the nature of their nuclei: prokaryotic and eukaryotic cells. Prokaryotic cells consist of a closed region with a simple internal organization without nucleus surrounded by a plasma membrane. Eukaryotic cells, unlike prokaryotic cells, are composed of many compartments (also called organelles), each one with a specific purpose in the cell. The cell structure and its organelles are schematically represented in Figure 1.1.

The plasma membrane is the external layer which delimits the cell and se-

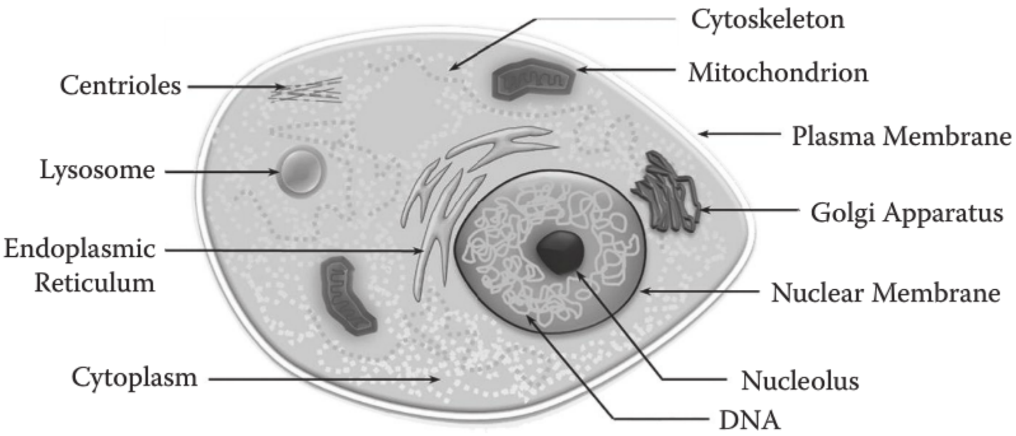

Cytoskeleton
Centrioles
Mitochondrion
Plasma Membrane
Lysosome
Golgi Apparatus
Endoplasmic Reticulum
Nuclear Membrane
Cytoplasm
Nucleolus
DNA

FIGURE 1.1: Structure and main components of a eukaryotic cell [NLM, 2014]

lectively allows materials to enter and leave. The region of the cell surrounding the nucleus till the plasma membrane is the cytoplasm. The cytoplasm comprises the cytosol (fluid) and the remaining organelles. The largest organelle is the cytoskeleton, which forms a network of long fibers that constitutes the structure and shape of the cell. The cytoskeleton also participates in the division and movement functions of the cell. The endoplasmic reticulum is composed of several branches and tubules. The main function of the endoplasmic reticulum is to collaborate in the processing and transporting of molecules created by the cell. Ribosomes are responsible for interpreting the genetic instructions to produce proteins. Ribosomes can be attached to the endoplasmic reticulum or freely floating in the cytoplasm. The Golgi apparatus helps in the excretion process, that is, stores the molecules processed by the endoplasmic reticulum and transports them outside. Similarly, lysosomes are responsible for the recycling of bacterias invading the cell, toxic substances or useless components. Mitochondria are in charge of obtaining energy to all the processes the cell develops. They are sophisticated organelles with their own genetic material (independent of the nucleus) and replication mechanics. Centrioles are paired barrel-shaped organelles which help determine the locations of the nucleus and other organelles within the cell. They also control that the processes of mitosis and meiosis (cell division) are adequately carried out. Finally, the nucleus serves as a 'commander' giving instructions for the growth, maturation, division or death of the cell. The nucleus holds the heredity material in the deoxyribonucleic acid (DNA) and it is surrounded by the nuclear membrane. This membrane serves as a barrier that regulates interchanges between the nucleus and the cytoplasm.

1.2.2 The Deoxyribonucleic Acid (DNA)

The deoxyribonucleic acid, or DNA, is a polymer that holds the genetic material in almost all the organisms. DNA contains the information about how, when and where to produce each kind of protein. DNA is mainly located in the nucleus (nuclear DNA) but it could also be found to a lesser extent in the mitochondria (mitochondrial DNA or mtDNA). DNA is structurally formed by two

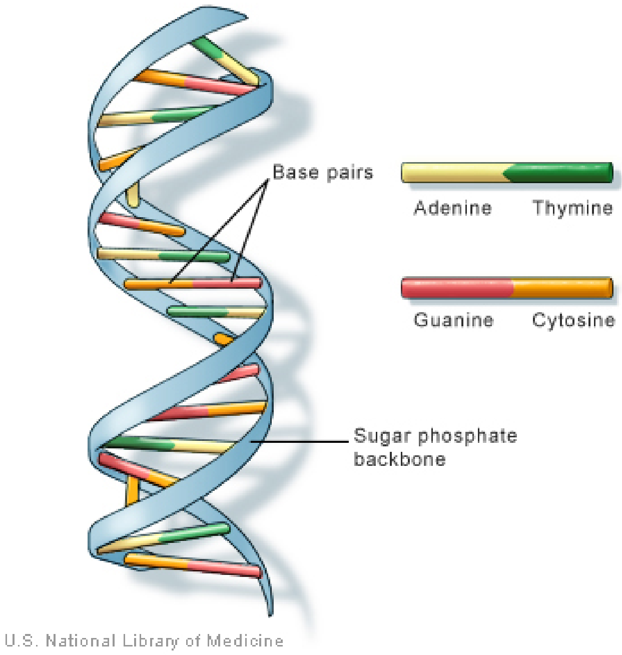


FIGURE 1.2: Diagram of the double helical strand in DNA formed by the A-T and G-C base pairs interaction by hydrogen bonds (extracted from [NLM, 2014])

long helical strands composed of monomers called nucleotides. Each nucleotide consists of a cyclic organic base, a sugar molecule and a phosphate molecule. The sugar and phosphate form the helical backbone of each strand whereas the bases join both strands. This double-helix DNA structure was firstly described by the Nobel prize winners James Watson, Francis Crick et al. [1953]. Human DNA is usually containing about 3×10^9 bases, with 99% of similarity in all human beings. Moreover, nearly every cell in the human body has the same DNA [NLM, 2014].

Four different cyclic organic bases are included in DNA: adenine (A), guanine (G), cytosine (C) and thymine (T). This bases are combined in the two strands to form base pairs by joining A with T and C with G through hydrogen bonds (Figure 1.2). The order of these bases is considered the instructions for building and maintaining an organism. This double helix plays a key role in the DNA replication, since each strand acts as a template to duplicate the sequence

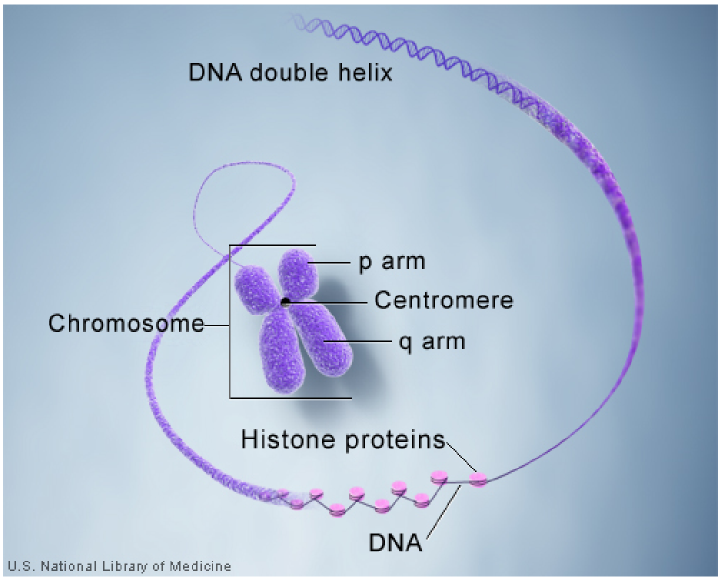


FIGURE 1.3: Chromosome structure formed by packaged DNA and histone proteins [NLM, 2014].

bases. Therefore, during cell division, each new cell will contain one strand from the original cell and a newly synthesized copy (this is called semi-conservative replication).

DNA is stored in the nucleus of the cell in several thread-like structures called chromosomes. Typically, each chromosome has a structure formed by a junction point or centromere and two differenced sections called arms (Figure 1.3). The centromere position helps to determine the chromosome shape and the gene locations. The arms are divided in two sections: the shortest section is known as *p-arm* (*p* for *petit* in French) whereas the longest one is *q-arm* (*q* for *queue* in French). The chromosomal structure is supported by the histones, proteins which allow DNA to coil several times around them forming the chromosome. In humans, each cell normally contains 23 pairs of chromosomes.

DNA contains the genes, which are the regions of a chromosome with the basic physical and functional unit of heredity. Most genes contain the information for synthesizing specific proteins. Genes are habitually composed by portions of sequence that codes the amino acids of the protein (*exons*) together with other non-coding portions (*introns*) [NHGRI, 2014]. The gene sizes can vary from hundreds to more than 2 million of base pairs (bp). For instance, genes in simple organisms like the yeast (*Saccharomyces cerevisiae*) have an average of 2,000 bp whereas more complex organisms like humans reach 10,000-20,000 bp per gene, in average [Cooper and Hausman, 2000]. Anyway, the human genome contains relatively short genes like the *Histone H1a* (HIST1H1A) with 781 bp or larger genes like the *Dystrophin* (DMD) with $2.2{\times}10^6$ bp. Although most individuals from the same species share the same genes, there are genes (around 1%) providing particular characteristics for each individual, i.e. the body shape or the hair color. Several forms of the same gene with small differences are known as alleles.

1.2.3 The Central Dogma: *from genes to proteins*

As stated, genes hold the necessary information to synthesize proteins. Proteins are large, complex molecules which have critical functions in organisms. They are responsible of the main work in cells, but also are needed for the regulation, structure an function of tissues and organs [NLM, 2014]. Proteins are formed by smaller units called amino acids which are linked one to another forming chains. There are 20 different amino acids in proteins. Amino acids are usually classified in several groups according to their chemical properties [Mathews et al., 2000]: uncharged polar amino acids (*Glycine, Alanine, Proline, Valine, Leucine, Isoleucine and Methionine*); aliphatic non-polar amino acids (*Serine, Threonine, Cysteine, Asparagine and Glutamine*); positively-charged basic amino acids (*Lysine, Arginine and Histidine*); aromatic amino acids (*Phenylalanine, Tryptophan and Tyrosine*); and negative-charged acid amino acids (*Aspartate and Glutamate*). The functionality and the three-dimensional structure of each protein is directly associated with the sequence of amino acids within the protein.

The creation of proteins from genes is a complex and carefully performed procedure within each cell. There are two major processes (transcription and translation) which constitute one of the most fundamental principles of the molecular biology, also called *the Central Dogma* (see Figure 1.4). Together, both processes constitute the whole process of gene expression.

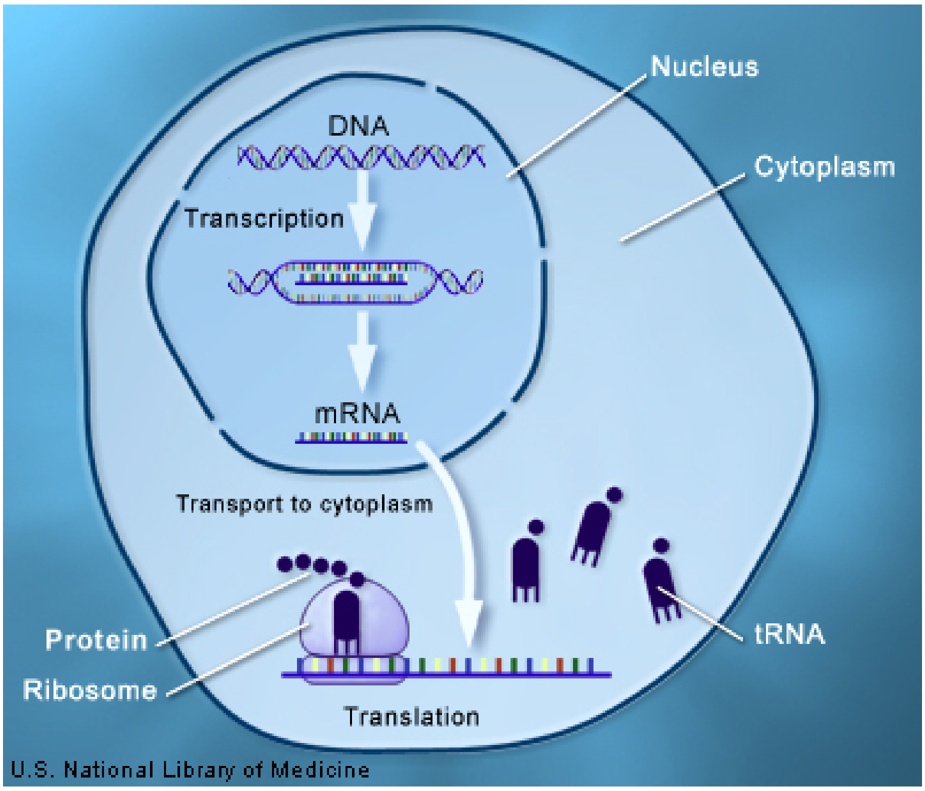

Nucleus
DNA
Cytoplasm
Transcription
mRNA
Transport to cytoplasm
Protein
Ribosome
tRNA
Translation
U.S. National Library of Medicine

FIGURE 1.4: Schematic processes of transcription and translation inside a cell (Central Dogma) [NLM, 2014].

Gene expression follows a stepwise mechanism: Firstly, during the transcription, the entire gene is copied into a similar single-stranded molecule called ribonucleic acid (RNA). Both RNA and the DNA contain the same chain of nucleotide bases, although the *thymine* bases (T) are substituted by *uracil* bases (U) in the RNA. This process is carried out by a large enzyme called RNA polymerase in the cell nucleus. A specific type of RNA called messenger RNA or mRNA, which contains the information to synthesize a protein, is then transferred out of the nucleus into the cytoplasm. Before being transfered to the cytoplasm, this mRNA must be matured through several processes which include the excision of the non-coding regions in the gene (introns). This excision process is called *splicing*. Depending on how this mRNA maturation is carried out, an array of different mRNAs can be generated (*alternative splicing*).

After that, in the process of translation, the ribosome interacts with the mature mRNA in the cytoplasm to read its bases and convert them into amino acids. Each set of three bases, called codon, usually codes one specific amino acid. However, several codons can code the same amino acid (*codon degeneracy*). In addition to code the *Methionine* amino acid, the codon 'AUG' determines the position where the translation starts. Also, some special codons ('UGA', 'UAA' and 'UAG') indicate to the ribosome the end of the protein (stop codons). The part of the gene between the start and stop codons is known as *Open Reading Frame* (ORF). The correspondence between mRNA codons and amino acids is shown in Table 1.1. Another type of RNA called transfer RNA (tRNA) assembles the protein, one amino acid at a time. Amino acids are then assembled into the protein until the ribosome finds a *stop* codon.

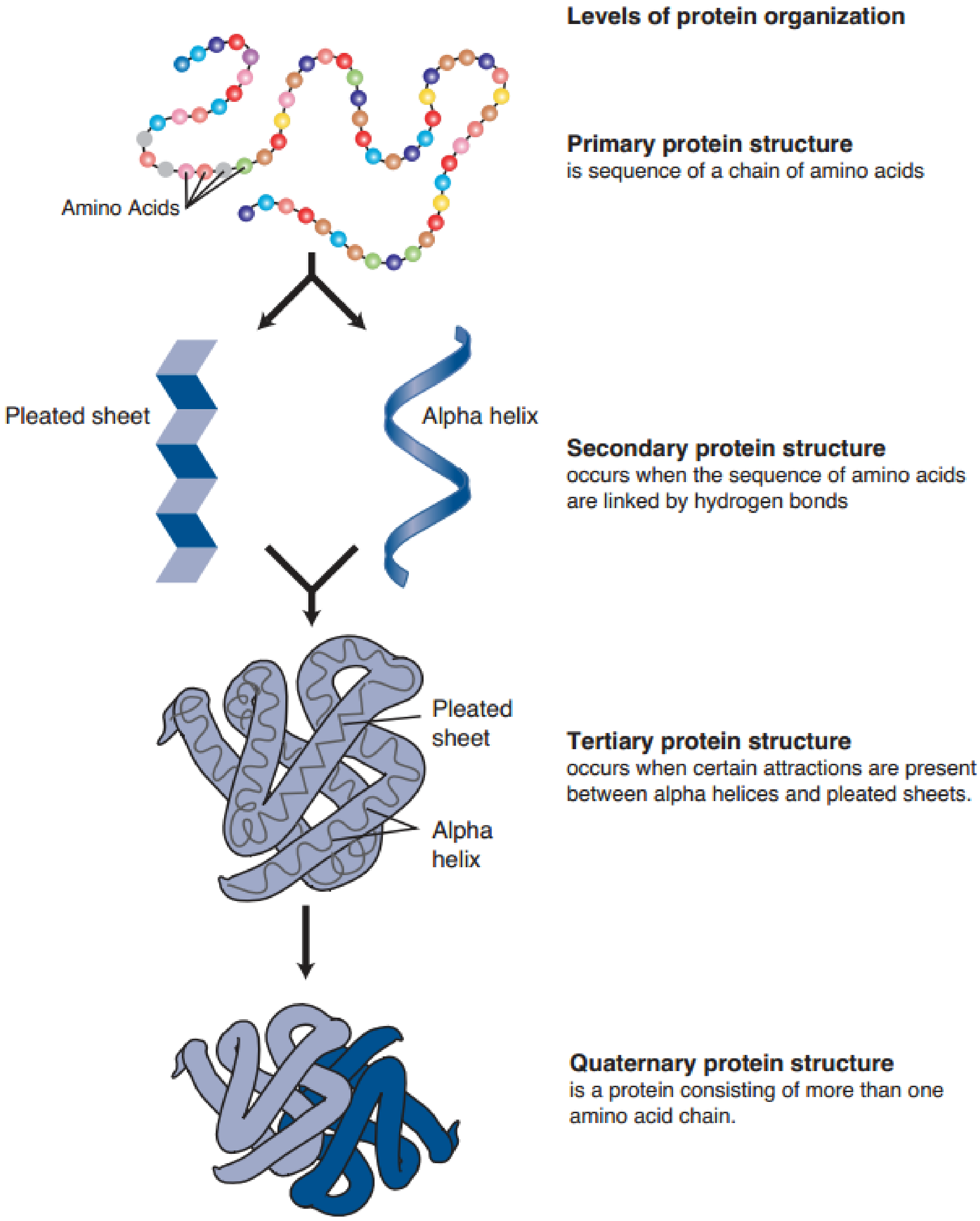
The assembling of proteins is made by folding the amino acid sequence in a particular structure. This structure is usually divided into four differentiated levels: primary, secondary, tertiary and quaternary structures (see these structural levels graphically in Figure 1.5). The structure in a protein is closely related to the role that the protein will carry out. Proteins play a variety of roles in the cell, including structural (cytoskeleton), mechanical (muscle), biochemical (enzymes), or cell signaling (hormones).

1.2.4 Gene mutations

A mutation consists in a permanent modification in the DNA sequence of a gene. Mutations can occur in a single base or in a larger section in the chromosome. When the variation is occurring in one nucleotide, this mutation is also known as single-nucleotide polymorphism (SNP). A mutation can be passed from parents to children (hereditary mutation, if it happens in the sexual cells) or be acquired during the lifetime (*de novo* and somatic mutations). De novo mutations occur just after fertilization in egg or sperm cells whereas somatic mutations could occur during the whole life. Therefore, while hereditary and *de novo* mutations usually affect every cell in an organism, the somatic mutations could happen on individual cells caused by environmental factors or mistakes in DNA replication.

TABLE 1.1: Correspondence between codons and their coded amino acids (AAs). The 20 amino acids and the stop codons are shown.

Amino Acid	**Symbol**	**Codon**	**Amino Acid**	**Symbol**	**Codon**
Methionine	Met/M	AUG	Tryptophan	Trp/W	UGG
Asparagine	Asn/N	AAU AAC	Aspartate	Asp/D	GAU GAC
Cysteine	Cys/C	UGU UGC	Glutamine	Gln/Q	CAA CAG
Glutamate	Glu/E	GAA GAG	Histidine	His/H	CAU CAC
Lysine	Lys/K	AAA AAG	Phenylalanine	Phe/F	UUU UUC
Tyrosine	Tyr/Y	UAU UAC	Isoleucine	Ile/I	AUU AUC AUA
Alanine	Ala/A	GCU GCC GCA GCG	Glycine	Gly/G	GGU GGC GGA GGG
Proline	Pro/P	CCU CCC CCA CCG	Threonine	Thr/T	ACU ACC ACA ACG
Valine	Val/V	GUU GUC GUA GUG	Arginine	Arg/R	CGU CGC CGA CGG AGA AGG
Leucine	Leu/L	UUA UUG CUU CUC CUA CUG	Serine	Ser/S	AGU AGC UCU UCC UCA UCG
Stop	–	UAA UAG	Stop	–	UGA



Levels of protein organization

Primary protein structure
is sequence of a chain of amino acids

Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds

Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.

Quaternary protein structure
is a protein consisting of more than one amino acid chain.

FIGURE 1.5: Levels of protein structures. There are four different levels in the protein conformation from the lineal sequence (primary structure) to the most complex association of structures (quaternary structure). This diagram was extracted from the National Human Genome Research Institute [NHGRI, 2014].

Depending on the type of mutation, the effects in the coded protein and its functionality could vary (Figure 1.6). Synonymous mutations (also called silent mutations) modify one DNA base pair without affecting the generated amino acid because the varying codon is associated to the same amino acid. In missense mutations, the change of one DNA base pair leads to the substitution of one amino acid for another in the protein. Finally, nonsense mutations change one DNA base pair producing one stop codon and prematurely ending the protein sequence (shorter and probably useless protein). Both missense and nonsense mutations are considered non-synonymous mutations.

Other possible alterations in genes are the insertion and deletion processes. An insertion adds a DNA base in the DNA chain modifying amino acids in the protein from this insertion on. Contrarily, deletions remove a DNA base pair reducing the gene length and altering the protein amino acids in a similar way.

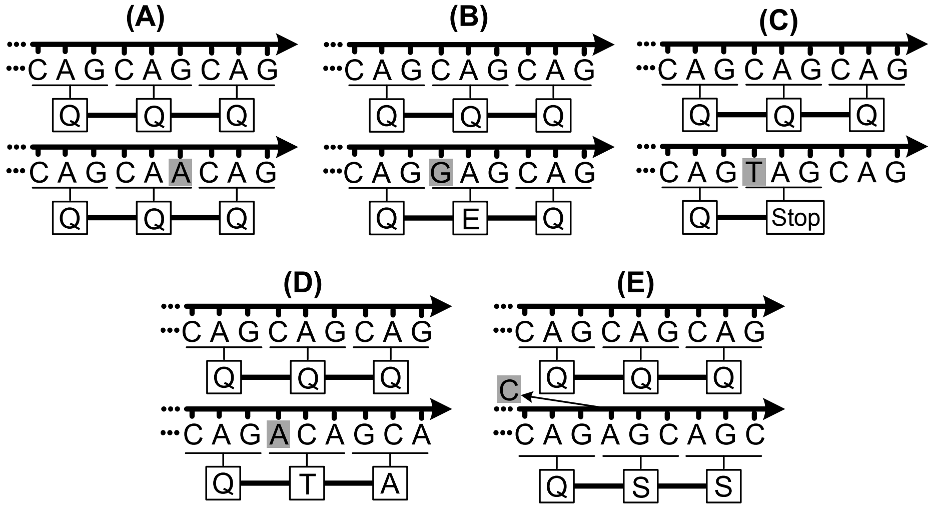


FIGURE 1.6: Type of mutations and alterations in DNA sequences (the mutated bases are in shading). (A) Synonymous mutations do not vary the coded amino acid because the resulting codon codes the same amino acid; (B) Missense mutations produce the modification of one amino acid ; (C) Nonsense mutations convert a codon into the stop codon and, therefore, the corresponding protein is shortened; (D) One DNA base pair is inserted in the sequence (insertion); (E) One base pair is removed from DNA sequence (deletion).

1.3 Advances in massive DNA sequencing

In this section, the main advances in DNA sequencing are presented, mainly focused on the Human Genome Project and Next-Generation Sequencing (NGS) technologies. The progress in sequencing has supposed a breakthrough in molecular biology retrieving a huge volume of data and motivating the increase of acquired knowledge.

1.3.1 Human Genome Project (HGP)

The Human Genome Project (HGP) [Watson, 1990] was an international consortium to determine the DNA sequences of the whole human genome, identifying its corresponding genes. This project officially began in 1990 and was finished by 2003 [Collins et al., 2003]. The HGP has accurately provided 3×10^9 of DNA base pairs corresponding to around 20,000-25,000 human genes. It has also sequenced other organisms such as mouse, fruit fly or yeast. These sequences can also be useful to analyze similarities and differences with the human genes, discovering common sequences and functions among several organisms. In addition to the DNA sequences, the project identified the locations of many genes and their main structures and organizations. The HGP organization also provided some free software tools to make easier the analysis of the data.

The work of the HGP has supposed a turning point to allow researcher learning more about genes, proteins and their functionalities. The HGP has then facilitated the increase of biological knowledge which implies a great impact in the areas of biomedicine, biotechnology and life sciences. Moreover, this project has caused the necessity of more complex tools to efficiently study the huge amount of provided information, which has enormously promoted the multidisciplinary areas of computational biology and bioinformatics [Collins et al., 2003].

1.3.2 Next generation sequencing (NGS)

As a consequence of the Human Genome Project, the determination of DNA nucleotides in genes (DNA sequencing) has experienced a remarkable advance in recent years. The huge amount of DNA genomes that are being sequenced has a great impact in the study of genetics and, more specifically, genetic variations and disorders. These studies provide the necessary information to understand the evolutionary relationships between sequences or the mechanisms of numerous diseases, even allowing the discovery of novel genetic tests or personalized therapies in patients [Dua and Chowriappa, 2012].

In this context, experts and researchers have been increasingly focused on the development of advanced technologies to rapidly sequence DNA genomes. The growth of these technologies has supposed the reduction of sequencing costs, a significant improvement of the accuracy in sequences and an increased throughput in sequencing. Therefore, the cost of an individual genome sequencing has been exponentially reduced to almost 4,000$ in January 2014 (see cost evolution in Figure 1.7).

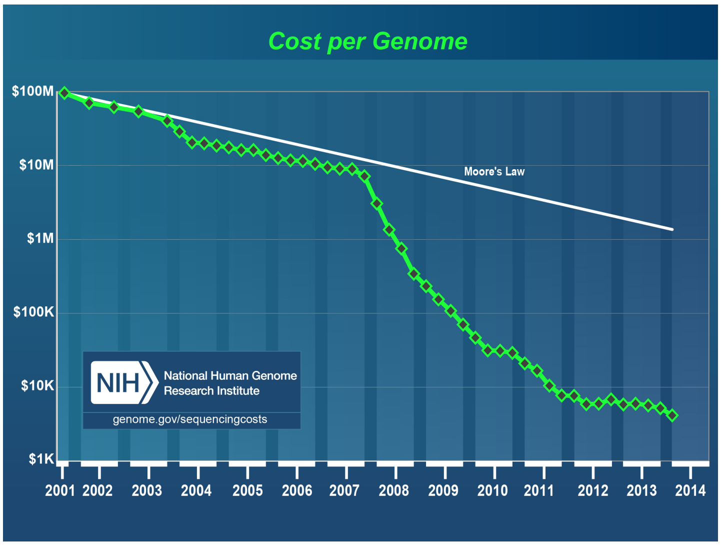


FIGURE 1.7: Cost of sequencing for one whole genome. The sequencing cost has been substantially reduced from early 2008 [National Human Genome Research Institute (NHGRI), 2014].

Originally, Sanger sequencing technology [Sanger et al., 1977] was used to determine genetic sequences but it was very expensive and time-consuming. Although this technology has been improved, it is only useful today for short DNA sequences. More recently, the so-called next-generation sequencing (NGS) technologies have sped up the process achieving a complete human genome in a few hours (days, at most) reducing costs drastically. The ability of these technologies to process millions of bases in parallel has also significantly increased the throughput in sequences.

Three commercial technologies are mainly considered in NGS, namely 454 sequencer (Roche), Illumina Genome Analyzer (formerly Solexa) and SOliD sequencer (Applied Biosystems). These technologies address different chemistry and amplification processes for sequencing. Consequently, they lead to different performances depending on particular metrics: the number of base pairs (bp) for each read sequence fragments (reads), the number of read bases per machine run or the time cost.

Thus, the most recent Roche 454 sequencers achieve 600bp-1,000bp per read, 1Gb per run and 20-24 hour for each run. In the case of Illumina sequencers, shorter reads of 200bp-400bp are obtained but with a total of 10-600 Gb per run. Illumina is today producing over the 90% of all sequencing data. Finally, the SOliD sequencer provides the shortest reads (less than 85 bp) but returning more than 200Gb per run. Both Illumina and SOLiD sequencers could last several days per run. Roche 454 sequencers take advantage of longer reads, though they are the least accurate technology (0.03-0.1% of wrong bases). On another hand, SOLiD achieves most accurate sequences (0.01% of wrong bases) whereas higher throughput and lower costs are obtained with Illumina. Nevertheless, both technologies still have to deal with considerable run times and short reads.

More recent technologies are trying to drastically reduce the run time maintaining the read length and error rate. For instance, Ion Proton has proposed a novel technology with runs of about 2 hours, with read lengths of 200bp and 10Gb per run [Rothberg et al., 2011]. Moreover, the PacBio RS technology has achieved what they called 'real-time' sequencing with reads of 3,000-15,000bp in just 15 minutes. However, this technology must still improve its high error rates. Finally, Oxford's nanopore has announced the commercialization of their promising nanopore sequencing developed in the last years [Rusk, 2013]. The nanopore technology will facilitate the sequencing process even offering a memory key-size unit that can be directly plugged into a laptop. It is also expected that this system could scan the whole genome in only 15 minutes and a very low cost [Ku and Roukos, 2013]. These last technologies are gathered in what is already considered the third generation of NGS.

The success of these technologies has expanded the possibilities of new analyses, i.e. *de novo* sequencing or resequencing. These techniques allow to characterize new genomes with no references or to discover significant variances and SNPs in already known genomes. Furthermore, the study of differential expression and alternative splicing is also addressed with by means of the RNASeq technique. All these analyses directly imply the handling of high amount sequence reads, requiring more efficient assembly and alignment approaches.

However, although NGS is providing many advantages, it has continually caused computational challenges to manage the generated data: more specialized instruments, more computational requirements for analysis and more advanced computers and databases for storage. Consequently, the bioinformatics field is currently becoming indispensable to solve these challenges.

1.4 Bioinformatics

Bioinformatics is considered a multidisciplinary research area at the interface between computational and biological sciences. A variety of definitions can be found in the literature but, according to the U.S. National Institute of Health (NIH), bioinformatics is formally defined as: research, development or application of computational tools and approaches for expanding the use of biological, medical, behavioral or health data, including those to acquire, store, organize, archive, analyze, or visualize such data [Huerta et al., 2000].

This subject differs from a related field known as *computational biology*. Again according to the NIH, computational biology considers the development and application of data-analytical and theoretical methods, mathematical modeling and computational simulation techniques to the study of biological, behavioral, and social systems. Therefore, while bioinformatics focused on the sequence, structural and functional analysis of genes their corresponding products, computational biology encompasses all biological areas involving any kind of computation.

Bioinformatics includes two complementary fields: *(i)* the development of software tools and databases to handle and store biological data and *(ii)* the application of these tools and databases to generate novel biological knowledge [Xiong, 2006]. Three wide areas are mainly taking advantages of these databases and tools: molecular sequences analysis, molecular structural analysis and molecular functional analysis.

The sequence analysis covers sequence alignments, sequence database searching, motif and pattern discovery, reconstruction of evolutionary relationships and genome assembly and comparison. Structural analysis focuses on protein structure prediction, classification and comparison. Finally, functional analysis includes gene expression profiles, protein-protein interactions (PPI), protein subcellular location prediction or metabolic pathways reconstruction. Specifically, this dissertation will be guided on the area of sequence analysis and sequence alignments (see Chapter 2 for details about sequence analysis).

Bioinformatics not only plays a key role for genomic and molecular biology research, but also is having a major relevance on many areas of biotechnology and biomedical sciences. For instance, applications in knowledge-based drug design, forensic DNA analysis or agricultural biotechnology are taking advantage of advances in bioinformatics. There is then no doubts that bioinformatics is a powerful field which is still revolutionizing biological research.

1.5 Bioinformatics Resources and Databases

As previously commented, bioinformatics areas have become even more necessary due to the huge amount of biological information that is being generated by the NGS techniques. These technologies have been growing the available data at an exponential rate. This information is stored and distributed in diverse databases across the world. Consequently, it is essential to know the main databases and sources where the information is provided and how they should adequately be consulted.

Databases in biology present some particular attributes that should be considered in order to retrieve data adequately: highly heterogeneous data, large volume of data, dynamical resources and not standardized information [Dua and Chowriappa, 2012]. Firstly, the complexity of biology has led to the appearance of diverse databases associated to different data types and data schemes: genome sequence databases, gene expression databases, protein sequence databases, protein structure databases, protein-protein databases or other annotation databases. Additionally, these databases hold large amount of data not only considering genomic and proteomic information but also expanding it with graphics, images, experimental measures, etc. Moreover, data are continually being modified, extended and updated with new features and releases (dynamical resources). Finally, biological databases are far from the standardization. The contained information is still ambiguous, with several synonymous terms and data formats which make difficult to interrelate the information and interpret it correctly. This lack of standardization affects to the creation of adequate management applications and data integration [Dua and Chowriappa, 2012].

There is a huge amount of databases in bioinformatics (see a summary in Table 1.2). Following, those more relevant and well-known will be described in detail. In this dissertation, some of these databases have been essential for the development of heterogeneous datasets of features associated to protein sequences.

TABLE 1.2: Summary of several highly used biological databases. Database names, content and URLs to be accessed are shown.

Category	**Content**	**Name**	**Data Source**
Genome	DNA sequences	ENA	http://www.ebi.ac.uk/ena/
		GenBank	https://www.ncbi.nlm.nih.gov/genbank/
	Gene Expression	GEO	http://www.ncbi.nlm.nih.gov/geo/
		Array-Express	https://www.ebi.ac.uk/arrayexpress/
Proteins	Protein sequences and annotation	Uniprot	http://www.uniprot.org/
	Protein families and domains	Pfam	http://pfam.xfam.org/
		InterPro	https://www.ebi.ac.uk/interpro/
	Protein Interactions	DIP	http://dip.doe-mbi.ucla.edu/
		IntAct	http://www.ebi.ac.uk/intact/
	Protein 2^{nd} Structure	DSSP	http://swift.cmbi.ru.nl/gv/dssp/
	Protein 3D Structure	PDB	http://www.rcsb.org/pdb/
	Structure Classification	CATH	http://www.cathdb.info/
Annotation	Molecular Attributes	GO	http://www.geneontology.org/
	Diseases and Genetic Attributes	OMIM	http://www.ncbi.nlm.nih.gov/omim
	Genome Classification and Pathways	KEGG	http://www.genome.jp/kegg/

1.5.1 GenBank

GenBank [Benson et al., 2013] is the NIH genetic sequence database. It collects publicly available DNA sequences for almost 260,000 formally described species. Genbank currently includes almost 1.6×10^{11} base pairs in approximately 1.7×10^{8} sequences (April 2014 release). From 80s to the present, the number of bases in GenBank has doubled approximately every 18 months. Sequences are submitted by individual laboratories and their qualities are reviewed before being published. Besides, GenBank is in continuous collaboration with the ENA database and the DNA DataBank of Japan (DDBJ) to daily exchange DNA data [Nakamura et al., 2013].

Data can be retrieved by using the NCBI Entrez system, which allows to relate data with other genomic and proteomic databases. Thus, Entrez can be applied in GenBank to obtain information about, among others, gene sequences and annotations (*Nucleotide*), gene expression (*GEO*) or gene products (*Protein*). Additionally, searches of sequence similarities can also be performed in GenBank by means of BLAST (see details about BLAST procedure in Chapter 2). The information can be extracted from GenBank through any of these databases in a plain text file by using the following generic URL:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi
?db=<*database*>&id=<*uid*>&rettype=gb

where the <*database*> field must be filled with a specific database in the Entrez system and the <*uid*> corresponds to either a single identifier or a comma-delimited list of identifiers related to the specified database. Thus, a record for the sequence and annotation of a specific gene in GenBank format could be retrieved from the Entrez *Nucleotide* database. In this case, depending where the gene is annotated, the GenBank accession number, the GenInfo (GI) identifier or the sequence reference in RefSeq can be used. For instance, for the human gene *ABL2*, the RefSeq reference (*NM_007314*) or the GI identifier (*GI:209862798*) could be considered to retrieve the full record from:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide
&id=NM_007314&rettype=gb
http://eutils.ncbi.nlm.nih.gov/entrez/eutils/efetch.fcgi?db=nucleotide
&id=209862798&rettype=gb

The previous URLs provide a plain text file in the GenBank format (*.gb* extension) and the information is distributed in several lines, each specified by a header. The main headers in the GenBank file are (see an example in Table 1.3):

- The **LOCUS** is formed by a line including the locus name, the sequence length, the molecule type, a GenBank classification and the date of the last modification. Although the locus name was originally designed to differentiate entries with similar sequences, now it can just refer the accession number. Regarding the GenBank classification, it indicates the division in GenBank to which the gene belongs. GenBank establishes 18 different divisions to classify sequences according to the species: rodent (ROD), primate (PRI), other mammalian (MAM), other vertebrates (VRT), bacterial (BCT), viral (VRL), etc.
- The following lines give a detailed description of each entry. Specifically, the definition of the sequence (**DEFINITION**), the accession and version of the entry (**ACCESSION** and **VERSION**), information about the corresponding organism (**SOURCE**) or the bibliography references (**REFERENCE**) are mainly provided. The *REFERENCE* field is complemented with other auxiliary headers like *AUTHORS*, *TITLE*, *JOURNAL* or *PUBMED*.
- The **FEATURES** field provides some annotations for the genes or gene products and their significant regions. This fields usually includes subheaders to describe each region in the gene, e.g. *source*, *gene*, *exon*, *misc_feature*, etc.
- The last field, **ORIGIN**, may be left blank, may appear as *Unreported* or may give a local pointer to the sequence start. The sequence data begin below this field.

In case the specific gene identifier (GI or RefSeq) is unknown, these identifiers can be previously obtained from the Entrez system by using the gene name. For instance, for the human gene *ABL2*, a XML file with a list of GI identifiers related to the different variants of this gene can be retrieved from:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?
db=nucleotide&term=ABL2+AND+Homo+Sapiens[Organism]

where each field in the search is included in the *term* definition of the URL separated by the operators AND/OR and spaces may be replaced by '+' signs. The type of each field is expressed in brackets (e.g. [Gene+Symbol], [All+Fields], [Organism], etc). The resulting XML file contains the tag <*IdList*> with the list of GI identifiers obtained according to this search.

TABLE 1.3: Summary of the main fields included in a GB file from GenBank. Examples for the ABL2 gene are shown.

Header	**Example**	
LOCUS	LOCUS	*NM_007314 12244 bp mRNA PRI 03-MAY-2014*
DEFINITION	DEFINITION	*Homo sapiens c-abl oncogene 2, non-receptor tyrosine kinase*
ACCESSION	ACCESSION	*NM_007314*
VERSION	VERSION	*NM_007314.3 GI:209862798*
SOURCE	SOURCE	*Homo sapiens (human)*
		...
REFERENCE	REFERENCE	*1 (bases 1 to 12244)*
	AUTHORS	*Bianchi C, Torsello B et al.*
	TITLE	*One isoform of Arg/Abl2...*
	JOURNAL	*Exp.Cell Res. 319(13), (2013)*
	PUBMED	*23707396*
		...
FEATURES	FEATURES	*Location/Qualifiers)*
	source	*1..12244*
		/mol_type="mRNA"
		/chromosome="1"
		/map="1q25.2"
	exon	*1..444*
		/gene="ABL2"
		/gene_synonym="ABLL; ARG"
		/inf="alignment:Splign:1.39.8"
		...
ORIGIN	ORIGIN	
		1 gctcggtggt ttaaagatgg ...
		61 gctcggtggt ttaaagatgg ...
		...
	//	

1.5.2 Gene Expression Omnibus (GEO)

Originally, the Gene Expression Omnibus (GEO) project [Edgar et al., 2002] was designed to store and provide gene expression data from microarrays. More recently, GEO has also distributed other high-throughput functional genomics data, for instance, from next-generation sequencing. This database is maintained by the NCBI and, as previously stated, it is part of the Entrez system. Each record in GEO typically incorporates image data (from microarray), expression data and annotation data. This information can also be analyzed in the latest version of GEO with a web application based on R (GEO2R) [Barrett et al., 2013].

GEO classifies three different kind of records supplied by submitters depending on the information they provide:

1. ***GEO Platform***: Each record includes a summary of the array or sequencer platform with a data table with the array template. A stable and unique GEO accession number ('GPLxxx' format) is assigned to each platform record.
2. ***GEO Sample***: This record describes the conditions of an individual sample, its mean features and how it is manipulated. Each Sample record is assigned a stable and unique GEO accession number ('GSMxxx' format).
3. ***GEO Series***: These records link a group of samples and provide the description of the whole study. Series records are identified by a stable and unique GEO accession number with the format 'GSExxx'.

All these records are freely available in the GEO database to be consulted or downloaded. They can easily be accessed by constructing a URL formatted as follows:

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?acc=<*accession*>
&targ=<*type*>&view=<*info*>&form=<*format*>

where <*accession*> can be any valid GEO accession or list of accessions, i.e. GLPxxx, GSMxxx or GSExxx, <*type*> indicate the type of record of the previous accessions (self, gsm, glp, gse or all), <*info*> determines the required information (brief, quick, data or full) and <*format*> selects the output format (text, html or xml). For instance, a full Series record for a mRNA expression profiling of human immune cell subsets (*GSE28491*) can be consulted as:

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?
acc=GSE28491&targ=self&view=full&form=text

This record is downloaded in a plain text file with the SOFT (*Simple Omnibus Format in Text*) format. The SOFT files can hold both data tables with gene expression and accompanying descriptive information. They usually provide a header to indicate the introduced information. For instance, the SOFT file from the previous URL of the *GSE28491* record includes a section with several descriptive data (*!Series_title, !Series_geo_accession, !Series_contributor*, etc), a section with the GEO Sample accessions considered in this Series record (*!Series_sample_id*) and additional information like the platform specifications or supplementary files, e.g. the raw CEL files of the microarray (*!Series_platform_id, !Series_supplementary_file, etc*). Following, an extract of this SOFT file is presented:

```
^SERIES = GSE28491
!Series_title = mRNA expression profiling of human
               immune cell subsets (HUG)
!Series_geo_accession = GSE28491
!Series_status = Public on Jan 31 2012
!Series_contributor = Florence,,Allantaz
...
!Series_sample_id = GSM705402
!Series_sample_id = GSM705403
...
!Series_supplementary_file = ftp://ftp.ncbi.nlm.nih.gov/
      geo/series/GSE28nnn/GSE28491/suppl/GSE28491_RAW.tar
!Series_platform_id = GPL570
!Series_platform_organism = Homo sapiens
!Series_platform_taxid = 9606
```

Each of the samples in this Series record could then be individually consulted. For instance, the first sample in the *GSE28491* record, *GSM705402*, can be retrieved from:

http://www.ncbi.nlm.nih.gov/geo/query/acc.cgi?
acc=GSM705402&targ=self&view=full&form=text

In this case, a SOFT file is again downloaded with descriptive information but also a table with the expression data of this sample. Additionally, the Series records in which this sample has been included are now specified instead of the GEO samples (*!Sample_series_id* header). Thus, a summary of the analogous SOFT file for the *GSM705402* sample is following shown:

```
^SAMPLE = GSM705402
!Sample_title = B cells rep1 mRNA (HUG)
!Sample_geo_accession = GSM705402
...
!Sample_series_id = GSE28491
!Sample_series_id = GSE28492
...
#ID_REF =
#VALUE = log2 RMA signal intensity
!sample_table_begin
ID_REF VALUE
231211_s_at 5.17940009
1560369_at 4.904921685
235614_at 4.415833251
1552256_a_at 5.266629994
```

These three primary records are also incorporated to two additional upper-level kind of records: GEO DataSets and GEO Profiles. ***GEO DataSets*** is a study-level database which stores descriptions of all original submitter-supplied records (Series record) as well as their corresponding curated datasets. These records are accessed with the accessions 'GDSxxx'. On the other hand, ***GEO Profiles*** is a gene-level database which provides the expression for an individual gene across all Samples in a DataSet.

The GEO DataSets are also downloaded from GEO database in a SOFT file. However, given that they incorporate the expression profiles of several samples and the file can be quite large, it is compressed and available from a FTP server with the generic URL:

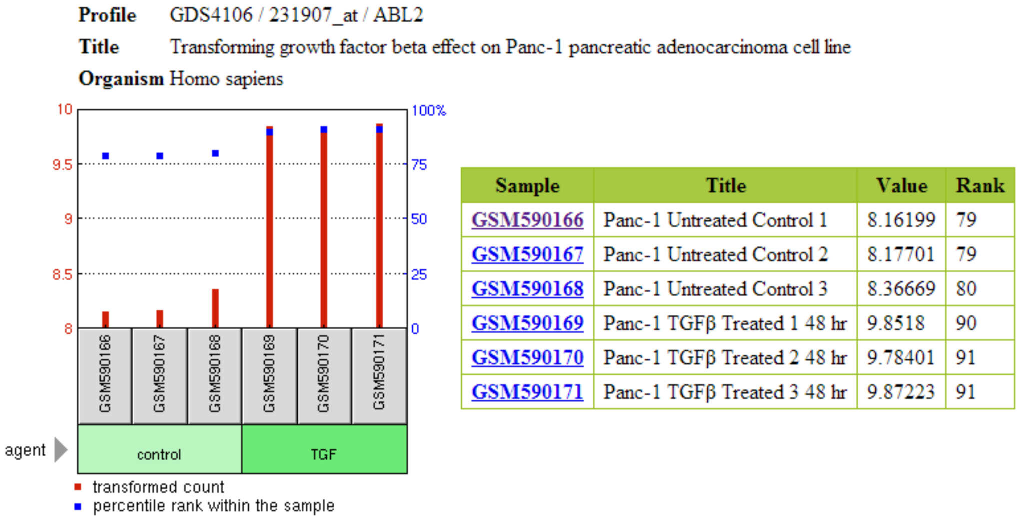
ftp://ftp.ncbi.nlm.nih.gov/geo/datasets/<*range*>/<*accession*>
/soft/<*accession*>_full.soft.gz

where <*range*> determine the directory name created by replacing the last three digits of the accession by "nnn" (e.g. *GDS4nnn* for the *GDS4106* accession).

The main drawback of these records is that they need to previously know the accession of a Sample, a Series or a DataSets to obtain the information. For this reason, GEO Profiles also gives the possibility of searching expression profiles by many other attributes, e.g. keywords, gene symbols, gene names, GenBank accession numbers, etc. This search can be directly performed from the website, but it is also possible to generate an URL with the Entrez system format to retrieve the Profile accessions associated to several conditions. For instance, the list of Profile accessions that provide information about the *ABL2* human gene related to the pancreatic adenocarcinoma can be obtained from the XML file in the URL:

http://eutils.ncbi.nlm.nih.gov/entrez/eutils/esearch.fcgi?db=geoprofiles
&term=abl2[Gene+Symbol]+AND+Homo+Sapiens[Organism]
+AND+Pancreatic+adenocarcinoma[All+Fields]

where, similarly to the search in GenBank (see above), each field in the *term* definition of the URL is separated by the operators AND/OR and spaces are replaced by '+'. A XML file is returned with the list of GEO Profile accessions contained under the tag <*IdList*>. For instance, the first GEO Profile related with the previous GSE28491 Series (79091962 accession) shows the gene expression profile of the *ABL2* gene in a GEO DataSet record related with the pancreatic adenocarcicoma (see results provided by this GEO Profile in Figure 1.8.



Profile GDS4106 / 231907_at / ABL2
Title Transforming growth factor beta effect on Panc-1 pancreatic adenocarcinoma cell line
Organism Homo sapiens

Sample	Title	Value	Rank
GSM590166	Panc-1 Untreated Control 1	8.16199	79
GSM590167	Panc-1 Untreated Control 2	8.17701	79
GSM590168	Panc-1 Untreated Control 3	8.36669	80
GSM590169	Panc-1 TGFβ Treated 1 48 hr	9.8518	90
GSM590170	Panc-1 TGFβ Treated 2 48 hr	9.78401	91
GSM590171	Panc-1 TGFβ Treated 3 48 hr	9.87223	91

FIGURE 1.8: Example of a standard GEO Profile output. This profile has been obtained for the 79091962 accession by searching the *ABL2* human gene related to the pancreatic adenocarcicoma in the database of GEO Profiles. Six samples have been analyzed for this gene (3 control + 3 treated patients).

1.5.3 Uniprot

Uniprot or Uniprot Knowledge Base (UniprotKB) [The UniProt Consortium, 2014] consists of a wide repository of proteins with accurate, consistent and rich annotation. Depending on the origin and quality of resources, the database is divided in two groups. The first one, *Swiss-Prot*, stores manually annotated information from literature or highly accurate computational analysis. The second, *TrEMBL*, provides automatically annotated data. The *Swiss-Prot* information has been carefully assessed whereas *TrEMBL* has not been manually reviewed.

Uniprot also contains other complementary databases with a particular purpose: UniRef, UniParc and UniMES. UniRef [Suzek et al., 2007] gathers groups of sequences (clusters) to speed up sequence similarity searches. On the other hand, UniParc [Leinonen et al., 2004] tracks the sequence identifiers to easily follow modifications in revised or obsolete sequences. Finally, UniMES is a repository specialized in metagenomic and environmental sequences.

The access to a specific entry in Uniprot is relatively easy. The user should

TABLE 1.4: Several alternative and formats in Uniprot to access the information associated to the human protein called *Tyrosine-protein kinase ARG*. Depending on the link, different information can be retrieved from this protein.

Link	**Type**	**Data**
http://www.uniprot.org/uniprot/ABL2_HUMAN http://www.uniprot.org/uniprot/P42684	Web	Full Annotation
http://www.uniprot.org/uniprot/ABL2_HUMAN.txt http://www.uniprot.org/uniprot/P42684.txt	Text	Full Annotation
http://www.uniprot.org/uniprot/ABL2_HUMAN.xml http://www.uniprot.org/uniprot/P42684.xml	XML	Full Annotation
http://www.uniprot.org/uniprot/ABL2_HUMAN.fasta http://www.uniprot.org/uniprot/P42684.fasta	Text	Sequence
http://www.uniprot.org/uniprot/ABL2_HUMAN.gff http://www.uniprot.org/uniprot/P42684.gff	Text	Protein Description

determine the dataset name where the entry is searched (e.g. uniprot, uniref, uniparc, etc) and the entry identifier. The information related to this entry is then located at **http://www.uniprot.org/<*dataset*>/<*identifier*>**.

When working with this database, it is usual to retrieve information in a structured format, namely XML or tab-separated text. For instance, the human *Tyrosine-protein kinase ARG* (corresponding to the *ABL2* gene), which is identifies by the *Swiss-Prot* name *ABL2_HUMAN* or the accession *P42684*, can be indistinctly found in Uniprot in several formats, as presented in Table 1.4. If the full annotation included in the tab-separated text file is considered (2^{nd} row in Table 1.4), data are organized in lines, with the first column indicating the kind of information included in each line (headers). The most useful headers (see summary in Table 1.5) to find the different annotation fields of a protein are:

- The ***ID*** and ***AC*** headers respectively indicate the *Swiss-Prot* name and main accessions of this protein. Several synonymous accessions can be provided in different *AC* lines or separated by ';'. The *ID* header also incorporates the total number of amino acids (AA) in the consulted protein.
- The ***GN*** header indicates the name of the gene(s) that code the consulted protein. Additionally, the GN line can contain other synonymous names or references to, for instance, the ORF name.

TABLE 1.5: Examples of principal headers in Uniprot tab-separated text format for the *ABL2_HUMAN* protein.

Header	Information	Example
ID	Name and sequence length	*ID ABL2_HUMAN Reviewed; 1182 AA.*
AC	Uniprot Accessions	*AC P42684; A0M8X0; B7UEF2;*
GN	Gene Name	*GN Name=ABL2; Synonyms=ABLL, ARG*
DR	Database cross-references	*DR PDB; 2ECD; NMR; -; A=163-268.* *DR GO; GO:0005829; C:cytosol;* *DR Pfam; PF08919; F_actin_bind; 1.*
FT	Data table of features	*FT HELIX 168 170* *FT STRAND 174 177* *FT CONFLICT 343 344 NL -> TI* *FT MOTIF 658 660 Nuclear signal* *FT REGION 694 930 F-actin-binding*
SQ	Sequences in proteins	*SQ SEQUENCE 1182 AA; 128343 MW;* *MGQQVGRVGE APGLQQPQPR GIRGSSAARP* *SGRRRDPAGR TTETGFNIFT QHDHFASCVE*

- The ***DR*** header provides information from external data resources that is related to this entry. For instance, annotation associated to Pfam domains, Gene Ontology terms or PDB structures are labeled under this header (these databases are described in detail below).

- The ***FT*** lines give some supplementary sequence properties. They habitually indicate the kind of feature being presented (special regions and motifs, secondary structure, sequence conflicts or variants, etc) in the second column of the tab-separated text as well as the initial and final amino acids in the sequence for this feature.

- The ***SQ*** header indicates a multi-line section which includes the whole protein sequence. The first line shows the total number of amino acids and other complementary data. After that, the sequence is presented in several lines with 60 amino acids per line, in groups of 10 amino acids (to simplify, only two lines and three groups per line are shown in Table 1.5).

Other possible headers are related with supplementary comments (***CC***), additional descriptive information (***DE***), the publication date (***DT***), organism information (***OS, OC, OG***) or bibliography reference (***RN, RP, RC, RX, RG, RA***), among others.

As appreciated, since Uniprot collects information from a great amount of other databases and resources, it results in an important tool to researches in bioinformatics. Therefore, useful conversion between different types of annotation and features can be performed with this database. Uniprot contains, for example, widely accepted biological ontologies, classifications and other cross-references. Thus, if we are just interested in finding the correspondence of an identifier in a specific database to another one (mapping tool), the following generic URL can be applied:

http://www.uniprot.org/mapping/?from=<*input*>
&to=<*output*>&query=<*queryID*>

were <*input*> and <*output*> are referred to the input and output databases in the conversion, e.g. Uniprot (*ACC+ID*), PDB (*PDB_ID*), GeneBank (*EMBL_ID*), Entrez ID (*P_ENTREZGENEID*), etc. The <*queryID*> identifier must be provided in terms of the <*input*> database. For instance, to determine the PDB identifiers related to the previous *P42684* protein, the mapping list can be retrieved at:

http://www.uniprot.org/mapping/?from=ACC+ID&to=PDB_ID&query=P42684

1.5.4 Pfam

The Pfam database [Finn et al., 2014] is a complete collection of manually curated protein families. Each family is represented by a set of aligned sequences and hidden Markov models (HMMs) to transform these alignments into a scoring system. Families are performed by detecting regions in proteins with a significant degree of similarity which could suggest homology. With this representation, common regions in families can be identified as functional regions, also called domains. Identifying the domains present in a protein can provide insights into the function of that protein. Additionally, related family entries are also gathered together forming clans. This relationship is determined by similarity of sequences and their profile-HMM.

Pfam is divided into two levels depending on the quality of the families: Pfam-A and Pfam-B. Pfam-A families are retrieved from *Pfamseq*, an underlying database of sequences based on the UniprotKB database. These families are highly accurate and carefully curated. Specifically, each family in Pfam-A provides a curated alignment containing the most representative sequences of the family, a profile HMM built from this alignment to score it and an automatically generated alignment containing all the family sequences. On another hand, Pfam-B includes unannotated and lower quality families which are automatically generated from clusters in ADDA database. Although of lower quality, Pfam-B families are used to identify possible functionally conserved regions not found in Pfam-A.

Pfam not only can access information directly from families or domains, but also it allows us to consult the domains associated to clans, proteins or tertiary structures (PDB). Therefore, queries including identifiers for these elements can be retrieved from Pfam website:

http://pfam.xfam.org/family/<*family*>
http://pfam.xfam.org/clan/<*clan*>
http://pfam.xfam.org/protein/<*protein*>
http://pfam.xfam.org/structure/<*pdb*>

with <*family*>, <*clan*>, <*protein*> and <*pdb*> indicating the identifier of the query in the Pfam, Uniprot or PDB databases. Besides using this web format, domains for a specific element can also be computationally retrieved from Pfam in XML format. The XML format is usually more useful when we are interested in handling and locally working with the provided information. For instance, the following generic URL can be used to retrieve the domains of a specific protein in XML format:

http://pfam.xfam.org/protein/<*protein*>?output=xml

Both the protein name or accession in Uniprot can be included in the <*protein*> field , thus helping to associate Pfam with this previously described database. For instance, for the protein *Tyrosine-protein kinase ARG*, the domain annotation can be downloaded from:

http://pfam.xfam.org/protein/P42684?output=xml
http://pfam.xfam.org/protein/ABL2_HUMAN?output=xml

The provided XML file follows a particular structure which makes easier to find and to manage the required information. Specifically, according to the standard XML format, several tags can be found in the downloaded file associated to each Pfam feature. The most important tags, which has been summarized in Table 1.6, are:

- <**entry**>: this tag defines the main features related to the query. In case a protein is consulted, this tag provides attributes about the database from which the protein identifier have been retrieved (*db* attribute), the release of this database (*db_release*) and the accession and the name of the consulted protein (*accession* and *id* attributes, respectively).
- <**sequence**>: this tag introduces the sequence associated to the consulted protein. Several attributes about this sequence are additionally provided: length of the sequence (*length*), codes to verify and assure the integrity of the data (*md5* and *crc65* attributes) and the version of this sequence (*version*).
- <**match**>: these tags are incorporated inside a more general <*matches*> tag. Each of these tags indicates a different domain in the protein. The accession, the identifier and the type of domains are usually added as attributes (*accession*, *id* and *type*, respectively). Each domain is usually identified by an accession with the format 'PFxxxxx' for Pfam-A domains or 'PBxxxxxx' for Pfam-B. These tags can also incorporate child tags (<*location*>) to determine the specific position of the specified domain.

TABLE 1.6: Principal tags in XML files downloaded from Pfam for a specific protein. Examples for the *ABL2_HUMAN* protein are shown.

Tag	Attributes	Example

<**entry**>	entry_type, db, db_release, id, accession	`<entry entry_type="sequence"`
		`db="uniprot" db_release="2012_06"`
		`accession="P42684" id="ABL2_HUMAN">`
		`...`
		`</entry>`

<**sequence**>	length, md5, crc64, version	`<sequence length="1182" md5="4ec88`
		`a661c26c4267a606e83cb51d1b8"`
		`crc64="ED93869BC2B14FAA"`
		`version="1">`
		`  MGQ...HGP`
		`</sequence>`

<**match**>	accession, id, type	`<match accession="PF00017" id="SH2"`
		`type="Pfam-A">`
		`...`
		`</match>`

<**location**>	start, ali_start, end, ali_end, hmm_start, hmm_end, evalue, bitscore	`<location start="173" end="248"`
		`ali_start="173" ali_end="248"`
		`hmm_start="1" hmm_end="77"`
		`evalue="6.2e-21" bitscore="84.20"/>`

- <**location**>: this tag is always nested in a <*match*> tag. It expands the information associated to each domain. Specifically, the initial and final positions in the sequence, the family alignment and the associated HMM are respectively included in the attributes *start*, *end*, *ali_start*, *ali_end*, *hmm_start* and *hmm_end*. Additionally, two quality measures are added: E-value (*evalue*) and the HMM score in bits (*bitscore*). The E-value measures the significance of the Pfam family against Uniprot database whereas the score is related to the quality of the alignment in the family.

1.5.5 IntAct

IntAct [Orchard et al., 2014] provides a freely available, open source database system and analysis tools for molecular interaction data. All interactions are collected by either curated from the literature or from direct user submissions. Published interactions must be previously reviewed and accepted by a senior curator.

This database currently includes almost 450,000 different interactions. These interactions mainly derive from protein-protein interactions (PPI) but also protein-small molecule (including phospholipids), protein-nucleic acid and protein-gene loci interactions are considered. All entries contain a detailed description of the experimental conditions in which the interaction was observed and references to UniprotKB (underlying database). It has also collaborated with the Gene Ontology annotation (GOA) project (see below this database) to annotate the binary pairs in their interactions.

As previous databases, IntAct also allows us to retrieve the information in a tab-separated text file in order to make computationally easier its management. The generic URL which is used in this case is:

http://www.ebi.ac.uk/Tools/webservices/psicquic/intact/
webservices/current/search/interactor/<*accession*>

where <*accession*> can be the identifier or name of a protein in Uniprot. For instance, the interactions that involve the example of the *Tyrosine-protein kinase ARG* (P42684) can be retrieved from:

http://www.ebi.ac.uk/Tools/webservices/psicquic/intact/
webservices/current/search/interactor/P42684

This URL provides a tab-separated text file according to the PSI-MITAB format [Kerrien et al., 2007] with all the interactions in which the protein is involved. Specifically, this PSI-MITAB file includes in separated columns the following information (see summary in Table 1.7):

- The two first columns are the identifiers for the two interacting elements or interactors. These identifiers can be related with the Uniprot database (proteins) or the RefSeq accession (genes). The database from which the identifier is extracted is also indicated in the corresponding column. Therefore, these columns should be in the format *database:accession*.
- The two following columns (3^{rd} and 4^{th}) determine other alternative identifiers for each interactor. Several identifiers can be included separated by '|'. These identifiers maintain the format *database:accession*.
- The 5^{th} and 6^{th} columns provide the alias name and alias type of the interactors. Multiple alias are also separated by '|'. In this case, each alias is specified by the format *database:alias(type)*.
- This column (7^{th}) gives the identifier of the method applied to detect the interaction. This method is usually specified among several terms in the PSI-MI controlled vocabulary ('psi-mi' database) and it has the format *database:identifier(name)*.
- The following two columns (8^{th} and 9^{th}) are related with the publication in which the interaction has been shown. Specifically, the first author and the publication identifier (e.g. in PubMed) is respectively added.
- The taxonomy identifiers for both interactors are presented in the 10^{th} and 11^{st} columns. These identifiers are associated with the organism to which the interactors belong. These fields have the format *taxid:identifier(organism name)*. This field can also be left empty ('-') if it is not relevant.
- The 12^{nd} column determines the type of the obtained interaction. This column has the format *database:identifier(type)*. As the interaction types are taken from the PSI-MI ontology, the database name always is set to 'psi-mi'.

TABLE 1.7: Summary of fields in a PSI-MITAB file provided from the IntAct database. The example column is extracted from the first interaction with the *P42684* protein query.

Column	**Content**	**Example**
1,2	Interactor identifiers	`uniprotkb:P62993` `uniprotkb:P42684`
3,4	Alternative identifiers	`intact:EBI-401755 |uniprotkb:Q63059...` `intact:EBI-1102694|uniprotkb:A0M8X0...`
5,6	Alias names	`uniprotkb:GRB2(gene name)|...` `uniprotkb:ABL2(gene name)|...`
7	Detection method	`psi-mi:"MI:0081"(peptide array)`
8	Author	`Wu et al.(2007)`
9	Publication identifier	`pubmed:17474147|imex:IM-11903|...`
10,11	Taxonomy identifier	`taxid:9606(Homo sapiens)` `taxid:9606(Homo sapiens)`
12	Interaction type	`psi-mi:"MI:0915"(physical association)`
13	Source database	`psi-mi:"MI:0469"(IntAct)`
14	Interaction identifier	`intact:EBI-1963658|imex:IM-11903-1150`
15	Confidence score	`intact-miscore:0.40`

- The 13^{rd} field presents the database (name and identifier) from which the interaction has been retrieved. Multiple source databases can be separated by '|'. This column has the format *database:identifier(source).*
- The interaction identifier according to the previously stated database is presented in the 14^{th} column. This field is represented by *database:identifier* and several identifiers can be concatenated by '|'.
- Finally, the 15^{th} column provides the confidence score for the proposed interaction and, optionally, for other fields like authors, interaction method, etc. The higher score the more confidence is considered.

1.5.6 Protein Data Bank (PDB)

The Protein Data Bank (PDB) [Berman et al., 2000] consists of a large collection of 3D structures from biological molecules, including both proteins and nucleic acids. It includes most organisms such as bacterias, yeast, plants, flies, some animals and human. The structural information is obtained by expert biologists using typical experimental techniques as X-ray crystallography or nuclear magnetic resonance (NMR) spectroscopy.

Each PDB record can be downloaded in a plain text file with the information about how each residue is spatially distributed. This information is represented in coordinates for each atom together with crystallographic structure factors and NMR experimental data. Additionally, PDB also includes other annotation data like molecule's name, primary and secondary structures, references to sequence databases and bibliographic citations. Specifically, each entry in PDB can be downloaded from the generic URL:

http://www.rcsb.org/pdb/downloadFile.do?fileFormat=pdb
&compression=NO&structureId=<*pdbID*>

where <*pdbID*> determines the structure identifier. Each line in the PDB file provides a header to indicate the information presented in such line. A wide number of headers can be found in PDB. A summary of these headers is presented in Table 1.8. Following, the most relevant ones, which are used during this dissertation, are described [wwPDB, 2008]:

- Several headers are applied for general description of each entry. Thus, the first line of the entry (*HEADER*) contains the PDB identifier, its classification and the date of deposition. Also, the description of the experiment used for discovering this structure or the macromolecular contents are respectively included in the headers *TITLE* and *CMPND*. Finally, the *EXPDTA* header determines the technique used to determine the structure.

- Information related to the protein sequence is also incorporated to the PDB file. The sequence itself is stored under several lines with the *SEQRES* header. Additionally, the reference to the database from which the sequence has been obtained is indicated in the *DBREF* header.
- The secondary structure associated with each entry is also annotated in the PDB file. Three standard substructures are identified in independent lines (*SHEET*, *HELIX* and *TURN* headers). It also determines the initial and final amino acids for each substructure and its positions in the sequence. The information is organized in the following order: serial number of the substructure, substructure identifier, number of strand the sheet (only for *SHEET*), initial residue, initial position, final residue, final position and substructure class.
- The structural definition is collected by determining the coordinates of each atom. Specifically, the *MODEL* header specifies the model number in case of multiple structures are associated to a single entry. Therefore, the following lines to a *MODEL* header will contain the *ATOM* header. The *ATOM* lines present the orthogonal coordinates (x, y, z) in Amströngs for each atom of each residue. They also determine the occupancy (alternative conformations in the atoms) and temperature factor for each atom. The eleven fields in the *ATOM* line represent (in this order): atom serial number, atom name, residue, structure chain, residue position, x coordinate, y coordinate, z coordinate, occupancy, temperature factor and charge of the atom.

Knowing the molecular structure is essential to understand its main function. This knowledge is then applied to other studies for health, diseases and drug development. The PDB database provides structures from short DNA regions or small proteins to complex macromolecules like the ribosome.

Currently, several tools, algorithms or databases annotating structural data use PDB information. Thus, it is important to highlight that other described databases such as Uniprot, Pfam or DSSP have already crossed their entries with the corresponding PDB structures [Joosten et al., 2011].

TABLE 1.8: Examples of principal headers in the PDB file for the structure of the *Ubuquitin* protein (1UBI).

Header	Example
HEADER	HEADER CHROMOSOMAL PROTEIN 03-FEB-94 1UBI
TITLE	TITLE SYNTHETIC STRUCTURAL AND BIOLOGICAL
	TITLE 2 STUDIES OF THE UBIQUITIN SYSTEM.
CMPND	CMPND MOL_ID: 1;
	CMPND 2 MOLECULE: UBIQUITIN;
	CMPND 3 CHAIN: A;
	...
EXPDTAD	EXPDTAD X-RAY DIFFRACTION
SEQRES	SEQRES 1 76 MET GLN ILE PHE VAL LYS THR LEU
	SEQRES 2 76 THR GLY LYS THR ILE THR LEU GLU
	...
HELIX	HELIX 1 H1 ILE 23 GLU 34 1
	HELIX 2 H2 LEU 56 TYR 59 5
	...
SHEET	SHEET 1 BET 5 GLY 10 VAL 17 0
	SHEET 2 BET 5 MET 1 THR 7 -1
	...
TURN	TURN 1 T1 THR 7 GLY 10 TYPE I
	TURN 2 T2 GLU 18 ASP 21 TYPE I
	...
MODEL	MODEL 1
ATOM	ATOM 1 N MET A 1 27.34 24.29 2.68 1.0 4.7 N
	ATOM 2 CA MET A 1 26.38 25.36 2.89 1.0 9.5 C
	...

1.5.7 DSSP

The Define Secondary Structure of Proteins (DSSP) program [Kabsch and Sander, 1983] was designed to determine the secondary structure for proteins. It then provides a database with standardized secondary structure assignments for all the protein entries in PDB.

DSSP calculates the secondary structure by means of the information provided by PDB entries about the 3D structure of a protein. The spatial position for each protein is read and the secondary structure is determined by the hydrogen bond energy between atoms. Therefore, DSSP does not predict secondary structures. Instead of that, it just retrieves the information from 3D structures and, thereby, a valid PDB entry must be available to obtain its corresponding secondary structure.

The DSSP program can be downloaded and locally executed. A plain text file (DSSP format) with the secondary structure is returned by the DSSP program from a standard PDB file as input. Anyway, the secondary structures for all the PDB entries can also be directly downloaded via FTP from the generic URL:

ftp://ftp.cmbi.ru.nl/pub/molbio/data/dssp/<*pdbID*>.dssp

where <*pdbID*> specifies the PDB identifier. For instance, the secondary structure of the PDB entry 1UBI (*Ubuquitin* protein) can be downloaded from:

ftp://ftp.cmbi.ru.nl/pub/molbio/data/dssp/1ubi.dssp

This URL provides a plain text file in the DSSP format. This file first contains some descriptive information directly copied from the PDB file: *HEADER, COMPND, TITLE*, etc (see PDB format above). It also determines some statistics extracted from the secondary structure calculation. The second half of the file contains a tab-separated table with the calculated secondary structure information per residue. Each column in this file is described in Table 1.9.

The main column in this DSSP file, *STRUCTURE* (4^{th} column), classifies each amino acid according to the secondary structure to which it belongs. Briefly, the outputs in the first subcolumn can be: α-helix (H), 3_{10}-helix (G), π-helix (I), β-bridge (B), β-strand (E), hydrogen bonded turn (T) and bend (S). Additionally, it also determines supplementary information in other subcolumns: if the residue could start, be inside or end an helix structure ('A>', number with the position in the helix or '<' symbols, respectively), if the α-torsion could be negative or positive ('+' or '-' symbols) or the parallel or anti-parallel class of β-bridges ('b' or 'B' symbols, respectively).

TABLE 1.9: Column headers provided by DSSP for the secondary structure description. These columns are part of the standard DSSP file, together with information retrieved from PDB.

Column	Header	Content
1	`#`	The residue number counted by DSSP.
2	`RESIDUE`	The residue number specified by PDB and chain identifier.
3	`AA`	One letter code for the corresponding amino acid.
4	`STRUCTURE`	Several subcolumns related to the type of secondary structure (one letter code for the structure) and additional labels for each type.
5,6	`BP1` `BP2`	The first and second bridge pair candidate (the following letter indicate the sheet).
7	`ACC`	Accessibility of the residue (surface area expressed in square Angstrom that can be accessed by a water molecule).

8-11	`N-H-->O`
	`O-->H-N`
	`N-H-->O`
	`O-->H-N`
		The H-bond energy with another residue where the current residue is either acceptor or donor. There are two columns for each type to allow bifurcated H-bonds.

12	`TCO`	The cosine of the angle between C=O of the current residue and C=O of previous residue.
13,14	`Kappa` `ALPHA`	Virtual bond angles (bend and dihedral angles) defined by the C-alpha atoms of residues
15,16	`PHI` `PSI`	Peptide backbone torsion angles.

17-19	`X-CA`
	`Y-CA`
	`Z-CA`
		The C-alpha atom coordinates

1.5.8 Gene Ontology (GO)

The Gene Ontology (GO) [Ashburner et al., 2000] is a controlled vocabulary providing a complete set of terms to describe representative attributes and annotation data for gene products and their functionalities. The ontology mainly covers three domains: the parts of the cell and extracellular environments (cellular component), the principal functions of gene products at the molecular level (molecular function) and the procedures and events accomplished by one or more ordered assemblies of molecular functions (biological process). Each of these ontologies is totally specie-independent. They are structured in independent directed acyclic graphs (DAGs) where the nodes define the terms associated to gene product properties and connections represent relationships between terms.

Gene Ontology allows the annotation of gene products, assigning the defined properties (terms) at different level according to the depth of knowledge of each entry. Additionally, each annotated term includes an evidence code to indicate how the annotation to a particular term is supported. Although evidence codes do reflect the type of work or analysis described in the cited reference which supports the GO term, they are not necessarily a classification of types of experiments/analyses. Manually-assigned evidence codes fall into four general categories: experimental (EXP, IDA, IPI, etc), computational analysis (ISS, ISO, SA, etc), author statements (TAS and NAS), and curatorial statements (IC and ND). Additinoally, only one evidence code is dedicated for automatically-assigned terms (not determined by a curator), namely the *Inferred from Electronic Annotation* (IEA) evidence.

Several tools are available in order to consult the annotation of ontological terms in particular gene products, e.g. QuickGO [Binns et al., 2009]. QuickGO is a fast web-based browser that provides a powerful method for searching across all GO database the terms of a specific gene product. Since it has been developed by the collaboration of both GO and Uniprot teams, QuickGO is a helpful tool to retrieve the GO terms associated to a protein extracted from Uniprot. Thus, in addition to the web application, the following generic URL can be applied to retrieve a tab-separated text file with the GO terms:

http://www.ebi.ac.uk/QuickGO/GAnnotation?
protein=<*uniprotID*>&format=tsv

with <*uniprotID*> being the Uniprot identifier of a protein or a list of identifiers for several proteins (separated by ','). For instance, for the protein *Tyrosine-protein kinase ARG*, the domain annotation can be downloaded from:

http://www.ebi.ac.uk/QuickGO/GAnnotation?protein=P42684&format=tsv

TABLE 1.10: Principal column headers in the tab-separated text file provided by QuickGO for the GO annotation of a protein. Some examples included in the *Tyrosine-protein kinase ARG* protein annotation are presented.

HEADER	INFORMATION	EXAMPLES
DB	Database from which the protein is originated.	`UniProtKB`
ID	Protein identifier in the native database.	`P42684`
Splice	Isoform identifier to annotate a splice variant.	`-, 1, 2, ...`
Symbol	Symbol corresponding to the protein ID.	`ABL2`
Taxon	Taxonomic identifier for the consulted species.	`9606`
Qualifier	Possible modification of the interpretation of an annotation.	`colocalizes_with` `contributes_to`
GO ID	Unique and stable identifier of the GO term.	`GO:0005515`
GO Name	GO term name which matches the GO identifier.	`protein binding`
Reference	PubMed reference a GO_REF identifier which contain the data supporting this annotation.	`PMID:12893824` `GO_REF:0000037`
Evidence	Code determining the evidence of the GO terms assignation.	`IPI,IEA, TAS, IDA,...`
With	Additional identifier to support annotations using certain evidence codes.	`UniProtKB:P07203` `InterPro:IPR000719`

Aspect	Sub-ontology to which the GO term belongs	`Function(F)`
		`Component (C)`
		`Process(P)`

Date	Date on which the annotation was made	`20071108`
Source	Database which provides the annotation of this GO term	`Uniprot,Ensembl, Reactome,InterPro, ...`

This tab-separated file introduce information from both, the original database of the protein Uniprot and the GO terms associated to this protein. Each line in the file corresponds to a different GO term annotated for the protein query. The fields (column headers) are summarized in Table 1.10.

1.6 Conclusions

In this chapter, a wide introduction about some important concepts in biology and molecular biology has been presented. These definitions aim to describe the framework where this dissertation is defined and the main challenges that are currently being managed in the bioinformatics field.

The Human Genome Project (HGP) and NGS techonologies have been highlighted as a turning point where computational solutions and tools have become essential due to the volume of biological data retrieved and the need of efficient studies. Consequently, bioinformatics and computational biology have positively been affected and applications for molecular sequence analysis, structural analysis and functional analysis are increasingly being developed.

An important consequence of the retrieval of such volume of data is the necessity of suitable repositories and databases in order to store them. A wide range of databases have been implemented in recent years, some of which have been described in this chapter. These databases are usually specialized in specific kind of data, from genomic or proteomic sequences to complex structural information or molecular annotation.

In following chapters, this dissertation will focus on one particular aspect in bioinformatics: sequence alignments and analysis. This field has directly been involved by the increase of biological data, mainly because both HGP and NGS technologies are primarily providing sequence data. Therefore, as described in Chapter 2, more complex computational tools and algorithms are today essential to obtain efficient analysis and retrieve important properties from DNA and protein sequences.

Chapter 2

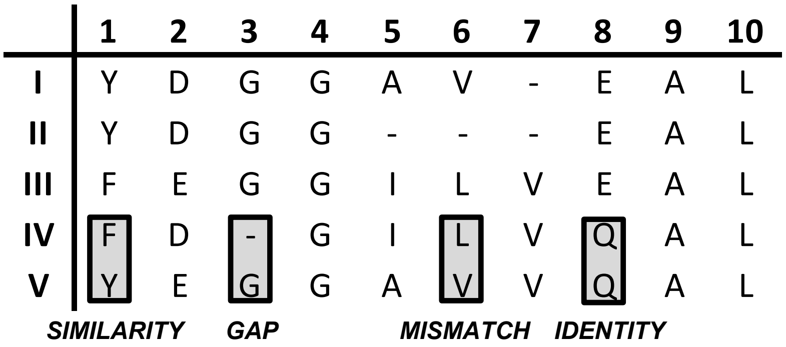
Introduction to Multiple Sequence Alignments

2.1 Background

In biology, a sequence is defined as one-dimensional continuous molecule composed of monomers covalently linked within a biopolymer. These biological sequences could be related to DNA, RNA or protein according to the underlying molecule type. As commented in the previous chapter, DNA molecules are built by chains of nucleotides which are represented by a four-letter alphabet {A, G, C, T} ({A, G, C, U} for RNA molecules). Likewise, proteins are built from sequences of 20 different amino acids, each one represented by one-letter symbol (see Table 1.1 in Chapter 1). Each element of these sequences is habitually named as residue. Therefore, sequence alignments are defined as formal comparisons of these molecular chains.

The main goal when aligning sequences is to identify the highest possible number of conserved regions across the compared sequences. It is well-known that these conserved regions between sequences likely suggest the existence of a shared ancestor (In biology, similarity due to common ancestors is called homology) [Doolittle, 1981]. This comparative task plays an essential role in the discovery of evolutionary relationships among DNA, RNA and protein sequences [Fitch, 1966].

To perform these comparisons, sequences are represented in alignments by their corresponding alphabets (see an example in Figure 2.1). Thus, residues exactly matching in alignments are defined as *identities* whereas different residues that share some physico-chemical features are called *similarities*. Additionally, those aligned residues which are totally different are considered *mismatches*. Similarities and mismatches usually model possible mutations between sequences (more likely with similarities than mismatches). Finally, gaps can be also incorporated in alignments to increase the percentage of common regions aligned. Gaps are usually represented by the '*' or '-' symbols. They represent the biological processes of deletion or insertion (see section 1.2.4 in the previous chapter for details). These definitions are graphically shown in the alignment example of Figure 2.1.



	1	2	3	4	5	6	7	8	9	10
I	Y	D	G	G	A	V	-	E	A	L
II	Y	D	G	G	-	-	-	E	A	L
III	F	E	G	G	I	L	V	E	A	L
IV	F	D	-	G	I	L	V	Q	A	L
V	Y	E	G	G	A	V	V	Q	A	L

FIGURE 2.1: Standard representation of a sequence alignment. Five sequences (I-V) are being aligned in this case.

Different methods are applied to evaluate the quality of an alignment depending on the number and kind of identities, similarities, mismatches and gaps. Generally, matrices are defined to score alignments, thus being called scoring matrices. These matrices contain one row and one column for each symbol in the associated alphabet. Each matrix cell then represents the score obtained by aligning each row and column residues. Therefore, an alignment is evaluated as the sum of scores for all the aligned pairwise residues. Two matrices are traditionally defined according to the provided scores: identity matrices (Figure 2.2-A) or similarity matrices (Figure 2.2-B).

Identity matrices only score positively pairs of residues being identical (identities). On the other hand, similarity matrices score positively identities and similarities whereas gaps and mismatches are penalized. Similarity matrices are used rather than identity matrices because they consider scores according to how residues are evolutionarily related (evolutionary distance). However, these matrices could lead to some noise in the alignment construction. For protein sequences, two sets of similarity matrices have been widely used: Dayhoff's matrices [Dayhoff et al., 1979] and BLOSUM [Henikoff and Henikoff, 1992]. Dayhoff's ones are based on the *Point Accepted Mutation* (PAM) concept. PAM considers the likelihood of a mutation between each two aligned amino acids during a specific evolutionary interval. Scores in these matrices are then calculated as the logarithm of this mutation probability. On another hand, BLOSUM (BLOcks SUbstitution Matrix) is derived from short aligned sequences in the database

(B)

C	12																			
S	0	2																		
T	-2	1	3																	
P	-3	1	0	6																
A	-2	1	1	1	2															
G	-3	1	0	-1	1	5														
N	-4	1	0	-1	0	0	2													
D	-5	0	0	-1	0	1	2	4												
E	-5	0	0	-1	0	0	1	3	4											
Q	-5	-1	-1	0	0	-1	1	2	2	4										
H	-3	-1	-1	0	-1	-2	2	1	1	3	6									
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	6								
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5							
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6						
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5					
L	-6	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	6				
V	-2	-1	0	-1	0	-2	-2	-2	-2	-2	-2	-2	-2	2	4	2	4			
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9		
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	
W	-8	-2	-5	-6	-	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17
	C	**S**	**T**	**P**	**A**	**G**	**N**	**D**	**E**	**Q**	**H**	**R**	**K**	**M**	**I**	**L**	**V**	**F**	**Y**	**W**

(A)

	A	**C**	**G**	**T**
A	1	0	0	0
C	0	1	0	0
G	0	0	1	0
T	0	0	0	1

FIGURE 2.2: (A) Identity matrix for DNA alphabet (A, C, G and T). (B) Dayhoff similarity matrix with PAM 250 (20% of identities). This matrix is built for the 20 amino acids in proteins, grouping them according to their physico-chemical properties.

BLOCKS [Pietrokovski et al., 1996]. This database contains sequence alignments from short conserved regions (blocks) of protein families. Therefore, BLOSUM is able to represent, in a more explicit way than PAM, distant relationships between amino acids. Different BLOSUM matrices can be obtained depending on the percentage of similarity in blocks. For example, BLOSUM82 or BLOSUM62 matrices were built using sequences with more than 82% and 62% of similarity, respectively.

Alignments can also be classified according to the way sequences are aligned. In this case, two different alignments can be differentiated: *global* or *local alignments*. In the global ones, sequences are considered as a whole and alignments try to match the full sequences. These alignments are usually achieved using score matrices with integer scores and they are useful for highly related sequences. In local alignments, sequences are divided into smaller blocks to find better regional alignments. In this case, score matrices usually include real scores [Attwood and Parry-Smith, 2003]. The previously described PAM matrices are applied with global alignments whereas BLOSUM is based on the local ones.

Finally, alignments can also be divided into *pairwise sequence alignments* or *multiple sequence alignments* (MSAs). The earliest tools were designed specifically focused on the alignment of two sequences (pairwise alignments). These tools took advantage of dynamic programming procedures in order to consider all feasible alignment. Each possible alignment is evaluated by using typical scoring matrices like Needleman-Wunsch (global) [Needleman and Wunsch, 1970] or Smith-Waterman (local) [Smith and Waterman, 1981] and the most accurate one is then obtained. However, when more than two sequences were to be aligned (multiple alignment), the high computational cost made unapproachable the use of dynamic programming. Consequently, other approaches like progressive algorithms or consistency-based methods have increasingly been developed to built MSAs (see section 2.2.2 for details).

MSAs have been traditionally applied in other bioinformatic and biomedical tasks like structure modeling [Guex and Peitsch, 1997], functional predictions [Sonnhammer et al., 1998], phylogenetic analysis [Feng and Doolittle, 1987] and sequence database searching [Altschul et al., 1994]. More recently, MSA tools have also been successfully applied to predictions of protein structures [Gelly et al., 2011], estimations of phylogenetic trees [Liu and Warnow, 2014], predictions of mutations [Hicks et al., 2011] or RNA-RNA interactions [Li et al., 2011].

Moreover, the development of novel experimental techniques, such as NGS and high-throughput technologies (see NGS technologies in section 1.3.2), has prompted a great demand of MSAs in the last years. Because these techniques provide mainly new nucleotide sequences and their subsequent protein products, MSA tools usually help to extract biological meanings from such information. Therefore, current MSAs should be capable of dealing with and efficiently analyzing the massive amount of information generated by these former techniques using advanced computational approaches based on well-known artificial intelligence and machine-learning techniques, such as support vector machines (SVMs) [Allahviranloo and Recker, 2013], genetic algorithms [Orobitg et al., 2013] or neural networks [Kukic et al., 2014] . Furthermore, MSA methodologies are taking advantage of NGS and high-throughput experiments by retrieving functional, structural and genomic data from well-known sequences to obtain more accurate alignments in a reasonable time [Kemena and Notredame, 2009]. Taking all these ideas into account, MSAs are becoming one of the most powerful and useful procedures in current molecular biology [de Juan et al., 2013, Li and Homer, 2010].

Subsequently, this dissertation mainly focuses on the development of intelligent computational systems for multiple sequences alignments in proteins. In this chapter, the main specifications and advances in MSAs will be presented. Firstly, the standard procedures to design MSA tools will be described. Next, a comparative study between some of these tools will be performed. Finally, as conclusions for this study, the current challenges in MSA will be addressed together with the objectives proposed in this dissertation in order to solve them.

2.2 Sequence alignment methodologies

In this section, the standard algorithms and procedures designed to align sequences will be described. Some relevant and well-known tools using these algorithms will be introduced to illustrate each procedure. These tools are sometimes referred to as *aligners*.

Classically, alignment algorithms have dealt with the trade-off between speed and accuracy. As previously commented, biological sequences were originally aligned by using dynamic programming techniques which provided the optimal alignment [Needleman and Wunsch, 1970, Smith and Waterman, 1981]. Nevertheless, due to the computational cost of these procedures, other more efficient strategies prevailed: progressive algorithms [Hogeweg and Hesper, 1984] or consistency-based methods [Gotoh, 1990]. Both approaches have also been combined with other relevant computational strategies to obtain more accurate alignments. More recently, other sophisticated tools have even considered heterogeneous data from amino acid sequences (domains, structures or homologies) to increase the quality of the alignments [Armougom et al., 2006, Pei and Grishin, 2007]. Such additional features enrich the alignment information designing more realistic solutions. In the following subsections, these alignment approaches will be described in detail.

2.2.1 Pairwise alignments

2.2.1.1 Classical methods with dynamic programming

The dynamic programming is a classical iterative procedure which aims to build the best final solution of a problem trying to previously solve similar smaller problems [Bellman, 1956]. Particularly in MSAs, dynamic programming individually evaluates each pair of possibly aligned residues to find the best combination and, therefore, the optimal final alignment. A complete example of the dynamic programming procedure in pairwise alignments is shown in Figure 2.3. Given two sequences X and Y, a matrix is first obtained by scoring with a specific score scheme each pair of possible aligned residues in these sequences. In this example, matches (m) are positively scored ($m = 1$) whereas mismatches (mm) and gaps (gp) are penalized ($mm = -1$ and $gp = -2$). Subsequently, an accumulated matrix is formed by scoring each cell from previous neighboring cells. Formally, the score S in the position (i, j) of the accumulated matrix is recursively calculated as:

$$S(i, j) = max \begin{cases} S(i-1, j) + gp \\ S(i-1, j-1) + a \\ S(i, j-1) + gp \end{cases} \tag{2.1}$$

where

$$a = \begin{cases} m & if \quad X(i) == Y(j) \\ mm & otherwise \end{cases} \tag{2.2}$$

with $X(i)$ and $Y(j)$ denoting the residues in the positions i and j for X and Y sequences, respectively. The optimal alignment is finally built by following the best possible path in the accumulated matrix (see Figure 2.3).

Needleman and Wunsch [1970] algorithm was the first optimal method based on dynamic programming to align two sequences. This method focused on global alignments and, therefore, it usually includes large gap extensions between homologous regions in sequences. Later, Smith and Waterman [1981] proposed an algorithm to find local alignments until obtaining an optimal solution. Both methodologies are continually being refined [Rognes, 2011].

However, these procedures are increasingly time-consuming according to the number and the length of sequences. Consequently, they cannot generally be applied for current aligners although they are still incorporated as part of them.

2.2.1.2 Heuristic methods

Heuristic methods are presented as an alternative when massive searches are necessary for sequence databases. These methodologies are called *heuristic* because they make previous assumptions about the final alignment to achieve a speed-up [Soh et al., 2012]. These assumptions are very reasonable when the homology among sequences is strong, but current heuristic methods actually need many more key parameters [Soh et al., 2012].

The most famous heuristic method is the Basic Local Alignment Search Tool (BLAST) [Altschul et al., 1990]. BLAST finds regions of local similarity between sequences. The program is able to compare nucleotide or amino acid sequences

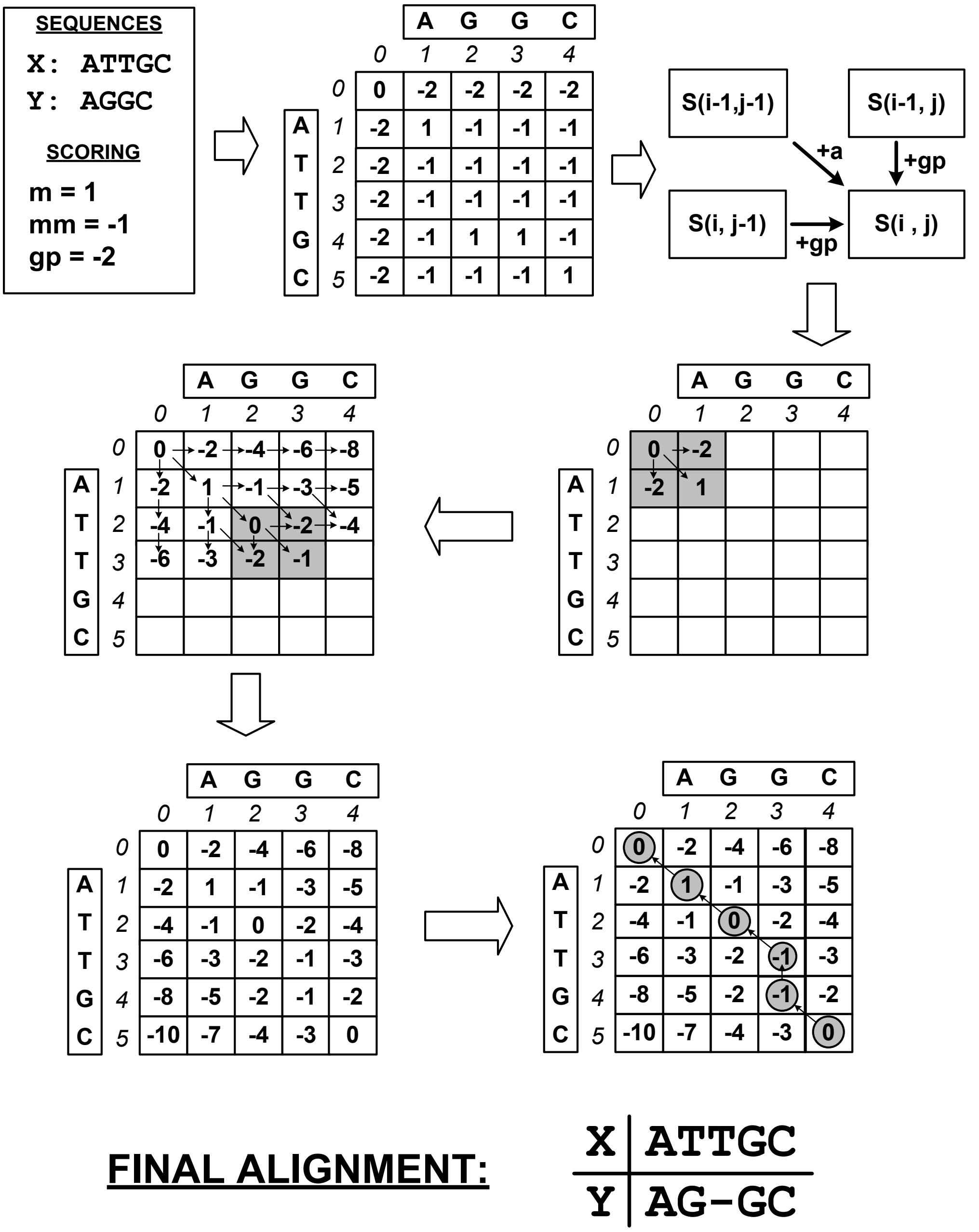

SEQUENCES
X: ATTGC
Y: AGGC
SCORING
m = 1
mm = -1
gp = -2
S(i-1,j-1)
S(i-1, j)
S(i, j-1)
S(i , j)
+a
+gp
+gp
FINAL ALIGNMENT:
X | ATTGC
Y | AG-GC

FIGURE 2.3: Alignment construction using dynamic programming. The scoring matrix is calculated according to the proposed score scheme ($m = 1$, $mm = -1$ and $gp = -2$). Next, the accumulated matrix is performed by adding the maximum score from the three previous neighboring cells according to the Equation 2.1 (+a if maximum score comes from position (i,j) and +gp, otherwise). The arrows show from which cell each score has been obtained. Finally, the best alignment is determined by identifying the back path with the best total scores. When the path moves diagonally a pair of residues are aligned whereas horizontal or vertical movements introduce gaps. This figure has been interpreted from [Xiong, 2006].

to a huge amount of sequences in databases and calculates their statistical significance of matches. BLAST has continually been updated. For instance, the *Position-Specific Iterative* BLAST (PSI-BLAST) [Altschul et al., 1997] provided a adapted version to find more distant sequences for multiple sequence alignments in proteins. PSI-BLAST derives a position-specific scoring matrix (PSSM) to give more accurate scores to MSAs by using standard BLAST. Moreover, Boratyn et al. [2012] have recently developed a more accurate and speed-up protein sequence search called *Domain Enhanced lookup time accelerated* BLAST (DELTA-BLAST). The word size or gap scores are some of the required parameters in BLAST for finding homologies in sequence. The word size represents the length for the initial exact match whereas two gap scores are usually given for opening a gap region and for extending gap regions (penalty scores).

Another analogous method, called FASTA, was the first database similarity search tool developed. Similarly to BLAST (FASTA preceded BLAST in the similarity search field), FASTA finds matches for a shorter blocks of identical residues by using a hashing procedure. FASTA can achieve more sensitivity than BLAST but it is considerably slower [Xiong, 2006]. Moreover, FASTA established a standard file format for the storage of sequences that is still massively used.

2.2.2 Multiple sequence alignments

2.2.2.1 Progressive algorithms

Progressive algorithms are one of the classical procedures to align multiple sequences. Initially, the algorithm performs pairwise alignments for each two sequences, using any scoring matrix based on evolutionary similarity (similarly to dynamic programming). Although matrices like BLOSUM or PAM are also applied, pairwise alignments of two sequences X and Y can be simply scored by the distance between sequences calculated as:

$$D_{pair}(X,Y) = 1 - \frac{M_{XY}}{min(L_X, L_Y)} \tag{2.3}$$

where M_{XY} denotes the number of matches between the sequences X and Y whereas L_X and L_Y define the length of such sequences.

A guide tree built by clustering strategies subsequently then determines the order in which sequences are added to the growing MSA. For each tree node, a pairwise *sequence/sequence*, *sequence/profile* or *profile/profile* alignment is progressively performed [Blackburne and Whelan, 2013]. Note that a profile (A) is defined as a partial alignment previously obtained. The distance in Equation 2.3 can be extended for general pairwise alignments in the guide tree as:

$$D(A,X) = \frac{\sum_{i=1, A_i \neq X}^{N} D_{pair}(A_i, X)}{N-1} \tag{2.4}$$

where A_i denotes each sequence in the profile A previously aligned in the tree, X represents the new sequence to align and N is the total number of sequences. The alignments in the guide tree are performed by typical scoring matrix like PAM or BLOSUM (this scoring matrix is configurable and it depends on the choosen progressive method). This procedure is graphically shown with an example in Figure 2.4.

Progressive algorithms are still widely applied today due to their flexibility in order to be complemented with other techniques. Each tool then allows to configure its own tree computing algorithm or change the scoring matrices and weighting systems. Moreover, progressive approaches achieve high quality when sequences are highly related.

However, the progressive methodologies could sometimes result in inaccurate alignments because of some mistakes produced in initial pairwise alignments with less related sequences or noisy input data. These errors are usually

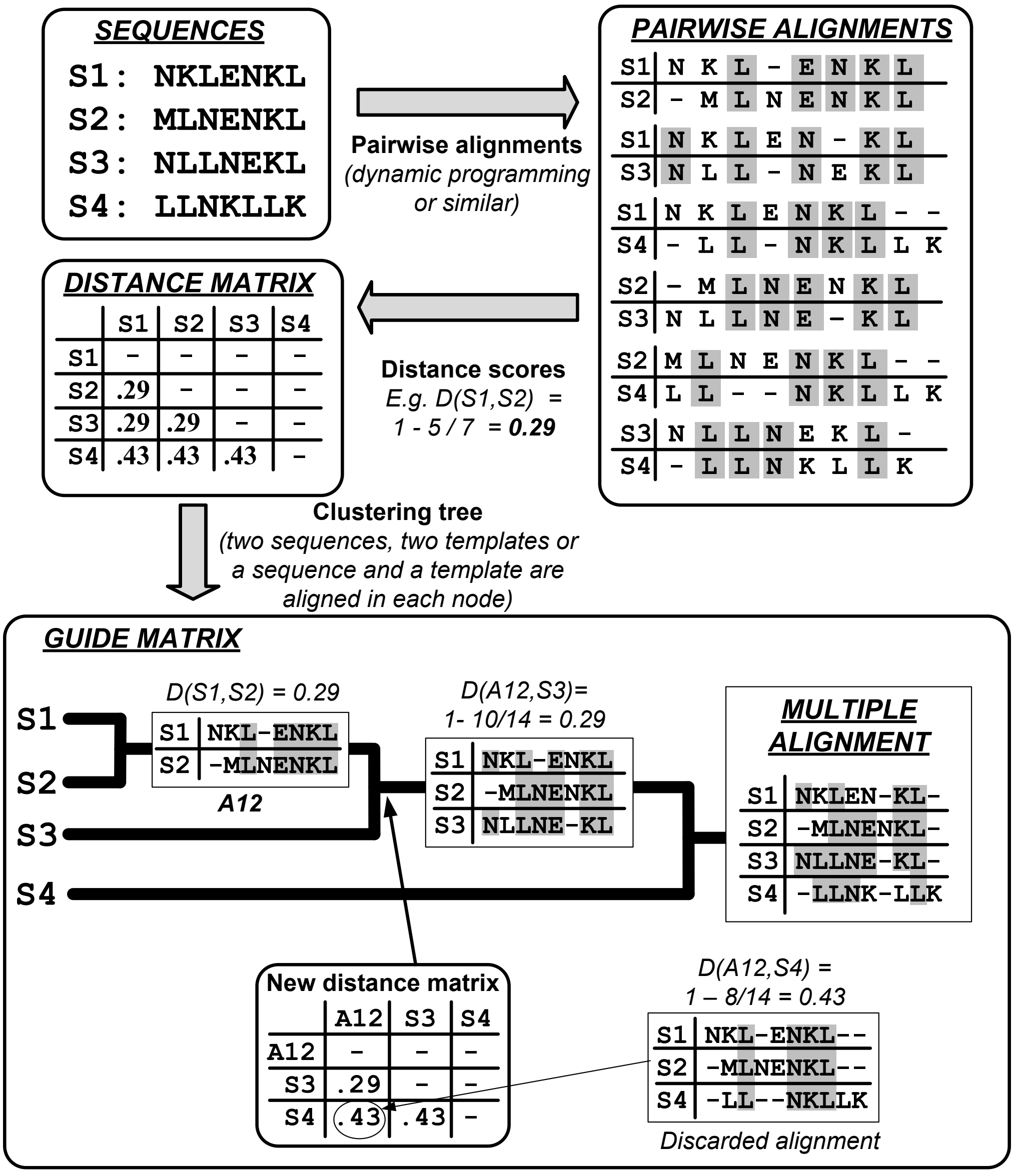

SEQUENCES
S1: NKLENKL
S2: MLNENKL
S3: NLLNEKL
S4: LLNKLLK
Pairwise alignments
(dynamic programming or similar)
PAIRWISE ALIGNMENTS
S1 N K L - E N K L
S2 - M L N E N K L
S1 N K L E N - K L
S3 N L L - N E K L
S1 N K L E N K L - -
S4 - L L - N K L L K
S2 - M L N E N K L
S3 N L L N E - K L
S2 M L N E N K L - -
S4 L L - - N K L L K
S3 N L L N E K L -
S4 - L L N K L L K
DISTANCE MATRIX
S1 S2 S3 S4
S1 – – – –
S2 .29 – – –
S3 .29 .29 – –
S4 .43 .43 .43 –
Distance scores
E.g. D(S1,S2) = 1 - 5 / 7 = 0.29
Clustering tree
(two sequences, two templates or a sequence and a template are aligned in each node)
GUIDE MATRIX
S1
S2
S3
S4
D(S1,S2) = 0.29
S1 NKL-ENKL
S2 -MLNENKL
A12
D(A12,S3)= 1- 10/14 = 0.29
S1 NKL-ENKL
S2 -MLNENKL
S3 NLLNE-KL
MULTIPLE ALIGNMENT
S1 NKLEN-KL-
S2 -MLNENKL-
S3 NLLNE-KL-
S4 -LLNK-LLK
New distance matrix
A12 S3 S4
A12 - - -
S3 .29 - -
S4 .43 .43 -
D(A12,S4) = 1 – 8/14 = 0.43
S1 NKL-ENKL--
S2 -MLNENKL--
S4 -LL--NKLLK
Discarded alignment

FIGURE 2.4: Typical alignment procedure with the progressive algorithms. Four sequences (S1-S4) are aligned in this case. All possible pairwise alignments are first performed with a dynamic programming procedure (for clarity, matches in alignments are shadowed). A distance matrix between each two sequences is then calculated to build the guide tree (the Equation 2.4 is applied in this case). The guide tree to progressively align pairwise sequences, pairwise partial alignments (profiles) or sequence/profile is then designed. The distance matrix is updated in each node considering the previously created profiles (partial alignments). The final multiple sequence alignment is generated following the guide tree until including all the initial sequences.

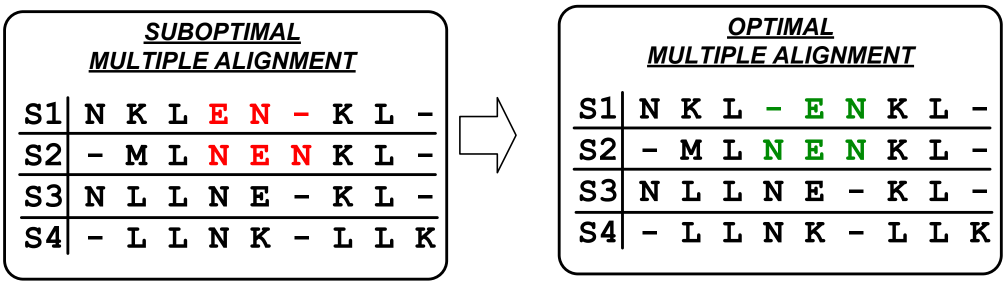


FIGURE 2.5: Comparison between a suboptimal multiple alignment and its corresponding optimal solution. The second alignment is more accurate (see red vs. green residues) than the progressive solution.

propagated by the guide tree and they can cause degraded or suboptimal solutions. In fact, the example in Figure 2.4 provides a suboptimal alignment (see the corresponding optimal alignment in Figure 2.5). In order to minimize these problems, current strategies usually work in two cycles: (i) alignments are obtained by the standard procedure; and (ii) the final alignment is optimized by rebuilding the guide tree.

Following, some well-known progressive algorithms used throughout this dissertation are going to be described in detail:

ClustalW (http://www.clustal.org/) [Thompson et al., 1994] is one of the most widespread progressive algorithms. The procedure shown in Figure 2.4 corresponds to this tool. It designs a tree-computing algorithm to find the final alignment by means of distance scores and a particular gap weighting scheme. The hierarchical clusterings used by ClustalW to allow a fast tree construction are based on the *Unweighted Pair Group Method with Arithmetic Mean* (UPGMA) [Sokal, 1958] or the *Neighbor Joining* (NJ) algorithm [Saitou and Nei, 1987]. ClustalW has recently been replaced by Clustal Omega [Sievers et al., 2011]. This new version uses seeded guide trees and hidden Markov models (HMMs) profile-profile techniques to generate alignments. Clustal Omega achieves more biologically meaningful multiple sequence alignments, even for divergent sequences.

Muscle (http://www.drive5.com/muscle/) [Edgar, 2004] stands out for its high speed to align sequences. It develops a strategy based on three stages: *(i)* a very fast progressive alignment is built with a k-mer counting strategy to estimate the distances; *(ii)* the previously built tree is improved with an iterative algorithm; *(iii)* the alignment is finally refined by a tree-dependent partitioning approach. This final step allows to repeatedly choose different tree branches in order to realign and optimize their corresponding profiles.

Kalign (http://msa.sbc.su.se/) [Lassmann and Sonnhammer, 2005] employs the Wu and Manber [1992] string-matching algorithm to improve distance scores of classical progressive approaches. Kalign can work with more distant and longer sequences, obtaining more accurate alignments in less computational time. For instance, Kalign performs alignments ten times faster than ClustalW.

MAFFT (http://mafft.cbrc.jp/alignment/server/) [Katoh et al., 2002] performs an improved progressive technique to significantly reduce the computational time. MAFFT includes two different procedures: *(i)* the Fast Fourier Transform (FFT) is first applied to speedily identify common homologies between sequences and *(ii)* a novel scoring method to reduce the computational cost and to achieve more accurate alignments, even in distant sequences or sequences with insertions, is implemented. MAFFT has also been improved by the FFT-NS-i algorithm. FFT-NS-i calculates low-quality pairwise distances, constructs a tentative MSA, recalculates refined distances from this MSA, and finally performs a second progressive alignment [Katoh and Toh, 2008]. Although FFT-NS-i could bring faster alignments, it usually implies the decrease of alignment quality.

Reticular Alignment (http://phylogeny-cafe.elte.hu/RetAlign/) [Szabo et al., 2010] or RetAlign is a more recent progressive tool. This program implements a progressive corner-cutting algorithm. This algorithm is based on a network where a set of optimal and suboptimal alignments are represented, instead of the classical scoring matrix. An accurate configuration of that network together with a correct tuning of its parameters produces the identification of the most accurate alignments in the network.

2.2.2.2 Consistency-based methodologies

As previously explained, progressive algorithms could perform suboptimal alignments when sequences are less related (low percentage of identities). Although some progressive solutions have proposed to simultaneously include all sequences instead of pairs, these procedures frequently lead to unapproachable computational costs [Attwood and Parry-Smith, 2003].

Consequently, consistency-based algorithms were designed to simultaneously consider more sequences in a reasonable time [Gotoh, 1990]. The main objective of these methods is to collect in a library information not only from previous pairwise alignments but also from other sequences involved in the multiple alignment. In this way, consistency algorithms estimate if the pairwise alignments are consistent with the final result. This library can then be used to guide a progressive alignment, but avoiding the mistakes that were produced in progressive approaches.

A consistency-based algorithm is usually implemented in several stages (see example in Figure 2.6). Firstly, each possible pairwise alignment is performed by using a fast progressive algorithm (Figure 2.6 (B)). These pairwise alignments are then considered to create the primary library. The primary library assigns a weight to each pair of aligned residues in the pairwise alignment. Specifically, each pair of aligned residues is weighted according to the identities in the pairwise alignment where these aligned residues are found (Figure 2.6 (C)). Formally, given two aligned sequences X and Y, the weight for each pair of aligned residues X_i and Y_i in such sequences is calculated as:

$$W(X_i, Y_j) = \begin{cases} \dfrac{M_{XY}}{min(L_X, L_Y)} \times 100 & if\ X_i, Y_j\ are\ aligned \\ 0 & otherwise \end{cases} \tag{2.5}$$

where M_{XY} denotes the number of matches in the alignment whereas L_X and L_Y are the lengths of sequences X and Y respectively. Following, the weights in the primary library are extended by checking how these pairs of residues are aligned with the remaining sequences. The new weights for the extended library are obtained as:

$$\begin{aligned} W_{ext}(X_i, Y_j) =& W(X_i, Y_j) + min\{W(X_i, Z_{i,1}), W(Y_j, Z_{j,1})\}+ \\ &+ ... + min\{W(X_i, Z_{i,N}), W(Y_j, Z_{j,N})\} \end{aligned} \tag{2.6}$$

where X_i and Y_j are the possible pairs of residues in sequences X and Y and $Z_{i,1}, ...Z_{i,N}$ and $Z_{j,1}, ...Z_{j,N}$ are the residues aligned with X_i and Y_j respectively in the N remaining sequences. The process of designing the extended library is shown in Figure 2.6 (D). For instance, let us consider the case of sequences $S1$ and $S2$. The initial weight associated to a specific pair of aligned residues between these sequences is defined as $W(S1, S2)$ (e.g. first residues 'T' in both sequences). According to Equation 2.5, the weight for this pair of residues is set to 89. Additionally, the weight in the extended library also considers how $S1$ and $S2$ sequences are aligned through $S3$ and $S4$. For sequence $S3$, since both 'T' residues in $S1$ and $S2$ are aligned with the same residue in $S3$, it can be concluded that there is also an alignment of the two 'T' residues through sequence $S3$. Therefore, the minimum of $W(S1, S3)$ and $W(S2, S3)$ is added to the initial weight for this pair of residues (Equation 2.6). Since $W(S1, S3) = 82$ and $W(S2, S3) = 94$, the library weight is increased in 82, giving a total of 171 (89+82). However, not all the remaining sequences provide information to the extended weight. For instance, since the pair of aligned 'T' residues in $S1$ and $S2$ are not aligned through sequence $S4$ (see Figure 2.6 (D)), there is no contribution of $S4$ to the alignment between $S1$ and $S2$. This process is repeated for each pair of residues and each pairwise alignment in the system. Therefore, the weight associated to each pair of aligned residues is the sum of the primary weight and all the remaining weights obtained from the examination of alignments with the other sequences (third sequences).

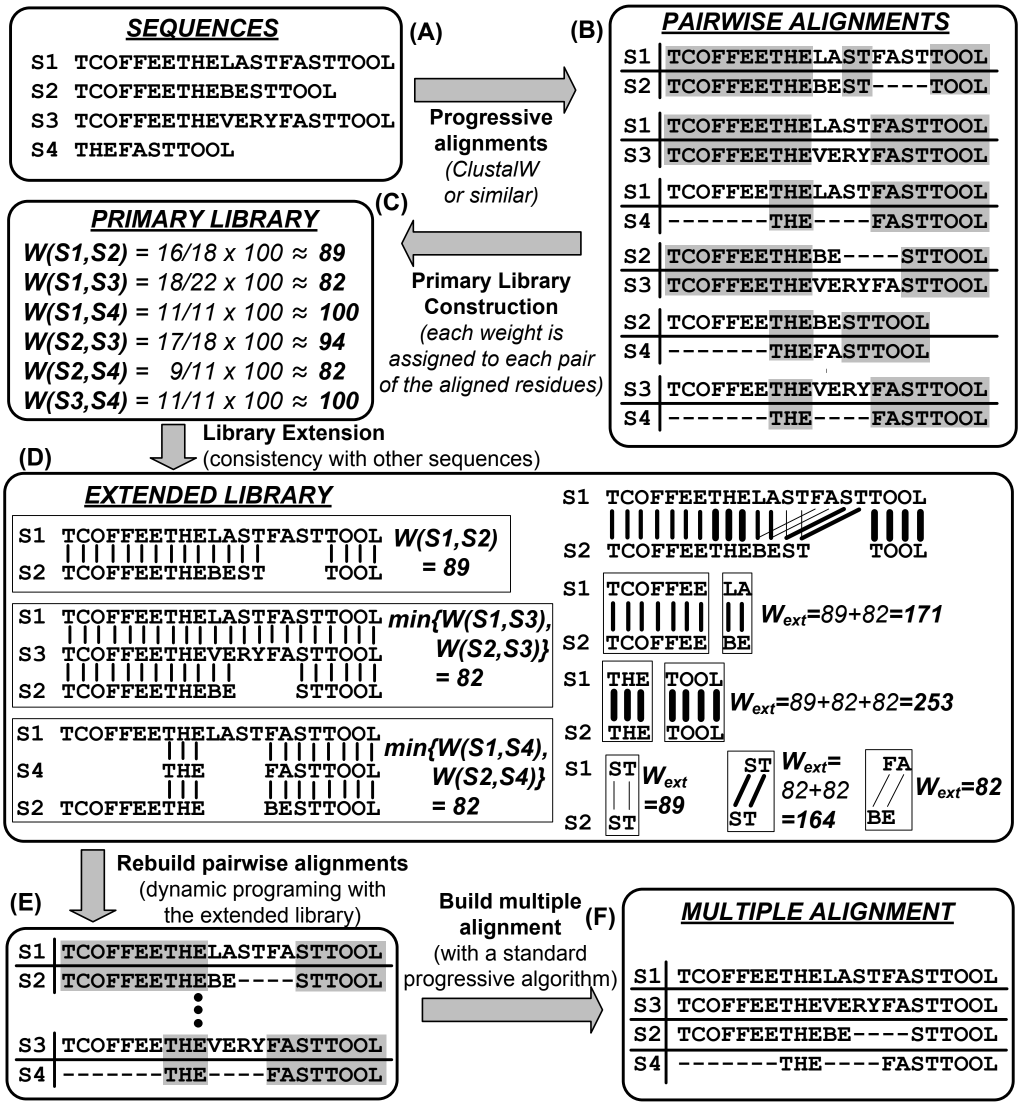

SEQUENCES
S1 TCOFFEETHELASTFASTTOOL
S2 TCOFFEETHEBESTTOOL
S3 TCOFFEETHEVERYFASTTOOL
S4 THEFASTTOOL
(A)
Progressive alignments (ClustalW or similar)
(B)
PAIRWISE ALIGNMENTS
S1 TCOFFEETHELASTFASTTOOL
S2 TCOFFEETHEBEST----TOOL
S1 TCOFFEETHELASTFASTTOOL
S3 TCOFFEETHEVERYFASTTOOL
S1 TCOFFEETHELASTFASTTOOL
S4 -------THE----FASTTOOL
S2 TCOFFEETHEBE----STTOOL
S3 TCOFFEETHEVERYFASTTOOL
S2 TCOFFEETHEBESTTOOL
S4 -------THEFASTTOOL
S3 TCOFFEETHEVERYFASTTOOL
S4 -------THE----FASTTOOL
(C)
PRIMARY LIBRARY
W(S1,S2) = 16/18 x 100 ≈ 89
W(S1,S3) = 18/22 x 100 ≈ 82
W(S1,S4) = 11/11 x 100 ≈ 100
W(S2,S3) = 17/18 x 100 ≈ 94
W(S2,S4) = 9/11 x 100 ≈ 82
W(S3,S4) = 11/11 x 100 ≈ 100
Primary Library Construction (each weight is assigned to each pair of the aligned residues)
Library Extension (consistency with other sequences)
(D)
EXTENDED LIBRARY
S1 TCOFFEETHELASTFASTTOOL
S2 TCOFFEETHEBEST TOOL
W(S1,S2) = 89
S1 TCOFFEETHELASTFASTTOOL
S3 TCOFFEETHEVERYFASTTOOL
S2 TCOFFEETHEBE STTOOL
min{W(S1,S3), W(S2,S3)} = 82
S1 TCOFFEETHELASTFASTTOOL
S4 THE FASTTOOL
S2 TCOFFEETHE BESTTOOL
min{W(S1,S4), W(S2,S4)} = 82
S1 TCOFFEETHELASTFASTTOOL
S2 TCOFFEETHEBEST TOOL
S1 TCOFFEE LA
S2 TCOFFEE BE
W_ext=89+82=171
S1 THE TOOL
S2 THE TOOL
W_ext=89+82+82=253
S1 ST
S2 ST
W_ext =89
ST
ST
W_ext= 82+82 =164
FA
BE
W_ext=82
Rebuild pairwise alignments (dynamic programing with the extended library)
(E)
S1 TCOFFEETHELASTFASTTOOL
S2 TCOFFEETHEBE----STTOOL
S3 TCOFFEETHEVERYFASTTOOL
S4 -------THE----FASTTOOL
Build multiple alignment (with a standard progressive algorithm)
(F)
MULTIPLE ALIGNMENT
S1 TCOFFEETHELASTFASTTOOL
S3 TCOFFEETHEVERYFASTTOOL
S2 TCOFFEETHEBE----STTOOL
S4 -------THE----FASTTOOL

FIGURE 2.6: Typical alignment procedure with consistency-based algorithms (in this case, T-Coffee). (A) Four input sequences to be aligned. (B) All possible pairs of sequences are aligned by using a progressive algorithm (e.g. ClustalW) (identities are shadowed). (C) The primary library is performed by weighting each pair of aligned residues according to the number of identities in the pairwise alignment (see Equation 2.5). (D) The initial library is extended considering the consistency of the previous pairwise alignments through the remaining sequences. For simplicity, only the case of $S1-S2$ pairwise alignment is shown. Here, it is checked if each pair of aligned residues is also aligned through $S3$ and $S4$. In these cases, the weights in the library for these pairs of residues are increased (see Equation 2.6) (see explanation in text for detail). Lines in the alignments indicate the degree of consistency that each pair of residues achieves (the more consistency, the wider lines). (E) The pairwise alignments are rebuilt with dynamic programming using the extended library weights for the scoring matrix. Thus, the possible mistakes in the previous progressive method are avoided by introducing the consistency with other sequences. (F) The final multiple alignment is performed with a standard progressive algorithm but including the consistent pairwise alignment previously rebuilt.

Once the library extension is complete for all the pairwise alignments, the weights in the library are applied to rebuild them using conventional dynamic programming (see Figure 2.6 (E)). In this case, the scoring matrix in dynamic programming is determined by the weight that each pair of residues has assigned in the library. Therefore, pairwise alignments are now consistent with the final alignment. In fact, the $S1$-$S2$ alignment has been corrected after using the extended library. At last, this final alignment is carried out with an standard progressive alignment (Figure 2.6 (F)).

This standard consistency-based procedure can be adapted depending on the applied scheme. More recent solutions are performing other more complex algorithms to score or penalize better each pair of residues. Thus, libraries in consistency-based algorithms can be modified and improved to obtain more efficient alignments.

Generally, the consistency solutions provides more accurate alignments than progressive algorithms, even with evolutionarily distant sequences. Since consistency methods consider a library to store alignment scores, the methodology applied to pairwise alignments and the final multiple alignment are totally independent. Nevertheless, consistency approaches produce a higher computational cost than progressive ones. N times more CPU is usually required (with N being the number of sequences).

Some consistency-based tools used during this dissertation are more exhaustively described following:

T-Coffee (http://www.tcoffee.org/) [Notredame et al., 2000] is considered the reference in consistency-based methods and it follows the standard procedure. In fact, the example in Figure 2.6 was performed with T-Coffee. In the first stage, T-Coffee obtains the pairwise alignments by combining two fast progressive approaches: ClustalW [Thompson et al., 1994] to provide global alignments and Lalign [Huang and Miller, 1991] to provide local information. These alignments from both approaches are incorporated to the primary library with a simple process of addition (adding the weights for those pair of residues aligned in both cases). More recently, T-Coffee has also included other more complex algorithms for the pairwise alignments like 'Proba_Pair' which is based on hidden Markov models from ProbCons (see below ProbCons description). Additionally, T-Coffee gives also the possibility of selecting which progressive algorithm is used for the final progressive algorithm (Mafft, ClustalW, Kalign, etc). Moreover, T-Coffee implements novel score schemes for the library weights, even based on additional properties like structures or homologies [Di Tommaso et al., 2011].

ProbCons (http://probcons.stanford.edu/) [Do et al., 2005] performs another well-known consistency-based algorithm. This approach takes advantage of hidden Markov models (HMMs) to optimize the classical scoring schemes in the library. ProbCons proposes a bi-phasic scheme to penalize gaps and mismatches in alignments. This scheme reports significant improvements in CPU time and alignment accuracy compared with the classical consistency procedure. Generally, methods based on HMM profiles like ProbCons also outperform other alignment methods in terms of the structural superposition quality [Kemena and Notredame, 2009].

Fast Statistical Alignment (FSA) (http://fsa.sourceforge.net/) [Bradley et al., 2009] proposes a statistical analysis framework to build the pairwise alignments. FSA estimates gap and substitution parameters which model the insertion and deletion processes in sequences through a paired hidden Markov model (HMM). This estimation is based on the sequence annealing algorithm by Schwartz and Pachter [2007] for constructing a multiple alignment from pairwise comparisons. Thus, FSA reduces the computational cost, even for a huge amount of sequences. However, the alignment accuracy is excessively affected owing to the excessive number of included gaps.

2.2.2.3 Algorithms with additional features

Methodologies that have been described before only used sequence information to perform alignments. However, neither progressive nor consistency-based algorithms were able to obtain optimal alignments when sequences are weakly related (less than 20% of identity) [Kemena and Notredame, 2009]. Therefore, sequence data is not considered enough to obtain high quality alignments in these cases.

Complementary information is then necessary to efficiently solve these kind of alignments. Consequently, recent MSA tools are increasingly developing algorithms which retrieve more biological data for alignments. Moreover, given that genomic and proteomic databases are rapidly increasing their resources due to emerging techniques in NGS and high-throughput experiments, they can provide useful information for this purpose.

Information related to homologies, domains, motifs or secondary and three-dimensional structures is usually included in novel alignment algorithms to achieve more accurate solutions. Nevertheless, these solutions usually entail an excessive consumed time and the improvements are just relevant in very specific cases with extremely distant sequences. Furthermore, though these methods can be run when some of the additional data are unavailable or unknown, they could provide inefficient alignments.

Following, some of these algorithms including new resources that are considered throughout this thesis are described in detail:

3D-Coffee [O'Sullivan et al., 2004] **and Expresso** [Armougom et al., 2006] are two packages integrated in the T-Coffee environment (http://www.tcoffee.org/). They introduce structural information to the standard consistency-based T-Coffee. This information is retrieved from the structural templates returned by the PDB database [Berman et al., 2000]. 3D-Coffee and Expresso differ in the origin of templates: while Expresso estimates the structural templates associated to the sequences, 3D-Coffee templates are specified by the user, allowing to add more accurate or not published templates (including models and handmade structures).

These tools include two novel pairwise comparisons to be considered for the library design: *sequence vs structure* and *structure vs structure*. Firstly, a threading method called FUGUE [Shi et al., 2001] predicts if a structure can be associated to a sequence. FUGUE classifies sequences into 64 different structural profiles depending on the main and secondary chain of the molecular structure, the solvent accessibility or the hydrogen bonding status. This algorithm retrieves the information from structural profiles provided by homologous alignments in the HOMSTRAD database [Mizuguchi et al., 1998].

In the second comparison, each two structures associated to the pairwise sequences are by default compared through the classical structure alignment protein (SAP) algorithm [Taylor and Orengo, 1989]. SAP permits fast structure alignments by avoiding the structural superposition. Specifically, each sequence is described by this algorithm according to the spatial distance from every particular residue (center) to the remaining residues in the sequence. The distances are calculated by using the spatial position of each residue which is extracted from PDB templates. These distances are considered invariant structural environments and they are able to accurately align sequences. Therefore, SAP performs several scoring matrices by calculating the distances between residues of two sequences centering them in two particular residues. These matrices are subsequently accumulated in a substitution matrix. Finally, both sequences are aligned with a dynamic programming procedure but considering scores in the substitution matrix, which provide information about the spatial distance between sequences and, therefore, the similarity of their structures. This procedure is briefly described in Figure 2.7.

The obtained pairwise alignment by applying structural information are incorporated to the consistency-based library of standard T-Coffee. Therefore, the extended library in the consistency-based process not only includes information about identities between each two sequences but also their similarity in a structural level.

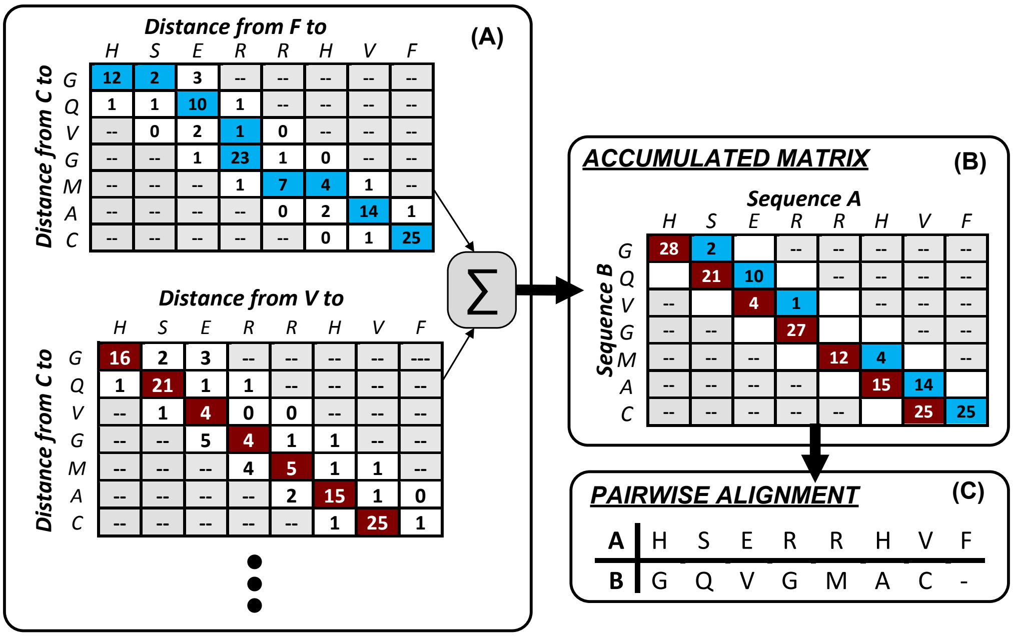


FIGURE 2.7: Pairwise alignment based on structural information with SAP algorithm (Scheme adapted from Taylor and Orengo [1989]). (A) Scoring matrices based on the distance among residues are created, centring each sequence in a specific residue. In this case, 56 scoring matrices should be calculated according to the 56 possible pairs of residues (for simplicity, only matrices centring in the residue pairs F-C and V-C are shown). (B) The optimal pathways from scoring matrices (colored cells) are accumulated in a substitution matrix. (C) The final alignment is rebuilt from the accumulated matrix according to the best final pathway.

Promals (prodata.swmed.edu/promals/) [Pei and Grishin, 2007] is another well-known tool including additional data. In this case, the algorithm retrieves homological data, combining sequences and homologies in profiles through hidden Markov models (HMMs). The complete procedure is schematically shown in Figure 2.8. First, Promals classifies similar sequences (more than 60% of identities) in groups, pre-aligning each group with a fast progressive algorithm (groups *A*,*B*,*C* and *D* in Figure 2.8). A representative sequence (usually, the longest one) is then selected from each group (R_A, R_B, R_C and R_D, respectively). For each representative sequence, homological regions against other sequences in the UNIREF90 database [Suzek et al., 2007] are searched with PSI-BLAST [Altschul et al., 1997]. Additionally, the secondary structure of each representative sequence and its previously determined homologies is estimated applying

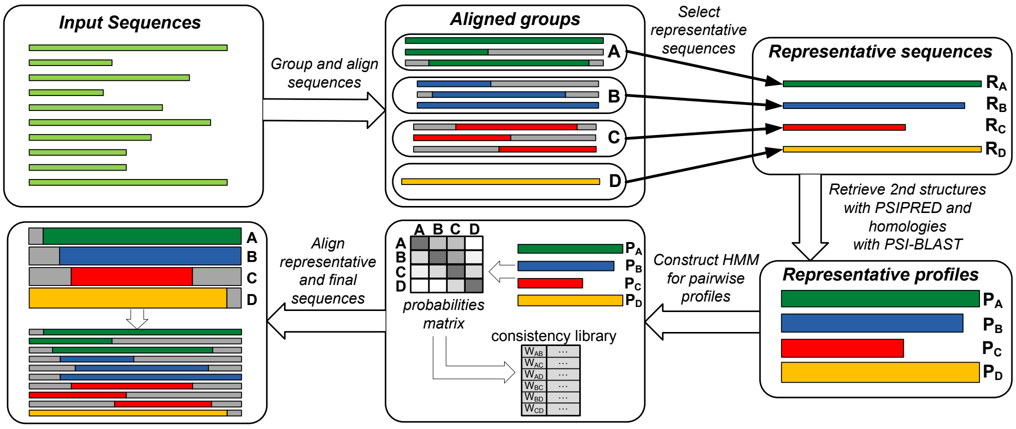

Input Sequences
Group and align sequences
Aligned groups
A
B
C
D
Select representative sequences
Representative sequences
R_A
R_B
R_C
R_D
Retrieve 2nd structures with PSIPRED and homologies with PSI-BLAST
Representative profiles
P_A
P_B
P_C
P_D
Construct HMM for pairwise profiles
A B C D
probabilities matrix
consistency library
W_{AB} ...
W_{AC} ...
W_{AD} ...
W_{BC} ...
W_{BD} ...
W_{CD} ...
Align representative and final sequences
A
B
C
D

FIGURE 2.8: Promals algorithm is graphically represented. Sequences are firstly aligned by groups (>60% of identities), selecting one representative sequence from each group. Homological information and secondary structure predictions are retrieved respectively from PSI-BLAST and PSIPRED for each representative sequence. This information is then incorporated to profiles for each sequence. Each pair of profiles are compared by designing a profile-profile HMM to determine the probability of aligning these profiles according to their homologies and secondary structures. This probabilities is incorporated to the consistency library. Finally, representative sequences and their groups are aligned. This diagram has been interpreted from Pei and Grishin [2007].

PSIPRED [Jones, 1999].

Following, a profile is derived for each representative sequence including the PSI-BLAST alignment (which defines homological regions in the sequences) and the secondary structure predictions (profiles are denoted as P_A, P_B, P_C and P_D in 2.8). A profile-profile HMM is then performed for each two representative sequences to determine the probabilities of homological matches between each two representative sequences (probability matrices). These matrices are used to determine the weights in the consistency library, which is applied to determine the pairwise alignment between each two representative sequences. Thus, the information associated to homologies in the representative sequences is incorporated to the consistency-based method. Finally, the representative sequences and their corresponding sequence groups are aligned with the consistency-based method.

This tool achieves significant improvements in alignments at the expense of an excessive computational cost. Running PSI-BLAST and PSIPRED and the transformation of HMMs to consistent weights are the most time-consuming phases. Later, Promals has been also extended with the incorporation of profiles with structural information creating the Promals3D tool [Pei et al., 2008].

2.3 Multiple sequence alignment benchmarks

Several benchmarks have been designed to allow the standardization of different MSA tools. These benchmarks consist of a heterogeneous dataset of sequences specially gathered to be aligned. They also provide the optimal alignments that should be obtained for the provided dataset. These optimal alignments are usually known as *references* or *gold standard* alignments. Therefore, these references can be compared against those obtained by standard MSA tools in order to determine the quality of their alignments. Following, some of the most used benchmarks will be explained in detail.

2.3.1 Oxbench

Oxbench [Raghava et al., 2003] provides an environment with several families of sequences together with their reference alignments. Oxbench also incorporates an evaluation software to assess the accuracy in multiple sequence alignments. The dataset is built according to the domain families retrieved from protein structural domains in the 3Dee database [Siddiqui et al., 2001]. The main Oxbench dataset, also called *Master dataset*, is composed of 672 families of sequences. All sequences in the Master dataset contains known three-dimensional structures. These families are also divided into subsets depending on the type of alignment analysis: *(i)* subset for pairwise alignments (273 families); *(ii)* subset of multiple alignments (399 families); and *(iii)* subset for short alignments (590 families). Note that several of these subsets could contain common families.

2.3.2 Prefab

Prefab (Protein REFerence Alignment Benchmark) [Edgar, 2004] includes a collection of more than 1900 alignments. In this benchmark, two unrelated proteins are first aligned by an structural method. Subsequently, each of these sequences is used to query a database by PSI-BLAST, retrieving high related sequences. The two targets and the resulting sequences are then combined in a multiple sequence alignment in order to build the reference alignments. The reference quality is assessed on the original pair of sequence, by comparing with their structural alignment. Prefab also contains an evaluation software (Q score) to calculate the accuracy of other MSA tools against its reference alignments.

2.3.3 BAliBASE

BAliBASE [Thompson et al., 1999] is one of the most widespread alignment benchmark. This benchmark defines 218 sets of sequences that are properly prepared to be aligned by MSA tools. These sequences have been carefully extracted from PDB [Berman et al., 2000], though not all sequences have annotated structures. These sequences were organized in six subset according to their families and similarities (see Table 2.1):

- **RV11** is formed by equidistant sequences, where sequences have less than 20% of identities and less than 35 insertions. It includes all protein families with more than 4 available structures annotated in the PDB database. This group contains 38 sets of sequences.
- **RV12** subset includes families not included in the previous group with at least 4 equidistant sequences and identity percentages between 20% and 40% (also excluding large insertions). This subset is formed by 44 sets of sequences.
- **RV20** subset considers families including sequences with more than 40% of identities and at least one known structure but with a highly divergent sequence (*orphan* sequence). This group consists of 41 sets of sequences.

TABLE 2.1: Subsets of sequences provided by BAliBASE benchmark

Subset	Number of sets	Identity percentage
RV11	38	<20%
RV12	44	20%-40%
RV20	41	>40%
RV30	30	>40%
RV40	49	>20%
RV50	16	>20%

- **RV30** selects sequences from different subfamilies that share more than 40% of identity between all their sequences but less than 25% between different subfamilies. RV30 contains a total of 30 sets of sequences.
- **RV40** sequences share more than 20% of identity but containing large terminal insertions. RV40 is the larger subset with 49 sets of sequences.
- **RV50** is formed by sequences that share more than 20% of identity but containing a large amount of internal insertions. This final group is composed by 16 sets of sequences.

This subset classification plays a key role to study the differences in terms of quality for MSA tools, specially when less evolutionarily related sequences are aligned. Moreover, BAliBASE also provides a set of handmade optimal alignments (references) for its sets of sequences. A specific score (BAliscore) is then defined to evaluate the obtained alignments against these references. This score represents the number of coincidences between the provided alignment and the reference one (typical sum-of-pairs or SP score). BAliscore is habitually used to perform the accuracy of MSA tools. For all these reasons, BAliBASE benchmark and BAliscore will be widely employed in this dissertation.

2.4 Comparison of MSA methodologies

As previously described, each MSA tool or aligner proposes a solution based on particular conditions or certain features (see Section 2.2.2 for details). Consequently, biologists and researchers do not still agree with a generally accepted solution. Although some benchmarks have been developed trying to unify criteria in the selection of the most suitable way to align sequences, this is currently an open issue and there is not just one acceptable tool [Ortuño et al., 2011].

To confirm this assumption, the 218 BAliBASE sets of sequences explained above are applied here for the analysis of different MSA methodologies. These sequences are aligned by using several tools in order to compare the obtained alignments against the references provided by BAliBASE. The ten aligners previously described according to three different approaches have been considered: ClustalW, Muscle, Kalign, MAFFT and RetAlign (Progressive algorithms); T-Coffee, ProbCons and FSA (Consistency-based methods); and 3D-Coffee and Promals (Algorithm with additional features). The whole dataset then contains 2180 alignments (218 sets of sequences aligned by 10 different methods). All the aligners are executed with their default configurations in a Ubuntu 12.04 Linux machine with Intel Core i5-2410M @ 2.3GHz, 4GB RAM.

The BAliscore is then computed for each alignment in order to evaluate its quality. Therefore, higher BAliscore values determine more accurate alignments, since they are more similar to the reference BAliBASE alignments. A BAliscore value of 1 indicates that the obtained alignment is identical to the reference (the best possible alignment). Thus, the Figure 2.9 depicts the average Baliscore obtained by each individual tool in every subset of BAliBASE. Additionally, alignments are also compared in terms of their computational cost. In this case, MSA tools are gathered according to the used alignment approach (Figure 2.10).

From Figure 2.9, it is noted that progressive methods (blue colors) are generally the least accurate ones, together with the consistency-based algorithm FSA. The two worst aligners, ClustalW and FSA, also show a strong dependence on the kind of aligned sequences (high variability depending on the BAliBASE subset). Moreover, algorithms with additional data (red colors) achieve significantly better accuracies (BAliscore values) when sequences are evolutionarily more distant (RV11). However, these differences are slightly appreciated against consistency-based methods like T-Coffee or ProbCons when sequences are highly related (remaining subset). Consequently, it can be suggested that

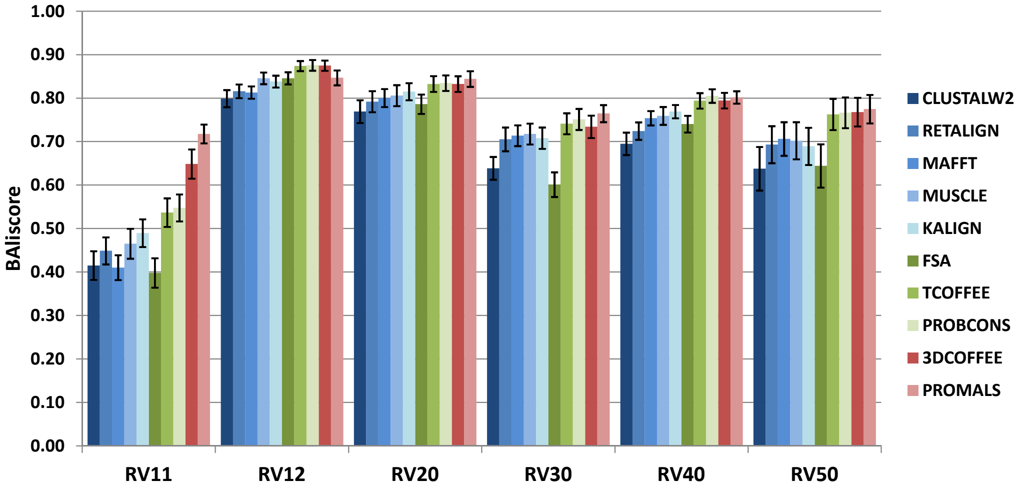


FIGURE 2.9: Average BAliscore values together with standard errors obtained for each MSA tool in every BAliBASE subset. Progressive approaches, consistency-based algorithms and algorithms with additional data are highlighted in blue, green and red color schemes, respectively.

more complex algorithms are only useful in some special situations. It is also observed that not a single method reaches the 90% of accuracy and only particular cases in RV12 and RV20 exceed the 80%, in average. These accuracies could currently be insufficient for some alignments.

To assess these results, the BAliscore values have been statistically analyzed for the ten aligners through a non-parametric test called Wilcoxon signed-rank test [Wilcoxon, 1945]. The Wilcoxon test provides pairwise comparisons between each two aligners in order to validate if there are significant different between them. If the obtained p-values were lower than a significance level, the null hypothesis of methods being statistically identical could be rejected and, therefore, the significant differences are proved. In this case, the significance level is set to 0.05. Additionally, the Wilcoxon test provides the *Z-Score* measure, which represents the distance between the two compared methods. Formally, the *Z-Score* is calculated as:

$$z = \frac{|\sum_{i=1}^{N} S(x_{2,i}, x_{1,i}) \cdot R_i| - 0.5}{\sigma} \tag{2.7}$$

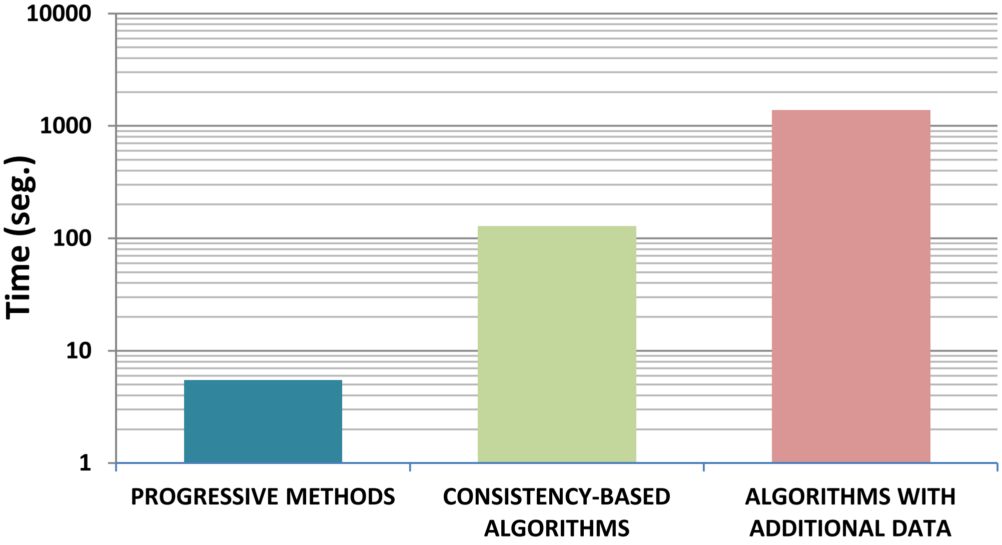


FIGURE 2.10: Average time per alignment used by the different alignment approaches (progressive methods, consistency-based algorithms and approaches with additional data) to align the BAliBASE dataset (218 sets of sequences). Note that time is represented in logarithmic scale.

where N denotes the number of samples with different values between the two methods ($x_{2,i} - x_{1,i} \neq 0$), being $x_{2,i}$ and $x_{1,i}$ the values in the *i-th* sample. R_i defines the rank of the sample *i-th* among the N samples ordered according to the differences between the values of the two compared methods (starting with the smallest difference). Finally, σ and the sign function S are calculated as:

$$\sigma = \sqrt{\frac{N(N+1)(2N+1)}{6}} \tag{2.8}$$

$$S(x_{2,i}, x_{1,i}) = \begin{cases} 1 & if \quad x_{2,i} > x_{1,i} \\ -1 & if \quad x_{2,i} < x_{1,i} \\ 0 & if \quad x_{2,i} = x_{1,i} \end{cases} \tag{2.9}$$

The Wilcoxon test results for this study are depicted in Table 2.2. The *Z-Score* column shows here information about which aligner is forming better alignments. A positive *Z-Score* indicates that the first method ('MSA method (1)' column) outperforms the second one ('MSA method (2)' column) whereas a negative *Z-Score* is obtained otherwise. Additionally, higher absolute *Z-Score* values determine more significant differences between methodologies.

According to the obtained p-values, it can be confirmed that algorithms including additional data (Promals and 3D-Coffee) does not statistically outperform other faster methods like T-Coffee or ProbCons (p-values>0.05). It has been also observed that these tools with additional data only improve the other algorithms when alignments for less related sequences are required. For instance, as previously suggested by Figure 2.9, these tools indeed statistically improve the remaining ones only when the BAliBASE RV11 subset is considered (<20% of identity). For this specific subset, the null hypothesis is rejected in the Wilcoxon test (p-values<0.05) and, therefore, the differences between algorithm with additional data and T-Coffee or ProbCons are considered statistically significant. This situation could imply again that only in particular cases algorithms with additional data are really necessary.

Additionally, no differences were found among progressive algorithms. Only ClustalW is statistically outperformed by Muscle and Kalign (p-values<0.05). Regarding consistency-based tools, it is unexpectedly observed that FSA is not statistically outperformed by other consistency-based tools (e.g. p-values of 0.051 or 0.069 against Muscle and Kalign, respectively). These results are due to the large variability of accuracy (BAliscore values) in FSA alignments, which produce that differences cannot be considered significant although FSA was previously determined as one of the worst aligners.

The time computed by aligners is also analyzed. According to Figure 2.10, it is noticed that approaches with additional data spend significantly more time, increasing almost exponentially with respect to progressive algorithms. These differences are mainly caused because the associated data must be previously retrieved and analyzed. Even though the information is already locally available and no databases need to be consulted, the time is kept high due to this information must still be analyzed and adapted to the algorithm. The computational cost in consistency-based methods could also be considerable, although

TABLE 2.2: Wilcoxon test for each pairwise comparison between MSA aligners

MSA method (1)	MSA method (2)	Z-Score	P-value	<0.05
ClustalW	RetAlign	-1.510	**0.131**	**No**
	Mafft	-1.612	**0.107**	**No**
	Muscle	-2.979	0.003	Yes
	Kalign	-2.901	0.004	Yes
	FSA	-0.949	**0.343**	**No**
	T-Coffee	-4.988	0.000	Yes
	ProbCons	-5.287	0.000	Yes
	3D-Coffee	-5.867	0.000	Yes
	Promals	-6.374	0.000	Yes
RetAlign	Mafft	-0.108	**0.914**	**No**
	Muscle	-1.613	**0.107**	**No**
	Kalign	-1.420	**0.156**	**No**
	FSA	0.487	**0.626**	**No**
	T-Coffee	-3.757	0.000	Yes
	ProbCons	-4.110	0.000	Yes
	3D-Coffee	-4.665	0.000	Yes
	Promals	-5.138	0.000	Yes
Mafft	Muscle	-1.514	**0.130**	**No**
	Kalign	-1.308	**0.191**	**No**
	FSA	0.551	**0.581**	**No**
	T-Coffee	-3.694	0.000	Yes
	ProbCons	-4.038	0.000	Yes
	3D-Coffee	-4.584	0.000	Yes
	Promals	-5.091	0.000	Yes
Muscle	Kalign	0.207	**0.836**	**No**
	FSA	1.955	**0.051**	**No**
	T-Coffee	-2.153	0.031	Yes
	ProbCons	-2.471	0.013	Yes
	3D-Coffee	-3.006	0.003	Yes
	Promals	-3.391	0.001	Yes
Kalign	FSA	1.820	**0.069**	Yes
	T-Coffee	-2.433	0.015	Yes
	ProbCons	-2.812	0.005	Yes
	3D-Coffee	-3.342	0.001	Yes
	Promals	-3.827	0.000	Yes
FSA	T-Coffee	-3.981	0.000	Yes
	ProbCons	-4.278	0.000	Yes
	3D-Coffee	-4.845	0.000	Yes
	Promals	-5.310	0.000	Yes
T-Coffee	ProbCons	-0.386	**0.700**	**No**
	3D-Coffee	-0.812	**0.417**	**No**
	Promals	-1.208	**0.227**	**No**
ProbCons	3D-Coffee	-0.440	**0.660**	**No**
	Promals	-0.842	**0.400**	**No**
3D-Coffee	Promals	-0.316	**0.752**	**No**

approaches like FSA have made an effort to drastically reduce it (at the expense of decreasing the accuracy).

Given that the average time exponentially increases reaching almost unapproachable values for more sophisticated tools, it can be again suggested that, only in special cases with less related sequences, additional data is clearly useful. Therefore, it can be expected that new trends in MSA will aim to integrate the major amount of biological information to increase the alignment qualities, but also trying to significantly reduce the used time.

2.5 Principal conclusions of the MSA comparison

Some relevant conclusions can then be revealed from this comparison. First, currently used aligners are clearly far from the highest possible qualities (average accuracies do not reach <90% for all the studied aligners and only some of them exceed <80%). Therefore, it is clear that alignment methodologies still have a plenty scope of improvement that can be exploited.

Secondly, it has been shown that the most complex algorithms are not always providing the best alignments. These methodologies are actually useful only when sequences are evolutionarily less related. Additionally, since these algorithms considerably increase the computational costs, it is critical to know if it is really worth running them. In case they do not provide enough accuracy, it is maybe possible to use less complex algorithms which drastically reduce the computational time.

Finally, though the BAliBASE references have been used here to evaluate alignments, there is no consensus about which is the best way to evaluate alignments when no references are available [Blackburne and Whelan, 2012]. Consequently, there are currently a great amount of feasible score schemes but the evaluation of alignments is still one of the main challenges in MSAs [Li and Fang, 2012].

2.6 Objectives of the Dissertation

The previous comparative study and the evidence that multiple sequences alignments are still in process of improvement have motivated the different algorithms that will be presented in this thesis. Specifically, several objectives related with MSAs are proposed as the principal basis of this dissertation:

- To determine a new algorithm to a priori infer the aligner which should provide the best alignment for a particular set of sequences. This methodology aims to estimate the accuracy that each aligner could reach before the alignment is performed in order to decide which is the most suitable tool. Since the proposed algorithm can be considered a regression problem, a least-squares support vector machine (LS-SVM) will be implemented to solve it. The system will integrate a dataset of features related to the sequences being aligned from heterogeneous biological data sources. This method will then provide predicted accuracies according to 23 representative features for 10 different aligners. The most significant features to a priori estimate the accuracy will be chosen by a feature selection procedure. This prediction could avoid the need for using more time-consuming algorithms when they are not really returning meaningful differences.
- To develop advanced schemes to evaluate multiple sequence alignments based on a set of heterogeneous biological features. The main objective here is to provide an efficient system which accurately evaluates alignments when no references are available. This advanced algorithms will integrate as much biological information as possible for the determination of alignment quality. In this case, the features will be associated with the alignments being evaluated (instead of only sequences, as the previous objective). A feature selection procedure will be also implemented to determine which features could be more related to the MSA quality. In this case, several regression approaches will be proposed and compared. The most promising schemes will be also compared against other evaluation scores in the literature.

- To implement a novel multi-objective algorithm for the optimization of multiple sequence alignments. A genetic approach focused on the non-dominated sorting genetic algorithm (NSGA-II) will be proposed. A novel scheme will be proposed to codify alignments as well as efficient mutation and crossover operators. The algorithm seeks to optimize alignments previously built by fast aligners by considering three different objectives. These objectives will be related to the structural information in proteins, the conserved regions in sequences and the percentage of gaps. Taking into account the three objectives, the genetic algorithm will integrate the most accurate regions in previous alignments in order to progressively evolves toward improved alignments. Given the multi-objective approach, a set of optimal alignments (Pareto front) will be provided by this algorithm.

2.7 Structure of the Dissertation

From this chapter on, the dissertation is arranged into three different parts according to the proposed objectives. Chapter 3 is dedicated to the problem of predicting the expected accuracy from aligners in a particular set of sequences before the alignment is designed. This chapter introduces the theoretical concepts related to the computational strategies considered for this problem, such as the LS-SVM algorithm or the feature selection with the minimum-Redundancy maximal-Relevance (mRMR) algorithm. The proposed algorithm is compared at the end of this chapter against another similar tool called AlexSys. Chapter 4 proposes a novel scoring method to accurately evaluate alignments by means of different biological features. In this case, several regression approaches applied to this problem (regression trees, bootstrap aggregation trees, LS-SVMs or Gaussian Processes) are described in detail. The normalized mutual information feature selection (NMIFS) is also introduced to choose the most relevant features. A statistical comparison of these scores against classical schemes (BLOSUM, PAM, etc) and more complex approaches (Facet, PredSP, etc) is finally presented. Chapter 5 describes the proposed multi-objective algorithm for the optimization of sequence alignments. In this case, the principal genetic algorithm concepts are introduced as well as some special considerations for the NSGA-II implementation. The optimization achieved with this algorithm is statistically analyzed in this chapter together with the comparison against other genetic aligners and 3D-Coffee.

Finally, the main conclusions of the dissertation are discussed in Chapter 6. In order to complement this dissertation, a short curriculum vitae of the PhD candidate is provided at the end to make easier the inquiry of the main publications supporting the contributions of this thesis.

Chapter 3

PAcAlCI: A priori Prediction of Accuracy Alignment based on Computational Intelligence

3.1 Motivation and goals

As previously commented in Chapter 2, current MSA methodologies do not always provide consistent solutions, since alignments become increasingly difficult when dealing with remotely related sequences [Li and Fang, 2012]. Moreover, the high number of existing MSA tools makes really hard to decide which one is the most appropriate for a particular set of protein sequences. As widely known, these algorithms directly depend on particular features of the sequences, causing relevant variations on the accuracy of the alignments [Blackburne and Whelan, 2013].

Therefore, there is no consensus about which is the best way to align sequences or the most adequate criterion to follow. Some systems have developed meta-method approaches [Muller et al., 2010, Wallace et al., 2006], which align sequences with several aligners to subsequently determine the most accurate one. However, although these meta-methods increase the accuracy in alignments, they require the computation of all the alternative aligners which could result in an excessive computational time. Consequently, to the best of our knowledge and after a careful bibliographic revision it is not clear yet how to know, in advance, which aligner is the most suitable for a specific set of sequences.

Trying to solve this problem, a novel algorithm to predict a priori the accuracy that several aligners will obtain when aligning a particular set of sequences is presented in this chapter. This system has been called *Prediction of Accuracy in Alignments based on Computational Intelligence* (**PAcAlCI**) [Ortuño et al., 2013]. PAcAlCI proposes an advanced intelligent system based on the extraction of a heterogeneous dataset of biological features. These features, associated to the sequences to be aligned and their corresponding proteins, are carefully extracted from well-known curated databases. Thus, the main goal of PAcAlCI is to estimate the most promising methodologies according to the estimated accuracies. These aligners are then proposed as the most suitable candidates to accurately align each set of sequences. Taking into account these objectives, PACAlCI has been developed in four main modules (see Figure 3.1) as follows:

1. A complete dataset of alignments is built by aligning several sets of sequences from the BAliBASE benchmark with ten different alignments tools. These ten methodologies are some of the most applied tools in the literature and their usefulness has been widely proved. The accuracy of these alignments is then obtained by using the BAliscore measured from BAliBASE (*Input Dataset* module).
2. A set of 23 heterogeneous features related to the sequences is retrieved from several databases. Specifically, these biological features are associated with the proteins codified by the sequences (*Feature Extraction* module). These features are converted into quantitative measures to facilitate their inclusion in the subsequent algorithm.
3. A feature selection algorithm based on the minimal-Redundancy-Maximal-Relevance (mRMR) criterion is applied to determine those features that are considered especially relevant for the prediction of accuracies in alignments (*Feature Selection* module). The most significant features are progressively included in a subset which is used by the subsequent algorithm. The subset of features which obtains the optimal prediction is then selected for our algorithm.
4. An algorithm for the estimation of the accuracy in alignments is finally designed based on Least-Squares Support Vector Machines (*LS-SVM Prediction* module). This algorithm learns from already known accuracies to predict the one for new sets of sequences through the previously extracted features from the sequences. This algorithm is first trained and validated by using a cross-validation procedure and the quality measures provided by BAliscore. Subsequently, the most suitable methodologies are determined in each particular set of sequences according to the predicted accuracies for all the proposed aligners.

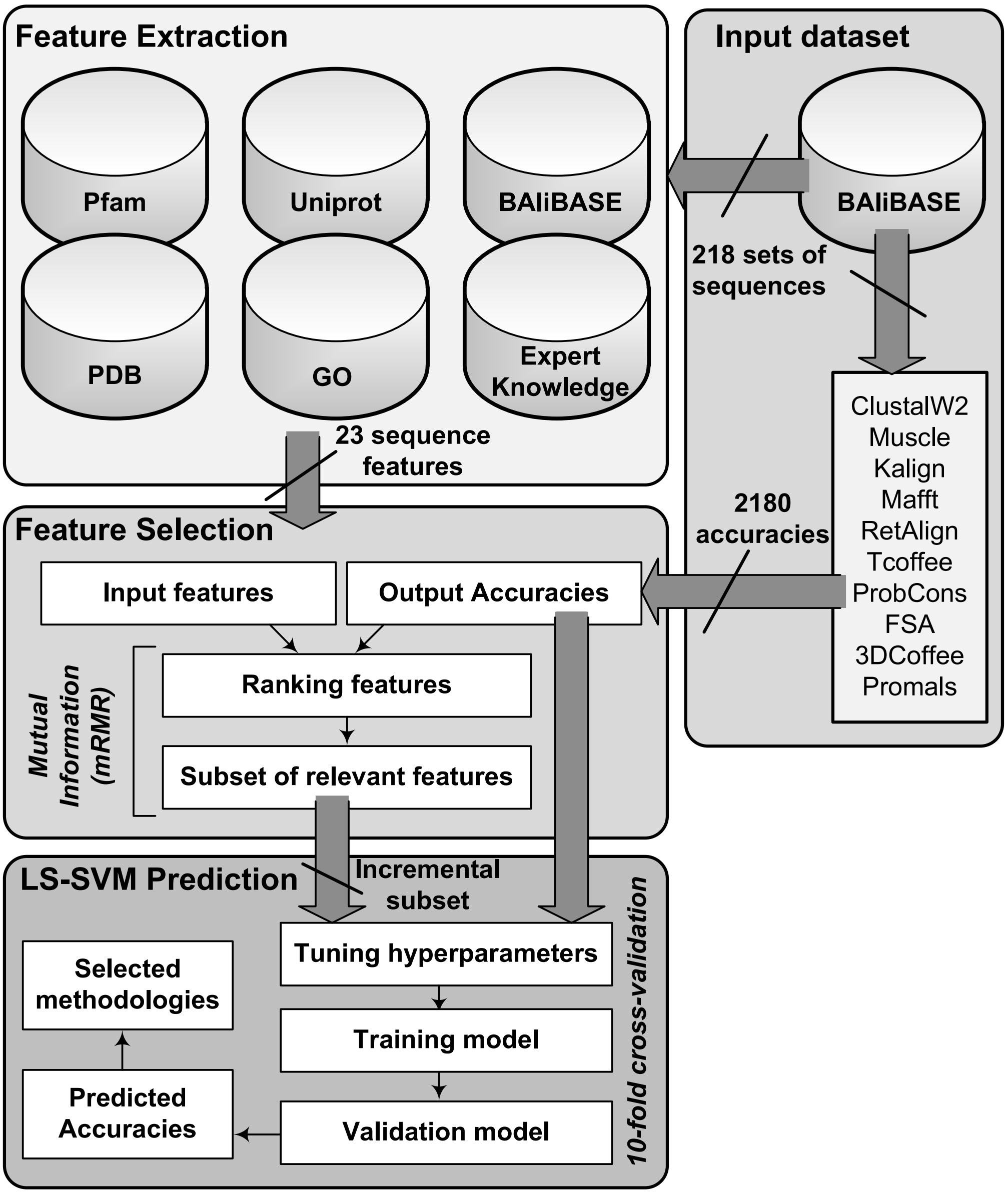

Feature Extraction
Pfam
Uniprot
BAliBASE
PDB
GO
Expert Knowledge
Input dataset
BAliBASE
218 sets of sequences
ClustalW2
Muscle
Kalign
Mafft
RetAlign
Tcoffee
ProbCons
FSA
3DCoffee
Promals
23 sequence features
2180 accuracies
Feature Selection
Input features
Output Accuracies
Mutual Information (mRMR)
Ranking features
Subset of relevant features
LS-SVM Prediction
Incremental subset
Tuning hyperparameters
Training model
Validation model
10-fold cross-validation
Selected methodologies
Predicted Accuracies

FIGURE 3.1: PAcAlCI scheme. The architecture is developed into four modules: *(i)* the *Input Dataset* module composes the alignments from 10 different tools; *(ii)* the *Feature Extraction* module consults and retrieves 23 features from well-known databases, *(iii)* the *Feature Selection* module determines the most relevant features for the prediction of accuracy; and *(iv)* the *LS-SVM Prediction* module performs the final accuracy estimation and selects the most suitable alignment tools according to such estimations.

The PAcAlCI system was completely implemented with Matlab$^{\circledR}$ (R2010b version). The PAcAlCI functions and source code is freely available and can be downloaded at **http://www.ugr.es/~fortuno/pacalci.htm**.

This chapter is then structured as follows: the different modules of PAcAlCI are presented and explained in detail in the next sections (*Input Dataset* module in section 3.2, *Feature Extraction* module in section 3.3, *Feature Selection* in section 3.4 and, finally, *LS-SVM Prediction* module in section 3.5). Furthermore, the performed experiments and results of this algorithm are shown in the section 3.6 together with a qualitative comparison with another similar tool, called AlexSys [Aniba et al., 2010].

3.2 Creation of a dataset of multiple sequence alignments

The first step in the process of designing PAcAlCI consists in creating a wide dataset of sequences and alignments. The dataset used to implement this algorithm has been previously presented in this dissertation for the comparison of alignments tools in the introductory chapter of multiple sequence alignments (section 2.4). This dataset is created from the BAliBASE sequences and ten different aligners. Briefly, following sections remind some of the main characteristics about the construction of this dataset.

3.2.1 BAliBASE benchmark

As commented in Chapter 2, several datasets and benchmarks have usually been developed to standardize the comparison of alignment results, e.g. Oxbench [Raghava et al., 2003], HOMSTRAD [Mizuguchi et al., 1998], Prefab [Edgar, 2004] or BAliBASE [Thompson et al., 1999]. In this case, the BAliBASE benchmark (v3.0) was chosen [Thompson et al., 2005]. It is important to highlight the principal properties of this benchmark that have been taken into account for the proposed PAcAlCI implementation:

- BAliBASE defines a total of 218 sets of sequences that were manually curated. These sets of sequences are provided in FASTA format, an standard text format file for sequences. Thus, they are specifically prepared to be directly aligned since this usually is the input format for alignment tools. Note that these sequences are mainly retrieved from the Protein Data Bank (PDB) [Berman et al., 2000].
- This benchmark also provides a set accurate reference alignments (*gold standard*) which have been manually refined by using expert knowledge. These references allow us to compare and evaluate the alignments obtained by other tools. Specifically, BAliBASE calculates a typical sum-of-pairs (SP) score called BAliscore to evaluate such alignments. BAliscore compares the reference alignment against alignments provided by other tools and determines the accuracy of such tools. Therefore, the BAliscore values are applied in PAcAlCI to learn from them and allow the posterior estimation of accuracies when no reference alignments are available.
- Sequences in BAliBASE are accurately classified in six different subsets according to specific properties. In this case, we are interested in the identity percentage among sequences provided by each group (see Table 2.1 for details). In fact, the subset to which each set of sequences belongs is considered one of the properties in our feature dataset (section 3.3)
- Sequences in BAliBASE are well-annotated since they are extracted from well-known databases, namely PDB or Uniprot. Moreover, information about these sequences can also be found in other databases like Pfam, Gene Ontology or DSSP. Besides, BAliBASE identifies sequences in each set by standard accessions like the PDB name or the Uniprot identifier. Therefore, the access to other standard databases is relatively simple given that both PDB and Uniprot systems provides useful tools to interrelate their entries with other database accessions.

These properties make BAliBASE an appropriate benchmark for the subsequent extraction of biological feature and the implementation of the PAcAlCI prediction system.

3.2.2 Multiple sequence alignment tools

To complete the dataset of alignments, it is also necessary to build several alignments from the previously described sets of sequences. Specifically, similarly to section 2.4, ten MSA tools are selected to generate the alignments of this dataset. These ten tools have been chosen because of their recognized usefulness that makes them widely applied in the literature [Blackburne and Whelan, 2013, Kemena and Notredame, 2009]. It is also important to remind that these tools are classified according to their implemented strategy: progressive techniques, consistency-based methods or algorithms including additional information. Among the progressive methods, ClustalW [Thompson et al., 1994], Muscle [Edgar, 2004], Kalign [Lassmann and Sonnhammer, 2005], Mafft [Katoh et al., 2002] and RetAlign [Szabo et al., 2010] are chosen. Three consistency-based approaches are also included, namely T-Coffee [Notredame et al., 2000], ProbCons [Do et al., 2005] and Fast Statistical Alignment (FSA) [Bradley et al., 2009]. Finally, the two methodologies with additional data presented in Chapter 2, namely Promals [Pei and Grishin, 2007] and 3D-Coffee [O'Sullivan et al., 2004] are also considered. All these tools have already been widely explained in section 2.2.2. These programs were run with their default configurations. A brief summary of the ten applied MSA tools is presented in Table 3.1.

This dataset includes a heterogeneous group of MSA tools. We have considered for this work those widely used aligners, but trying to include diverse properties: different computational complexity and consumed time, several strategies, use of some additional data, etc. The goal is to determine if these properties could affect in the accuracy of the alignments when different types of sequences have to be aligned. If so, PAcAlCI will determine which tool is more suitable to align each particular set of sequences taking into account the properties of every tool. For instance, according to the comparative study in section 2.4, techniques like progressive and consistency-based approaches were not achieving an acceptable accuracy with evolutionarily less related sequences whereas more complex methods achieved higher accuracies at the expense of an excessive computational time. Nevertheless, this consumed time was considered un-

TABLE 3.1: Summary of the applied methodologies. Ten different methodologies were run to align multiple sequences. Their versions and the applied strategies are also shown.

METHOD	VERSION	TYPE
ClustalW [Thompson et al., 1994]	2.0.10	Progressive
Muscle [Edgar, 2004]	3.8.31	Progressive
Kalign [Lassmann and Sonnhammer, 2005]	2.04	Progressive
Mafft [Katoh et al., 2002]	6.85	Progressive
RetAlign [Szabo et al., 2010]	1.0	Progressive
TCoffee [Notredame et al., 2000]	8.97	Consistency-based
ProbCons [Do et al., 2005]	1.12	Consistency-based
FSA [Bradley et al., 2009]	1.15.5	Consistency-based
3DCoffee [O'Sullivan et al., 2004]	8.97	Additional Data
Promals [Pei and Grishin, 2007]	vServer	Additional Data

approachable and improvements could be significant only in a few cases with clearly distant sequences [Kemena and Notredame, 2009].

A total dataset of 2180 alignments is then generated to be applied in PAcAlCI (218 sets of sequences aligned by 10 different tools). Besides, their corresponding 2180 accuracies are calculated using the BAliscore program provided by BAliBASE. These BAliscore values are considered in PAcAlCI as an accurate measure of the quality for alignments.

3.3 Databases and feature extraction

In this section, a set of features associated with the BAliBASE sequences and their corresponding proteins are extracted from well-known biological databases. Such databases are consulted to obtain useful data which complement the sequences and to build a complete set of features. The final extracted dataset is composed by 23 biological features.

Some features related to sequences (domains, types of amino acids or structures) have already been successfully included in other similar knowledge-based systems [Aniba et al., 2010, Nuin et al., 2006]. However, additional features have been incorporated here in order to complement this dataset taking into account

TABLE 3.2: Summary of the extracted features. 23 features are retrieved from different databases. (1) These features are calculated as the percentage of amino acids (AA) with that specific feature. (2) These features are calculated as the number of occurrences per sequence.

	FEATURE	SOURCE	RANGE	TYPE
f_1	Sequences	BAliBASE	[4, 142]	Integer
f_2	Average length	BAliBASE	[66, 1630]	Real
f_3	Variance length	BAliBASE	$[0, 2.47\times10^6]$	Real
f_4	AA in α-helix(1)	UniProt	[0, 1]	Real
f_5	AA in β-strand(1)	UniProt	[0, 1]	Real
f_6	AA in transmemb.(1)	UniProt	[0, 1]	Real
f_7	Domains(2)	Pfam	[0.0, 6.7]	Real
f_8	Shared Domains(2)	Pfam	[0.0, 117.1]	Real
f_9	GO terms(2)	GO	[0.0, 8.7]	Real
f_{10}	MF-GO terms(2)	GO	[0.0, 5.2]	Real
f_{11}	CC-GO terms(2)	GO	[0.0, 2.5]	Real
f_{12}	BP-GO terms(2)	GO	[0.0, 4.1]	Real
f_{13}	Shared GO terms(2)	GO	[0.0, 201.9]	Real
f_{14}	3D-Structures(2)	PDB	[0.0, 3.1]	Real
f_{15}	Seq. w/ 3D-structures	PDB	[0, 1]	Real
f_{16}	Shared 3D-structures(2)	PDB	[0.0, 0.8]	Real
f_{17}	Polar AA(1)	Biochemistry	[0, 1]	Real
f_{18}	Non-polar AA(1)	Biochemistry	[0, 1]	Real
f_{19}	Basic AA(1)	Biochemistry	[0, 1]	Real
f_{20}	Aromatic AA(1)	Biochemistry	[0, 1]	Real
f_{21}	Acid AA(1)	Biochemistry	[0, 1]	Real
f_{22}	Reference subset	BAliBASE	[1, 6]	Integer
f_{23}	MSA Method	—	[1, 10]	Integer

other studies such as protein interaction predictions [Wu et al., 2006] or protein model classifications [Roslan et al., 2010]. Therefore, a more complete feature environment is presented in this chapter in order to study its significance in sequence alignments (see the dataset summary in Table 3.2).

Depending on the specific property that is extracted, several databases described in section 1.5 are consulted: Uniprot for protein information [The UniProt Consortium, 2014], Pfam for protein domains [Finn et al., 2014], PDB for protein structures [Berman et al., 2000] or Gene Ontology for molecular annotation [Ashburner et al., 2000]. Following, the full feature dataset is presented, indicating for each feature the database from which they have been retrieved and its nomenclature in their feature list.

Note that these features are subsequently converted to quantitative measures. To formally define these quantitative measures, the following annotation is generally applied: N defines the number of sequences whereas L_i indicates the length of the *i-th* sequence of a particular set of sequences.

3.3.1 Features from BAliBASE sequences

Besides a benchmark, BAliBASE [Thompson et al., 2005] can also be considered the first consulted database, since it provides the sequences that are aligned. Some statistical measures associated with each set of sequences are extracted from BAliBASE:

- **Number of sequences** (f_1): the number of sequences (N) included in each set of sequences is a key property for this purpose. The number of sequences is directly related to the complexity of the alignment and, therefore, to the probability of having less accurate alignments. Moreover, the computational time that each method will spend to obtain the alignment could exponentially increase with this parameter. Therefore, they could reach unapproachable time or low accuracies depending on the number of sequences.
- **Average length of sequences** (f_2): the length of sequences could also give essential information to determine the accuracies of the alignments. Similarly to the previous feature, longer sequences could lead to more complex alignments and extremely high computational time in some tools. Thus, tools providing local alignment methodologies could align longer sequences better whereas shorter sequences could be aligned as global alignments. The feature is mathematically expressed as:

$$f_2 = \mu_L = \sum_{i=1}^{N} \frac{L_i}{N} \tag{3.1}$$

- **Variance of the sequence length** (f_3): another statistical measure associated with the sequence length is the variance. The variance represents how divergent the sequences are in terms of their lengths (length variability). A higher variance implies the necessity of a more efficient gap scheme in the tool, since more gaps are required to complete the alignment. Other properties involved with this parameters are the insertion and deletion processes. This feature is highly variable for sequences in BAliBASE. It can be formally defined us:

$$f_3 = \sigma_L^2 = \sum_{i=1}^{N} \frac{(L_i - \mu_L)^2}{N} \tag{3.2}$$

3.3.2 Features from Uniprot

Uniprot [The UniProt Consortium, 2014] is considered an essential database to extract protein information given that it provides accurate, consistent and rich data about proteins. Additionally, thanks to the standardization of this database, proteins can be accessed from different accessions and cross-references with other databases are easily found in Uniprot. In this case, we are interested in properties related to protein locations or secondary structure of their sequences. Sequences in Uniprot are usually complemented with the secondary structure in which each amino acid is involved. Two basic secondary structures are provided: α-helix or β-strand. Additionally, Uniprot also allows to the exact location of proteins in the cell, even determining the position of some special amino acids in particular regions, e.g. in the transmembrane. Therefore, the following features are then calculated from this database:

- **Percentage of amino acids in α-helix structures** (f_4): higher percentages of amino acids in a well-defined α-helix structure could indicate the presence of similar secondary structures between sequences. This property could likely lead to more related sequences or similar regions. This percentage is calculated as:

$$f_4 = \frac{\sum_{i=1}^{N} |\alpha_i|}{\sum_{i=1}^{N} L_i} \tag{3.3}$$

where $|\alpha_i|$ denotes the number of amino acids in α-helix structure for the *i-th* sequence.

- **Percentage of amino acids in β-strand structures** (f_5): similarly to the previous one, the percentage of amino acids in β-strand structures are measured. Similar secondary structure can again be associated with more similar sequences. In this case, the feature is formally defined as:

  $$f_5 = \frac{\sum_{i=1}^{N} |\beta_i|}{\sum_{i=1}^{N} L_i} \tag{3.4}$$

  where $|\beta_i|$ is now representing the number of amino acids in β-strand structure for the *i-th* sequence.

- **Percentage of amino acids in the transmembrane region** (f_6): proteins or partial blocks in proteins located in the transmembrane region could suggest similar functionality or structures. Although BAliBASE does not incorporate many sequences within the transmembrane region, to identify these regions could be useful to know if any of the aligners provides better accuracies in this situation. Similarly to the previous two properties, the percentage of transmembrane regions is calculated as:

  $$f_6 = \frac{\sum_{i=1}^{N} |T_i|}{\sum_{i=1}^{N} L_i} \tag{3.5}$$

  where $|T_i|$ defines the number of amino acids in transmembrane for the *i-th* sequence.

3.3.3 Features from Pfam

As explained in section 1.5, Pfam [Finn et al., 2014] stores protein domains which consist in common functional regions in families. Finding domain similarities in sequences could then imply the existence of regions with related functionality. Therefore, this common functionality can be useful to understand how some sequences must be efficiently aligned or how close sequences are in their families. The domain properties that are performed from Pfam database are:

- **Number of domains per sequence** (f_7): the number of domains in a sequence could represent an acceptable measure about how well this sequence is annotated in terms of domains. The dependence between the domain annotation of sequences and the quality of the alignment could be determined with this property. Formally, this feature is calculated as:

$$f_7 = \sum_{i=1}^{N} \frac{|D_i|}{N} \tag{3.6}$$

  where $|D_i|$ defines the number of Pfam domains found in the *i-th* sequence.

- **Number of shared domains per sequence** (f_8): this feature relates the domains in different sequences. It measures the number of domains in each sequence that also belong to another sequence in the set. That is, the number of times that two sequences include the same domain is counted and divided by the number of sequences. These common domains between each pair of sequences are formally measured as:

$$f_8 = \frac{|\bigcap_{\substack{i,j=1 \\ j \neq i}}^{N} (D_i, D_j)|}{N} \tag{3.7}$$

  where (D_i, D_j) are the sets of domains in the *i-th* and *j-th* sequences, respectively and $\cap(D_i, D_j)$ are the common domains in both sequences with '| |' indicating the cardinality of the set.

3.3.4 Features from Gene Ontology

Briefly, Gene Ontology (GO) [Ashburner et al., 2000] provides controlled vocabularies for the annotations of molecular attributes. Three different ontologies can be found in GO: molecular function (MF), cellular component (CC) and biological process (BP). Since GO project defines a complete description of processes and activities in proteins, this information could significantly improve the information about the sequences in the proposed system. Features extracted from GO are:

- **Number of GO terms per sequence** (f_9): this feature gives a measure about how well-annotated are the sequences in GO. Formally, this feature is defined as:

$$f_9 = \sum_{i=1}^{N} \frac{|GO_i|}{N} \tag{3.8}$$

  with $|GO_i|$ being the number of GO terms for the *i-th* sequence. To understand the dependence on alignments of each ontology separately (MF, CC and BP), this feature was also recalculated for each ontology independently (the three following features).

- **Number of MF terms per sequence** (f_{10}): this feature only considers those terms which are included in the *molecular function* ontology. The *molecular function* terms refer to the functionality of proteins in a molecular level. The feature is measured as:

$$f_{10} = \sum_{i=1}^{N} \frac{|GO_{MFi}|}{N} \tag{3.9}$$

- **Number of CC terms per sequence** (f_{11}): in this case, terms in the *cellular component* ontology are considered. The parts of the cell involved and other extracellular enviroment are described with this feature. The *cellular* *component* feature is calculated as:

$$f_{11} = \sum_{i=1}^{N} \frac{|GO_{CCi}|}{N} \tag{3.10}$$

- **Number of BP terms per sequence** (f_{12}): finally, the number of *biological process* terms are measured in this feature. The *biological process* terms define the processes and events totally or partially carried out by the proteins. This feature is obtained as:

$$f_{12} = \sum_{i=1}^{N} \frac{|GO_{BPi}|}{N} \tag{3.11}$$

- **Number of shared GO terms per sequence** (f_{13}): in a similar way as the f_8 feature in Pfam database, the number of terms in a sequence that also belong to another one is calculated. This feature can provide information about proteins taking part of similar functionality and processes or being located in the same cellular regions. This feature is measured as:

$$f_{13} = \frac{|\bigcap_{\substack{i,j=1 \\ j \neq i}}^{N} (GO_i, GO_j)|}{N} \tag{3.12}$$

where (GO_i, GO_j) is a pair of sets with the terms of the *i-th* and *j-th* sequences, respectively. Likewise, $\cap(GO_i, GO_j)$ are the common GO terms in both sequences with '| |' indicating the cardinality of the set.

3.3.5 Features from the Protein Data Bank

The Protein Data Bank (PDB) includes information about experimentally determined 3D-structures of each protein. The structural data is relevant in proteins given that it is strongly related with the functionality of the protein and it is evolutionarily more conserved than sequences. Additionally, this information plays a key role to determine if tools using additional information could be useful. For instance, 3D-Coffee requires structural data to build the alignment and its accuracy could be affected by the lack of these structures. Therefore, this dataset has been complemented with the following features related to the tertiary structures:

- **Number of 3D structures per sequence** (f_{14}) in the same way as previous cases, the number of structures that are found in PDB for each sequence is measured. This feature could be an evidence of how well-annotated the sequences are in terms of their tertiary structure. This measurement is defined as:

$$f_{14} = \sum_{i=1}^{N} \frac{|PDB_i|}{N} \tag{3.13}$$

with $|PDB_i|$ being the number of tertiary structures in the *i-th* sequence.

- **Percentage of sequences with 3D structure** (f_{15}): in this case, the number of sequences that include at least one 3D-structure is calculated. This feature allows to know if sequences in each set are well-defined in PDB. This percentage is expressed as follows:

$$f_{15} = \sum_{i=1}^{N} \frac{\mathcal{F}(|PDB_i|)}{N} \tag{3.14}$$

where

$$\mathcal{F}(|PDB_i|) = \begin{cases} 1 & if \quad |PDB_i| > 0 \\ 0 & otherwise \end{cases} \tag{3.15}$$

- **Number of shared 3D structures per sequence** (f_{16}): the common structures between each pair of sequences are also considered. This feature expresses the structural relationship between sequences. In a similar way as previous features, these relationships are calculated as the times that two sequences share one 3D structure. This is formally calculated as:

$$f_{16} = \frac{|\bigcap_{\substack{i,j=1 \\ j \neq i}}^{N} (PDB_i, PDB_j)|}{N} \tag{3.16}$$

with (PDB_i, PDB_j) being a pair of sets with the 3D structures of the *i-th* and *j-th* sequences, respectively and $\cap(PDB_i, PDB_j)$ the shared structures in both sequences ('| |' indicates the cardinality of the set).

3.3.6 Additional features

Apart from these databases, other resources have been considered in order to complete the dataset of features. For instance, the chemical classification of amino acids previously described in section 1.2.3 [Mathews et al., 2000] has been applied. This kind of classification has been proved to be useful in alignments for similar prediction tools, e.g. AlexSys [Aniba et al., 2010]. Thus, the following features were calculated:

- **Percentage of polar uncharged amino acids** (f_{17}): This group is composed by the amino acids Glycine (G), Alanine (A), Proline (P), Valine (V), Leucine (L), Isoleucine (I) and Methionine (M).

$$f_{17} = \frac{\sum_{i=1}^{N} |AA_{polar,i}|}{\sum_{i=1}^{N} L_i} \tag{3.17}$$

- **Percentage of non-polar aliphatic amino acids** (f_{18}): This group is composed by the amino acids Serine (S), Threonine (T), Cysteine (C), Asparagine (N) and Glutamine (Q).

$$f_{18} = \frac{\sum_{i=1}^{N} |AA_{nonpolar,i}|}{\sum_{i=1}^{N} L_i} \tag{3.18}$$

- **Percentage of basic positively-charged amino acids** (f_{19}): The amino acids Lysine (K), Arginine (R) and Histidine (H) form this group.

$$f_{19} = \frac{\sum_{i=1}^{N} |AA_{basic,i}|}{\sum_{i=1}^{N} L_i} \tag{3.19}$$

- **Percentage of aromatic amino acids** (f_{20}): Phenylalanine (F), Tyrosine (Y) and Tryptophan (W) are considered the aromatic amino acids.

$$f_{20} = \frac{\sum_{i=1}^{N} |AA_{aromatic,i}|}{\sum_{i=1}^{N} L_i} \tag{3.20}$$

- **Percentage of negatively-charged amino acids** (f_{21}): finally, the negatively-charged group considers the Aspartate (D) and Glutamate (E) amino acids. This group is also called *acid* amino acids.

$$f_{21} = \frac{\sum_{i=1}^{N} |AA_{acid,i}|}{\sum_{i=1}^{N} L_i} \tag{3.21}$$

In this chemical classification, each $|AA_{x,i}|$ represents the number of amino acids of the x group in the *i-th* sequence.

Additionally, **the subset in BAliBASE** (f_{22}) to classify the sets of sequences mainly according to the identity in their families (see table 2.1) was also considered. This feature represents a discrete value indicating the group or subset to which the set of sequences belongs. Note that there are six different possible groups: RV11, RV12, RV20, RV30, RV40 and RV50. The feature can then determine how the identity percentage affects the accuracy provided by each aligner.

Finally, the MSA tool being executed (see section 3.2 for details) is the last included feature. Since the goal here is to predict the accuracy of each method according to all these features, **the selected tool** (f_{23}) is an essential feature in the proposed algorithm. This feature is then a discrete variable with ten possible values (ten different tools). This feature is also applied to determine the most suitable methods according to the predicted accuracies.

To complete this dataset, each set of 23 features (**input variables**) is related to its corresponding accuracy (**output variable**). As explained before, the accuracy is determined by using the BAliscore from BAliBASE. It defines a quality measure comparing the obtained alignments against the gold-standard references.

3.4 Selection of features based on mutual information

When working with a wide dataset of features, it could be interesting to previously study the importance of each feature with regard to the output variable to be predicted, in this case, the accuracy in alignments. The relevance of each feature is usually performed by using a feature selection procedure. These procedures evaluate the dependence of every feature against the output value (accuracies) and rank them according to a quality criterion. As a consequence, feature selection algorithms reduce the number of features, filtering out those ones that are irrelevant or do not complement the information already provided by other features.

In order to analysis the relevance of the 23 features previously defined, a feature selection procedure is developed in this section. In this regard, a brief overview about feature selection procedures (section 3.4.1), a description of the mutual information measure (section 3.4.2) and, more specifically, the minimum-Redundance-Maximum-Relevance (mRMR) algorithm which is used in PAcAlCI (section 3.4.3) are presented.

3.4.1 Feature selection procedures

Normally, two standard algorithms are classified in the literature to select features depending on the way they work: filter methods or wrapper methods [John et al., 1994]. The first ones, filter methods, are designed to evaluate each feature directly from the data, without any feedback from the subsequent prediction algorithm. They then select the most relevant features through a preprocessing procedure. On the other hand, wrapper methods evaluate the goodness of the selected features by using the posterior prediction algorithm. They provide a subsets of feature to the predictors and receive their performance as feedback (usually, in terms of accuracy or quality).

Both methodologies have also been combined in other researches in order to achieve a more accurate selection [Peng et al., 2005]. In these cases, a filter method is firstly applied to obtain the relevance of the features. This method is then assessed with the wrapper method using a posterior support vector machine (SVM) model. In fact, a combination between filter and wrapper is used in this chapter.

3.4.2 Mutual Information definition

Traditionally, several metrics like correlation or mutual information have been used to establish the goodness of each feature for selection algorithms [Bins and Draper, 2001, Cover and Thomas, 2006]. Mutual information (MI) is a widely used approach and it is considered a good indicator of relevance between variables. The main advantages of the mutual information compared with others measures are:

1. MI is able to accurately detect any relationship between features included in the dataset.
2. MI is invariant under space transformations produced in the included features [Estevez et al., 2009, Kullback, 1997].

However, calculating the mutual information can be difficult, and the performance of a feature selection algorithm depends on the accuracy of the mutual information [Babich and Camps, 1996]. MI for two random continuous variables x and y (two features) can be expressed as:

$$I(x,y) = \int\int p(x,y) log \frac{p(x,y)}{p(x)p(y)} dxdy \tag{3.22}$$

where $p(x)$ and $p(y)$ represent the marginal probability density functions (*pdf*) of both variables x and y; and $p(x,y)$ defines the joint probability density function. The MI for continuous variables must be approximated through estimations of density with, for example, a Parzen Gaussian Window procedure [Babich and

Camps, 1996]. This equation can also be reformulated as a summation if X and Y are defined as discrete variables:

$$I(x, y) = \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x, y) log \frac{p(x, y)}{p(x)p(y)} dxdy \tag{3.23}$$

where $p(x, y)$ is now the joint probability mass function. In this case, $\mathcal{X}$ and $\mathcal{Y}$ represent the complete alphabet of the discrete variables x and y, respectively.

Despite the previously commented drawbacks, mutual information is a widely used measure for feature selection procedures. In fact, the mutual information is applied in the chosen feature selection algorithm called minimal-Redundancy-Maximal-Relevance [Peng et al., 2005].

3.4.3 Minimal-Redundancy-Maximal-Relevance (mRMR) algorithm

Feature selection algorithms based on mutual information aim to find the largest dependency between a subset of features and the output variable. This dependency, also called maximal dependency, can be defined in terms of mutual information as:

$$max\{D(S, y)\}; D(S, y) = I(S, y) = \int\int p(S, y) log \frac{p(S, y)}{p(S)p(y)} dS\, dy \tag{3.24}$$

where y represents the output variable and S defines a subset of selected features from the complete set of features F ($S \subset F$). However, the estimation of the multivariate density functions $p(S, y)$ and $p(S, y)$ is often hard to implement when a large number of features is being considered.

For this reason, the minimal-Redundancy-Maximal-Relevance (mRMR) method defined by Peng et al. [2005] proposes an equivalent form to estimate the maximal dependence. mRMR presents a two-stage feature selection procedure:

first, a simple filter method is applied to estimate the feature dependence by using the mRMR criterion based on mutual information; and, secondly, a wrapper method is considered to determine a compact subset of features by using backward and forward selections until the candidate set of features minimizes the estimation error in the output variable.

More specifically, the mRMR criterion is composed by two conditions according to the mutual information of features (instead of only the maximal dependency): *(i)* maximum relevance of a subset of features according to the output variable (equivalent to the maximal dependency) and *(ii)* minimal redundancy within the selected subset of features. These conditions are formulated as follows:

$$max\{D(S,y)\}; D(S,y) = \frac{1}{|S|}\sum_{x_i \in S} I(x_i, y) \tag{3.25}$$

$$min\{R(S)\}; R(S) = \frac{1}{|S|^2}\sum_{x_i, x_j \in S} I(x_i, x_j) \tag{3.26}$$

where y and S again define the output variable and the candidate subset of selected features, respectively. In this case, $D(S,y)$ is called the **relevance** of a specific subset against the output variable whereas $R(S)$ represents the **redundancy** with the already selected features. Peng et al. [2005] have proved that the combination of these two conditions is equivalent to the maximal dependence when one feature is selected at one time (first-order incremental method). Therefore, the mRMR criterion is obtained combining both constraints. The operator ϕ is defined to combine $D(S,y)$ and $R(S)$ to optimize both simultaneously:

$$max\{\phi(D,R)\}; \phi = D - R \tag{3.27}$$

The mRMR method has been selected in PAcAlCI because it obtains a great accuracy in a reduced time, even compared with other wrapper algorithms [Estevez et al., 2009]. Moreover, discrete and continuous random variables are both taken into account in this feature selection algorithm. Such property is essential in the proposed set of features, since both types of variables were included (*real* and *integer* types in Table 3.2). The output accuracy is also defined as a continuous variable.

Therefore, features in the full dataset are progressively selected in descent ordering according to the mRMR criterion to be included in the subsequent system for the accuracy estimation. Consequently, the subset of features minimizing the posterior estimation error is finally considered by PAcAlCI.

3.5 Prediction algorithm based on Least-Square Support Vector Machine

This section presents a brief introduction to the learning methodology used in this chapter, Least Squares Support Vector Machine (LS-SVM) [Suykens et al., 2003]. As previously described, LS-SVM is applied to predict the accuracy that several alignment tools could provide for a specific set of sequences before these sequences are aligned. This prediction is performed by training the LS-SVM algorithm with a dataset of input features and the real accuracies provided by these tools. The aim of this algorithm is to determine which methodologies are more reasonable in terms of accuracy in order to obtain the best possible result in the alignment.

Following, a detailed definition of LS-SVM is presented. Next, the assessment and MSA tool selection procedures that have been followed in this system are also described.

3.5.1 Least-Square Support Vector Machines (LS-SVMs)

LS-SVMs are reformulations of standard SVMs, closely related to the regularization networks and the Gaussian processes, but additionally emphasizing and exploiting primal-dual interpretations from optimization theory [Suykens et al., 2003]. The LS-SVM algorithm was designed for solving both classification and regression problems. For classification problems, other similar paradigms such as SVM have been presented in the literature as more effective methods. However, LS-SVMs have shown excellent performance in different applications for regression problems. Since the variable to be predicted in this work is based on the accuracies of the alignments (continuous variable), the application of LS-SVM would be an effective and faithful solution.

The cost function of this paradigm is based on a regularized least squares function with equality constraints, requiring to solve a linear system (such as Karush-Kuhn-Tucker) instead of the quadratic programming (QP) problem provided by the SVM case. This linear system is usually solved by iterative methods like the Conjugate Gradient (CG) algorithm [Hestenes and Stiefel, 1952]. Since this paradigm specially suites well for function approximation (regression problems), the function to model can be then represented according to its primal weight space as follows:

$$\hat{y} = \vec{w}^T \phi(\vec{x}) + b \tag{3.28}$$

where $\vec{w}^T$ and b define the parameters of the model, the function $\phi(\vec{x})$ converts the input space into a higher dimensional feature space, and $\vec{x}$ is a n-dimensional vector including the inputs x_i ($x_i, i = 1, \ldots, n$). Since an approximation problem is addressed, the equation to optimize is formulated as:

$$min_{\vec{w},b,e} J(\vec{w}, e) = \frac{1}{2}\vec{w}^T \vec{w} + \gamma \frac{1}{2} \sum_{i=1}^{N} e_i^2 \tag{3.29}$$

where γ represents the regularization hyper-parameter, N is the number of observations and e_i defines the equality constraints as:

$$e_i = y_i - \hat{y}(\vec{x}_i), i = 1, \ldots, N \tag{3.30}$$

This optimization problem is then solved in dual space, finding the λ_i and b coefficients from:

$$\hat{y} = \sum_{i=1}^{N} \lambda_i K(\vec{x}, \vec{x_i}) + b \tag{3.31}$$

where the kernel function $K(\vec{x}, \vec{x_i})$ is defined as the dot product between the $\phi(\vec{x})$ and $\phi(\vec{x_i})$ mappings. Whether Gaussian kernels are considered, the kernel function $K(\vec{x}, \vec{x_i})$ takes the form:

$$K(\vec{x}, \vec{x_i}) = exp\left[-\left(\frac{\left\|\vec{x_i} - \vec{x}\right\|}{\sqrt{2}\sigma_i}\right)^2\right] \tag{3.32}$$

where σ_i is considered the second hyper-parameter of the problem. As in SVM, the performance of LS-SVM is affected by the values of the hyper-parameters: the width of the kernel (σ_i) and the regularization parameter (γ). The regularization parameter also controls the overfitting instead of decreasing the number of kernels [Rossi et al., 2006]. LS-SVM models using Gaussian kernels are similar to the Radial Basis Function Networks (RBFN) in SVMs but additionally defining an RBF node per data point. These RBF kernels were used in the proposed implementation.

In this specific system, the LS-SVM was applied from the implementation provided in the Matlab® library called *LS-SVMLab* [De Brabanter et al., 2011]. This library consists of different functions for tuning hyper-parameters, training with a specific feature dataset or assessing the designed LS-SVM model. Therefore, the tuning of the hyper-parameters for the RBF kernels is done by this library. Specifically, this tuning is performed by a multidimensional unconstrained non-linear optimization. This optimization finds the optimal hyper-parameters via the Nelder and Mead [1965] algorithm, which does not require any gradient information. Additionally, a cross-validation procedure is also incorporated into this optimization in order to improve the obtained hyper-parameters.

3.5.2 LS-SVM assessment

The proposed LS-SVM model is validated by a 10-fold cross-validation procedure. This procedure randomly divides the complete dataset (2180 problems) into 10 subset of 218 problems. Nine subsets are then applied to train the proposed system. The training procedure includes the most relevant features and the posterior accuracy that was obtained for each problem in the subset. Thus, hyper-parameters are tuned and the LS-SVM model is estimated. Subsequently, the last subset is used to validate the estimated LS-SVM model. The accuracies from such subset are predicted and compared with those already known. The training and validation procedures are repeated ten times, each one leaving for validating different subsets.

The predicted accuracies in the validation phase are assessed by their errors against real accuracy values. The prediction error is measured by means of the *mean relative error* (MRE). The whole LS-SVM prediction is validated by the MRE of the complete dataset. The MRE is calculated as:

$$MRE = \frac{1}{L}\sum_{s=1}^{L} E_R(s) = \frac{1}{L}\sum_{s=1}^{L} \frac{|a_s - \widehat{a_s}|}{a_s} \tag{3.33}$$

where a_s defines the BAliscore value (real accuracy indicator) for the set of sequences s whereas $\widehat{a_s}$ represents the value predicted by the LS-SVM model for the same set of sequences. Thus, $E_R(s)$ determines the *relative error* for the set of sequences s. Additionally, L indicates the total number of predictions made by the LS-SVM model. Since a 10-fold cross-validation procedure is being performed, the total number of predictions L is equivalent to the number of sets of sequences in the dataset.

The MRE also determines the most adequate subset of features chosen in the previous feature selection procedure according to the mRMR criterion. Since the MRE determines the estimation error in our system, the features selected for the LS-SVM model which provides the lower MRE value are then considered the optimal subset of features for the accuracy estimation in PAcAlCI.

3.5.3 Selection of the most suitable methodologies

After the accuracy estimation, PAcAlCI determines the MSA tools that are considered suitable to be used for the alignment of each specific set of sequences. The selection of these tools is done taking into account the previously predicted accuracies of the ten proposed MSA tools for this set of sequences.

Taking into account the previous error value (MRE), a confidence interval is proposed to select the most suitable methodologies. This confidence value (σ_s) is calculated by means of the higher predicted value for a specific set of sequence (the higher accuracy) and the MRE obtained in the prediction. Therefore, those methodologies whose accuracies exceed the confidence value for this specific set of sequences are selected. The σ_s confidence value is measured as:

$$\sigma_s = max\{\widehat{A_s}\} \cdot (1 - MRE) \tag{3.34}$$

where $\widehat{A_s}$ defines a vector including those accuracies which are predicted for all methodologies in the set of sequences. Therefore, those tools whose predicted accuracy is included in the range $[max\{\widehat{A_s}\}, max\{\widehat{A_s}\} - \sigma_s]$ are considered accurate enough to be selected as suitable tools. Thus, a subset of accurate methodologies is provided to the user according to the predicted values of accuracy.

3.6 Experiments and Results

In this section, the results for the PAcAlCI system with the proposed dataset are presented. The dependence of the features with the output accuracy is first analyzed in order to determine the optimal subset of features. Next, the estimation of accuracies in PAcAlCI is studied in detail together with the posterior selection of methodologies. Since it is observed a higher estimation error for inaccurate alignments, a filtering process is also proposed in PAcAlCI. Finally, the PAcAlCI program is qualitatively compared against another similar tool called AlexSys [Aniba et al., 2010].

3.6.1 Determination of significance for biological features

As described above, the mRMR feature selection algorithm [Peng et al., 2005] is applied to determine the optimal subset of relevant features. That procedure returns a ranking of features according to their dependence with the calculated accuracies. An increasingly higher subset of features is progressively included in the posterior prediction algorithm. The resulting ranking is shown in Table 3.3.

According to this ranking, the most relevant features related to the accuracy in alignments are *the number of domains per sequence* (f_7) and *the selected alignment tool* (f_{23}). Regarding the first one, domains can be considered a measure of how deeply sequences are known. Domains are also associated with functional relationships and they involve more conserved regions in sequences. However, it is surprising to observe that these conserved regions do not have to be similar among sequences. Indeed, despite the fact that this feature has been considered the most relevant, the feature measuring the common domains among sequences (f_8, *shared domains*) has not been as relevant as the previous one ($15th$ position). Then, although the occurrences of domains are an acceptable measure to determine the accuracy in the alignments, it is not strictly necessary that these domains are shared in the aligned sequences. On the other hand, the second feature is an essential variable because it is including obligatory information. This feature must always be considered in order to know for which methodology the prediction is done, developing a more robust and coherent system of prediction in PAcAlCI.

The features associated with sequences, *the number of sequences* (f_1) and *the average/variance of the sequence length* (f_2, f_3), were also ranked among first positions in the ranking: $3rd$, $4th$ and $6th$ features, respectively. These features have been highlighted because the ability to obtain accurate alignments directly depends on the sequence properties. Thus, sets with more and more longer sequences are habitually harder to align and more complex methodologies could be necessary. Also, high variable sets according to their sequence lengths are

TABLE 3.3: Feature ranking obtained by the mRMR procedure taking into account their dependence against the accuracies in the alignments by means of the maximal relevance and minimal redundancy criteria.

RANK	FEATURE	
1	f_7	Domains
2	f_{23}	MSA Method
3	f_1	Sequences
4	f_2	Average length
5	f_{22}	Reference subset
6	f_3	Variance length
7	f_5	Amino acids in β-strand
8	f_{21}	Acid amino acids
9	f_{17}	Polar amino acids
10	f_{19}	Basic amino acids
11	f_9	GO terms
12	f_{18}	Non-polar amino acids
13	f_{20}	Aromatic amino acids
14	f_{14}	3D-Structures
15	f_8	Shared Domains
16	f_4	Amino acids in α-helix
17	f_{10}	MF-GO terms
18	f_{13}	Shared GO terms
19	f_{12}	BP-GO terms
20	f_{11}	CC-GO terms
21	f_{15}	Sequences with 3D-structures
22	f_6	Amino acids in transmembemrane
23	f_{16}	Shared 3D-structures

also difficult to align and require more complex gap schemes in the methodologies to model deletion and insertion processes. Finally, it is important to highlight that these feature are easily calculated since they are measured directly from the sequences, without accessing any database. Nevertheless, not only features considering sequence properties but also including amino acid information are found in the first half of the ranking. For instance, features such as *types of amino acids* ($f_{17} - f_{21}$) or *the secondary structure* (f_4, f_5) are acceptably classified in this ranking. They have been considered significant because they provide complementary information about the composition and conformation of sequences. Thus, they have also resulted in helpful data to efficiently predict some similarities in sequences.

Additionally, analyzing the occurrences of proposed features in the BAliBASE sequences could help to understand how the PAcAlCI program will work in different situations. Thus, it is widely known that BAliBASE sequences have often well-annotated secondary structures (α-helix or β-strand), domains or GO terms. Nevertheless, there are a few set of sequences from BAliBASE where some of these properties are not available in the consulted databases; thereby, cases without this information are also considered in PAcAlCI. Consequently, the accuracy for novel sets of sequences not including some of the specified features could still be accurately estimated, returning the most adequate methods for these sequences. On the other hand, other features are clearly less frequent in BAliBASE sequences (e.g. *transmembrane regions*). In this cases, the significance obtained for these features is considered irrelevant and the selection procedure usually discards these feature (the *transmembrane amino acids* feature was ranked in the 22*nd* position).

3.6.2 Estimation of accuracies for MSA tools

In the following step, features previously analyzed are progressively added to the subsequent LS-SVM model. PAcAlCI then estimates the accuracy that each methodology returned for every set of sequences. Specifically, PAcAlCI is performed using an incremental combination of features in ascendant relevance order according to the previous mRMR ranking (Table 3.3). Such combination is applied by adding one feature at a time. Finally, a 10-fold cross-validation procedure is performed to assess the algorithm with each subset of feature. For every incremental subset, the prediction error (MRE) was calculated for the training and validation sets. Therefore, the subset of feature that minimizes the prediction error is considered the optimal subset for the estimation of accuracies in the system.

The evolution of MRE for each combination of features is depicted in Figure 3.2. This evolution shows that the error progressively decreases with regard to the number of features included in the prediction. However, an almost minimal value is reached at around 10 features. The prediction MRE is then kept around

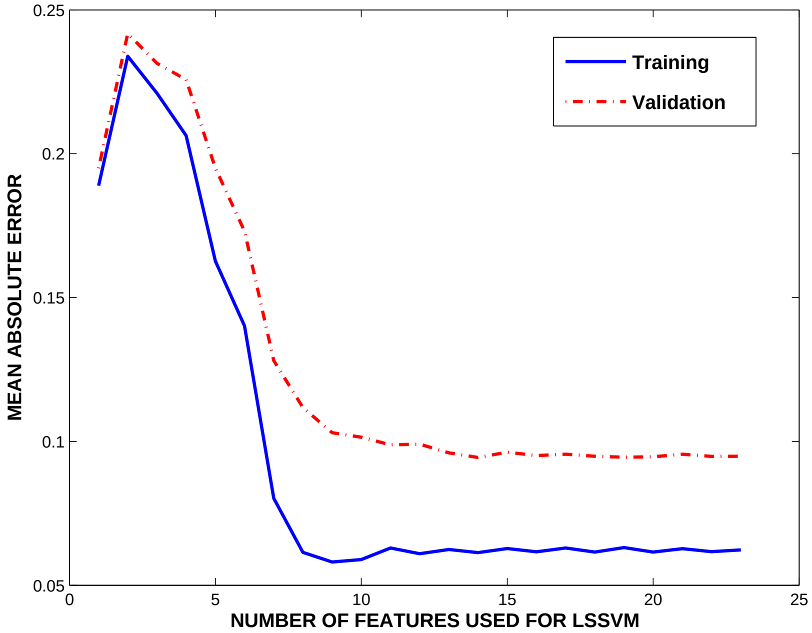


FIGURE 3.2: Evolution of the mean relative error. The number of features progressively increases in ascendant relevance order. The training and validation errors are shown.

6% for the training and 10% for the validation data. So, it could be suggested that all features are not necessary to obtain the optimal prediction. A smaller number of features can then be used to perform PAcAlCI without lack of accuracy. Specifically, the 10 most relevant features are considered for the final LS-SVM model implemented in PAcAlCI. Taking into account these 10 features, the predicted accuracies of four representative BAliBASE sets of sequences are shown in Table 3.4. The total MRE value returned by PAcAlCI with these 10 features is 0.0587 for the training set and 0.1012 for validation. This error is distributed along the 2180 predicted accuracies as shown in Figure 3.3.

3.6.3 Filtering alignments with low accuracies

Analyzing more deeply the prediction with 10 relevant features, it is observed that higher error values are less frequent (see detail in Figure 3.3). Moreover, these errors are usually associated with less accurate alignments. Alignments

TABLE 3.4: Example of predicted accuracies for four different sets of sequences. Predicted accuracies are compared to those real ones (BAliscore values) obtained by each methodology. Values in bold represent those tools that are selected for each example according to their accuracies and the confidence interval (see section 3.5.3 for details). Therefore, these tools are considered suitable for aligning these sequences. The prediction relative error is also shown (see $E_R(s)$ definition in Equation 3.33).

Reference set	Method	Real Acc.	Pred. Acc.	Rel. Error
RV11 4^{th}	3DCoffee	**0.786**	**0.648**	0.176
	Promals	**0.748**	**0.707**	0.055
	ProbCons	0.623	**0.684**	0.097
	TCoffee	0.612	0.598	0.024
	Muscle	0.600	0.384	0.360
	Kalign	0.573	0.617	0.077
	Mafft	0.526	0.625	0.188
	FSA	0.439	0.416	0.053
	RetAlign	0.388	0.277	0.287
	ClustalW2	0.196	0.529	1.699
RV11 20^{th}	3DCoffee	**0.854**	**0.735**	0.139
	Promals	**0.817**	**0.800**	0.020
	Mafft	0.692	0.652	0.058
	ProbCons	0.681	0.699	0.027
	TCoffee	0.654	0.604	0.077
	ClustalW2	0.652	0.579	0.113
	RetAlign	0.633	0.527	0.168
	Kalign	0.600	0.682	0.137
	Muscle	0.592	0.604	0.020
	FSA	0.532	0.631	0.186
RV40 24^{th}	Promals	**0.692**	**0.626**	0.096
	Mafft	**0.676**	**0.692**	0.023
	Kalign	**0.631**	0.562	0.110
	3DCoffee	0.575	0.611	0.064
	TCoffee	0.575	0.556	0.033
	ProbCons	0.568	0.598	0.053
	FSA	0.533	0.533	0.000
	Muscle	0.514	0.515	0.002
	RetAlign	0.511	0.552	0.080
	ClustalW2	0.496	0.438	0.117
RV50 10^{th}	Promals	**0.865**	**0.786**	0.092
	Mafft	**0.795**	**0.754**	0.052
	ProbCons	**0.794**	**0.755**	0.050
	3DCoffee	**0.781**	**0.806**	0.031
	TCoffee	**0.779**	0.711	0.088
	Kalign	0.737	**0.750**	0.017
	FSA	0.591	0.641	0.085
	Muscle	0.529	0.709	0.340
	RetAlign	0.511	0.631	0.235
	ClustalW2	0.483	0.577	0.194

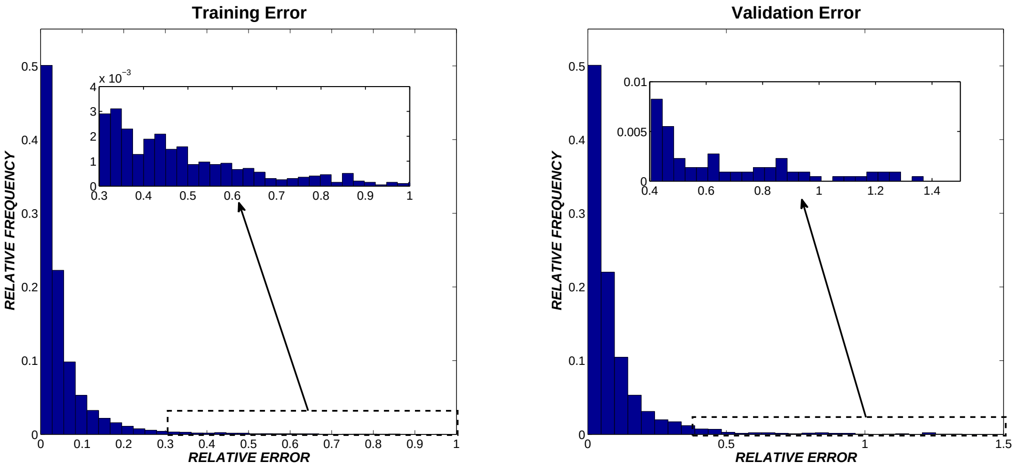


FIGURE 3.3: Distribution of relative errors for training and validation sets. The corresponding LS-SVM prediction was performed using 10 features. The *relative frequency* is represented in the Y-axis (number of samples with their errors in each range normalized by the total number of samples in the dataset). In the case of the validation error The highest errors with less frequency are enlarged to be appreciated in detail.

with low accuracies are less meaningful in our algorithm, as the MSA tools which perform them will not be considered adequate and, therefore, they will not be selected. Consequently, these alignment could be even removed from the PAcAlCI implementation only leaving the alignment from those methodologies that provides high accuracies for each particular set of sequences.

Considering this observation, PAcAlCI is redesigned in order to include a minimal accuracy value as threshold (α). Then, the LS-SVM model in PAcAlCI is only implemented with those alignments that exceed this threshold, filtering out the remaining ones. This filtering approach has achieved a significant improvement in the subsequent prediction. For instance, for $\alpha = 0.5$, the dataset included in PAcAlCI was reduced by 12% (265 alignments showed an BAliscore value inferior to 0.5). In this case, the MRE value using 10 input features decreases to 0.0340 in the training set and to 0.0608 in the validation. That is, error values are reduced by >2% and >4% for the training and validation processes respectively. Although other α thresholds have also been addressed, results for $\alpha = 0.5$ provides the least estimation errors. Besides, this threshold has already been considered in other systems to differentiate between accurate and inaccu-

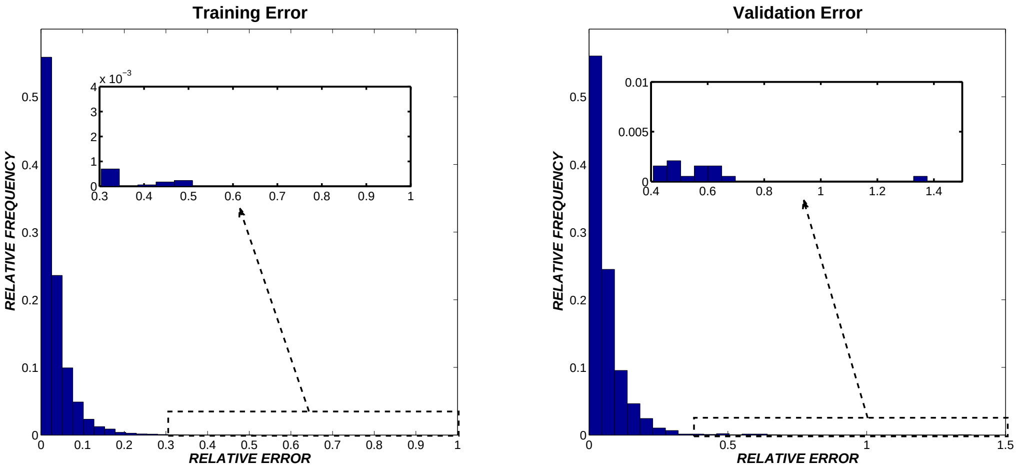


FIGURE 3.4: Distribution of relative errors for training and validation processes by filtering out low accurate alignments. The *relative frequency* is represented in the Y-axis (number of samples with their errors in each range normalized by the total number of samples in the dataset). The highest errors with less frequency have now been almost removed. The LS-SVM prediction is then improved, avoiding predictions with high errors ($\alpha = 0.5$).

rate alignments [Aniba et al., 2010].

The MRE error has been optimized with this procedure because inaccurate alignments, which led to highly wrong estimations, have been previously filtered (see the new distribution of errors in Figure 3.4). Prediction errors can now be considered low enough to adequately determine differences between methodologies and, therefore, to decide the most suitable tools for each specific set of sequences. The selection of the MSA tools is also positively affected by the filtering process given that the wrong estimations in the accuracy could lead to the selection of false positive MSA tools.

3.6.4 Selection of the most suitable MSA tools

After determining the optimal subset of features and improving the accuracy estimation with a filtering process, the most promising MSA tools can be selected according to the proposed features. Unlike other few researches in the literature [Anderson et al., 2011, Aniba et al., 2010] where just the best method is predicted, a group of highlighted methodologies is selected in PAcAlCI. This approach has been considered more realistic because several tools can obtain quite similar alignments for each specific set of sequences, without significant differences in their predicted accuracies. In these cases, PAcAlCI will provide a wider range of methodologies that can be used.

As previously commented, a confidence interval is defined to decide those methodologies which acceptably align a set of sequences. The confidence interval covers those accuracy values that are higher than a confidence value σ_s (see its formal definition in section 3.5.3). Those methodologies whose accuracies exceed such confidence value were chosen as candidate methods.

In order to assess the selection procedure, the confidence interval is applied to both BAliscore values (real accuracies) and accuracies predicted by PAcAlCI. Two sets of suitable methodologies are then retrieved (real and predicted sets). The number of selected methodologies is variable for each set of sequences, as it depends on how similar alignments are obtained by the different tools. Both groups (real and predicted ones) are then compared in order to determine how many methodologies included in the real set are correctly selected in the predicted set (see the Venn's diagrams for four representative sets in Figure 3.5). In order to assess the performance of the proposed prediction, the *precision* indicator is calculated. This measure is determined as the percentage of methods that are correctly selected (*true positive* or TP) according to all the methods selected by the prediction (*true positive* and *false positive* or TP+FP). This indicator was also calculated to assess the performance of the AlexSys tool (see comparison in the following section). Therefore, a total precision of 83.6% is obtained when performing PAcAlCI for 10 features and without α threshold. Moreover, when inaccurate alignments are filtered with $\alpha = 0.5$, the precision value increases to 85.9%. Therefore, it can be suggested that the proposed system is usually determining an acceptable group of outstanding methodologies.

Another interesting analysis is to determine how the selection of methodologies is affected by the chosen biological features. Thus, as shown in Figure 3.5, methodologies including additional information, namely 3DCoffee and

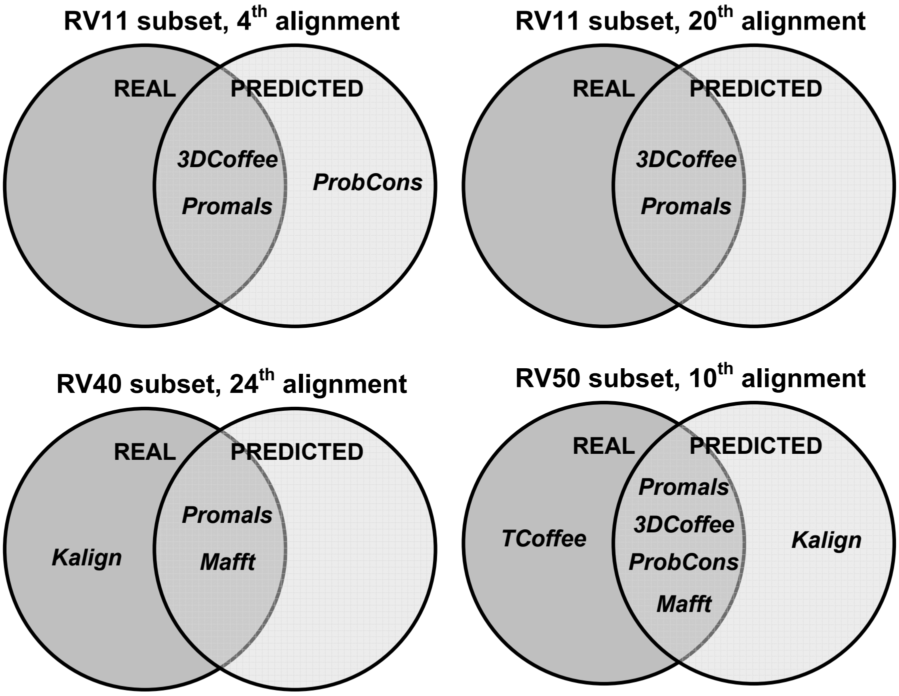

RV11 subset, 4th alignment
REAL
PREDICTED
3DCoffee
Promals
ProbCons
RV11 subset, 20th alignment
REAL
PREDICTED
3DCoffee
Promals
RV40 subset, 24th alignment
REAL
PREDICTED
Kalign
Promals
Mafft
RV50 subset, 10th alignment
REAL
PREDICTED
TCoffee
Promals
3DCoffee
ProbCons
Mafft
Kalign

FIGURE 3.5: Intersection of real and predicted suitable methodologies (Venn diagrams) corresponding to the four alignments whose accuracies are shown in Table 3.4.

Promals, are frequently selected in PAcAlCI for alignment with less evolutionarily related sequences (RV11 subset, <20% identity). In these cases, more commonly used aligners (ClustalW, Kalign or Muscle) are inappropriate, as they are not able to build accurate enough alignments. However, these more complex tools (3DCoffee and Promals) do not always outperform other simpler but faster methods when sequences are more related. For instance, other methodologies as Mafft, T-Coffee, Kalign or ProbCons have also been selected by PAcAlCI in other sets of sequences (see examples for RV40 and RV50 subsets in Figure 3.5). Consequently, we could again suggest that the prediction algorithm is working as expected.

Other additional conclusions can be deduced from this study. Apart from 3D-Coffee and Promals, those datasets including more than two domains per sequence usually selects Kalign as one of the suitable methods (in the 80.95% of cases), whereas Mafft is considered appropriate for datasets with less domains

(78% of datasets selected it). Regarding the number and the length of sequences, large sets (>50 sequences or >400 amino acids of average length) usually pick Mafft or, to a lesser extent, Kalign (90.17% and 71.9%, respectively), while ProbCons is chosen for shorter datasets (62.89%). Finally, Kalign also suits when the sequence length in the set has a high variability (a difference of more than 100 amino acids in average among sequences) and ProbCons for low variability (69.05% and 65.89% of cases, respectively).

3.6.5 Qualitative comparison against AlexSys

Although there are other expert systems to select the appropriate MSA tools for particular sequences [Anderson et al., 2011, Thompson et al., 2006], PAcAlCI is compared to AlexSys [Aniba et al., 2010], since it performs a more similar strategy (see qualitative comparison in Table 3.5). AlexSys proposes a decision-tree algorithm to predict whether sequences are *strongly* or *weakly* aligned according to each specific methodology. This criterion is established in AlexSys by defining a threshold for the accuracies of the alignments: those alignments exceeding an accuracy of 0.5 are considered *strong* whereas those with lower accuracies are considered *weak*. The best method among those classified as *strong* is then inferred according to their success probability or their required CPU time.

However, the quantitative comparison of both methodologies is hard. Although both algorithms develop similar machine learning approaches, their objectives are quite different. Given that AlexSys defines a *strong* vs. *weak* prediction, a classification problem with binary solution is being addressed. This binary prediction can be quite subjective in some cases. Since accuracies over 0.5 are already classified as *strong*, quite distant accuracies, e.g. 0.5 and 0.9, are considered identical for the AlexSys approach. In a different way, PAcAlCI firstly predicts real accuracy values, hence, a regression problem is being addressed. Thus, directly predicting the accuracies provides a relevant improvement in order to decide before whether it is worth aligning with a specific methodology. Anyway, the most suitable methods are also selected in PAcAlCI considering the best predicted accuracies. According to Aniba et al. [2010], AlexSys correctly

TABLE 3.5: Qualitative comparison between two similar tools (PAcAlCI and AlexSys) to select adequate sequence aligners. The performance and main attributes of both procedures are shown.

Feature	**PAcAlCI**	**AlexSys**
number of aligners	10	6
benchmarks	BAliBASE (218 sets)	BAliBASE + OxBench (218 + 672 sets)
kind of problem	regression (real)	classification (binary)
machine learning strategy	LS-SVM	Decision Trees
values of prediction	Accuracies	Weak (accuracy$<$ 0.5) Strong (accuracy$>$ 0.5)
prediction rate	83.6% (α = 0) 85.9% (α = 0.5)	45.0% (1^{st}*aligner*) 45.5% (2^{nd}*aligner*)

predicts the best aligner in a 45% of its test alignments. In another 45.5% of the alignments, the best aligner corresponds to the second predicted method. In general, these prediction rates are quite similar in PAcAlCI taking into account that a wider range of methodologies can be selected in this case (83.55% or 85.89% of methods are correctly selected depending on the α threshold). Regarding the number of considered tools, PAcAlCI gives the chance of selecting a larger number of methodologies (ten approaches against the six of AlexSys), including more sophisticated ones as 3DCoffee or Promals.

Despite these differences, both methods may be considered complementary, as both perform accurate classifiers but in different contexts. AlexSys only provides one tools which is likely to be one the the two most adequate aligners whereas PAcAlCI estimates a group of suitable alignments. In any case, the final decision of selecting the most suitable methodology among the proposed ones can rely on the final user of this system. These tools have been performed to support this decision, providing information about which methodology is likely to build an alignment of high quality. Other criteria that have not been directly included in PAcAlCI, such as the complexity of parameters to configure the methodologies or the required time could also be taken into account in order to choose the correct methodology among the group of selected ones. Thus, the user could consider other properties to decide the used aligner according to his own requirements and criteria.

3.7 Conclusions and Recommendations

In this chapter, a novel program called PAcAlCI for the a priori estimation of the accuracy in several alignments and the posterior selection of the most appropriate MSA tools according to these accuracies has been presented. This program has been performed by a carefully extracted dataset of biological features and a machine learning algorithm based on LS-SVM models.

The final implementation of PAcAlCI has considered a subset of 10 relevant features considering, for example, information related to domains, sequences or secondary structures. Additionally, a filtering procedure has been performed to remove the alignments that were inaccurate and were incorporating high errors in the prediction algorithm.

Finally, some additional instructions about how PAcAlCI works can be found at the website http://www.ugr.es/~fortuno/pacalci.htm or the associated publication [Ortuño et al., 2013]. Please, note that, for a successful running of the PAcAlCI program, it is necessary to have the following considerations:

- This program requires as unique input variable the set of sequences to be aligned. This set of sequences must be incorporated in a file in the standard FASTA format and the path for this file is provided to the PAcAlCI function. Given that PAcAlCI has been trained with BAliBASE sequences, it is necessary to identify the sequences included in the FASTA file with their Uniprot accession or the PDB identifier.
- It is important to remind that PAcAlCI requires the usage of the *LS-SVMLab* library [De Brabanter et al., 2011]. This library must then be downloaded and incorporated to the Matlab® path.
- The PAcAlCI program returns two different variables: *(i)* the ten proposed methodologies and their corresponding predicted accuracies; and *(ii)* the methodologies that have been considered relevant for the query of sequences.

Chapter 4

Regression models for an intelligent estimation of the MSA quality

4.1 Introduction and motivation

The accurate evaluation of multiple sequence alignments (MSAs) is still an important challenge in bioinformatics which has not been properly solved yet. Although several scores have continuously been proposed with this purpose, they do not agree about which are the most accurate alignments or how they must be adequately evaluated.

As previously stated, MSAs have been traditionally scored by using weighted matrices like PAM [Dayhoff et al., 1979] or BLOSUM [Henikoff and Henikoff, 1992]. Briefly, these matrices are normally associated with the probability of finding specific mutations between each pair of possible amino acids. These matrices are still widely applied in the initial stages of several MSA tools, mainly to obtain preliminary pairwise alignments in progressive and consistency-based methodologies. However, since these matrices only take into account the sequence information (nucleotides or amino acids), they do not achieve a sufficiently accurate score, specially with less related sequences [Liu et al., 2009].

For this reason, more recent scores are trying to improve the classical weighted matrices by adding supplementary biological information. It has already been shown in this dissertation (see Chapter 2, section 2.4) that considering data like homologies or protein structures to build alignments can positively affect their quality. Therefore, these features could also be helpful to determine such quality. For instance, novel scores like the *Contact Accepted mutatiOn* (CAO) [Lin et al., 2003] or STRIKE [Kemena et al., 2011] determine the molecular contacts between amino acids in the 3D structure to estimate the quality of alignments. In the same way, other alternative evaluations add a set of features related to secondary structure, homologies and distribution of gaps [Ahola et al., 2008, Kececioglu and DeBlasio, 2013]. These new algorithms take advantage of estimating their scores by learning from the evaluation tools of standard benchmarks like BAliscore in BAliBASE [Thompson et al., 2005] or Q-Score in OxBench [Raghava et al., 2003]. It is important to remind that these benchmarks evaluate alignments by determining their similarity with a set of manually obtained reference alignments. Therefore, new scoring algorithms seek to estimate the alignment quality trying to be highly correlated to the benchmark evaluation but being also applicable when there is no reference alignments.

In this chapter, several novel scoring algorithms are presented and analyzed to evaluate MSAs. Since the principal goal is to estimate a quality score similar to those evaluations from benchmarks described above, a regression problem is again being addressed. In this case, several mathematical and supervised learning approaches are going to be proposed and compared to determine the most efficient score. The procedures of the different scoring systems are schematically presented in Figure 4.1. Although these algorithms follow a scheme analogous to the PAcAlCI tool presented in Chapter 3, the objectives for both systems are quite different. While the main objective in PAcAlCI was to estimate the alignment accuracy to know which alignment tool may be better for each specific set of sequences before aligning them, the scoring schemes presented here aim to estimate the quality of already performed alignments, independently of the used tool. Following, the common properties and main differences between both processes are clarified:

- **Alignment Dataset:** The datasets in both cases are composed of the 218 sets of sequences provided by BAliBASE. In the PAcAlCI tool, the input dataset only included the sequences in BAliBASE whereas the subsequent alignments were only applied to calculate the quality of each set of sequences with each particular tool. In this case, the corresponding alignments obtained from the same 10 methodologies are directly added to the dataset. Moreover, the dataset of alignments presented in this chapter has been extended with additional sets of sequences from the OxBench benchmark [Raghava et al., 2003]. Such benchmark incorporates to the system the alignments corresponding to other 336 sets of sequences. Therefore, each MSA tool aligns a total of 554 sets of sequences.
- **Feature Extraction:** Both proposals take advantage of an heterogeneous dataset of features accurately retrieved. These features are mostly collected from the same databases: Uniprot [The UniProt Consortium, 2014], Pfam

[Finn et al., 2014] or PDB [Berman et al., 2000]. However, although both datasets are based on similar biological properties, the features proposed here are calculated for the obtained alignments instead of only sequences. Anyway, some features related to the size of the sets of sequences are identically included in both datasets.

- **Feature Selection:** Although both systems include a feature selection module based on mutual information, the *Normalized Mutual Information Feature Selection* (NMIFS) algorithm [Estevez et al., 2009] has been chosen here instead of mRMR. This feature selection normalizes the standard mutual information measurement trying to avoid the inclusion of irrelevant features.
- **Regression:** A range of different regression algorithms are presented and deeply analyzed in this chapter instead of only the LS-SVM model. The main purpose here is to compare them in order to determine which one could result in a more efficient scoring scheme. Specifically, the following regression approaches are considered in addition to LS-SVM: regression trees, bootstrap aggregation trees and Gaussian processes.

The chapter is then structured as follows. Section 4.2 describes the dataset of sequences that is retrieved from the OxBench and the BAliBASE benchmarks and their corresponding alignments. The measurements that are proposed to build the dataset of features related to these alignments are then presented and explained in detail in section 4.3. In Section 4.4, the applied feature selection procedure is defined and widely described. Section 4.5 is dedicated to the definition of the four implemented regression models for the prediction of accuracies in alignments. Section 4.6 is focused on describing the implementation and validation of the scoring schemes. In section 4.7 an additional procedure for the validation of the proposed scoring schemes based on pairwise comparisons of alignments is explained in detail. The main findings and experimental results are then shown in section 4.8. Finally, the most important conclusions retrieved from this chapter are highlighted in section 4.9.

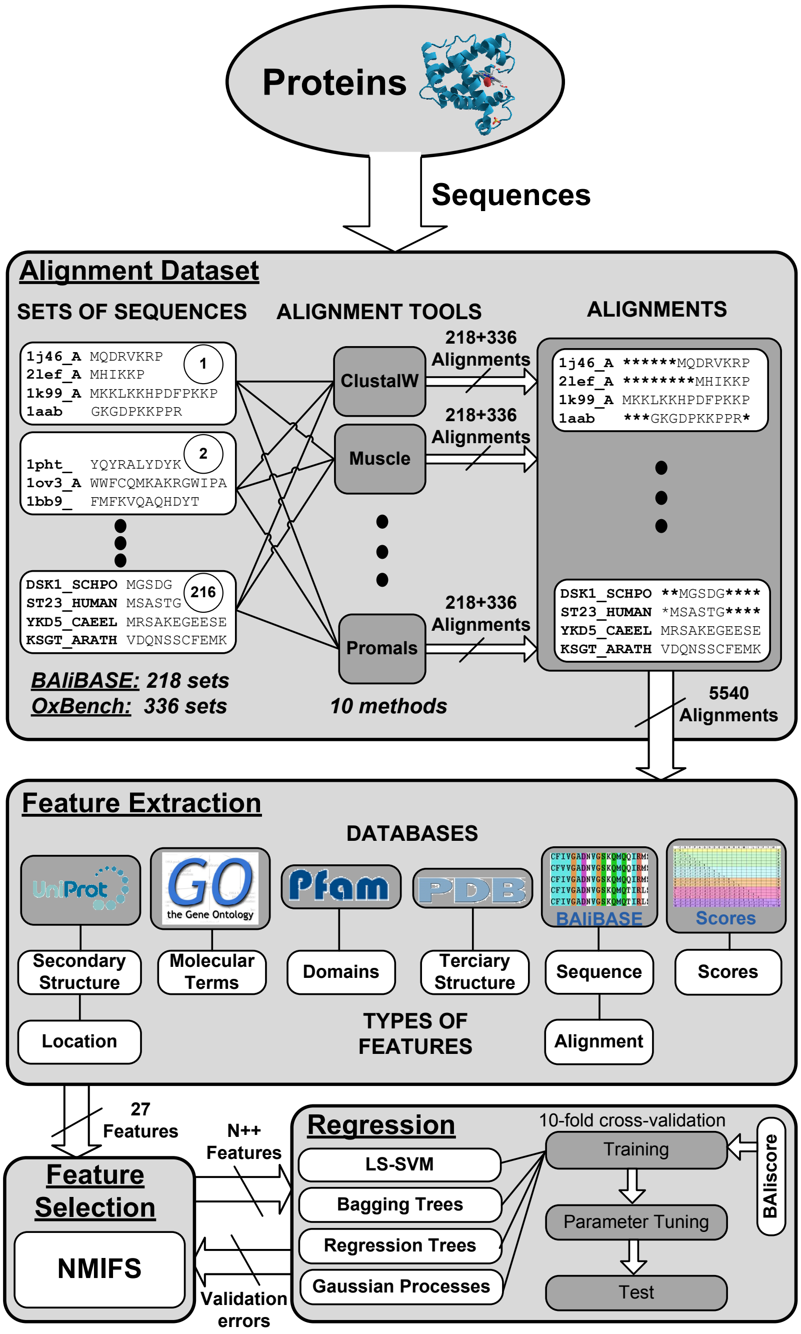

Proteins
Sequences
Alignment Dataset
SETS OF SEQUENCES
ALIGNMENT TOOLS
ALIGNMENTS
1j46_A MQDRVKRP
2lef_A MHIKKP
1k99_A MKKLKKHPDFPKKP
1aab GKGDPKKPPR
1
1pht_ YQYRALYDYK
1ov3_A WWFCQMKAKRGWIPA
1bb9_ FMFKVQAQHDYT
2
DSK1_SCHPO MGSDG
ST23_HUMAN MSASTG
YKD5_CAEEL MRSAKEGEESE
KSGT_ARATH VDQNSSCFEMK
216
ClustalW
Muscle
Promals
218+336 Alignments
218+336 Alignments
218+336 Alignments
1j46_A ******MQDRVKRP
2lef_A ********MHIKKP
1k99_A MKKLKKHPDFPKKP
1aab ***GKGDPKKPPR*
DSK1_SCHPO **MGSDG****
ST23_HUMAN *MSASTG****
YKD5_CAEEL MRSAKEGEESE
KSGT_ARATH VDQNSSCFEMK
BAliBASE: 218 sets
OxBench: 336 sets
10 methods
5540 Alignments
Feature Extraction
DATABASES
UniProt
GO the Gene Ontology
Pfam
PDB
BAliBASE
Scores
Secondary Structure
Molecular Terms
Domains
Terciary Structure
Sequence
Scores
Location
TYPES OF FEATURES
Alignment
27 Features
N++ Features
Feature Selection
NMIFS
Validation errors
Regression
10-fold cross-validation
LS-SVM
Bagging Trees
Regression Trees
Gaussian Processes
Training
Parameter Tuning
Test
BAliscore

FIGURE 4.1: Flowchart of the full proposed implementation. The regression procedures were implemented including several stages: *(i)* The *Alignment Dataset* is constructed by aligning the BAliBASE and OxBench sets of sequences with 10 different aligners, *(ii)* the *Feature Extraction* retrieves features associated with several biological properties from different sources and databases, *(iii)* the *Feature Selection* applies the NMIFS algorithm to determine the subset of the most relevant features and *(iv)* four different regression models are performed to estimate the quality of alignments in the *Regression* stage.

4.2 Construction of the alignment dataset

The dataset of alignments considered in this chapter is based on the previously defined one for the PAcAlCI tool. This dataset was composed by 218 sets of sequences provided by the BAliBASE benchmark [Thompson et al., 2005], each one aligned by using ten different tools. The aligners were carefully selected among well-known and widely used MSA tools including progressive algorithms (ClustalW, Muscle, Kalign, Mafft and RetAlign), consistency-based methods (T-Coffee, ProbCons and FSA) and tools with additional data (3D-Coffee and Promals). The original dataset was then composed by a total of 2180 alignments (see section 3.2 for details about this dataset).

In addition to this original dataset, the OxBench benchmark [Raghava et al., 2003] is also incorporated here. Similarly to BAliBASE, OxBench is an environment with several families of sequences together with their reference alignments (see section 2.3 for details). Specifically, the subset of 336 accurate OxBench sets of sequences included in the *Bench* suite [Edgar, 2009] are considered. *Bench* is a repository which collects the most accurate protein sequence alignments of several benchmarks, e.g. OxBench or BAliBASE. The sequences from OxBench are also aligned with the ten MSA tools commented above. Therefore, the new extension of the alignment dataset is composed by 3360 alignments.

An extended dataset of 5540 different alignments (218+336 sets of sequence by 10 tools) is then applied in this chapter. Similarly to PAcAlCI, these alignments are also evaluated by the BAliscore tool, in order to determine their similarity with respect to the reference alignments (both BAliBASE or OxBench references). These evaluations are the basis (output variable) to apply the regression approaches which estimate the quality scores for alignments.

4.3 Dataset with the extraction of alignment features

As previously commented, the PAcAlCI tool (Chapter 3) included a dataset of 23 features to calculate measurements related to the sequences in each set before being aligned (see Table 3.2). A similar dataset is then created here but, in this case, the proposed features are based on the alignments. Nevertheless, the three first features in PAcAlCI which were related to the size of the sets of sequences are also considered in the new dataset (see section 3.3.1 for details). Specifically, these features were related to the number of sequences in each set (f_1), the average length of these sequences (see f_2 in Equation 3.1) and, finally, the variance of the length (see f_3 in Equation 3.2).

In addition to that, since the aim in this work is to determine the quality of each alignment, other 24 features more related to the alignments are proposed. These features are carefully designed trying to be specially relevant for the estimation of qualities in alignments. To the best of our knowledge and after a careful revision of the literature, such a wide feature dataset has not been applied before for alignment evaluations.

The additional 24 features are obtained from the same databases as the PAcAlCI dataset: Pfam [Finn et al., 2014], PDB [Berman et al., 2000] and Uniprot [The UniProt Consortium, 2014]. Each database is consulted to retrieve information about particular biological properties (see *'Feature Extraction'* stage in Figure 4.1). Just as a reminder, Pfam provides information about functional regions in protein sequences (domains), Uniprot gives the secondary structures of the proteins and tertiary structures are retrieved from PDB. The Gene Ontology database has not been applied here because the terms in this vocabulary are not specified for particular regions in the sequences and, therefore, no information can be retrieved from alignments.

Some other typical measurements associated with the quality of alignments are also extracted, namely percentages of gaps, identities or totally conserved columns. In addition, some of the most used evaluation systems and quality scores in alignments are added: PAM [Dayhoff et al., 1979], BLOSUM [Henikoff and

Henikoff, 1992], RBLOSUM [Styczynski et al., 2008], GONNET [Gonnet et al., 1992] or STRIKE [Kemena et al., 2011].

The whole feature dataset is then composed of 27 heterogeneous features. These new features are summarized in Table 4.1. Most of them are calculated as the percentage of pairwise aligned amino acids that share the same value of a specific property (*matches*). The *match* function of a general property p can be defined as:

$$match(p_a, p_b) = \begin{cases} 1 & if\ p_a = p_b\ and\ p_a \neq 0 \\ 0 & otherwise \end{cases} \tag{4.1}$$

where p_a and p_b represent the specific values of the property p for two amino acids a and b which have been aligned. The p_a and p_b values are zero if the property is unknown or unannotated for these amino acids, or the position contains a gap. Considering an alignment with N sequences and L columns (length of the alignment), the maximum number of possible pairwise matches (M_T) is calculated independently of the considered property as:

$$M_T = L \cdot \frac{N!}{2!(N-2)!} = \frac{L \cdot N \cdot (N-1)}{2} \tag{4.2}$$

Both the *match* function and M_T are taken into account in the definition of most of the new features. A common notation is used to calculate all the features for a generic alignment: L determines the length of the alignment and N the number of sequences. Following, a description of the features applied to construct the novel dataset is presented according to each consulted source or database.

4.3.1 Features directly extracted from alignments

Several simple measurements have been traditionally calculated from alignments as possible criteria related to the quality of such alignments. Three of these measurements have been included in the novel dataset:

TABLE 4.1: Summary of the 27 features associated with alignments. The three first features ($f_1 - f_3$) are incorporated from the PAcAlCI dataset (see Table 3.2). *Matches* refer to the percentage of paired amino acids with a particular property in common that have been aligned (Equation 4.1).

	FEATURE	SOURCE	RANGE	TYPE
f_1	# Sequences	Sequences	[4, 142]	Int
f_2	Average length	Sequences	[43, 1630]	Real
f_3	Variance length	Sequences	$[0, 2.47\times10^6]$	Real
f_4	Identities	Alignments	[0, 1]	Real
f_5	Gaps	Alignments	[0, 1]	Real
f_6	Totally conserved columns	Alignments	[0, 1]	Real
f_7	α-helix 2^{nd} structure matches	Uniprot	[0, 1]	Real
f_8	β-strand 2^{nd} structure matches	Uniprot	[0, 1]	Real
f_9	$Turn$ 2^{nd} structure matches	Uniprot	[0, 1]	Real
f_{10}	Unknown 2^{nd} structure matches	Uniprot	[0, 1]	Real
f_{11}	Secondary structure matches	Uniprot	[0, 1]	Real
f_{12}	Pfam-A domain matches	Pfam	[0, 1]	Real
f_{13}	Pfam-B domain matches	Pfam	[0, 1]	Real
f_{14}	Total domain matches	Pfam	[0, 1]	Real
f_{15}	Domain clan matches	Pfam	[0, 1]	Real
f_{16}	3D Contact matches	PDB	[0, 1]	Real
f_{17}	Polar AA matches	Biochemistry	[0, 1]	Real
f_{18}	Non-polar AA matches	Biochemistry	[0, 1]	Real
f_{19}	Basic AA matches	Biochemistry	[0, 1]	Real
f_{20}	Aromatic AA matches	Biochemistry	[0, 1]	Real
f_{21}	Acid AA matches	Biochemistry	[0, 1]	Real
f_{22}	AA Type matches	Biochemistry	[0, 1]	Real
f_{23}	BLOSUM62	Scores	$[-1.7, 39.7]\times10^6$	Real
f_{24}	RBLOSUM62	Scores	$[-1.7, 39.7]\times10^6$	Real
f_{25}	PAM250	Scores	$[-1.9, 30.5]\times10^6$	Real
f_{26}	GONNET	Scores	$[-19.5, 30.5]\times10^6$	Real
f_{27}	STRIKE	Scores	[-0.3 6.3]	Real

- **Percentage of identities** (f_4): the identity percentage is the simpler and more traditionally way of evaluating alignments. It basically counts the number of coincidences in the aligned amino acids. Although this value does not provide an accurate enough measure of quality by itself, it could be an useful information if it is integrated in the feature dataset together with other complementary features. This value is expressed as:

$$f_4 = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(s_{ij}, s_{ik})}{M_T} \tag{4.3}$$

where s_{ij} and s_{ik} are the residue symbols in the i-th position of the j-th and k-th sequences, respectively.

- **Percentage of gaps** (f_5): the overuse of gaps to increase the identity percentage in alignments can lead to a negative effect, even reducing the real quality of the alignment [Nozaki and Bellgard, 2005]. Therefore, the number of gaps in an alignment could be useful to help estimate this correct quality. This feature is formally calculated as:

$$f_5 = \sum_{i=1}^{L} \sum_{j=1}^{N} \frac{isGap(s_{ij})}{N \cdot L} \tag{4.4}$$

where s_{ij} determines the symbol found in the i-th position of the j-th sequence in the alignment. The function *isGap* for a general symbol s in the alignment is defined as:

$$isGap(s) = \begin{cases} 1 & if\ s = \text{'-'}\ (gap) \\ 0 & otherwise \end{cases} \tag{4.5}$$

- **Percentage of totally conserved columns** (f_6): this measure is also considered in several algorithms as a good criterion for the quality of alignments. It could help penalize those alignment with partially accurate regions but not completely aligned columns, which could result in worse qualities [Mirarab and Warnow, 2011]. This feature is determined as:

$$f_6 = \sum_{i=1}^{L} \frac{totalColumn(S_i)}{L} \tag{4.6}$$

where S_i represents the full i-th column ($S_i = \{s_{ij}\}\ \forall j = 1 \ldots N$) in the alignment and the function $totalColumn(S_i)$ is defined as:

$$totalColumn(S_i) = \begin{cases} 1 & if\ s_{ij} = s_{i1}\ \forall j = 2, \ldots, N \\ 0 & otherwise \end{cases} \tag{4.7}$$

4.3.2 Alignment features extracted from Uniprot

Uniprot [The UniProt Consortium, 2014] is again consulted mainly to determine, in this case, the coincidences of identical secondary structures in the sequences of the alignments. In addition to the two basic secondary structures (α-helix and β-strand) previously considered, a new type of structure has been incorporated in the dataset: *turn* structures. Therefore, the following measurements are calculated:

- **α-helix matches** (f_7): this feature determines the number of amino acids in a α-helix structures that have been aligned with other amino acids in an identical structure are determined. This value is formally expressed as:

  $$f_7 = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(\alpha_{ij}, \alpha_{ik})}{M_T} \tag{4.8}$$

  where α_{ij} and α_{ik} takes the value '1' whether the residues aligned in the i-th position of the j-th and k-th sequences belong to a α-helix structure and '0' otherwise.

- **β-strand matches** (f_8): similarly to the previous feature, this measurement specifies the percentage of pairwise aligned amino acids both included in a β-strand structure. This feature is measured as:

  $$f_8 = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(\beta_{ij}, \beta_{ik})}{M_T} \tag{4.9}$$

  with β_{ij} and β_{ik} determining if the amino acids in the i-th position of the j-th and k-th sequences belong to a β-strand structure ('1' for residues in β-strand and '0' otherwise).

- *Turn* **matches** (f_9): the *turn* structure property repeats a procedure analogous to the previous two features. Thus, the percentage of aligned amino acids sharing a similar *turn* structure is obtained as

$$f_9 = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(\omega_{ij}, \omega_{ik})}{M_T} \tag{4.10}$$

  where ω_{ij} and ω_{ik} are binary variables with the value *true (1)* if the residues in the i-th position of the respective j-th and k-th sequences are part of a *turn* structure and *false (0)* otherwise.

- **Unknown secondary structure matches** (f_{10}): in addition to the three previous types of secondary structure, those amino acids without an annotated secondary structure in Uniprot are also taken into account. Thus, the percentage of matches without a known secondary structure is obtained as:

$$f_{10} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(u_{ij}, u_{ik})}{M_T} \tag{4.11}$$

  where u_{ij} and u_{ik} determine if the residues in the i-th position of the j-th and k-th sequences contain an annotated secondary structure.

- **Total secondary structure matches** (f_{11}): finally, all the possible matches related with the three previous types and unknown secondary structure are gathered in a general feature. This feature is calculated as:

$$f_{11} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(ss_{ij}, ss_{ik})}{M_T} \tag{4.12}$$

  where ss_{ij} and ss_{ik} specify the type of secondary structure to which the amino acids of the j-th and k-th sequences in the i-th position belong. These variables can take the values {'H', 'B', 'T', 'U'} for α-helix, β-strand, *turn* or *unknown* secondary structures, respectively.

4.3.3 Alignment features extracted from Pfam

The domains in Pfam are generally associated to functional regions in the protein sequence. Therefore, the number of domain matches between sequences in the alignment could be related to the quality of such alignment. Since Pfam defines two domain classes (Pfam-A and Pfam-B) and the domains are also gathered in clans, the following features are considered in the proposed dataset:

- **Pfam-A domain matches** (f_{12}): this feature calculates the percentage of aligned amino acids which are in an identical Pfam-A domain. Pfam-A domains define the most accurate and carefully curated families. This percentage of matches is performed as:

$$f_{12} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(Da_{ij}, Da_{ik})}{M_T} \tag{4.13}$$

  where Da_{ij} and Da_{ik} include the Pfam-A domain associated to the amino acids in the i-th position of the j-th and k-th sequences in the alignment (in case the amino acids belong to a Pfam-A domain). Their values refer to the accession number of the amino acid ('PFxxxxx').

- **Pfam-B domain matches** (f_{13}): this measurement is equivalent to the previous one for Pfam-B domains. Pfam-B specifies the families of domains with low quality which are automatically generated. This feature is calculated as:

$$f_{13} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(Db_{ij}, Db_{ik})}{M_T} \tag{4.14}$$

  where Db_{ij} and Db_{ik} specify the Pfam-B domain associated to the amino acids in the i-th position of the j-th and k-th sequences (in case the amino acids belong to a Pfam-B domain). The Pfam-B accession number is applied in this case for each amino acid ('PBxxxxxx').

- **Total domain matches** (f_{14}): the two previous features are gathered in this new measurements. In this case, all the domain matches independently of the kind of families are considered. The value for this feature is calculated as:

$$f_{14} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(D_{ij}, D_{ik})}{M_T} \tag{4.15}$$

  with D_{ij} and D_{ik} being the domain to which each residue respectively belong, independently of the Pfam-A or Pfam-B classes.

- **Domain clan matches** (f_{15}): although two aligned amino acids do not belong to the same domain, their corresponding domains could be related, both being part of a common Pfam clan. Clans are groups of domains whose sequences have high similarity. Therefore, the percentage of aligned amino acids sharing the same clan is measured here as:

$$f_{15} = \sum_{i=1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} \frac{match(C_{ij}, C_{ik})}{M_T} \tag{4.16}$$

  with C_{ij} and C_{ik} defining the clans to which each amino acid respectively belongs. Clans are identified by their accession number ('CLxxxx').

4.3.4 Alignment features extracted from PDB

It has been shown in the PAcAlCI dataset that it is hard to find sequences in BAliBASE with common PDB structures and, therefore, this kind of feature is less predictive (see the ranking of the feature f_{16} in Table 3.3). Consequently, instead of features directly related with the PDB structure, a new feature associated to the contacts involved in such structure is proposed for this dataset:

- **Tertiary Contact matches** (f_{16}): two amino acids in the same sequence are said to be in contact if any of their atoms are close enough that a solvent molecule cannot be inserted between them [Connolly, 1983]. Similarly to the contact definition in Kemena et al. [2011], only contacts involving two amino acids separated by at least five amino acids are considered here to avoid the influence of secondary structures. Contact matches have been shown to be a good indicator of quality in alignment and they are the basis of some previous score algorithms [Kemena et al., 2011, Lin et al., 2003]. Therefore, a new features is proposed to calculate the number of contacts between two amino acids in each sequence that are correctly aligned with other two amino acids in contact from other sequences.

$$f_{16} = \frac{\sum_{i=1}^{L} \sum_{m=i+1}^{L} \sum_{j=1}^{N} \sum_{k=j+1}^{N} match(CT_{imj}, CT_{imk})}{M_{CT}} \tag{4.17}$$

where i and m determine the two positions of a possible contact in the same sequence whereas j and k are each two compared sequences in the alignment. Thus, CT_{imj} and CT_{imk} respectively indicate if there is a contact (1 for *true* and 0 for *false*) between two positions in the alignment (i and m) for each two sequences (j and k). Additionally, M_{CT} determines the maximum number of possible matches between contacts and it is obtained as:

$$M_{CT} = \sum_{j=1}^{N} \sum_{k=j+1}^{N} min\Big[\sum_{i=1}^{L} \sum_{m=i+1}^{L} CT_{imj} \, , \, \sum_{i=1}^{L} \sum_{m=i+1}^{L} CT_{imk} \Big] \tag{4.18}$$

4.3.5 Alignment features extracted from chemical properties

The classification of amino acids proposed by Mathews et al. [2000] according to their chemical properties is considered in this case to calculate the matches in the alignment (see this chemical classification in section 1.2.3). Therefore, five additional features related to this chemical classification are calculated as:

- **Matches of polar uncharged amino acids** (f_{17}):

$$f_{17} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(P_{ij}, P_{ik})}{M_T} \tag{4.19}$$

- **Matches of non-polar aliphatic amino acids** (f_{18}):

$$f_{18} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(NP_{ij}, NP_{ik})}{M_T} \tag{4.20}$$

- **Matches of basic positively-charged amino acids** (f_{19}):

$$f_{19} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(BA_{ij}, BA_{ik})}{M_T} \tag{4.21}$$

- **Matches of aromatic amino acids** (f_{20}):

$$f_{20} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(AR_{ij}, AR_{ik})}{M_T} \tag{4.22}$$

- **Matches of negatively-charged (acid) amino acids** (f_{21}):

$$f_{21} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(AC_{ij}, AC_{ik})}{M_T} \tag{4.23}$$

where *P*, *NP*, *BA*, *AR* and *AC* determine if a specific position in the alignment contains an amino acid respectively classified as polar, non-polar, basic, aromatic or acid (1 if *true*, 0 for $false$). Additionally, the matches of all these types of amino acids are also gathered in one extra feature:

- **Matches of chemical amino acid types** (f_{22}): all the pairwise aligned amino acids that share the same chemical properties are counted in this feature. This measurement is calculated as:

$$f_{22} = \sum_{i=1}^{L}\sum_{j=1}^{N}\sum_{k=j+1}^{N} \frac{match(ch_{ij}, ch_{ik})}{M_T} \tag{4.24}$$

where ch_{ij} and ch_{ik} indicate the type of amino acid in the i-th position of the j-th and k-th sequences in the alignment. These variables can take the values {'P', 'N', 'B', 'A', 'C'} for Polar, Non-polar, Basic, Aromatic and aCid amino acids, respectively.

4.3.6 Alignment features from additional scores

Finally, the proposed dataset is complemented by five of the most used evaluation schemes in the literature. These scores are computationally easy to obtain and they can remarkably contribute to improve the estimation of the alignment quality. The five score schemes are defined as:

- **BLOSUM62 score** (f_{23}) [Henikoff and Henikoff, 1992]: BLOSUM is based on local alignments of very conserved regions in protein families. This score can be calculated from different BLOSUM matrices depending on the percentage of similarity in the considered protein families. Thus, the matrix of the most standard BLOSUM score, BLOSUM62, is built using sequences with more than 62% of similarity.
- **RBLOSUM62 score** (f_{24}) [Styczynski et al., 2008]: RBLOSUM can be considered a bug-fixed version of BLOSUM. The fixed BLOSUM62 matrix is again considered in this case to calculate the RBLOSUM62 score.
- **PAM250 score** (f_{25}) [Dayhoff et al., 1979]: PAM matrix represents the likelihood of mutation between each two aligned amino acids during a specific evolutionary interval. Several matrices can be retrieved from PAM according to the rate of mutation considered. For instance, the highly used PAM250 matrix, which is applied here, estimates its values considering that 250 mutation could occur for each 100 amino acids (rate of 250%). This PAM250 score is usually applied for higher evolutionary distances between sequences (20% similarity). The PAM250 score is generally considered an acceptable score for alignment tools [Orobitg et al., 2013].

- **GONNET score** (f_{26}) [Gonnet et al., 1992]: GONNET proposes another score scheme based on matrices. The idea here is to determine am initial scoring matrix by aligning pairwise sequences with classical distance measures. However, this matrix is then iteratively refined by realigning sequences with the previous matrix and estimating a new matrix with the redesigned pairwise alignments.
- **STRIKE score** (f_{27}) [Kemena et al., 2011]: STRIKE evaluates alignments according to its estimation of contacts in protein structures. The contacts are estimated in STRIKE in the same way as defined for the feature f_{16}. In this case, only the pairs of amino acids which have been aligned in the same position as the estimated contacts of a sequence are evaluated. These pairs of amino acids are scored according to the STRIKE matrix, which is estimated considering the frequency of contacts in several sequence databases. Thus, STRIKE achieves a good performance of the quality in alignments, even being acceptably correlated with the BAliscore tool [Kemena et al., 2011].

4.4 Feature selection with normalized mutual information

As performed in PAcAlCI, the relevance of each feature in the new dataset can be analyzed by using mutual information. The mutual information (MI) calculates the dependence of features without any assumption about their underlying relationship [Cover and Thomas, 2006] (see the formal definition of mutual information in section 3.4.2).

In this case, the normalized mutual information feature selection (NMIFS) [Estevez et al., 2009] procedure was considered instead of mRMR. NMIFS defines a filter method based on a reformulated MI, which is normalized by the minimum entropy. Therefore, for two features x and y the normalized MI is obtained as:

$$NMI(x, y) = \frac{MI(x, y)}{min\{H(x), H(y)\}} \tag{4.25}$$

where $MI(x, y)$ determines the standard mutual information defined in Equation 3.22 for continuous features and Equation 3.23 for discrete ones. Besides, $H(x)$ and $H(y)$ represent the entropy of x and y. The entropy for a general feature x is then calculated as:

$$H(x) = -\int p(x) \log(x)\, dx \tag{4.26}$$

where $p(X)$ is the marginal probability density function (pdf) of the feature X.

The NMIFS selection procedure has been chosen here because it avoids some negative aspects of the mRMR selection. For instance, mRMR procedure may select redundant features after irrelevant features or it could even choose some irrelevant features before relevant ones. These drawbacks are reduced in NMIFS thank to the normalization of the mutual information. This normalization compensates for the MI bias toward multivalued features, and restricts its values to the range [0, 1] [Estevez et al., 2009].

Therefore, NMIFS proposes to substitute the redundancy measure in mRMR by using the average normalized MI. The redundancy of a particular feature x_i with respect to a subset of already selected features (S) was finally calculated according to the NMI definition as follows:

$$R(S, x_i) = \frac{1}{|S|} \sum_{x_s \in S} NMI(x_i, x_s) \tag{4.27}$$

where $|S|$ represents the cardinality of the subset of features. In this case, the redundancy takes values in [0,1]. A value 0 determines that the feature x_i is independent of the subset S and a value 1 indicates the strongest correlation between x_i and all features in S.

A feature x_i is then selected by the NMIFS algorithm if it maximizes the selection criterion determined by G as:

$$max[G];\ G = MI(x_i, y) - R(S, x_i) \tag{4.28}$$

where $MI(x_i, y)$ indicates the relevance of the feature x_i, calculated as the MI of x_i and the output variable y (maximum dependency criterion). The G measure is similar to the criterion proposed by mRMR (see Equation 3.27) but including the normalization of MI in the redundancy term $R(S, x_i)$.

Consequently, similarly to the feature selection procedure designed in PAcAlCI, the most relevant biological features are also selected here from the proposed dataset according to the NMIFS criterion. Each subset is progressively formed by including the following feature in decreasing order with respect to the G criterion. The subsets of features that optimize the posterior estimation errors in each subsequent regression algorithms are then considered.

4.5 Regression models

As previously commented, the purpose of this work is to automatically estimate the qualities of alignments with an score scheme approximately similar to BAliscore. Since the BAliscore is a continuous variable in the range [0, 1], this estimation of qualities is considered a regression problem. Statistically, regression is defined as the process of estimating the relationship between variables. This estimation can be provided by many techniques for modeling and analyzing the variables, mainly focusing on the relationship between a dependent real-value variable which has to be estimated (output) and one or more independent variables (inputs).

Several computational algorithms have been designed specially focused on regression problems. Some of these solutions are implemented in this chapter: regression trees, bootstrap aggregation trees, Gaussian processes and, again, LS-SVMs. Both regression and bagging trees are designed with the *Statistics* toolbox of Matlab® (R2010b version) whereas th LS-SVM model uses the *LS-SVMLab* library [De Brabanter et al., 2011]. On the other hand, Gaussian processes are designed by using the WEKA software [Hall et al., 2009]. These regression approaches are all validated by using a 10-fold cross-validation procedure and they are subsequently assessed with an independent test dataset.

From the four proposed regression solutions, the LS-SVM model has been previously considered for the implementation of PAcAlCI (Chapter 3). Therefore, a wide explanation of the LS-SVM regression model has already been presented in this dissertation, section 3.5. A detailed description of the remaining three regression models is then presented below.

4.5.1 Regression Trees

Decision trees are useful predictive models based on tree structures [Rokach, 2008]. These trees usually include several nodes representing functions related to each included feature in order to decide which should be the following node. Final nodes (leaves) provide the estimated output according to the followed path in the tree. For regression problems, leaves give a real output value depending on the considered path.

One classical but recognized approach implemented for decision trees is the classification and regression trees algorithm (CART) [Breiman, 1993]. CARTs are non-parametric binary decision trees that have shown an efficient and accurate performance, mainly because of their growing technique and error-complexity pruning [Rokach and Maimon, 2005]. Another important advantage of CART is the ability to perform regression trees.

The CART regression splits branches in the decision according to the mean-squared error (MSE) criterion in the training dataset. Thus, the fitting error associated with a specific node t in the tree is determined as:

$$E(t) = \frac{1}{N_t} \sum_{i=1}^{N_t} (y_i - k_t)^2 \tag{4.29}$$

where N_t specifies the number of samples that have reached the node t and y_i is the output value for each of these samples. Additionally, k_t determines the node constant, which is calculated as the average of output values of these samples:

$$k_t = \frac{1}{N_t} \sum_{i=1}^{N_t} y_i \tag{4.30}$$

In a binary tree as proposed by CART, the best split (s) to determine the two following sub-nodes of a node t is the resulting from minimizing the expression:

$$\Delta E(s,t) = E(t) - E(s,t) \tag{4.31}$$

where $E(s,t)$ represents the error of a specific split s in a node t and it is calculated as:

$$E(s,t) = \frac{N_{tL}}{N_t} E(t_L) + \frac{N_{tR}}{N_t} E(t_R) \tag{4.32}$$

with N_{tL} and N_{tR} defining the number of samples that are passed to the left and right sub-nodes (t_L and t_R) respectively.

Finally, the values in the leaves, which are considered the estimated output of the tree, are also calculated as the constant of a node (Equation 4.30) [Rokach and Maimon, 2005].

Since the regression tree is iteratively dividing the original training dataset into two subsets in each node, the error estimations in the nodes are progressively being performed with less samples. These small subsets lead to more unreliable estimations and, consequently, an overfitting of the training data. To avoid this overfitting, there are two possible alternative procedures: *(i)* growing the full tree and pruning after those leaves and branches considered as unreliable and *(ii)* determining a stopping criteria to finish the growth of the tree. In the first case, the pruning procedure proposed in the CART algorithm aims to achieve a trade-off between error and complexity measures of the tree (Error-Complexity pruning). In the second case, the minimum number of samples in the leaf or the minimum number of samples in nodes with more than one value per feature (impure node) can be considered as stopping conditions.

In the implementation proposed in this chapter using the *Statistics* toolbox from Matlab®, both minimum values of the second option are tuned as stopping criteria. Specifically, the minimum number of samples considered by each tree leaf is defined here as L_{min} whereas the minimum number of samples in impure nodes is called P_{min}. P_{min} should at least be set to the double of L_{min}.

4.5.2 Bagging Regression Trees

Bootstrap aggregation, also named *bagging*, is an ensemble learning algorithm based on the generation of several replicas of a specific learner [Breiman, 1996]. Each of these replicas are trained by randomly selecting a subset of samples from the training dataset. Each particular sample may appear repeated or not at several of these training subsets. In this case, an ensemble of several decision trees built by the previously described CART algorithm is applied. In addition to the use of different training subsets, each bagging tree could also select independent feature subsets for decisions on its nodes. The random selection of training subsets reduces the correlation between trees and increase the overall regression power. It is then generally accepted that the bagging approaches can reduce the instability and bias of individual trees and achieve more effective results [Martinez-Muñoz et al., 2009].

The bagging algorithm is then formed by a set of K regression trees $\{\mathcal{R}_k,\ k = 1, \ldots, K\}$ each consisting in a number of N_k samples of the full dataset. The aim is to use the $\{\mathcal{R}_k\}$ regression trees to get a better prediction than with only a single tree. Therefore, if each of these trees provides a prediction $\phi(x, \mathcal{R})$ for a sample s (set of features), the bagging algorithm estimates a set of K predictions $\{\phi(s, \mathcal{R}_k)\}$. Since the estimated output in this problem is a numerical and continuous variable, the final prediction is calculated as the average of $\phi(s, \mathcal{R}_k)$ over k:

$$\phi_A(s) = \sum_{k=1}^{K} \frac{\phi(s, \mathcal{R}_k)}{K} \tag{4.33}$$

Since the bagging model has been implemented from the *Statistics* toolbox in Matlab® (R2010b version), some particular parameters can be configured, namely the number of regression trees (K), the minimum number of samples required by each tree leaf (L_{min}, similarly to the previous regression tree) and the rate of samples S_k to be randomly selected for each tree. From these parameters, Breiman [1996] suggested that the number of regression trees K is not excessively decisive in the final prediction as long as a minimum number is reached (around 10 trees). Despite that, the three parameters L_{min}, K and S_k are empirically tuned here trying to reduce the prediction error as much as possible.

4.5.3 Gaussian Processes

Gaussian Processes (GPs) are defined as Bayesian machine learning approaches based on a particularly effective method for placing a prior distribution over the space of functions [MacKay, 1998, Rasmussen, 2006]. Therefore, GPs can be considered a collection of random variables which have a joint Gaussian distribution [Rasmussen, 2006]. Formally, the output of a GP is defined by its mean $m(\vec{x}_i)$ and covariance $K(\vec{x}, \vec{x}_i)$ functions:

$$y_i \sim \mathcal{N}(m(\vec{x}_i), K(\vec{x}, \vec{x}_i)) \tag{4.34}$$

where $\vec{x}$ and $\vec{x}_i$ denote pairs of input features and each y_i represents a joint

Gaussian random variable modeling the output value for each specific input $\{x_i, i = 1, \ldots, N\}$. In this specific case, a radial basis function (RBF) was specified for the covariance function $K(\vec{x}, \vec{x_i})$ similarly to the previously presented RBF kernel in the LS-SVM approach (Equation 3.32, Chapter 3). Note that this covariance function provides values close to the unity when variables are similar and decreases when they are more distant [Rasmussen, 2006]. Therefore, the covariance of the output value is then written as a function of the inputs.

However, for modeling more realistic situations, GPs consider that independent and identically distributed Gaussian noises should be incorporated to the random variables ($y'_i = y_i + \varepsilon$) and, therefore, their covariances:

$$K(y, y_i) = K(\vec{x}, \vec{x_i}) + \sigma_n^2 \delta \tag{4.35}$$

where σ_n^2 defines the distributed Gaussian noise level and δ represents the Kronecker delta which takes the value one if $\vec{x}$ and $\vec{x_i}$ are the same and zero otherwise.

This situation can be generalized for a dataset of features X which is split into training and validation subsets (X and X_*, respectively). The training set also provides a set of known output values including Gaussian noise which are notated as y'. Thus, given the subsets X and X_* and the output y', the predictive distribution y_* for a regression based on Gaussian processes can be defined as:

$$y_* | X, y', X_* \sim \mathcal{N}(m(y_*), K(y_*)) \tag{4.36}$$

with the mean and covariance functions being now expressed in terms of the two subsets and the known outputs as:

$$m(y_*) = K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} y \tag{4.37}$$

$$K(y_*) = K(X_*, X_*) - K(X_*, X)[K(X, X) + \sigma_n^2 I]^{-1} K(X, X_*) \tag{4.38}$$

where I represents the unity matrix and K is selected to be the same RBF kernel function as LS-SVMs (see Equation 3.32 for details).

As previously stated, the Gaussian processes in this chapter are designed by using the *GaussianProcesses* function provided by the WEKA software [Hall et al., 2009]. This function does not automatically perform hyper-parameter tuning but there are some configurable parameters to allow users tuning them manually. Specifically, in order to implement the proposed Gaussian process regression, two configurable hyper-parameters have been considered: width of the kernel (σ_i) (similarly to LS-SVM) and the noise level (σ_n^2).

4.6 Implementation and assessment of the scoring schemes

As depicted in the flowchart of Figure 4.1, the implementation and assessment of the four proposed scoring schemes is planned into three stages: training, parameter tuning and test. With this purpose, the dataset of alignments (5540 alignments) is randomly divided into two independent subsets. More specifically, two thirds of the dataset are considered for the training and parameter tuning stages whereas the remaining one third is kept for testing. Nevertheless, this division has been performed considering that all the ten alignments for the same set of sequences should be included at the same subset. Therefore, these subsets are firstly obtained from the original sets of sequences (554 sets), thus including 370 sets (two thirds) for the training and parameter tuning stages and 184 sets (one third) for the test. Extrapolating this division to the complete dataset, a total of 3700 alignments (370 sets aligned by 10 tools) is dedicated to the training and parameter tuning (**training/validation subset**) whereas the remaining 1840 are taking into account for testing (**test subset**).

After splitting the dataset into two independent subsets, the training and parameter tuning stages are performed. Similarly to PAcAlCI, a 10-fold cross-validation process is applied to train and validate each regression model. Briefly, this procedure divides the alignments of the training/validation subset into ten independent groups (370 alignments per group). Nine of these groups are then used to train each regression model whereas the remaining one is considered to validate and tune the parameters of each model. The procedure is repeated ten times alternatively leaving for validation each of the different groups.

The parameter tuning is performed by training each model with different combination of parameters and selecting those values which provide the lower error in the validation (considering the 10-fold cross-validation). The validation error is estimated in this case by the *mean absolute error* (MAE). The MAE for the validation dataset (S) is measured as:

$$MAE = \frac{1}{|S|} \sum_{s=1}^{|S|} |a_s - \widehat{a_s}| \tag{4.39}$$

where a_s defines the BAliscore value (real accuracy indicator) for the alignment s in the dataset whereas $\widehat{a_s}$ represents the value predicted by the considered regression model for the same alignment. $|S|$ determines the number of alignments in the validation dataset (S).

In addition to the parameters of each regression model, the optimal number of features is also determined in the parameter tuning stage. Thus, each regression model is trained with a increasingly larger dataset of features according to the relevance ranking provided by the NMIFS process. In this case, the optimal number of features is obtained taking into consideration both the minimization of the BAliBASE error calculated as explained in Equation 4.39 (MAE error) and the complexity of the model. The complexity of the model is considered here directly proportional to the number of features that have been included. Although it has been observed that the increase of features does not directly imply an increase in the computational cost of the model (excepting the bagging tree model), the time for the retrieval and calculation of the features included in the model must necessarily be taken into account.

Consequently, the optimal subset of features is performed according to a measure based on the Akaike Information Criterion (AIC) [Akaike, 1974]. The AIC calculates the quality of different models in order to compare them. This AIC measure is defined as:

$$AIC = 2k - 2\ln(L) \tag{4.40}$$

where L defines the *likelihood* function of the model and k the number of parameters. The model providing the smallest AIC value is considered the most adequate one. Since we are interested in compare the error of the model with the number of features, the criterion applied in this case has been reformulated as:

$$AIC_{error} = 2k_f + 2\ln(MAE) \tag{4.41}$$

where k_f is the number of features included in the model and MAE is the validation error. Since $ln(MAE)$ provides negative values, the sign has been modified accordingly. Lastly, in order to give to both terms (number of features and validation error) the same importance both terms in the addition of the Equation 4.40 are previously normalized to the range [0, 1].

Finally, after tuning the parameters and selecting the optimal number of features, each regression model is performed by training with the whole training/ validation subset (3700 alignments). The obtained regression models are then assessed by using the independent test subset of 1840 alignments. Thus, the different models are compared among them and with other scores schemes by determining the correlation of the predicted qualities with the real ones provided by BAliscore in such subset.

4.7 Assessment by comparisons of pairwise alignments

When comparing different scoring schemes to evaluate alignments, it is interesting to know the ability of each score to determine which of two or more alignments is the most accurate for the same sequences. Consequently, it is usual in this kind of system to assess the different scoring schemes taking into account this ability. More specifically, the accuracy values provided by each proposed scoring scheme for two different alignment of the same set of sequences are compared to decide which one is the most accurate. Subsequently, the BAliscore values for the same alignments validate if the decision has been correctly taken.

The pairwise comparisons are also performed with the results in the test subset (1840 alignments). Therefore, 10 different alignments for each of the 184 sets of sequences can be compared by pairs. These pairwise validations are then formally represented by the comparative functions between two alignments for the corresponding score scheme (T_S) and for the reference BAliscore (T_B) as:

$$T_S(A_i^k, A_j^k) = \begin{cases} 1 & if \quad S(A_i^k) > S(A_j^k) \\ 0 & otherwise \end{cases} \tag{4.42}$$

$$T_B(A_i^k, A_j^k) = \begin{cases} 1 & if \quad B(A_i^k) > B(A_j^k) \\ 0 & otherwise \end{cases} \tag{4.43}$$

where $S(A_i^k)$ and $B(A_i^k)$ are the alignment qualities estimated by each proposed score (S) and the reference BAliscore (B) for the alignment A_i^k. Specifically, A_i^k denotes the alignment obtained by the MSA tool i ($i = \{1, \ldots, 10\}$) for the k-th set of sequences in the dataset ($k = \{1, \ldots, 184\}$).

These comparisons are then assessed by applying the standard prediction measurements true-positive (TP), false-positive (FP), true-negative (TN) and false-negative (FN) defined as:

$$TP = \sum_{k=1}^{184} \sum_{i=1}^{10} \sum_{j=i+1}^{10} \mathcal{F}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big); \quad (T_B(A_i^k, A_j^k) = 1) \tag{4.44}$$

$$FP = \sum_{k=1}^{184} \sum_{i=1}^{10} \sum_{j=i+1}^{10} \overline{\mathcal{F}}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big); \quad (T_B(A_i^k, A_j^k) = 0) \tag{4.45}$$

$$TN = \sum_{k=1}^{184} \sum_{i=1}^{10} \sum_{j=i+1}^{10} \mathcal{F}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big); \quad (T_B(A_i^k, A_j^k) = 0) \tag{4.46}$$

$$FN = \sum_{k=1}^{184} \sum_{i=1}^{10} \sum_{j=i+1}^{10} \overline{\mathcal{F}}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big); \quad (T_B(A_i^k, A_j^k) = 1) \tag{4.47}$$

where the functions $\mathcal{F}$ and $\overline{\mathcal{F}}$ are defined as:

$$\mathcal{F}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big) = \begin{cases} 1 & if \quad T_S(A_i^k, A_j^k) = T_B(A_i^k, A_j^k) \\ 0 & otherwise \end{cases} \tag{4.48}$$

$$\overline{\mathcal{F}}\Big(T_S(A_i^k, A_j^k), T_B(A_i^k, A_j^k)\Big) = \begin{cases} 1 & if \quad T_S(A_i^k, A_j^k) \neq T_B(A_i^k, A_j^k) \\ 0 & otherwise \end{cases} \tag{4.49}$$

The standard sensitivity, specificity and accuracy measurements are then calculated to evaluate the performance of each scoring scheme:

$$sensitivity = \frac{TP}{TP + FN} \tag{4.50}$$

$$specificity = \frac{TN}{FP + TN} \tag{4.51}$$

$$accuracy = \frac{TP + TN}{TP + FN + FP + TN} \tag{4.52}$$

These three measurements are then applied to determine how each score is capable to decide which is the best possible alignment. Thus, higher values of sensitivity, specificity and accuracy indicate a better prediction ability.

4.8 Experimental Results

In this section, the results obtained for the four proposed regression models are presented. The outcomes of the tuning of parameters for each model and the feature selection based on NMIFS are firstly shown. Subsequently, the correlation of each novel scoring scheme is analyzed in the test subset. Finally, a comparison of pairwise alignments is carried out in order to determine how each score is able to differentiate which alignment is more accurate.

4.8.1 Tuning of parameters in the regression models

In order to implement the most accurate possible models for the four proposed regression algorithms, their main configuration parameters are optimized. These parameters are empirically determined trying to minimize the regression error in the validation by using a 10-fold cross-validation procedure. The values obtained for each model are shown in Table 4.2. It is important to remind that

TABLE 4.2: Parameters optimized for each regression methodology. These optimal values are determined for the final implementation of each approach, also including the optimal number of features.

APPROACH	PARAMETER	OPTIMAL
Regression Tree	Min. observations per leaf (L_{min}) Min. observations per impure nodes (P_{min})	$L_{min} = 3$ $P_{min} = 17$
Bagging Trees	Min. observations per leaf (L_{min}) Number of trees (K) Rate of samples per tree (S_k)	$L_{min} = 3$ $K = 25$ $S_k = 0.7$
LS-SVM	Regulation parameter (γ) Width of the kernel (σ_i)	$\gamma = 81.44$ $\sigma_i = 0.401$
Gaussian process	Noise level (σ_n^2) Width of the kernel (σ_i)	$\sigma_n^2 = 0.1$ $\sigma_i = 35$

these parameters were specifically tuned together with the selection of features (see following section for details).

For decision trees (regression and bagging), the minimum number of observations per leaf (L_{min}) is usually set to 5 by default whereas at least 10 trees (K) should be considered in bagging trees models to obtain an appreciable improvement. Nevertheless, ranges of {1,50} for K and {1,20} for L_{min} are taken into account in this case to find the optimal parameters. With respect to the number of observation in impure nodes, it is usually recommended to include at least twice the minimum observations per leaf ($P_{min} \geq 2 \cdot L_{min}$). However, a higher P_{min} value is determined to be the optimal one in this case, namely $P_{min} = 17$. Finally, the rate of samples considered for each bagging tree is set to 0.7.

For the LS-SVM model, the two hyper-parameters of the RBF kernel (γ and σ_i) are specified by using the greedy tuning function provided in De Brabanter et al. [2011]. Finally, the noise level and width of the kernel in Gaussian processes are optimized by determining the minimum error in the ranges $]0, 1]$ for σ_n^2 and $]1, 100]$ for σ_i.

4.8.2 Determination of the optimal subsets of features

The four previously proposed models are performed by using a dataset of 27 features (**input variables**), trying to estimate qualities similarly to the BAliscore (**output variable**). These features are related to the size of each set of sequences from BAliBASE and OxBench benchmarks as well as several properties of their corresponding alignments.

The selection of the most relevant features was performed together with the previous tuning of parameters, using in both cases the same 10-fold cross-validation procedure and the same training/validation subset. The NMIFS procedure is firstly applied to return a ranking of features, considering their significance with respect to the BAliscore. The obtained NMIFS ranking is presented in Table 4.3. From this table, it is remarkable that the two most significant features are related to the tertiary and secondary structure of the sequences. Thus, the matches found in the alignments between the contact of the tertiary structure (f_{16}) are ranked as the first feature whereas the total matches in all the types of secondary structures (f_{11}) are classified in the second position. These results support the idea that the alignment of similar tertiary and secondary structures is strongly related to the increase of the alignment quality.

As expected, some evaluation scores and measurements traditionally associated to the quality of the alignment are also ranked among the most significant features. Thus, the percentages of identities (f_4), gaps (f_5) and totally conserved columns (f_6) have been respectively ranked at the *4th*, *6th* and *7th* positions. Also, some scores like STRIKE (f_{27}) or PAM250 (f_{25}) are found in higher positions (*3rd* and *5th*, respectively) whereas other scores like RBLOSUM (f_{24}) or GONNET (f_{26}) are also ranked in the top half of the ranking (*10th* and *11st*, respectively). The top half of features in the ranking also included the remaining scores . Finally, other interesting features to highlight are the matches in the chemical amino acids types (f_{22}) or the variance length (f_3) which are ranked at the *8th* and *9th* positions. For instance, the variance length could be associated to less related sequences and, therefore, lower accuracies.

TABLE 4.3: Feature ranking obtained by the NMIFS procedure taking into account their dependence against the BAliscore values (real accuracies).

RANK	**FEATURE**	
1	f_{16}	3D Contact matches
2	f_{11}	Secondary structure matches
3	f_{27}	STRIKE score
4	f_4	Identities
5	f_{25}	PAM250 score
6	f_5	Gaps
7	f_6	Totally Conserved columns
8	f_{22}	Amino acid Types matches
9	f_3	Variance length
10	f_{24}	RBLOSUM score
11	f_{26}	GONNET score
12	f_{19}	Basic Amino acid matches
13	f_{20}	Aromatic Amino acid matches
14	f_{17}	Polar Amino acid matches
15	f_{14}	Total domain matches
16	f_{18}	Non-polar Amino acid matches
17	f_{23}	BLOSUM score
18	f_{21}	Acid Amino acid matches
19	f_9	Turn secondary structure matches
20	f_{12}	Pfam-A domain matches
21	f_2	Average length
22	f_{10}	Unknown secondary structure matches
23	f_7	α-helix matches
24	f_8	β-strand matches
25	f_{15}	Domain clan matches
26	f_{13}	Pfam-B domain matches
27	f_1	# Sequences

Taking this ranking into account, the four regression models were designed by using the 10-fold cross-validation in the training/validation subset and increasingly including one additional feature in order to determine the optimal subset of features. As previously described, a criterion based on the Akaike Information Criterion (AIC) is proposed to reach a trade-off between the minimization of error and the number of selected features, which is considered the complexity in our models (see section 4.6 for details).

The mean absolute error (MAE) and the adapted Akaike criterion (AIC_{error}) values are represented in Figure 4.2 for each subset of features. The error in the LS-SVM model, regression tree and bagging trees dropped more drastically and before than in Gaussian processes. Likewise, error evolution settled earlier in regression and bagging tree procedures (between 5-10 features, approximately) than in LS-SVM or Gaussian Processes (10-15 features). The optimal MAE error is shown in the case of Gaussian processes but it is reached with a large number of features whereas the regression trees show the worst performances (higher errors) independently of the number of features. The bagging trees and the LS-SVM model obtain errors quite similar to Gaussian trees, specially with less features.

However, taking into account the adapted Akaike criterion (AIC_{error}), the LS-SVM model needs less features (7 features) than the Gaussian processes (9 features) or the regression and bagging trees (9 and 10 features, respectively) to reach the optimal subset of features. As observed, the AIC_{error} criterion does not determine the number of features with the minimal possible error. Instead of that, this criterion tries to provide a number of features where the achieved error is acceptable without excessively increasing the complexity of the models. According to the Figure 4.2, even though the regression and bagging trees specify a higher number of features (9 and 10 respectively), there are other local minima in AIC_{error}, e.g. with 5 features, that could be also considered. Nevertheless, as it can be appreciated in the error evolution representation, these two models have shown worse performance independently of the subset of features.

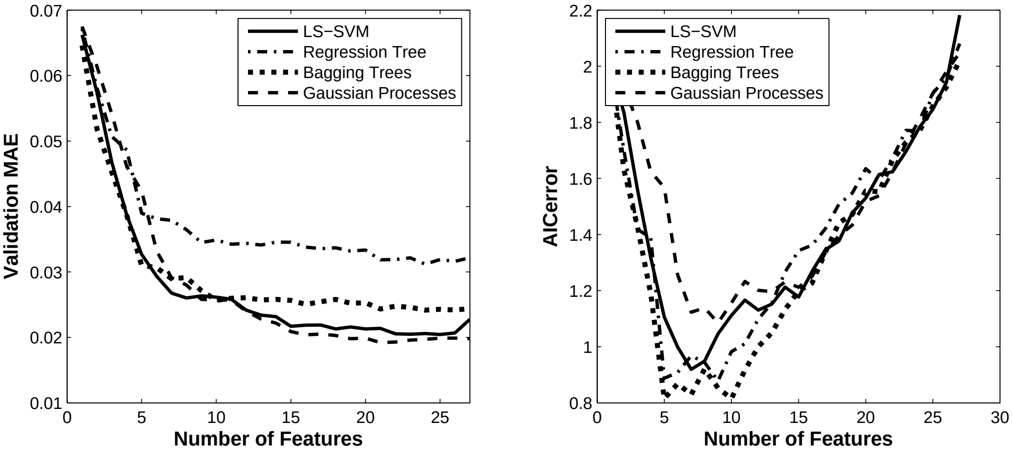


FIGURE 4.2: Error evolution and feature selection criterion according to the number of features included in the regression approaches. Both errors and AIC_{error} are calculated for the validation subsets in the 10-fold cross-validation process applied to the dataset of 3700 alignments. These errors are calculated as the mean absolute error (MAE) (see Equation 4.39). The AIC_{error} criterion determines the optimal number of features (minimal AIC_{error} value) for each model. This criterion based on Akaike Information Criterion (AIC) is calculated according to the Equation 4.41

Although the computational time is not being deeply analyzed in this chapter, the selection of features plays a key role to simplify and reduce the time in the models. Specifically, the feature reduction in the dataset implies a time optimization in the feature extraction. Thus, even though some time-consuming features have been included in these datasets because of their importance (e.g. *'Tertiary Structure'* type or *'STRIKE'* score), some other computationally expensive features such as particular secondary structure features (α-helix, β-strand or *turn* secondary structures) or features related to Pfam domains are excluded, thus saving significant time in the extraction stage.

4.8.3 Correlation between different score schemes and BAliscore

After validating and determining the optimal number of features and parameters for the four intelligent score estimations, they are finally performed by using the full validation set of 3700 alignments. In order to assess their performance and compare with other score schemes, these models are tested with the remaining 1840 alignments (test set). The quality values estimated by the advanced scores are then compared against the other popular score schemes that have been also included as features, namely BLOSUM62, RBLOSUM62, PAM250, GONNET and STRIKE. Therefore, it can be proved if these scores are improved when they are integrated together with other features in the proposed regression models.

Since BAliscore provided an accurate measure of quality with regard to the reference alignments, the correlation between each proposed score and the BAliscore is calculated. As shown in Figure 4.3, the proposed intelligent scores based on regression methodologies provide the highest correlation values ($R > 0.9$), significantly improving the correlation of the remaining scores. Among the proposed regression models, the bagging trees show the best correlation ($R = 0.947$). As can also be observed, PAM250, RBLOSUM62 and GONNET scores do not practically provide any correlation with BAliscore ($R < 0.12$). The lack of correlation with the real quality of the alignments could become an important drawback to those MSA tools building their alignments by using such scores. On the other hand, the STRIKE score provides a quite better correlation ($R = 0.722$) but it is also outperformed by the four regression proposals. It can be then suggested that the integration of relevant features and classical scores is useful to obtain better scoring schemes in order to accurately evaluate alignments.

4.8.4 Assessment of scores by comparing pairwise alignment

Since ten different alignments from the ten corresponding tools are provided in our dataset for the same sets of sequences, a pairwise comparison can be additionally performed in order to analyze the importance of each score scheme predicting which of two alignments is the best one. Therefore, the test set (1840 alignments) is also applied for this comparison. Alignments from each of these sets of sequences are then compared by pairs determining if one alignment is better or worse than the another one according to the provided scores. The alignment considered as the best by BAliscore for each pairwise comparison is chosen as the true best alignment (gold standard).

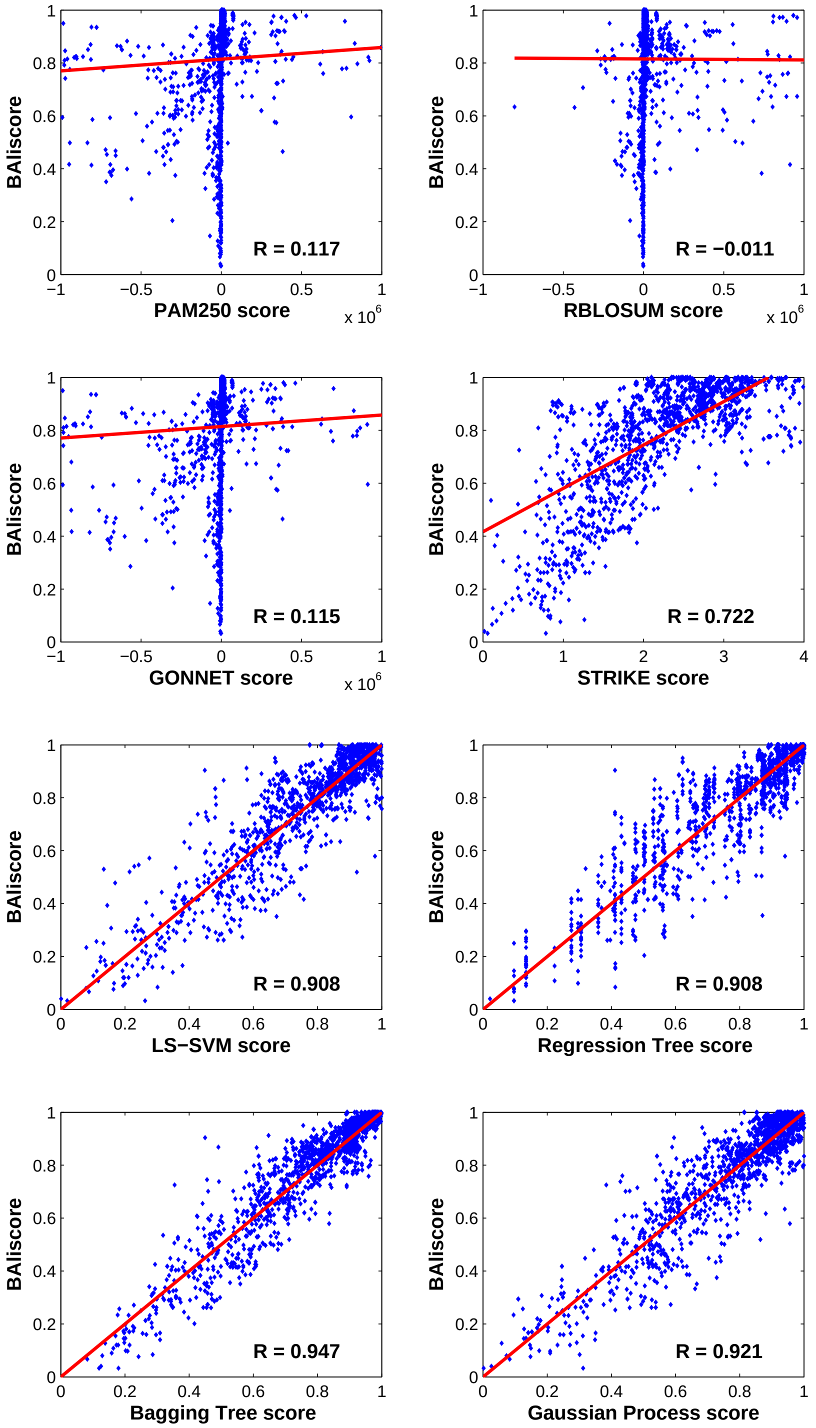


FIGURE 4.3: Graphical correlation between BAliscore and the remaining considered scores (PAM250, RBLOSUM62, GONNET and STRIKE) as well as the proposed advanced scores based on regression methodologies (LS-SVM, Regression Tree, Bagging trees and Gaussian Processes). For simplicity, BLOSUM62 is omitted as it returned a correlation similar to RBLOSUM62.

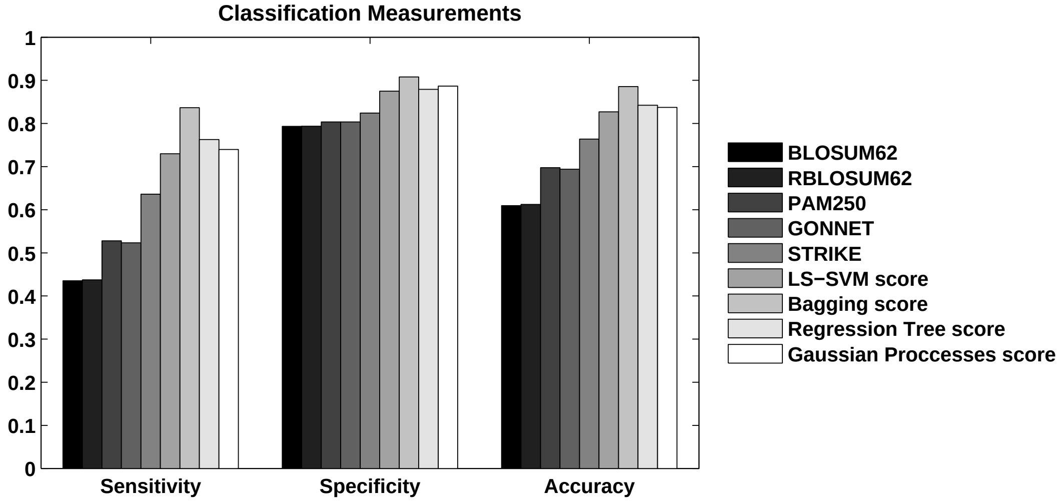


FIGURE 4.4: Sensitivity, specificity and accuracy obtained for the 8280 pairwise comparison of alignments in each score scheme. The values of each score are compared against the BAliscore value to determine these measurements.

Given that these comparisons are applied to 1840 alignments corresponding to 184 sets of sequences, a total of 8280 pairwise comparisons are performed for each score scheme (10 different alignments taken in pairs for the 184 sets of sequences in the test set). According to the provided definition (see section 4.7 for details), a pairwise comparison is considered true (positive or negative) when a specific score scheme agrees with the BAliscore selecting the same best alignment in such comparison. Otherwise, if both scores differ, the comparison is considered false.

In order to compare the different scores as well as their ability to decide the best alignment, the sensitivity, specificity and accuracy measurements are obtained. Therefore, the prediction qualities from the different scoring schemes are depicted in Figure 4.4. According to these three measurements, the proposed advanced scores again outperform the remaining standard scores, being the STRIKE score the only one able to reach acceptable qualities. It is then proved again that classical scores like PAM, BLOSUM or GONNET are not excessively useful to evaluate and compare alignments since they are not capable to accurately differentiate which of two alignments is better.

Among the proposed estimations, the bagging trees model clearly shows the most powerful score to predict the best alignment (*sensitivity* = 0.837, *specificity* = 0.909 and *accuracy* = 0.886), similarly to the correlation results. However, these results do not have to be necessarily related to the correlation with BAliscore. For instance, although the scoring scheme based on Gaussian processes showed the second best correlation in the previous results, this scheme is slightly outperformed in these pairwise comparisons by the regression trees, which provided the least correlated performance among the advanced scores (see the *accuracy* plot in Figure 4.4). Therefore, it can be suggested that a lower correlation with BAliscore does not necessarily imply a worse prediction of the best alignment.

4.9 Final conclusions

The need of more sophisticated systems to determine the quality of multiple sequence alignments is today essential to allow the construction of more accurate alignments. Therefore, several novel scoring algorithms for MSAs have been proposed in this chapter. These algorithms took advantage of several biological sources to build a dataset of heterogeneous features. Among these features, some other measurements and typical score schemes were also incorporated. The dataset of heterogeneous features was then integrated in several regression models such as regression trees, bootstrap aggregation trees, LS-SVMs and Gaussian processes to estimate the quality of MSAs.

The proposed evaluations were assessed by using a dataset composed by the BAliBASE and the OxBench benchmarks. Each of the proposed advanced scores have been performed by tuning their principal parameters and selecting the optimal number of included features. In addition to some standard MSA scores, features associated to secondary and tertiary structures in alignments were proved to be specially relevant to determine their quality. Subsequently, these schemes have been tested with an independent subset of alignment (test set) and they have been compared against other standard scores. A significant improvement in terms of the correlation with the BAliscore values (true qualities) was shown for the four regression models.

Finally, the advanced score schemes were evaluated by comparing alignments by pairs (pairwise comparisons). In this case, the proposed methods also showed a good performance in terms of sensitivity, specificity and accuracy. Additionally, it was also appreciated that the ability to predict the most accurate alignment does not have to be related to the results previously shown in the correlation with the BAliscore values.

Taking into account the correlation and pairwise comparison results, it can be concluded that the proposed scores are working as expected and, in general, they evaluate alignments in a better way than standard scores. The integration of several scores together with other supplementary biological features helps to improve the evaluation of MSAs, obtaining more realistic quality values. Therefore, the proposed models provide interesting evaluations that may be applied in the future for the design and optimization of novel MSA optimizers and MSA tools.

Chapter 5

MO-SAStrE: Multiobjective Optimizer for Sequence Alignment based on Structural Evaluation

5.1 Background, motivations and goals

In recent years, the increase of novel methodologies, mainly next-generation sequencing and high throughput experiments, has increased the number of existent tools to analyze and to align biological sequences. Since these techniques provide mainly new nucleotide sequences and their subsequent products, MSAs tools usually help to extract biological meanings from such information. However, as previously analyzed in Chapter 2 (section 2.4), current tools still provide partially optimal or suboptimal alignments. Moreover, although there are many MSA methodologies, each tool is usually based on particular strategies and their accuracy directly depends on the properties of the sequences to be aligned (see Chapter 3 for details). Consequently, there is no consensus about which method builds more accurate alignments or the most adequate way of aligning sequences [Nuin et al., 2006, Sierk et al., 2010].

Although the MSA tools often achieve suboptimal solutions, some specific regions within the alignments are usually more accurate than others depending on the sequence properties locally found at these particular regions. These suboptimal solutions have also a negative influence on posterior phylogenetic analyses based on alignments, as wrong phylogenetic trees are obtained when alignments are inaccurate [Wong et al., 2008]. For this reason, some MSA tools take advantage of jointly optimizing both phylogenetic trees and alignments [Ronquist et al., 2012, Westesson et al., 2012]. These methods aim to avoid the bias generated by guide trees in progressive methods, though they do not achieve good performances in terms of structure [Mirarab and Warnow, 2011].

In order to improve the alignment accuracies, recent MSAs tools are increasingly capable of dealing with and efficiently analyze the massive amount of sequence data. Several advanced computational approaches based on well-known artificial intelligence and machine learning algorithms are used with this purpose [Anisimova et al., 2010]: hidden markov models, support vector machines, decision trees, etc. Also, genetic algorithms (GAs) have been widely used to build and optimize MSAs [Gondro and Kinghorn, 2007, Naznin et al., 2011].

Another important challenge to take into account when dealing with MSA optimization is to provide an efficient evaluation method to measure the alignment accuracy. It is essential to consider a good objective function to be optimized and, consequently, to improve alignments. As previously introduced, MSA strategies have traditionally applied matrices like Point Accepted Mutation (PAM) [Dayhoff et al., 1979] or BLOSUM [Henikoff and Henikoff, 1992], which only consider nucleotide or amino acid information to evaluate every aligned pair of residues. Nevertheless, when the number of sequences increases or longer and more distant ones are included, alignments are more likely to be inaccurate using such scores [Liu et al., 2009]. In these cases, additional information is necessary to complement the scoring matrices.

Thus, recent evaluations tend to use more complex scores including supplementary biological features, such as homologies or protein structures. For instance, some score schemes benefit from homological profiles provided by PSI-BLAST [Altschul et al., 1994] to increase the accuracy evaluating alignments. Additionally, structural information is also used in other evaluation systems since structures are evolutionarily more conserved than sequences in proteins. These evaluations can then determine more distant relationships between sequences and, therefore, scores with structural information are better suited to evaluate alignments [Kemena and Notredame, 2009]. For example, Kececioglu et al. [2010] provided a novel scoring scheme to evaluate MSAs from their predicted secondary structures. Other scores, such as Contact Accepted mutatiOn (CAO) [Lin et al., 2003] and STRIKE [Kemena et al., 2011] scores also estimated the molecular contacts from tertiary structures in proteins to calculate alignment accuracies. However, although there are several possibilities to evaluate alignments, researchers and experts do not agree about what is the best way [Blackburne and Whelan, 2012].

So, a novel methodology for the optimization of multiple sequences alignments is developed in this chapter. This method is called ***M**ultiobjective **O**ptimizer for **S**equence **A**lignments based on **Str**uctural **E**valuations* (MO-SAStrE). A multiobjective genetic algorithm is proposed to take advantage of several evaluation schemes instead of only one score. It specifically defines three objectives which are used to evaluate alignments generated by the genetic algorithm:

1. Evaluation based on tertiary structural information by using the STRIKE score [Kemena et al., 2011].
2. Number full columns with the same aligned amino acid (totally conserved columns).
3. Percentage of positions in the alignment that do not include gaps (percentage of non-gaps).

The MO-SAStrE system aims to optimize alignments previously performed by fast but inaccurate MSA tools. The main goal is to obtain high quality alignments, even improving other similar genetic algorithms but also more complex non-genetic tools including structural information like 3D-Coffee [O'Sullivan et al., 2004]. Therefore, alignments from MO-SAStrE are finally compared with results shown by other known genetic and non-genetic alignment algorithms.

This chapter is structured as follows. Section 5.2 reviews some basic concepts and properties in evolutionary algorithms and multiobjective optimization. The architecture of the proposed optimizer is presented and deeply described in section 5.3. This architecture is additionally validated according to the assessment procedure explained in section 5.4. In section 5.5, the experimental procedure and the principal results obtained through our optimizer are presented. Also, the comparison against other genetic and non-genetic algorithm is detailed in this section. Finally, some important statements and conclusions about this approach are presented in section 5.6.

5.2 Bases of evolutionary and genetic algorithms

The Evolutionary Algorithm (EA) paradigm is based on the use of probabilistic search algorithms inspired by certain points in the Darwinian theory of evolution [Spears et al., 1993]. Several different techniques are grouped under the generic denomination of EA: Genetic Algorithms (GA), Evolutionary Strategies (ES) Evolutionary Programming (EP) and Genetic Programming (GP). However, the underlying purpose behind these techniques is similar: given a population of individuals, a natural selection is caused (survival of the fittest) increasing the fitness of the population [Eiben and Smith, 2008]. The essential features shared by all EAs are [Espejo et al., 2010]:

1. A group of candidate or partial solutions (population of individuals) is used instead of just one solution. This set of candidates can randomly be created or provided by the user.
2. A generational inheritance method. Genetic operators are applied to the individuals of a population to give birth to a new population of individuals (the next generation). The main genetic operators are crossover (recombination) and mutation. More specifically:
    - Crossover is applied to two or more individuals (the so-called parents) which exchange part of their genetic material resulting in one or more new individuals (children). This operator is said to be a *one-point* crossover if the two individuals are split into two regions by one random position and they exchange one of the regions. A *two-point* crossover can be also defined if the exchanged region is delimited by a starting and an ending position randomly selected.
    - Mutation randomly changes a small portion of the genetic material of one individual creating a new individual. This change can vary from only one element (gene) in the individual to a few consecutive elements (several genes).
3. A fitness-biased selection method. The quality of each individual is measured by a fitness function. The better the fitness of an individual, the higher probability of selecting this individual for the next generation. Therefore, an evolutionary process is being developed where the best individuals are likely to survive.

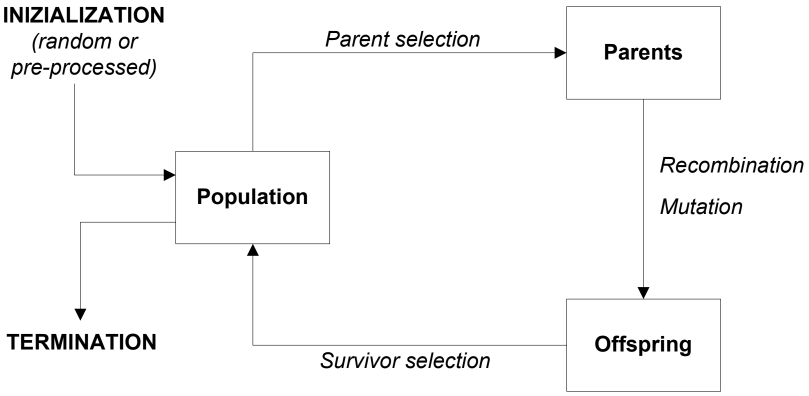


FIGURE 5.1: General scheme of an evolutionary algorithm as a flow-chart (Introduction to Evolutionary Computing, [Eiben and Smith, 2008])

Both new mutation and crossover individuals (offspring) competes with the old population for being included to the next generation based on their fitness. The application of these operators and the selection process lead to improve the fitness in progressive populations [Eiben and Smith, 2008]. The general scheme of an evolutionary algorithm is given in Figure 5.1.

In addition to the previously described processes, there are two other important components to take into account in EAs: the representation (also called codification) and the termination condition. The representation is responsible of converting a real problem into a context understandable by the EA. A encoding procedure must then be defined to represent individuals and facilitate their handling by operators and fitness functions. In a slightly different sense, the representation can also refer to the data structure used to define individuals [Eiben and Smith, 2008]. Thus, different variants of EAs are usually related to the way individuals are represented. For instance, Genetic Algorithms (GAs), which are considered in this chapter, are essentially EAs where individuals are encoded as strings over a finite alphabet. Regarding the termination condition, it defines the criterion given for finishing the optimization. This criterion is commonly chosen to fulfill a specific condition: a limit of generations, a threshold value under which the fitness improvement remains, a number of generation without fitness improvement, etc.

All these properties makes GAs extremely helpful in MSA because they can be implemented independently of the objective function [Naznin et al., 2011]. Thus, GAs can define multiple evaluations regardless any modification in the optimization procedure. Additionally, although evolutionary algorithms cannot compete in terms of speed with progressive alignments, they have the advantage of being able to correct the initially misaligned sequences in progressive methods [Gondro and Kinghorn, 2007]. Moreover, GAs can be easily parallelized, significantly reducing this excessive computational cost. Consequently, several methodologies have already been developed based on GAs to build and to optimize MSAs [Naznin et al., 2011, Taheri and Zomaya, 2009] (some of these methodologies are described in detail in section 5.4.3).

5.2.1 Multiobjective evolutionary optimization

There are so many problems with several goals that must be optimized simultaneously. These goals are often in conflict, since the optimization of one measure could imply unacceptable values for another goal. In these cases, a commitment among the proposed goals has to be reached.

This kind of problem is usually known as multiobjective optimization (MO) problem. MO strategies are generally more powerful than a simple weighted sum of objectives because they take advantage of exploring better the solution space in all objectives. In contrast to a weighted sum method, multiobjective approaches prevent from introducing bias in any of the problem objectives [Ombuki et al., 2006]. Besides, multiobjective algorithms do not need to derive specific weights. Thus, the influence of each objective can be separately analyzed from the several solutions.

Thus, MO problems do not provide a single solutions, but a set of equally accurate solutions (non-dominated) instead. The dominance criterion is applied to identify these equally acceptable solutions in a set according to the proposed objectives. Formally, for a set of objectives $f_1, f_2, ..., f_N$ that have to be simultaneously maximized, a specific solution x is said to dominate another solution y

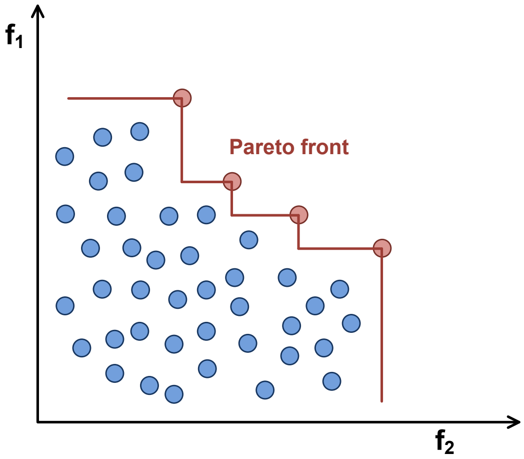


FIGURE 5.2: Graphical definition of Pareto front. For simplicity, only two objectives to be maximized (f_1 and f_2) are shown. As appreciated, solutions in the Pareto front (red circles) have not any other solution with better values in all the objectives (these solutions are non-dominated), whereas the remaining solutions (blue circles) have at least another solution with better values in all the objectives (these solutions are dominated).

(dominance relationship) if two main conditions are fulfilled:

- x solution is not worse than y solution in all of the objectives:

$$\forall i = 1, \ldots, N \; : \; f_i(x) \geq f_i(y) \tag{5.1}$$

- x solution is better than y solution in at least one objective.

$$\exists i = 1, \ldots, N \; : \; f_i(x) > f_i(y) \tag{5.2}$$

The subset of solutions that are not dominated by any other is called *Pareto front*. This definition is graphically represented in Figure 5.2.

EAs can be reformulated for MO problems in an easy and suitable way by defining a function fitness that takes into account different objectives simultaneously. In these cases, a subset of individuals (the Pareto front) is determined in the final set of optimal solutions. EAs applied to multiobjective problems are usually referred to as MOEAs or, specifically for GAs, MOGAs [Deb et al., 2002].

In recent years, several MOEAs have been proposed, being the most widely applied the SPEA2 [Zitzler et al., 2002] and the NSGA-II [Deb et al., 2002] approaches. In this chapter, the NSGA-II multiobjective procedure is used in this chapter. This procedure is deeply described in the following section.

5.2.2 Non-dominated Sorting Genetic Algorithm II (NSGA-II)

The Non-Dominated Sorting Genetic Algorithm (NSGA-II) scheme [Deb et al., 2002] is one of the most recognized methods for MO (>5,700 cites in ISI Web of Knowledge). NSGA-II defines a complete procedure for the fitness evaluation and the selection through the dominance concept and a crowding distance to include more diversity in the following generations (see NSGA-II scheme in Figure 5.3). More specifically, NSGA-II is mainly focused on two properties [Gacto et al., 2009]:

- The fitness function in NSGA-II evaluates each individual according to multiple Pareto fronts by means of a Pareto-compliant ranking method. Solutions in the current population are ranked in the following manner. All non-dominated solutions (optimal Pareto front) are ranked in the first position (F_1). A reduced population is then considered without the first ranked solutions. Next, the new non-dominated solutions in the reduced population are ranked in the second position (F_2). Thus, this procedure is progressively performed until all solutions are ranked in N Pareto fronts ($F_1, F_2, \ldots, F_N$). Consequently, a different rank is assigned to each solution where solutions with smaller ranks are viewed as being better than those with larger ranks.
- A elitist generation update procedure is implemented for the selection of individuals. When the next population is to be created, the current and offspring populations (P and O, respectively) are merged in a single extended population. The next population is constructed by choosing the best solutions from this merged population. This procedure is performed by progressively including in the next population the best non-dominated

Pareto fronts (first rankings). Depending on the population size, it is possible that only some solutions from the last included Pareto front (F_t) can be added to the next population. Therefore, each solution of this front is evaluated using the so-called *crowding distance* criterion. For a given solution, this measure calculates the distance between its adjacent solutions with the same rank in the objective space. Less crowded solutions with larger values of the crowding measure are viewed as being better than more crowded solutions with smaller values of the crowding distance. Thus, those solutions that are situated in areas little explored or, in other words, distant solutions are included from F_t, benefiting the diversity of solutions in the next population. Formally, for a specific solution $i \in F_t$, the crowding distance is calculated as:

$$d_i = \sum_{m=1}^{M} \frac{f_m^{i+1} - f_m^{i-1}}{f_m^{max} - f_m^{min}} \tag{5.3}$$

where M is the number of objectives; f_m^{i+1} and f_m^{i-1} are the nearest previous and posterior neighbors in the objective m; and f_m^{max},f_m^{min} are the maximum and minimum values of the objective m, respectively.

5.2.3 Performance assessment in multiobjective optimization

Comparing a multiobjective algorithm against other optimizers is often required in order to assess its quality. However, the evaluation and assessment of a multiobjective algorithm is usually a hard process, since there are no available approaches to simultaneously compare the quality of different solutions according to several objectives. Moreover, some multiobjective algorithms (e.g. MOGAs) have a stochastic nature. If they are applied several times to the same problem, a different set of solutions may be returned each time. Therefore, the stochasticity of this kind of algorithms makes even harder their assessment.

Current assessment approaches for MO algorithms mainly propose to convert the multiple objectives of the final set of solutions (optimal Pareto front) to

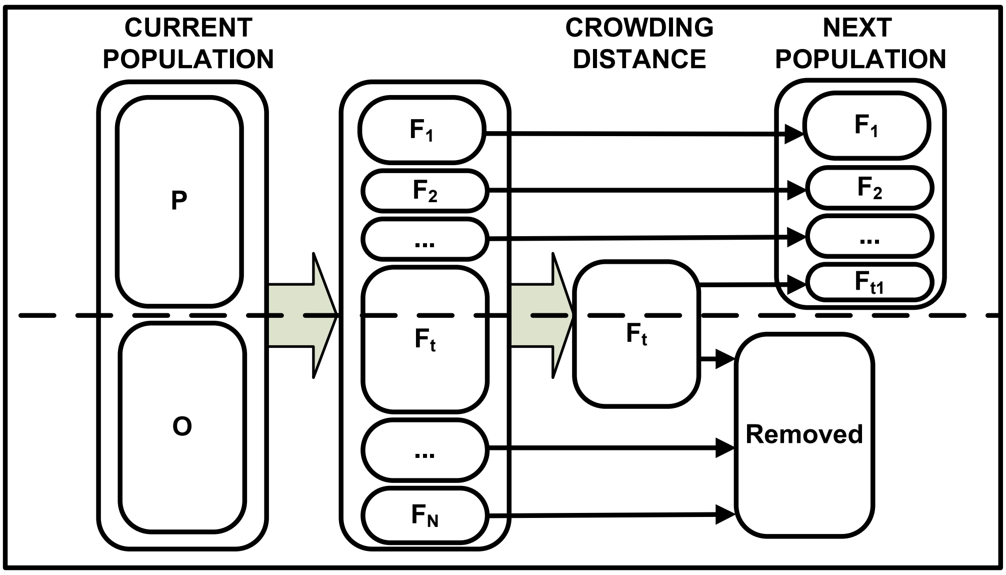


FIGURE 5.3: Pareto ranking and selection schemes in NSGA-II. The extended population in each generation is composed by the previous population (P) and offspring (O) created by operators. The population is classified in different Pareto fronts ($F_1, \ldots, F_N$) in order to obtain a non-dominated sorting. The first ranked solutions are included in the next population. The last selected solutions are determined by the crowding distance in the last considered front F_t (see Equation 5.3). This figure has been adapted from Deb et al. [2002].

a single real value (quality indicator), which is easier to compare [Zitzler et al., 2008]. In case of dealing with stochastic MOs, each compared algorithm should be run several times, so several quality values are obtained. Then, the resulting quality values of each algorithm have to be compared applying standard statistical tests.

Several quality indicators have been proposed in the literature to carry out these assessments. According to Zitzler et al. [2008], some relevant indicators are:

- The **dominance ranking** consists in ordering all runs for all compared algorithms according to the dominance concept (similarly to the fitness function described above for NSGA-II). A Pareto front F_1 is said to be *better* than a Pareto front F_2 ($F_1 \rhd F_2$) if for each solution y in F_2, there is at least one solution x in F_1 that is not worse than y in all the N objectives (in other

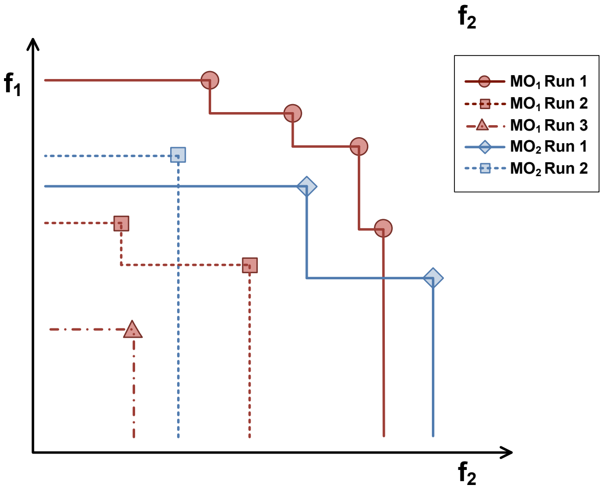


FIGURE 5.4: Two stochastic multiobjective optimizers are compared with regard to two objectives (f_1 and f_2) by using dominance ranking: MO_1 (red) and MO_2 (blue). In this case, both objectives must be maximized. According to the definition in Equation 5.4, none of the Pareto fronts is *better* than Runs 1 in MO_1 and MO_2 and, therefore, both front are assigned the lowest rank 1. However, all Pareto fronts are *better* than Run 3 in MO_1 and accordingly its rank is 5 (the worst one). Overall, the resulting ranks are $(1, 3, 5)$ for MO_1 and $(1, 2)$ for MO_2.

words, each y is *weakly dominated* by any x). Formally, if the objectives have to be maximized:

$$\forall y \in F_2, \exists x \in F_1 : f_i(x) \geq f_i(y) \; \forall i = 1, \ldots, N \tag{5.4}$$

Therefore, each Pareto front is ranked according to the number of Pareto fronts that are *better*. Lower rankings then represent better Pareto fronts than higher ones. An example of how fronts are ranked for two different MO algorithms (MO_1 and MO_2) is shown in Figure 5.4. The dominance ranking is useful as a first step to know the order of the proposed algorithms. However, this ranking indicator does not provide information about the differences between different Pareto fronts in terms of their quality [Zitzler et al., 2008].

- The **hypervolume** [Zitzler et al., 2007] calculates the portion of the objective space that is covered by the Pareto front with respect to a bounding point. The bounding point is the reference point to calculate the hypervolume and it must be located out of the all objective range. If all objectives

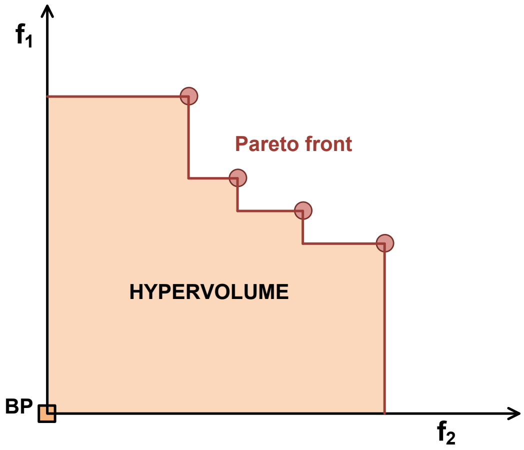


FIGURE 5.5: Graphical definition of the hypervolume for two objective (f_1 and f_2). The hypevolume value is calculated as the area (orange region) which is covered from the Pareto front (red line) to the bounding point (BP) (orange square). Since objectives are maximized in this case, the BP is located under the minimal objective range.

are minimized it is located over the maximum objective values whereas it is located under the minimum objective values when maximizing. In both cases, a MO algorithm outperforms another one when it returns a higher hypervolume value. The hypervolume indicator is graphically explained in Figure 5.5 for the maximization of two objectives (f_1 and f_2). Unlike the dominance ranking, the hypervolume indicator makes possible to determine the differences in quality between different Pareto fronts, even for those not dominated. However, these differences need to be assessed by any statistical test.

- The **attainment function** method summarizes the Pareto fronts from several runs in each MO algorithm in terms of the so-called empirical attainment function (EAF) [Fonseca et al., 2011]. The main idea in the attainment function is to determine the probability that a general solution x is *weakly dominates* by the Pareto front of the MO algorithm. This probability can be estimated for stochastic MO algorithms like the number of Pareto fronts (several runs) that dominates each solution x in the objective space, normalized by the total number of runs. Therefore, the objective space is cut in several regions according to the estimated probability (see graphical rep-

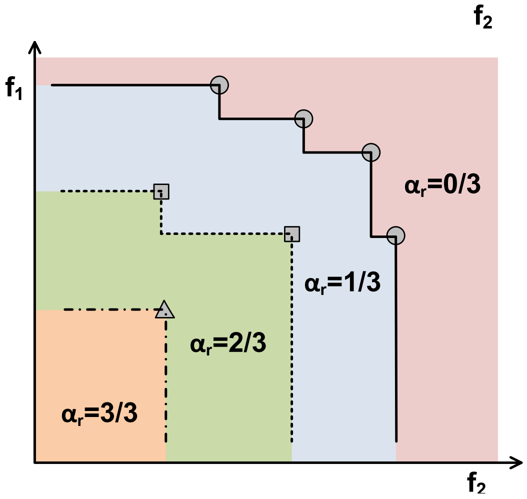


FIGURE 5.6: Graphical representation of the empirical attainment function (EAF). The EAF divides the objective space in regions according to the probability of being weakly dominated by the multiobjective optimizer. Four regions can be differentiated in this example: the orange region is weakly dominated by the three fronts and therefore it is assigned the probability 1 (3/3); the red region is not assigned probability (0/3) since it is not dominated by any front; the remaining blue and green regions are assigned probabilities 1/3 and 2/3 because they are weakly dominated by one and two fronts, respectively.

resentation in the example of Figure 5.6). The EAF for a solution x is then calculated as:

$$\alpha_r(x) = \frac{1}{r}\sum_{i=1}^{r} I(F_i \geq \{x\}) \tag{5.5}$$

where F_i are the different Pareto front in r runs and $\geq$ represents the weak dominance relationship explained in Equation 5.4. The indicator function $I(.)$ is true if the F_i weakly dominates x and false, otherwise. The comparison of two MOs is then performed by statistically comparing their both EAFs. The transformation from Pareto fronts to the EAFs keeps more information than other approaches like hypervolume or dominance ranking. However, this approach is computationally expensive and it is only applicable with a few objectives (ideally, two objectives) [Fonseca et al., 2011].

As previously commented, all these indicators must be additionally assessed by statistical tests in order to determine their significance comparing different runs. It is also important to note that all these measures have been described in case of all objectives are maximized, but they could be analogously applied for minimizing objectives.

5.3 MO-SAStrE Implementation

Multiple sequence alignments (MSAs) can be considered multiobjective problems since there is no consensus about how alignments should be adequately evaluated. Several criteria and score schemes are then being taken into account for this purpose [Nuin et al., 2006]. Additionally, including several suitable objectives can also provide more flexibility in the optimization procedure of MSAs. Consequently, a multiojective optimizer based on genetic algorithms (MO-SAStrE) is presented in this chapter. As previously introduced, MO-SAStrE is designed to include three different objectives: 3D structure evaluation with the STRIKE score, totally conserved columns and percentage of gaps in alignments. The multiobjective approach is developed through the previously described NSGA-II scheme [Deb et al., 2002], as it is an efficient and recognized method for MO problems. In addition to this multiobjective strategy, the proposed optimizer includes a own-designed representation (codification) to encode the alignments in an appropriate way for the optimization purpose. Novel mutation and crossover operators are also implemented taking advantage of the proposed representation.

The full procedure followed for MO-SAStrE is shown in Figure 5.7. First, some well-known, popular and fast aligners are applied to generate a partial population of suboptimal alignments (individuals). The initial population is then created filling the partial population to N individuals (where N defines the population size) by means of the crossover operator. Subsequently, as determined by standard genetic algorithms, the population is extended by the mutation and crossover operators (offspring), according to their assigned probabilities p_c and p_m respectively (see the 'Operators' stage in Figure 5.7). The best individuals are then selected from the extended population to be included in

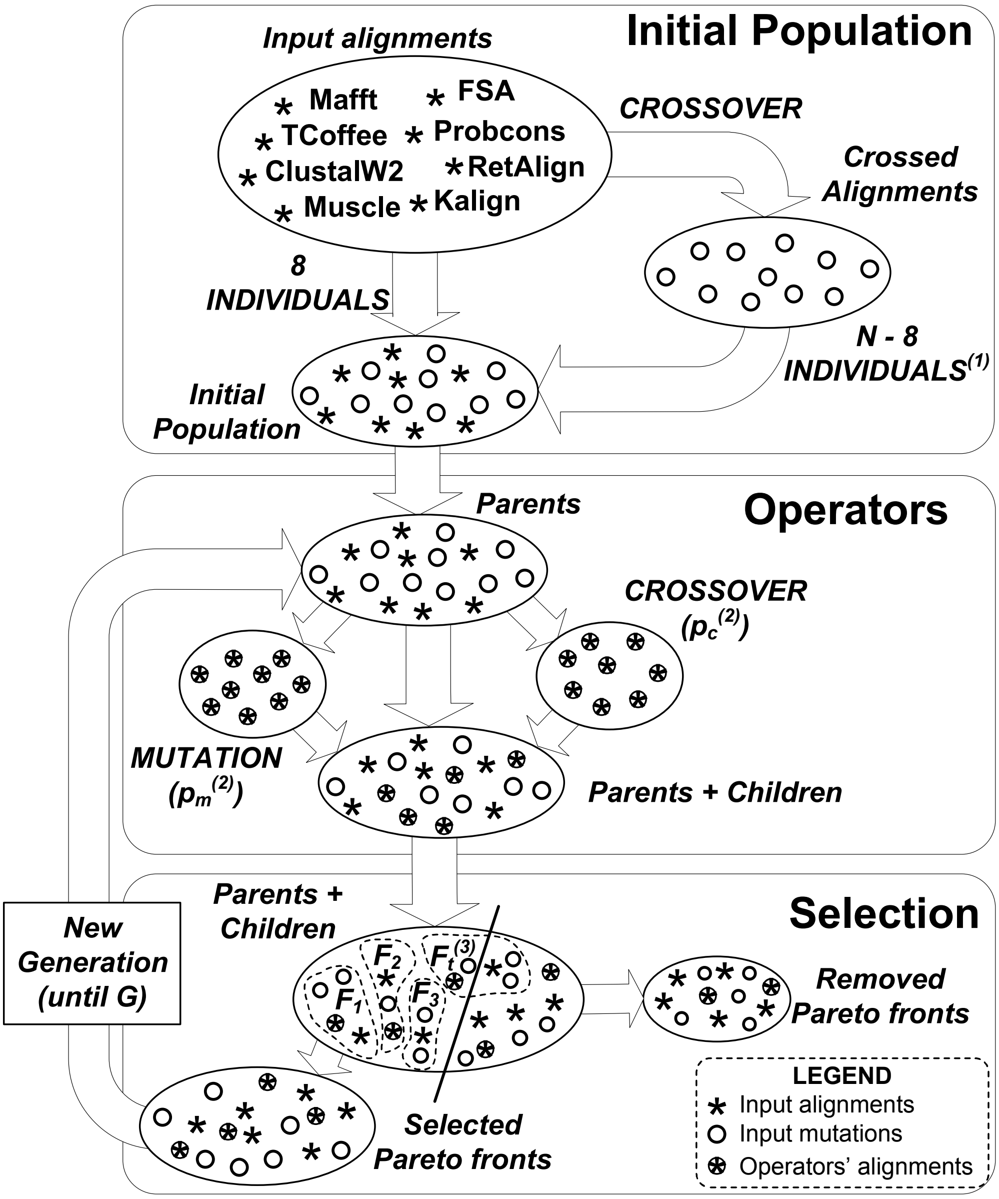

Initial Population
Input alignments
Mafft
FSA
TCoffee
Probcons
ClustalW2
RetAlign
Muscle
Kalign
CROSSOVER
Crossed Alignments
8 INDIVIDUALS
N - 8 INDIVIDUALS(1)
Initial Population
Operators
Parents
CROSSOVER (pc(2))
MUTATION (pm(2))
Parents + Children
Selection
New Generation (until G)
Parents + Children
F2
Ft(3)
F1
F3
Removed Pareto fronts
Selected Pareto fronts
LEGEND
Input alignments
Input mutations
Operators' alignments

FIGURE 5.7: MO-SAStrE flowchart. The initial population is performed by eight previous suboptimal alignments and filled with the crossover operator until N individuals $^{(1)}$ (*Initial population* stage). Each population is then extended by using the mutation and crossover operators, with p_c and p_m representing the crossover and mutation probabilities $^{(2)}$ (*Operators* stage). Finally, the selection procedure is done by using the NSGA-II selection approach (*Selection* stage). F_t defines the last included Pareto front, where individuals must be selected according to the crowding distance $^{(3)}$.

the next generation. This selection is progressively carried out as proposed by the NSGA-II procedure (section 5.2.2), taking the optimal non-dominated solutions (Pareto fronts) from the current population. If all individuals in the last included Pareto front (F_t in Figure 5.7) cannot be included, they must be selected according to the crowding distance (see equation 5.3 for details). This process is repeated during a number of generations in order to iteratively optimized the population. Finally, when the total number of generations (G) is reached or the Pareto front does not change in a specified number of consecutive generations, the optimal Pareto front in the last population is returned as the set of optimized alignments. This implementation of the NSGA-II approach was taken from the *Global Optimization* toolbox of Matlab® (R2010b version).

Following, the different processes in the MO-SAStrE implementation are described in detail: the representation of individuals (section 5.3.1), the initialization of the first population (section 5.3.2), the mutation and crossover operators (section 5.3.3), the fitness functions (section 5.3.4) and the termination condition (section 5.3.5).

5.3.1 MSA Representation

Other genetic algorithms implemented for MSA tools have encoded alignments by using the classical representation [Notredame and Higgins, 1996, Taheri and Zomaya, 2009]: the standard 20-letters alphabet for amino acids and the '-' symbol for gaps (see Figure 5.8(A)). However, it has been seen that this representation could lead to more complex and inefficient processes when working with the GA operators.

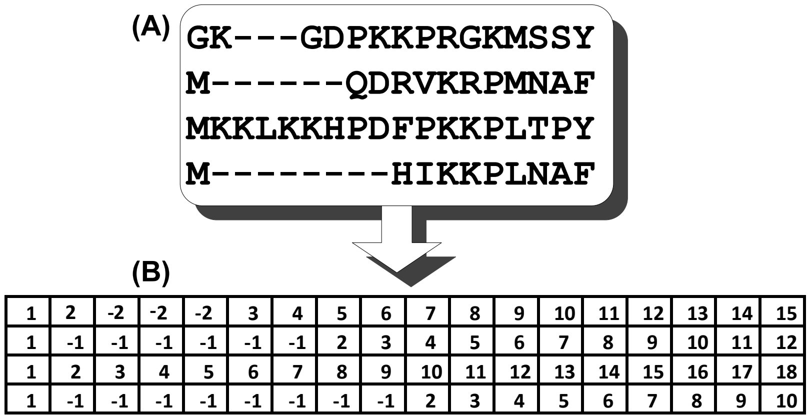
Consequently, a novel pseudo-codification is proposed here. In addition to the standard representation, alignments in MO-SAStrE are represented with a numerical matrix taking into account two main characteristics:

1. Each amino acid is encoded with its position in the corresponding sequence to which it belongs.

2. Gaps are encoded with the position of the last amino acids in the sequence where they have been included, but with a negative value.

Input alignments are encoded before they are included in the first population. This representation is defined as *pseudo-codification* because the original sequences are still needed in some stages of the optimization. Therefore, the whole optimization is done by using both encoded alignments (individuals) and their corresponding sequences. More specifically, the mutation and crossover operators are applied by using the pseudo-codification whereas the original sequences are only needed for the fitness evaluation. After the optimization ends, individuals are decoded again and therefore returned to the standard alignment representation. An example of the proposed codification is shown in Figure 5.8.

This pseudo-codification has a high impact in the posterior own-designed operators (specially, the crossover operator). Mainly, the proposed representation aims to easily identify those positions where the crossover can be applied. It avoids possible mistakes and difficulties performing the crossover, increasing the efficiency in the alignment management (see details in section 5.3.3).



```
GK---GDPKKPRGKMSSY
M------QDRVKRPMNAF
MKKLKKHPDFPKKPLTPY
M--------HIKKPLNAF
```

1	2	-2	-2	-2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	-1	-1	-1	-1	-1	-1	2	3	4	5	6	7	8	9	10	11	12
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
1	-1	-1	-1	-1	-1	-1	-1	-1	2	3	4	5	6	7	8	9	10

FIGURE 5.8: Pseudo-codification of alignments in MO-SAStrE. (A) Standard representation of a multiple sequence alignment. (B) Alignment encoded by a matrix of integer values: positions in their corresponding sequences (amino acids) and positions of the last amino acid in such sequences denoted with a negative sign (gaps).

5.3.2 Initialization of the population

Since an optimization procedure is proposed in this chapter, the initial population in MO-SAStrE is not randomly generated. Instead of that, a set of suboptimal alignments to be optimized is included in the population. Including previously obtained solutions to build the initial set in genetic algorithms has widely been shown in the literature to be efficient and better than random initializations [Dasgupta et al., 2009, Tsujimoto et al., 2009].

Therefore, alignments already determined by other tools are quickly generated to form the initial population. These alignments are obtained by fast but often inaccurate aligners. Some of the aligners previously analyzed in the comparison of Chapter 2 (section 2.4) are implemented in MO-SAStrE. Specifically, the fastest aligners in this previous comparison are considered. These aligners are just those ones implemented under progressive and consistency-based approaches. Aligners considering additional features (e.g. 3D-Coffee and Promals) are discarded since they are too time-consuming for the initialization process. Moreover, similarly to these complex tools, the aim of MO-SAStrE is to achieve optimized alignments using additional information related with the tertiary structure. In fact, MO-SAStrE will be subsequently compared with 3D-Coffee.

Alignments from eight different tools are then included in the initial population. Among progressive algorithms, ClustalW [Thompson et al., 1994], Muscle [Edgar, 2004], Kalign [Lassmann and Sonnhammer, 2005], Mafft [Katoh et al., 2002] and RetAlign [Szabo et al., 2010] are chosen in the initial population. Additionally, three algorithms based on consistency are also added in the population: T-Coffee [Notredame et al., 2000], FSA [Bradley et al., 2009] and ProbCons [Do et al., 2005]. All these tools have already been described in detail in Chapter 2, section 2.2.2.

All aligners are run with their default parameters, though they can be modified according to the user preferences. This specific initial population is chosen because alignments can be quickly constructed but they might need to be improved. Anyway, in case of more accurate initial alignments were provided, MO-SAStrE could even return better output alignments though the improvement range could be reduced.

5.3.3 Genetic algorithm operators

MO-SAStrE includes the two standard operators in GAs: mutation and crossover. These operations are applied to a subset of randomly chosen alignments from the population according to the probabilities p_m and p_c, respectively. The goal of these operators is to include in the population new alignments not considered before, taking into account previous alignments. These operators are run for each generation in the optimization procedure. Since sequences in alignments cannot be altered, the implementation of both operators might not be applied in its classical form. Instead of that, some modifications are introduced in their performances:

- The **mutation operator** only mutates gaps, in order to keep the order of amino acids. A random set of closed gaps (gap region) is then shifted to another random position in the same sequence. Two important aspects are introduced with the proposed definition of mutation:

    1. New variants of alignments not taken into account until now are introduced by this operator. These variants add different gap distributions that could improve the quality of the previous alignments.
    2. The gap region can sometimes be shifted to another position where all the other sequences also contain gaps. Therefore, full columns of gaps can be removed, thus reducing the number of gaps.

  A specific example of the mutation operation, including the gap reduction process, is shown in Figure 5.9.

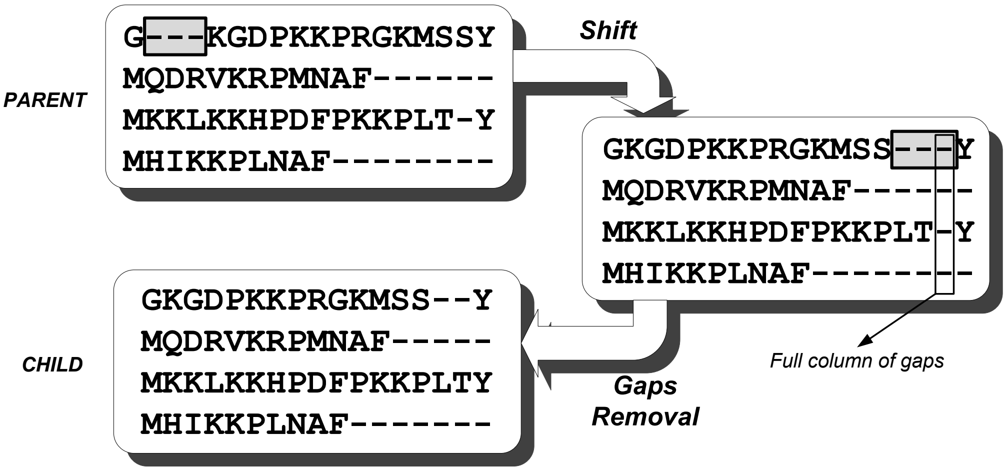

PARENT
G---KGDPKKPRGKMSSY
MQDRVKRPMNAF------
MKKLKKHPDFPKKPLT-Y
MHIKKPLNAF--------
Shift
GKGDPKKPRGKMSS---Y
MQDRVKRPMNAF------
MKKLKKHPDFPKKPLT-Y
MHIKKPLNAF--------
Full column of gaps
Gaps
Removal
CHILD
GKGDPKKPRGKMSS--Y
MQDRVKRPMNAF-----
MKKLKKHPDFPKKPLTY
MHIKKPLNAF-------

FIGURE 5.9: Mutation procedure. A region of closed gaps (shading area) is randomly chosen and shifted to another random position. Full columns of gaps are then removed if they are found.

- The **crossover operator** is designed as an one-point crossover. This operator is carefully redesigned in order to maintain the order in the sequences. It takes advantage of the pseudo-codification explained above in order to facilitate the crossing between parents. The proposed procedure, which is graphically explained in Figure 5.10, is performed as follows: *(i)* one column from one parent is randomly selected, dividing it into two blocks; *(ii)* the sequence positions in this selected column are identified in the second parent, but not necessarily in one single column;*(iii)* the second parent is also split into two blocks by these positions in different columns; *(iv)* finally, the blocks in both parents are crossed. In order to match blocks from both parents, those undefined positions in the second parent are filled with gaps. Thus, it can be assured that the obtained children have crossed the parents' regions without altering their sequences. The proposed crossover is considered the most important operator in MO-SAStrE in order to improve the input alignments. Since some aligners can provide more accurate alignments in some regions than others, this operation is essential for the optimization purpose. Even though some gaps are introduced to assemble the two parents, crossed children could iteratively incorporate the best sections from different parents, providing more accurate alignments.

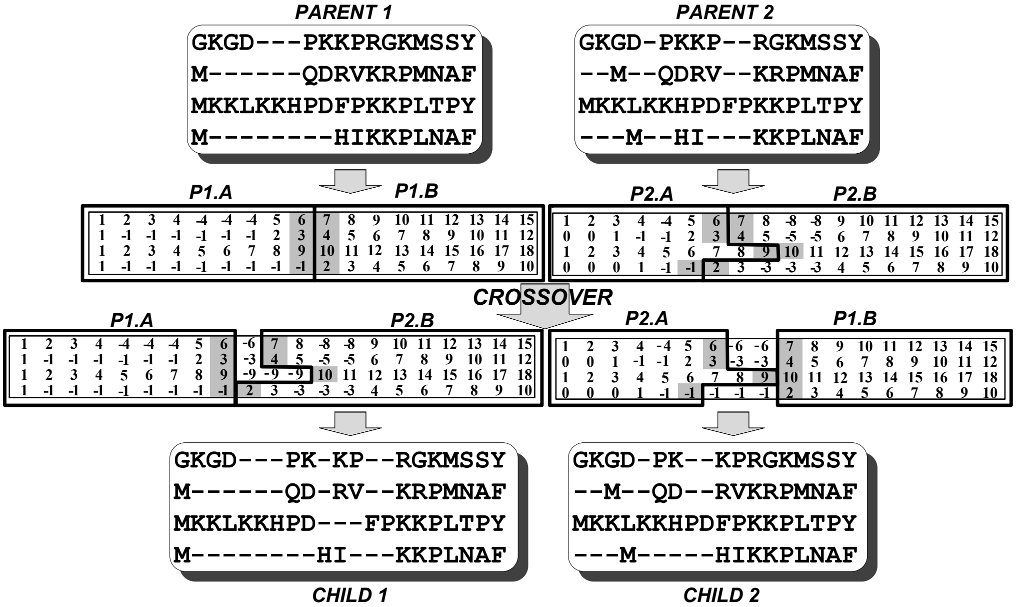

PARENT 1
GKGD---PKKPRGKMSSY
M------QDRVKRPMNAF
MKKLKKHPDFPKKPLTPY
M--------HIKKPLNAF
PARENT 2
GKGD-PKKP--RGKMSSY
--M--QDRV--KRPMNAF
MKKLKKHPDFPKKPLTPY
---M--HI---KKPLNAF
P1.A
P1.B
P2.A
P2.B
CROSSOVER
GKGD---PK-KP--RGKMSSY
M------QD-RV--KRPMNAF
MKKLKKHPD---FPKKPLTPY
M--------HI---KKPLNAF
CHILD 1
GKGD-PK--KPRGKMSSY
--M--QD--RVKRPMNAF
MKKLKKHPDFPKKPLTPY
---M-----HIKKPLNAF
CHILD 2

FIGURE 5.10: Crossover operator. Both standard and novel codification are shown (see the codification procedure in Figure 5.8). The first parent is divided into two blocks (P1.A and P1.B) according to a randomly selected column. Two blocks are also obtained from the second parent according to the same positions of the selected column (P2.A and P2.B). The blocks are crossed by filling with gaps in order to assemble them.

5.3.4 Multiobjective fitness function

Since MO-SAStrE is designed as a multiobjective algorithm, three complementary scores are included to evaluate each alignment: STRIKE score, percentage of totally conserved columns and percentage of non-gaps. These three objectives have been already successfully applied in this dissertation, namely in the dataset of features of the previously presented scoring schemes. In fact, it was proved that these three evaluations were highly relevant to determine the quality of alignments (see Chapter 4 for details).

The STRIKE score (f_1): this evaluation is a novel index for calculating alignment accuracies by using at least one known tertiary structure [Kemena et al., 2011]. The structural information is retrieved from the Protein Data Bank (PDB) [Berman et al., 2000]. STRIKE estimates the contacts of the sequence containing structural information by determining the distances between its amino acids. Specifically, two atoms is said to be in intramolecular contact when a solvent molecule cannot be inserted between their molecular surfaces [Connolly, 1983]. The distance between in two amino acids to determine if they are in contact are then calculated from the spatial position of atoms in the amino acids provided by the PDB structure. In order to avoid contacts produced by the secondary structure, STRIKE only considers those contacts that involve amino acids separated by at least five amino acids in the sequence. After estimating the contacts in one sequence according to its tertiary structure, the pairs of amino acids aligned in the same positions as such contacts are retrieved for the remaining sequences. Such pairs of amino acids are then scored according to a novel scoring matrix (STRIKE matrix) also based on structural information Kemena et al. [2011]. Specifically, the STRIKE matrix was performed taking into account the contacts found in a dataset of alignments from several databases. For each possible pair of amino acids x and y, the score $M_{x,y}$ in the STRIKE matrix is calculated as:

$$M(x,y) = 10 \times ln\left(\frac{v_{xy}}{v_x v_y}\right) \tag{5.6}$$

where v_{xy} is the frequency of contacts involving amino acids x and y and v_x and v_y are the single amino acid frequency in the considered dataset. Therefore, if a sequence T with known structures is applied to determine the STRIKE score of an alignment, it can be calculated as:

$$f_1 = \sum_{i=1;i\neq T}^{N} \sum_{j,k=1}^{L} M(x_{ij}, x_{ik}) \times IsContact(x_{ij}, x_{ik}) \tag{5.7}$$

where x_{ij} and x_{ik} are each pair of amino acids in the *i-th* sequence (different from T), N and L are the number the sequences and the length of the alignment, and the function $IsContact$ is defined as:

$$IsContact(x_{ij}, x_{ik}) = \begin{cases} 1 & if\ x_{ij}\ and\ x_{ik}\ are\ in\ contact \\ 0 & otherwise \end{cases} \tag{5.8}$$

In case of several available structures, the STRIKE score is separately calculated for each structure and the score is finally averaged. This evaluation permits to identify the accuracy in the alignments better than other well-known scores such as BLOSUM [Henikoff and Henikoff, 1992] or PAM [Dayhoff et al., 1979]. Moreover, the STRIKE score clearly outperforms the other evaluations when sequences are evolutionarily more distant. STRIKE score also shows a strong non-parametric correlation with the BAliscore values. That is, both BAliscore and STRIKE usually identify the same alignment as the best one when two different alignments are compared (in around 79% of cases) [Kemena et al., 2011]. Consequently, STRIKE plays a key role in the MO-SAStrE optimizer, evaluating the structural correctness in the alignment and providing an accurate score scheme for the improvement of alignments.

The percentage of totally conserved (TC) column (f_2): the TC measure is a classical score in MSAs. It takes into account the number of columns that are completely aligned with exactly the same amino acid:

$$f_2 = \sum_{i=1}^{L} \frac{\mathcal{TC}(x_i)}{L} \tag{5.9}$$

where the function $\mathcal{TC}(x_i)$ is defined as:

$$\mathcal{TC}(x_i) = \begin{cases} 1 & if\ x_{ij} = x_{i1}\ \forall j = 2, \ldots, N \\ 0 & otherwise \end{cases} \tag{5.10}$$

with x_{ij} denoting the element in the *i-th* position of the *j-th* sequence, x_i defining the entire column in the *i-th* position of the alignment and L the length of the alignment. The number of complete columns is a widely accepted evaluation applied by several benchmarks [Edgar, 2004, Thompson et al., 2005]. This objective is relevant in MO-SAStrE because some progressive methodologies usually favor partial alignment but not complete columns, producing suboptimal alignments [Mirarab and Warnow, 2011]. Additionally, this score also indicates more conserved or important regions in sequences.

The percentage of non-gaps (f_3): this score is included as the third proposed objective. This function measures the number of amino acids with respect to the number of gaps. Formally, for an encoded alignment (individual), this objective is calculated as:

$$f_3 = \sum_{i=1}^{N} \sum_{j=1}^{L} \frac{\mathcal{A}(c_{ij})}{N \cdot L} \tag{5.11}$$

where the function $\mathcal{A}(x)$ is defined as:

$$\mathcal{A}(c) = \begin{cases} 1 & if\ c > 0 \\ 0 & otherwise \end{cases} \tag{5.12}$$

with c_{ij} being the corresponding value in the *i-th* position of the *j-th* sequence for the encoded alignment, N the number of sequences and L the length of the alignment. This objective has a key role since some methodologies often overuse gaps in order to increase the identity percentage, what indeed can reduce the real quality of alignments [Nozaki and Bellgard, 2005]. Consequently, the proposed optimization tries to reduce the number of gaps, building more compact and realistic alignments.

Therefore, MO-SAStrE aims to optimize alignments according to a novel evaluation based on conserved structural information in sequences, but also reducing the number of gaps and keeping fully conserved sections. As notated, depending on the objective the proposed pseudo-codification or the original sequences could be necessary. Besides, it is important to note that the three objectives are designed to be maximized.

5.3.5 Termination condition

The last aspect to take into account in MO-SAStrE is the termination condition. This condition allows us to determine when the optimization process has finished or has converged to an optimal solution. In this case, although a maximum limit of generations (G) is also established, the optimization procedure stops if the best multiobjective fitness function (best Pareto front) does not change over a particular number of generations (S).

5.4 MO-SAStrE Assessment

The proposed multiobjective optimizer is tested through a dataset including several sets of sequences. Additionally, once the optimization is performed for this dataset, some statistical tests are included in order to compare with: *(i)* the input dataset of alignments, thus determining if the optimization was significant; and (ii) other aligners and genetic algorithms implemented for the same dataset.

5.4.1 Dataset of sequences

The BAliBASE benchmark (v3.0) [Thompson et al., 2005] is again proposed to be the dataset used for the multiobjective optimization. It is interesting to remind that this benchmark provides 218 sets of sequences specially designed to be aligned. In this case, some relevant advantages have been considered to choose this dataset:

- The sets of sequences in BAliBASE are classified according to, besides of other properties, the sequence identities. It can then be studied if MO-SAStrE is capable to optimize the alignments with sequences less related as well as those with higher identity (see identity percentages in Table 2.1 of Chapter 2). For instance, the first subset RV11 includes the less related sequences (< 20% of identity).

- The sequences included in this dataset were manually extracted from the Protein Data Bank (PDB) [Berman et al., 2000]. Therefore, they generally have known structures, which is essential for the performance of the STRIKE evaluation (first objective in MO-SAStrE).
- The reference alignments provided by BAliBASE and the BAliscore tool are useful to determine the real accuracy of the optimized alignments. Therefore, the references and the quality indicator BAliscore can be used to validate the proposed optimization and compare with other optimizers and aligners.
- The remaining tools that are compared in this chapter (mainly, other GA optimizers) also considers this dataset. This is an relevant advantage to have a fair comparison since these tools are unfortunately not freely available but the results provided in their corresponding publications for the BAliBASE dataset can be used instead.

5.4.2 Multiobjective and statistical assessment

MO-SAStrE is defined as a stochastic process, because the algorithm converges to different solutions when it is applied several times to the same problem. Consequently, several runs of the same problem must be carried out in order to statistically evaluate its performance.

As commented before, Zitzler et al. [2008] proposed several indicators to assess multiobjective stochastic optimizers: the hypervolume indicator, dominance rankings or the attainment function method (see section 5.2.3 for details). The main goal of these quality indicators is to reduce the provided scores (three objectives) of multiple optimal solutions (Pareto front) to one single score, making the algorithm easier to assess. In this sense, the hypervolume indicator (HV) [Zitzler et al., 2007] is selected for the validation of the proposed multiobjective optimizer. This option has been considered the optimal one for this algorithm because it returns more accurate values than dominance rankings but without the computational complexity of other indicators as the attainment function. Besides, as previously commented, differences between optimizers are more easily appreciable with the HV indicator.

However, given the stochastic nature of MO-SAStrE, the hypervolume is not enough to assess the optimization process. Since each problem in the dataset is run several times, several HV values are obtained for each problem. These values from the MO-SAStrE optimization are also compared with the initial HV provided by the previous aligners in the 218 problems. Non-parametric statistical test are commonly necessary to compare and to validate several runs in stochastic multiobjective approaches [Conover, 1999]. Specifically, the Mann-Whitney rank sum or the Kruskal-Wallis tests are recommended in the literature for multiobjective optimizers using the hypervolume indicator [Zitzler et al., 2008].

In this case, the rank sum test proposed by Mann et al. [1947] is considered. This non-parametric test precisely compares two independent algorithms, determining if the differences between them are relevant. Here, the Mann-Whitney test assesses if a significant improvement is achieved in the alignments optimized by MO-SAStrE against those ones initially constructed by other faster aligners. This comparison is performed for each of the 218 problems independently by running MO-SAStrE several times and using the resulting hypervolume values.

5.4.3 Other similar optimizers

The performance of MO-SAStrE is also evaluated by comparing with other evolutionary algorithms in MSAs, namely SAGA [Notredame and Higgins, 1996], MSA-GA [Gondro and Kinghorn, 2007], RBT-GA [Taheri and Zomaya, 2009] and VDGA [Naznin et al., 2011].

- **The Sequence Alignment Genetic Algorithm (SAGA)** [Notredame and Higgins, 1996] was one of the first genetic algorithms applied to MSAs. It develops a standard EA algorithm, providing a large number of operators that are gradually applied to obtain more accurate alignments. The operators are dynamically selected according to the evolution process in the GA. However, its main disadvantage is that the proposed dynamic scheduling of operators implies larger computational time.

- **The Multiple Sequence Alignment Genetic Algorithm (MSA-GA)** [Gondro and Kinghorn, 2007] designs a simple genetic algorithm for optimizing MSAs. Authors in MSA-GA suggest that the complex operators of SAGA are unnecessary and they only propose a simple crossover operator instead. This operator can work in two ways: crossing regions in the alignments similarly to MO-SAStrE (vertical crossover) or crossing one full sequence with gaps in the alignments (horizontal crossover). In addition, the so-called WSP score (Weighted Sum of Pairs) is applied for the fitness function. This score is the same considered in the ClustalW tool for the pairwise alignment evaluation.

- **The Rubber Band Technique Genetic Algorithm (RBT-GA)** [Taheri and Zomaya, 2009] is a recent optimizer that proposes a hybrid approach taking advantage of the Rubber Band technique (RBT) and a genetic algorithm. Here, each alignment is encoded by using a RBT environment. The RBT environment is inspired by the behavior of an elastic rubber band. This behavior is said to be analogous to the locations of biologically related regions in sequences. A simple *Sum-Of-Pairs* (SP) score with gap opening and extension penalizations is used for the fitness function. Their experimental results showed that the overall performance of RBT-GA was better than previous optimizers according to the BAliBASE benchmark Taheri and Zomaya [2009].

- **The Vertical Decomposition Genetic Algorithm (VDGA)** [Naznin et al., 2011] implements a vertical decomposition procedure in order to create several subsequences from original sequences. These subsequences are independently aligned by a guide tree approach. The genetic algorithm then reassembles the obtained sub-alignments in a new full alignment, also applying the WSPM score in the fitness function.

As it can be observed, the principal difference in MO-SAStrE with respect to these other genetic optimizers is a more complex fitness evaluation. Specifically, MO-SAStrE originally incorporates a multiobjective approach with one evaluation based on structures. Additionally, although the operators are similar between several methods, the proposed pseudo-codification contributes to a more efficient handling of these operators.

Other standard non-genetic aligners are included in addition for these comparisons:

- ClustalW [Thompson et al., 1994]
- MultAlign [Barton and Sternberg, 1987]
- *Pattern-Induced Multi-sequence Alignment* (PIMA) [Smith and Smith, 1992]
- PILEUP [Devereux et al., 1984]
- Dialign [Morgenstern et al., 1996]
- *Hidden Markov Model Training* (HMMT) [Eddy, 1995]
- PRRP [Gotoh, 1996].

These tools have been chosen because they have already been included in similar comparisons in the literature [Gondro and Kinghorn, 2007, Naznin et al., 2011]. Besides, 3D-COFFEE [O'Sullivan et al., 2004] is also considered order to evaluate MO-SAStrE against another aligner using structural information.

Since some of these tools are not freely available, the validation of MO-SAStrE against these other aligners has not been possible for the full proposed dataset. Instead of that, this comparison has been addressed by using the experimental results provided in the VDGA publication [Naznin et al., 2011]. In VDGA, BAliscore values of different problems in BAliBASE v2.0 were considered. Also, the same problems were previously included in the experimental results of the MSA-GA and RBT-GA publications [Gondro and Kinghorn, 2007, Taheri and Zomaya, 2009]. However, since BAliBASE v3.0 is applied in MO-SAStrE, a subset with the 20 common problems included in both versions of BAliBASE is taken.

Therefore, MO-SAStrE is statistically assessed by comparing it against each MSA tool for this subset. Since pairwise comparisons without repetitions are performed in this case, the non-parametric signed-rank Wilcoxon test is employed [Wilcoxon, 1945]. According to the Wilcoxon test, the MO-SAStrE optimizer is said to statistically outperform other aligner if its mean rank is significantly better. The Wilcoxon test has been applied and widely described before in this dissertation, namely section 2.4.

5.5 Experimental results and discussion

5.5.1 Selection of parameters in the genetic algorithm

The fist step in the validation of a genetic algorithm is to estimate the parameters that will maximize the optimization. In order to configure the proposed multiobjective algorithm, six different parameters must be provided: population size, number of generations, probabilities of mutation and crossover, the number of generations considered in the termination condition and repetitions per problem. Although some tuning algorithms and more complex system can be proposed to accurately optimize these parameters, they are selected in this case according to the standard values commonly used by genetic algorithms [Eiben and Smith, 2008]. The applied parameters are then summarized in Table 5.1.

First, the population size is set to 100 alignments (individuals). The same population size was also included in GA methods which will be compared, such as SAGA [Notredame and Higgins, 1996] or VDGA [Naznin et al., 2011]. On the other hand, a maximum limitation of 500 generations is defined. Nevertheless, if there is no changes in the best fitness function (optimal Pareto front) during 50 consecutive generations, the optimization stops (termination condition).

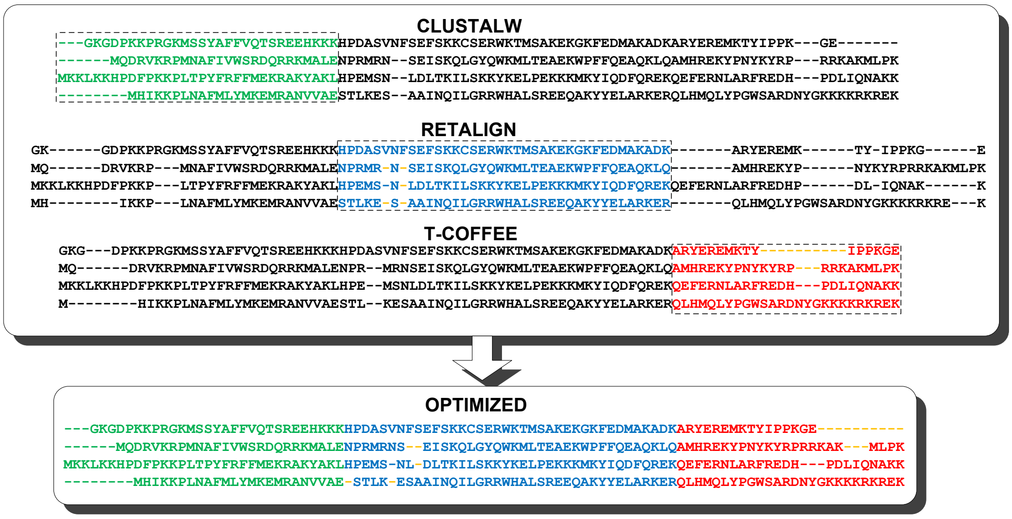
TABLE 5.1: MO-SAStrE parameter configuration. Six parameters were determined according to common values proposed by Eiben and Smith [2008] for genetic algorithms.

Parameter	**Value**
Population Size (N)	100
Number of generations (G)	500
Crossover probability (p_c)	0.8
Mutation probability (p_m)	0.2
Number of generations without changes (S)	50
Repetitions of each problem (r)	10

These two parameters are kept large enough to assure the convergence and optimization of the alignments. In fact, only in a few cases the total number of generations will not be reached.

After these two parameters are set, the operator probabilities are determined. Since the crossover has been considered the main operator for this optimization, it is assumed that its probability must be the same or higher than the mutation's one. Consequently, the following pair of probabilities 80%-20% was set for crossover (p_c) and mutation (p_m), respectively. These probabilities values are a standard combination for genetic algorithms [Eiben and Smith, 2008]. Other pairs of probabilities have also been tested (50%-50%, 60%-40% and 90%-10%) but differences in the optimization were not significant. This parameter configuration is then used to validate MO-SAStrE.

Finally, as the proposed optimizer is defined as a stochastic procedure, each problem must be run several times. In this case, each of the 218 problems in BAliBASE was optimized 10 times. The same number of runs was also included in the experiments of the VDGA [Naznin et al., 2011] and RBT-GA [Taheri and Zomaya, 2009] tools. A total of 2180 Pareto fronts were then obtained (10 solutions by 218 problems).



```
CLUSTALW
---GKGDPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKGKFEDMAKADKARYEREMKTYIPPK---GE-------
------MQDRVKRPMNAFIVWSRDQRRKMALENPRMRN--SEISKQLGYQWKMLTEAEKWPFFQEAQKLQAMHREKYPNYKYRP---RRKAKMLPK
MKKLKKHPDFPKKPLTPYFRFFMEKRAKYAKLHPEMSN--LDLTKILSKKYKELPEKKKMKYIQDFQREKQEFERNLARFREDH---PDLIQNAKK
--------MHIKKPLNAFMLYMKEMRANVVAESTLKES--AAINQILGRRWHALSREEQAKYYELARKERQLHMQLYPGWSARDNYGKKKKRKREK

RETALIGN
GK------GDPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKGKFEDMAKADK-------ARYEREMK------TY-IPPKG------E
MQ------DRVKRP---MNAFIVWSRDQRRKMALENPRMR-N-SEISKQLGYQWKMLTEAEKWPFFQEAQKLQ-------AMHREKYP------NYKYRPRRKAKMLPK
MKKLKKHPDFPKKP---LTPYFRFFMEKRAKYAKLHPEMS-N-LDLTKILSKKYKELPEKKKMKYIQDFQREKQEFERNLARFREDHP------DL-IQNAK------K
MH--------IKKP---LNAFMLYMKEMRANVVAESTLKE-S-AAINQILGRRWHALSREEQAKYYELARKER-------QLHMQLYPGWSARDNYGKKKKRKRE---K

T-COFFEE
GKG---DPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKGKFEDMAKADKARYEREMKTY----------IPPKGE
MQ------DRVKRPMNAFIVWSRDQRRKMALENPR--MRNSEISKQLGYQWKMLTEAEKWPFFQEAQKLQAMHREKYPNYKYRP---RRKAKMLPK
MKKLKKHPDFPKKPLTPYFRFFMEKRAKYAKLHPE--MSNLDLTKILSKKYKELPEKKKMKYIQDFQREKQEFERNLARFREDH---PDLIQNAKK
M--------HIKKPLNAFMLYMKEMRANVVAESTL--KESAAINQILGRRWHALSREEQAKYYELARKERQLHMQLYPGWSARDNYGKKKKRKREK

OPTIMIZED
---GKGDPKKPRGKMSSYAFFVQTSREEHKKKHPDASVNFSEFSKKCSERWKTMSAKEKGKFEDMAKADKARYEREMKTYIPPKGE----------
------MQDRVKRPMNAFIVWSRDQRRKMALENPRMRNS--EISKQLGYQWKMLTEAEKWPFFQEAQKLQAMHREKYPNYKYRPRRKAK---MLPK
MKKLKKHPDFPKKPLTPYFRFFMEKRAKYAKLHPEMS-NL-DLTKILSKKYKELPEKKKMKYIQDFQREKQEFERNLARFREDH---PDLIQNAKK
--------MHIKKPLNAFMLYMKEMRANVVAE-STLK-ESAAINQILGRRWHALSREEQAKYYELARKERQLHMQLYPGWSARDNYGKKKKRKREK
```

FIGURE 5.11: Optimized solution in MO-SAStrE for the first problem in BAliBASE (1^{st} alignment in RV11). In this case, the optimizer builds an alignment joining aligned blocks from ClustalW (green), RetAlign (blue), and TCoffee (red). Gaps are also shifted by the mutation operator (yellow).

5.5.2 Optimization results

As previously described, eight input alignments are introduced in MO-SAStrE for each of the 218 BAliBASE dataset. This tool mainly assembles these previous alignments progressively by using the crossover and mutation operators to randomly obtain optimized alignments. Therefore, each possible solution in MO-SAStrE is composed by several partial regions of previous alignments and gap shifts as shown in the example of Figure 5.11.

A set of non-dominated optimal alignments (Pareto front) is returned when MO-SAStrE finishes its optimization. These obtained alignments are equally good and it is not possible to decide which one is more accurate according to the three objectives. Therefore, the selection of the best alignment only depends on the objective the user considers more useful regarding the specific aligned sequences. In case the alignment with the best STRIKE score (first objective) is chosen, it would obtain more quality according to the structural information in the sequences. In addition, as previously stated, those alignments with higher STRIKE scores are usually improved in terms of accuracy according to the BAlis-

TABLE 5.2: Multiobjective scores for the 1^{st} alignment in the RV11 dataset. Evaluations for input alignments and for optimized MO-SAStrE alignments are represented. Although MO-SAStrE returned 30 alignments from the optimal Pareto front, five representative ones are shown to simplify.

SET	**METHOD**	**MO-SAStrE OBJECTIVES**		
		STRIKE	***NON-GAPS(%)***	***TC(%)***
Input Alignments	ClustalW	2.4544	89.84	1.04
	Muscle	2.6041	89.84	1.04
	Kalign	2.4404	87.12	3.03
	RetAlign	2.2210	79.13	2.75
	Tcoffee	2.5116	89.84	1.04
	ProbCons	2.5116	89.84	1.04
	Mafft	2.3893	87.12	1.01
	FSA	2.1857	69.00	0.80
Optimized Alignments	MO-SAStrE 1	2.4677	90.79	2.11
	MO-SAStrE 2	2.6441	89.84	5.21
	MO-SAStrE 3	2.6864	89.84	4.17
	MO-SAStrE 4	3.0544	82.93	3.85
	MO-SAStrE 5	3.1329	78.41	2.73

core tool [Kemena et al., 2011]. Otherwise, whether the alignment with the highest percentage of non-gaps is selected, a more compact and realistic alignment could be obtained. Finally, a higher number of totally conserved columns in alignments could provide a better quality in terms of the evolutionary homologies among sequences. For this reason, the Pareto fronts are hard to compare as multiple non-dominated solutions are considered and they should be assessed simultaneously according to their three evaluations. In this case, although alignments are equally optimized according to the three objectives, the alignments in the final optimal Pareto fronts are ordered according to their STRIKE score.

The results of the first problem in BAliBASE RV11 subset are presented in Table 5.2. Specifically, the objective values of the eight initial alignments and those from the optimal Pareto front provided by MO-SAStrE (five optimal alignments) are shown in this table. According to these results, the MO-SAStrE alignments outperform the input methodologies in at least one objective in this particular case. The initial and optimized Pareto fronts for the same problem are also graphically shown in Figure 5.12. As appreciated, the optimized front achieves

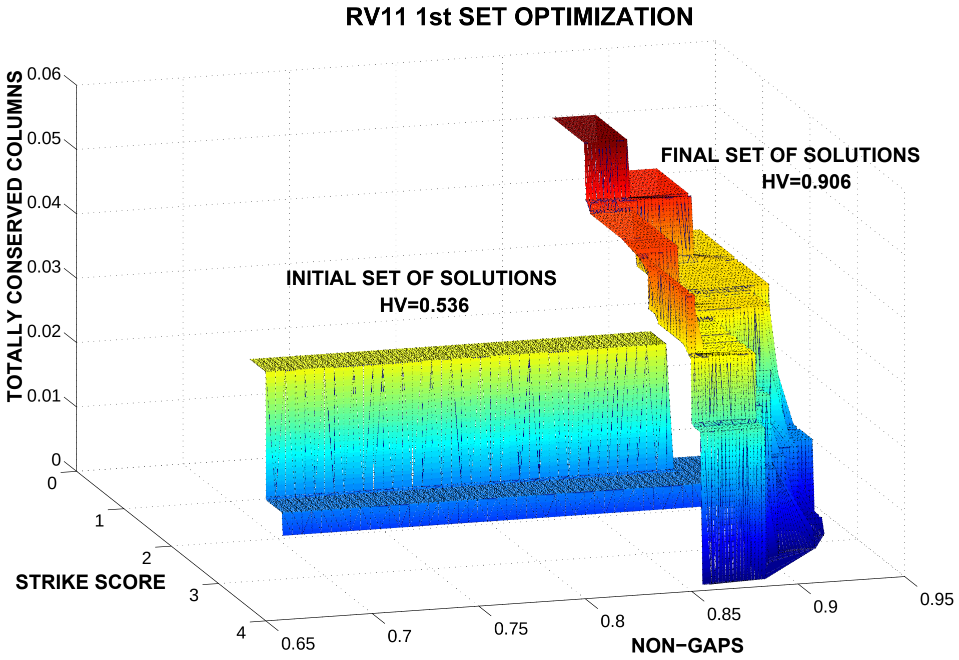


FIGURE 5.12: 3D surfaces for the input and optimized Pareto fronts in the 1^{st} problem of the BAliBASE RV11 subset.

quite higher values in the three objectives than the initial one.

An optimization procedure analogous to the one described for the first problem is carried out for the 218 problems. The averages and standard deviations of the three objectives for the complete dataset are shown in Table 5.3. Here, it can be observed that MO-SAStrE always achieves the best average values in all the three objectives. Additionally, MO-SAStrE also outperforms the input methodologies in terms of alignment accuracies (BAliscore values in Table 5.3). It can then be suggested that the proposed optimization is successfully working.

It is important to remark that this improvement is reached at expenses of a high computing time (~ 10 minutes in average). Despite that, the computational costs obtained by MO-SAStrE are acceptable taking into account that the simplest and fastest input methodologies were chosen in order to be subsequently optimized. Additionally, MO-SAStrE is kept in a range of time similar to more complex methodologies like 3D-Coffee. Moreover, although the improvement

TABLE 5.3: Average scores and standard deviation for the 218 BAliBASE problems optimized by MO-SAStrE.

METHOD	**MO-SAStrE OBJECTIVES**			**BAliscore**
	STRIKE	***NON-GAPS(%)***	***TC(%)***	
ClustalW	1.947±0.815	55.33±21.45	1.67±2.83	0.669±0.213
Muscle	2.181±0.734	52.39±21.05	1.88±2.88	0.724±0.197
Kalign	2.191±0.718	48.02±20.62	1.83±2.86	0.727±0.181
RetAlign	2.176±0.717	49.06±19.64	2.08±3.04	0.703±0.194
Tcoffee	2.131±0.760	46.31±22.52	1.84±2.88	0.760±0.177
ProbCons	2.183±0.723	43.94±22.54	1.83±2.90	0.770±0.169
Mafft	2.229±0.722	50.04±20.96	1.98±2.94	0.773±0.168
FSA	1.849±0.922	31.19±20.30	1.38±2.58	0.685±0.215
MO-SAStrE	2.374±0.619	58.51±21.53	2.44±3.25	0.794±0.152

of computational costs has not been considered a main goal here, MO-SAStrE can still be widely optimized with this purpose by applying different strategies of parallelization.

5.5.3 Hypervolume analysis

Although the previous results have suggested that MO-SAStrE acceptably improves the accuracy for input alignments, it is still necessary to assess this affirmation. As commented before, the hypervolume indicator (HV) proposed by Zitzler et al. [2008] is calculated to reduce the Pareto front of optimal solutions (3 objectives per solution) in a single measure (see details in section 5.2.3). Thus, the HV can be interpreted as a measure of quality which takes into consideration the three proposed objectives simultaneously.

Previously to the HV analysis, the three objectives are normalized to the range [0, 1] in order to give them the same weight in the objective space. Since the three independent objectives must be maximized here, the bounding point should be located at least in the minimum values of the three objectives. Consequently, the bounding point is set to the position (0, 0, 0). The HV values are then distributed in the range [0, 1]. Thus, better Pareto fronts would lead to higher HV values (the higher objectives, the more covered objective space in HV).

The HV values are firstly considered here to assure that each problem successfully converges to an optimized solution. For instance, the HV convergence in four representative problems is shown in the Figure 5.13 (three runs per problems are represented). As appreciated, the convergence is reached in a different number of generations, depending on the specific problem. Anyway, the four shown problems converge before the maximum limit of generation is reached. Similar convergence plots have been obtained for the remaining problems in the dataset. Consequently, it can be considered that the optimization is adequately converging.

The HV values obtained by the MO-SAStrE optimization can also be compared against the HV results from the initial tools (eight input alignments). Thus, the average improvement associated to each BAliBASE subset in terms of the HV indicator is depicted in Figure 5.14. In general, the alignments optimized in MO-SAStrE show strictly better HV values for all the 218 problems. More specifically, the HV values obtained by MO-SAStrE from the 10 runs of each problem always exceed the HV values from the initial alignments. It is also observed that the optimized alignments achieve an average improvement of 63.01% in the whole dataset according to HV values. Such an improvement even increases to 70.34% when dealing with less related sequences and alignments become more difficult (RV11 subset in BAliBASE).

Nevertheless, there are two problems where the improvement did not reach 10%: *1st* and *36th* sets in RV20. These two sets are harder to improve because they belong to a BAliBASE subset with higher similarity percentages and, therefore, the alignments from initial tools are already quite accurate. Moreover, these particular problems also include some special features such as higher number of sequences or highly divergent lengths, making more difficult the optimization.

5.5.4 Statistical assessment of MO-SAStrE

The MO-SAStrE optimizer has already been validated both graphically and in terms of hypervolume. In both cases, it has been observed that MO-SAStrE is

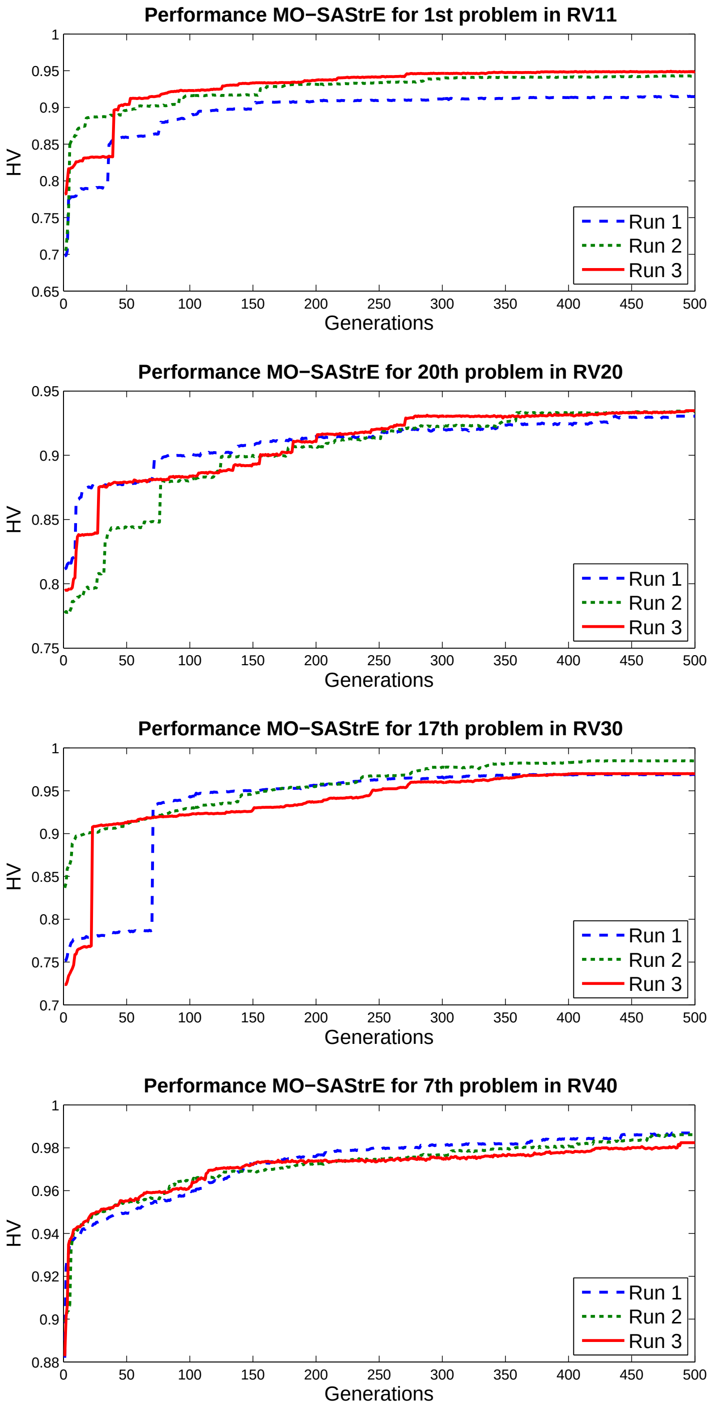


FIGURE 5.13: MO-SAStrE performance in terms of hypervolume (HV) progression (4 different problems). The increase of the HV indicator is represented with respect to the number of generations. For simplicity, only three runs per problem are shown.

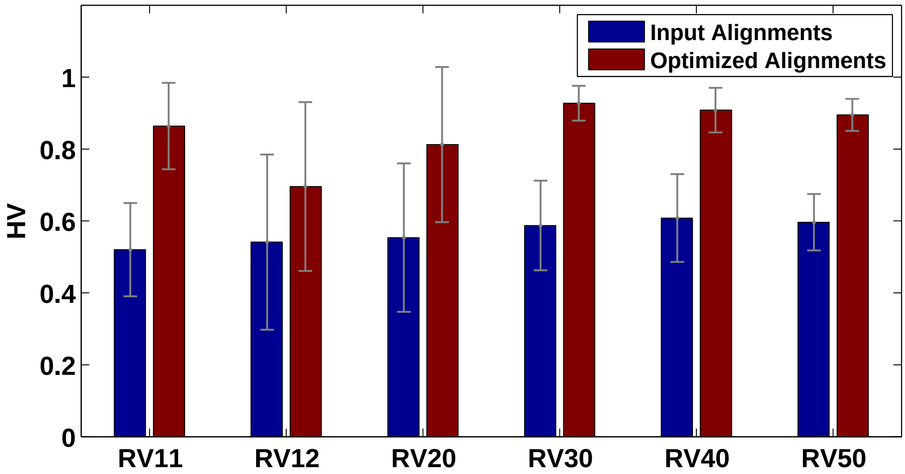


FIGURE 5.14: Average hypervolume values for each BAliBASE subset. HV values are shown for the input dataset and the optimized Pareto front of alignments.

generally improving the initial alignments from the initial methodologies. However, these studies are not considered enough for the assessment of the proposed approach. In addition to determining that alignments are improved, it is also necessary to statistically validate such improvements. The Mann-Whitney rank sum test [Mann et al., 1947] is then applied with this purpose. For each problem, this test can confirm whether the differences in HV values between two methods (in this case, the initial alignments and the optimized ones from 10 different runs) are significant. If the provided p-values are lower than the significance level (α), it can be rejected the null hypothesis of methods being identical. Consequently, the improvement is statistically confirmed. The significance level used to reject the null hypothesis in the 218 problems and to validate the improvement was set to $\alpha = 0.01$.

According to the proposed Mann-Whitney test, the complete dataset is considered significantly better, even those problems where improvements did not exceed 10%. Consequently, it can be confirmed that MO-SAStrE successfully optimized the 218 problems in the dataset in relation to the input methodologies, since the null hypothesis is rejected in all of them.

5.5.5 Comparison with other MSA methodologies

As previously introduced, MO-SAStrE is finally compared with four genetic algorithms applied to MSA: SAGA [Notredame and Higgins, 1996], MSA-GA [Gondro and Kinghorn, 2007], RBT-GA [Taheri and Zomaya, 2009] and DVGA [Naznin et al., 2011]. Other seven non-genetic methodologies, which are also included in comparisons of anterior genetic approaches, are also considered (see section 5.4.3 for detail). Since these algorithms are assessed with a subset of problems in BAliBASE 2.0. and MO-SAStrE applies BAliBASE 3.0, a subset of 20 problems common in both datasets is selected. The 3D-COFFEE algorithm [O'Sullivan et al., 2004] is also added in order to compare MO-SAStrE against another important aligner using structural information. These methodologies are compared to MO-SAStrE in two different studies, following the same experiments provided by Naznin et al. [2011].

5.5.5.1 MO-SAStrE vs. MSA-GA and VDGA

Firstly, MO-SAStrE is compared with a subset of methods that are included in both the MSA-GA and VDGA experiments. Both tools were assessed against ClustalW, which is also taken into account in this study. 3D-Coffee is additionally added to this comparison. To perform this study, the following considerations have been taken:

- MSA-GA defined two different configurations depending on whether the initial population is created with a prealign procedure or not.
- VDGA is configured according to the number of blocks in which each sequence is decomposed. Three decompositions are studied: 2 blocks (*Decomp_2*), 3 blocks (*Decomp_3*) and 4 blocks (*Decomp_4*).
- MSA-GA ran each problem five times instead of the ten considered in VDGA and MO-SAStrE. The best solution was then reported in all cases.

TABLE 5.4: Comparison with MSA-GA, VDGA, ClustalW and 3D-COFFEE. The BAliscore values are shown for 8 different BAliBASE sets. The two best scores are highlighted in bold.

Subset	**Dataset**	**MSA-GA**	**MSA-GA prealign**	**ClustalW**	**VDGA Decomp_2**	**VDGA Decomp_3**	**VDGA Decomp_4**	**3D-Coffee**	**MO-SAStrE**
RV11	19th	0.501	0.687	0.592	0.443	0.482	0.451	**0.812**	**0.716**
	27th	0.443	0.405	0.392	0.416	0.459	**0.464**	**0.530**	0.403
	31st	0.212	0.302	0.296	0.347	0.359	0.282	**0.675**	**0.544**
	38th	0.295	0.488	0.479	0.531	0.545	0.548	**0.783**	**0.808**
RV20	21st	0.755	0.758	0.757	0.857	0.863	0.853	**0.911**	**0.913**
	35th	0.761	0.768	0.766	0.847	**0.850**	0.839	0.823	**0.879**
RV30	30th	0.580	0.619	0.619	0.870	0.890	0.887	**0.909**	**0.918**
RV40	49th	0.710	0.635	0.630	0.330	0.542	0.478	**0.863**	**0.865**

- Both MO-SAStrE and 3D-COFFEE were run with all structures available in the PDB database for each specific set of sequences.

The proposed solutions are all evaluated with the BAliscore tool. Although a total of 26 problems were performed by MSA-GA and VDGA, a subset of them (eight problems) also included in BAliBASE v3.0 is taken to compare with MO-SAStrE.

Consequently, Table 5.4 shows the BAliscore results obtained from these methodologies (the two best BAliscore values are marked in bold). From these eight problems, MO-SAStrE outperforms MSA-GA and VDGA in a total of seven problems, while it obtains more accurate alignments than 3D-COFFEE in five out of the eight problems. From the beaten problems, MSA-GA (both with and without prealign process) and the three decomposition of VDGA are better in just one case, namely the *27th* dataset in the RV11 subset. However, although the most accurate alignment is not achieved in this problem, MO-SAStrE shows a quite close BAliscore value to MSA-GA and VDGA. On the other hand, 3D-COFFEE beats MO-SAStrE in three RV11 problems: *19th*, *31st* and, again, *27th*. In this case, it can be observed that MO-SAStrE alignments are more distant to 3D-COFFEE than those where MO-SAStrE wins 3D-COFFEE.

5.5.5.2 MO-SAStrE vs. SAGA, RBT-GA and VDGA

In the second comparison, MO-SAStrE is evaluated with SAGA, RBT-GA and, again, VDGA and 3D-COFFEE. This analysis also includes other strategies used in the VDGA and RBT-GA assessment processes, namely PRRP, ClustalW, Dialign, PIMA, HMMT and PILEUP. Following, some important considerations about these methodologies are clarified:

- Similarly to MO-SAStrE, both RBT-GA and VDGA applied 10 independent runs for each problem and the best alignment was taken for their comparison.
- PIMA includes two different ways to calculate the typical guide in progressive aligments: *maximum linkage* (ML) and *sequential branching* (SB).
- 3D-Coffee and MO-SAStrE consider the same structural information from PDB (all available structures).

The subset proposed by Taheri and Zomaya [2009] and Naznin et al. [2011] to assess their tools contained 34 problems from BAliBASE 2.0. Twelve of these problems are considered here, since they must again be shared in the both versions of BAliBASE.

The obtained BAliscore values are then presented in Table 5.5. From this table, it is observed that MO-SAStrE achieves one of the two best results in ten out of twelve problems. VDGA outperforms MO-SAStrE in two problems, namely *34^{t}h* in RV20 is beaten by the *Decomp_2* version and *28^{t}h* in RV30 by both *Decomp_3* and *Decomp_4*. On the other hand, 3D-COFFEE achieves better alignments in four problems. The problems where MO-SAStrE achieves worse alignments are closer to the best accuracy than those ones proposed by other methodologies, excepting 3D-COFFEE whose alignments are quite similar.

TABLE 5.5: BAliscore comparison with SAGA, RBT-GA, VDGA and other known methodologies. The BAliscore values are shown for 12 BAliBASE datasets. The two best scores are highlighted in bold.

Subset	**Dataset**	**PRRP**	**CLUSTALW**	**SAGA**	**DIALIGN**	**HMMT**	**PIMA** *(SB)*	**PIMA** *(ML)*	**MULTALIGN**	**PILEUP**	**RBT-GA**	**VDGA** *Decomp_2*	**VDGA** *Decomp_3*	**VDGA** *Decomp_4*	**3DCOFFEE**	**MO-SAStrE**
RV20	19^{th}	0.772	0.746	0.726	0.783	0.539	0.620	0.688	0.614	0.678	0.567	0.803	0.819	0.816	**0.827**	**0.825**
	21^{st}	0.711	0.761	0.623	0.576	0.530	0.393	0.386	0.566	0.702	0.660	0.857	0.863	0.853	**0.911**	**0.913**
	30^{th}	0.056	0.482	0.492	0.000	0.053	0.129	0.129	0.000	0.000	0.795	0.732	0.778	0.794	**0.901**	**0.911**
	32^{nd}	0.760	0.557	0.694	0.724	0.641	0.469	0.463	0.500	0.476	0.825	0.875	0.815	0.774	**0.928**	**0.917**
	34^{th}	0.404	0.484	0.498	0.262	0.423	0.390	0.561	0.593	0.278	0.745	**0.856**	0.829	0.742	**0.888**	0.855
	35^{th}	0.767	0.752	0.763	0.612	0.647	0.730	0.695	0.765	0.766	0.730	0.847	**0.850**	0.839	0.823	**0.879**
	37^{th}	0.363	0.192	0.282	0.350	0.141	0.183	0.211	0.192	0.159	0.755	0.717	0.751	0.781	**0.861**	**0.864**
	39^{th}	0.668	0.375	0.739	0.122	0.213	0.096	0.092	0.384	0.224	0.812	0.890	0.889	0.899	**0.920**	**0.912**
RV30	3^{rd}	0.128	0.163	0.186	0.000	0.006	0.000	0.000	0.000	0.110	0.180	0.383	0.453	0.408	**0.572**	**0.586**
	20^{th}	0.415	0.146	**0.585**	0.000	0.366	0.000	0.000	0.000	0.268	0.310	0.398	0.414	0.41	0.525	**0.590**
	21^{st}	0.139	0.130	0.269	0.139	0.037	0.083	0.148	0.241	0.083	0.350	0.469	0.481	0.526	**0.625**	**0.673**
	28^{th}	0.736	0.547	0.672	0.050	0.050	0.393	0.438	0.652	0.498	0.680	0.836	**0.866**	**0.866**	0.853	0.862

5.5.5.3 Statistical evaluation for both comparisons

Both comparisons (Tables 5.4 and 5.5) are joined together in order to estimate the significance of the MO-SAStrE performance. Specifically, the BAliscore results are compared with the Wilcoxon non-parametric statistical test. Briefly, the Wilcoxon test allow us to determine if there are significant differences between each two compared methodologies according to the results in independent problems. This test has been previously described in Chapter 2, section 2.4. The significance level for this statistical test is set to $\alpha = 0.05$.

The Wilcoxon test results for the comparison of MO-SAStrE against each other tool can be consulted in Table 5.6. According to the obtained p-values, MO-SAStrE shows significant improvements over the remaining methods, excepting 3D-COFFEE. These differences are even more meaningful when comparing with ClustalW and VDGA, since a larger number of problems (20 alignments) are included. It could then be stated that MO-SAStrE is actually improving former genetic optimizers.

Regarding the comparison with 3D-COFFEE, both 3D-COFFEE and MO-SAStrE provide similar alignments, since none statistically outperforms the other (Wilcoxon p-value>0.05). However, the *Z-score* value observed between the two methods suggests that MO-SAStrE is slightly better than 3D-Coffee according to the performed comparison (the negative *Z-score* indicates a slight improvement from MO-SAStrE).

Nevertheless, MO-SAStrE introduces some additional advantages with respect to 3D-COFFEE. Firstly, MO-SAStrE is able to work with only one structure thanks to the STRIKE score and the contact concept, whereas 3D-COFFEE requires at least two structures to build a pairwise structure superposition. Also, the alignment accuracy from 3D-COFFEE directly depends on the number of available structures, making it less useful when just a few structures are available. The influence of the number of structures over MO-SAStrE is less relevant. Additionally, MO-SAStrE provides more flexibility due to the fact that it

TABLE 5.6: Wilcoxon non-parametric test. Pairwise comparisons between MO-SAStrE and each other method. 'Sign+'/'Sign-' identifies the number of problems that MO-SAStrE respectively wins/loses other method . The 'Z-score' represents the distance between the two compared methods according to the obtained BAliscore (see Equation 2.7 in Chapter 2).

MSA tool	**Sign+**	**Sign-**	**Z-score**	**Pvalue**	**P<0.05**
MSA-GA	7	1	-2.381	0.017	yes
MSA-GA prealign	7	1	-2.381	0.017	yes
PRRP	12	0	-3.059	0.002	yes
SAGA	12	0	-3.059	0.002	yes
DIALIGN	12	0	-3.059	0.002	yes
HMMT	12	0	-3.059	0.002	yes
SB_PIMA	12	0	-3.059	0.002	yes
ML_PIMA	12	0	-3.059	0.002	yes
MULTALIGN	12	0	-3.059	0.002	yes
PILEUP8	12	0	-3.059	0.002	yes
RBT-GA	12	0	-3.059	0.002	yes
CLUSTALW	20	0	-3.920	0.000	yes
VDGA_Decomp2	18	2	-3.809	0.000	yes
VDGA_Decomp3	18	2	-3.510	0.000	yes
VDGA_Decomp4	18	2	-3.547	0.000	yes
3D-COFFEE	13	7	-0.579	0.562	no

includes other optimization criteria (multiobjective approach) in order to evaluate and improve the alignments in addition to structural information. In case no tertiary structures are available for sequences, 3D-Coffee should return the alignment with the standard consistency-based method T-Coffee [Notredame et al., 2000].

5.6 Conclusions and final statements

A novel algorithm called MO-SAStrE has been proposed to optimize multiple sequence alignments. This optimizer has been developed through the multiobjective genetic algorithm NSGA-II, specially based on a structure evaluation (STRIKE score) and other two complementary scores (totally conserved columns and percentage of non-gaps). This algorithm takes advantage of a wider range of evaluation measures than other similar methodologies and it provides a more sophisticated fitness function. For this algorithm, alignments previously obtained from eight methodologies (mainly progressive and consistency tools) were encoded using a novel representation and own-designed crossover and mutation procedures. The obtained alignments were built as an ensemble of the best aligned blocks from these solutions in order to adjust the sequences as precisely as possible. A complete dataset of problems from BAliBASE v3.0 was then applied. The results for this optimizer showed that the alignments could generally be improved similarly to other time-consuming aligners like 3D-Coffee or Promals. The hypervolume indicator and the Mann-Whitney test confirmed that MO-SAStrE significantly optimized the alignments in this dataset with regard to the input methodologies. Additionally, comparisons with other genetic and non-genetic approaches showed that MO-SAStrE can provide more accurate alignments according to the BAliscore measure (accuracy indicator).

We acknowledge that one of the main drawbacks of the proposed approach could be the limited availability of PDB structures. However, the pivotal relevance of structures to accomplish well-annotated sequences is beyond dispute. Thus, PDB is currently making a major effort to accurately annotate proteins' structures, which has been translated into an exponential increase in the last 10 years (99928 structures in April 2014). It is also known that current databases have admitted this relevance and they are currently being updated in order to include as many PDB structures as possible. For instance, the Pfam database [Finn et al., 2014] considers that the use of structural information will help to improve domain definitions and to increase coverage of sequences included in other databases. Thus, the Pfam database (release 27.0) already includes some structural annotation in more than 50% of its families, which represents >95% of the known PDB structures.

Additionally, it is important to remark that the main application of MSA tools is to infer several biological feature of unknown sequences by comparing them with those well-annotated. Then, at least one well-annotated sequence should ideally be added in each multiple sequence alignments, including at least some structural information. Anyway, in order to make the proposed approach more robust, MO-SAStrE implements an alternative objective for those cases where sequences lack any PDB structures. In those cases, the STRIKE objective is substituted by an easier evaluation such as the PAM250 score [Dayhoff et al., 1979]. Although this alternative is not the main goal here, it has been also checked that the multiobjective optimization using the PAM250 score could still be acceptably effective [Ortuño et al., 2012].

Chapter 6

Conclusions and Future Work

6.1 Conclusions

This section summarizes the general and most relevant conclusions of this dissertation, in addition to the specific conclusions of each contribution that have been previously stated at the end of their corresponding chapter.

6.1.1 Conclusions regarding multiple sequence alignments

This dissertation has been mainly focused on the study, analysis and optimization of multiple sequence alignments (MSAs). Consequently, some of the principal MSA methodologies and strategies existing in the literature have been introduced (Chapter 2) to understand the current state in this regard as well as its main problems and challenges.

A detailed comparison of ten different methodologies, which were considered throughout this dissertation, was performed to determine the advantages and drawbacks of each strategy. This comparison was applied according to the 218 sets of sequences from the BAliBASE benchmark and the differences between methodologies were validated by using the Wilcoxon non-parametric statistical test. Thus, some important conclusions were revealed with this study:

- The results of the comparison demonstrated that the accuracies in the alignments of these approaches are far from the optimal ones, even for the more recent and sophisticated tools like 3D-Coffee and Promals. It may then be concluded that MSAs have a wide range of improvement in terms of their accuracy that must still be exploited. Therefore, it can be said that the MSA methodologies are still an open issue in bioinformatics.
- In association with the previous conclusion, the computational time was another measure to take into account in MSAs according to this comparison. Although classical methods (ClustalW, Muscle, T-Coffee, etc) provided an acceptable time, it was likely to expect that more complex algorithms (3D-Coffee or Promals) would require more computational costs in

order to optimize the alignment accuracy. However, it was demonstrated that only in a few cases in which less related sequences were aligned, this improvement was truly significant. Therefore, we may conclude that the effectiveness of each method is strongly related to the sequences to be aligned. Then, it is not worthwhile to run these kind of complex methods when sequences are highly related since similar results are obtained with faster strategies.

- It was also noted that biologist and researchers do not either agree about a general accepted solution. There was not then a clear consensus about how it is more adequate to evaluate alignments. Thus, although the compared methodologies were mainly applying classical scoring scheme like BLOSUM, GONNET or PAM, it was observed that more recent tools tend to incorporate supplementary sources in addition to the aligned sequences like homologies or secondary and tertiary structure. Thus, it can be suggested that a more accurate scoring scheme including more heterogeneous information could clearly benefit future alignment tools.

These conclusions motivated the subsequent algorithms and methodologies that have been presented in this dissertation with the main objective of improving some of the challenges that have been previously concluded.

6.1.2 Conclusions about PAcAlCI

The Chapter 3 was devoted to the problem of predicting the most adequate methodology for the alignment of a specific set of sequences. As previously demonstrated, the accuracy that each alignment tool can provide directly depends on the sequences to be aligned. Therefore, a novel tool called ***P**rediction of **Ac**curacy **Al**ignment based on **C**omputational **I**ntelligence* (PAcAlCI) was proposed. This tool was based on the integration of heterogeneous biological features and the implementation of a regression model to estimate the possible accuracy of each tool before the alignment was performed. A dataset of 24 features associated with the proteins coded by the sequences and including, for instance, domains, molecular annotation, secondary or tertiary structures, was proposed. The features were extracted from important and highly consulted databases, namely Uniprot, Pfam, PDB and Gene Ontology as well as other features from specialized literature. The subsequent regression model was designed by means of the LS-SVM approach.

To prove the usefulness of PAcAlCI, the sets of sequences from the BAliBASE benchmark were aligned with 10 standard alignment tools. This dataset of sequences was applied to extract the proposed features and to train the LS-SVM regression model. The methodology was finally assessed by performing a 10-fold cross-validation process. Thus, the results obtained from this validation may lead us to the following important conclusions:

- The results demonstrated that the integration of the proposed heterogeneous dataset of features together with the LS-SVM regression model was effective to acceptably determine the accuracy of each aligner for each specific set of sequences. The provided accuracies was assessed by comparing with the Baliscore value from BAliBASE.
- Associated with the previous conclusion, it was also proved that the prediction of the accuracy was considered useful to estimate the subset of methodologies that are likely to provide the best accurate alignments. Depending on the set of sequence to be aligned, several methods were then estimated to be equivalent in terms of their accuracy. Thus, although the computational time of the methodologies was not directly considered, PAcAlCI may contribute to considerably decrease this computational time since it suggests alternative methods which may provide similar accuracies in less time.
- It must be also emphasized that not all features in the dataset were completely necessary. In fact, a subset of the 10 most relevant features was proved to be the optimal one to reach the minimal error in the accuracy prediction. From these features, three properties or groups of properties were highlighted: *(i)* number of domains found in sequences; *(ii)* features

directly related to the number and the length of the sequences; and *(iii)* features associated with the chemical types of the amino acids (acid, polar and basic ones). From the discarded features, it is notably remarkable the removal of important features associated, for instance, with tertiary structures or some GO ontologies.

- The prediction of adequate tools to align sequences showed results equivalent to a similar system called AlexSys. However, several advantages were incorporated in PAcAlCI that were not considered before: *(i)* PAcAlCI allows us to decide the most promising tool from a wider and more heterogeneous set of aligners; *(ii)* PAcAlCI provides a subset of tools that were determined to be almost equivalent in terms of their accuracy instead of only one tool; and *(iii)* PAcAlCI gives an estimation of the accuracy that can be obtained by each selected tool.

PAcAlCI was then proved to be a helpful tool for users that seek an adequate methodology to align their sequences. PAcAlCI may be considered as a guide for beginner users who need an efficient aligner but do not know the possible methodologies as well as for experts searching a powerful tool to quickly determine the most adequate aligner.

6.1.3 Conclusions about intelligent scores for the estimation of alignment qualities

The Chapter 4 was focused on the development of several advanced scoring schemes for the evaluation of alignments. These intelligent scores were designed by several regression models, namely Gaussian Processes (GP), Regression trees, Bagging trees and LS-SVMs. Similarly to the previous PAcAlCI tool, these proposed evaluation models took advantage of the integration of a heterogeneous dataset of biological features. In this case, the features were retrieved related to a wide dataset of alignments. Other standard scores like PAM, BLOSUM, RBLOSUM, GONNET or STRIKE were also integrated in our dataset of features to accurately evaluate the alignments. The BAliscore values were again taken into account as an accurate tool to train the four proposed regression models.

The ability of the proposed intelligent scores with integrated features to evaluate the quality of diverse alignments was proved by comparing against other more standard scores. These comparisons were performed by using a dataset of alignments provided by the BAliBASE and OxBench benchmarks and ten different MSA tools. Some important conclusions are considered from the findings presented in Chapter 4 about these advanced scores:

- The regression models are interesting approaches to predict the quality of alignments by integrating several features and learning from an accurate quality as BAliscore. Since BAliscore is able to provide alignment evaluations only for those sets of sequences with an accurate reference alignment (e.g. sequences from benchmarks like OxBench or BAliBASE), the regression models allowed us to learn of this tool and provide similar quality estimation for other possible alignments without available references.
- According to the feature selection procedure, it was demonstrated that a subset of relevant features could be enough to obtain acceptable quality predictions, without excessively increasing the complexity of the proposed scores. More specifically, it was shown that the alignment of similar tertiary structure (contacts) and similar secondary structures in the sequences were directly related to the quality of the alignments. Additionally, some of the included scores like STRIKE or GONNET as well as other important measurements directly associated with the alignments (percentage of identities, gaps or totally conserved columns) were also strictly necessary to obtain good performances in the four regression models. Therefore, the 7 most relevant features were taken into account by the LS-SVM proposal whereas the Gaussian processes and regression/bagging trees needed a wider number of them (10 features for bagging trees and 9 features for both Gaussian processes and regression trees).

- All the proposed regression models incorporating their corresponding subset of features outperformed the initial standard scores when running the test set. According to the correlation results, the advanced score based on bagging trees showed the most correlated qualities with the real values of BAliscore. Anyway, it can be concluded that the four proposals achieved high correlations values and all of them could be applied to obtain an accurate scoring scheme.

- In addition to the previous conclusion, the four proposed scores were also assessed in term of their ability to determine the best of two alignments (pairwise alignments comparisons). According to the results in this comparison, the four intelligent scores were generally able to adequately determine the best alignment (> 80% of accuracy), thus improving the results of other standard scores. Again, the best performance among the four proposal was the bagging tree model. However, it was proved that this comparison did not have to be related to the previously shown correlation. For instance, the regression trees showed better results in this comparison than Gaussian processes although its correlation with Baliscore was clearly worse.

- Taking into account both the correlation results and the comparison of pairwise alignments, it may be concluded that the four proposed scoring schemes are capable of adequately evaluating alignments. Therefore, it can be remarked that these approaches may be useful to be applied in other MSA strategies to optimize or build new alignments. These advanced scores could then help to obtain more biologically meaningful alignments since they are evaluated considering a wide range of biological properties.

6.1.4 Conclusions about MO-SAStrE

This part of the dissertation was devoted to the problem of improving the accuracy of MSAs by incorporating additional information. In this case, a multiobjective genetic algorithm called *Multiobjective* ***O****ptimizer for* ***S****equence* ***A****lignments based on* ***Str****uctural* ***E****valuation* (MO-SAStrE) was proposed to optimize alignments previously performed by other eight tools. More specifically, the multi-objective NSGA-II approach was performed based on three different fitness functions: STRIKE score, percentage of non-gaps and percentage of totally conserved columns. The STRIKE score, which is considered the main objective of the optimizer, evaluated alignments by incorporating information about the tertiary structure of the sequences. Thus, a more distant evolutionary relationship may be determined among sequences providing a more accurate alignment. Additionally, a novel pseudo-codification was implemented to allow the design of efficient and successful mutation and crossover operators.

The usefulness of MO-SAStrE was demonstrated by comparing with the input aligners which were included to be optimized. Moreover, an additional comparison with other optimizers based on genetic algorithms as well as other tool including structural information (3D-Coffee) was also performed. According to the results from these comparisons, some important conclusions can be drawn:

- The multi-objective approach was an appropriate methodology for the optimization of the alignments, mainly because although there is a great amount of scoring schemes to evaluate alignments, they do not often agree about their accuracies. Therefore, the trade-off between the three different scores proposed in MO-SAStrE became useful to jointly optimize the accuracy of alignments.
- The proposed codification gave an adequate representation of the alignment based on positions of the amino acids which was essential for the subsequent design of efficient operators. This representation was defined as a pseudo-codification because the original alignment was still necessary in some stage of the optimization like the fitness evaluation. Anyway, the alternate application of both this pseudo-codification and the original alignment did not have any appreciable influence in the accuracy or computational cost of the optimization process.

- The genetic algorithm showed its effectiveness to optimize the input alignment over other tools by incorporating an evaluation related to structural information. A significant optimization was appreciated for the whole dataset of sequences in BAliBASE. These improvements were obtained due to the implementation of the relevant mutation and crossover operators. Specifically, the crossover achieved the assembly of the most accurate regions of each initial alignment by progressively crossing them in a new alignment. In addition to that, the mutation operator redistributed the gaps trying to find alternative alignments that were not considered until then.

- According to the observed results, it is important to emphasize that the efficiency of MO-SAStrE was even more significant when aligning more distant sequences. This is a key conclusion due to the fact that these alignments are indeed the most difficult ones and, as previously stated, current tools could not obtain accurate enough alignments.

- The comparisons with the three genetic approaches MSA-GA, RBT-GA and VDGA as well as other classical non-genetic tools, confirmed that MO-SAStrE significantly outperformed these other tools, providing more accurate alignments according to the BAliscore measure (accuracy indicator). This improvement was statistically assessed by comparing the obtained result with the Wilcoxon non-parametric test.

- The MO-SAStrE optimizer obtained slightly better improvements (though not statistically significant) than other tool using similar information, namely 3D-Coffee. Moreover, it was proved that MO-SAStrE may work better than 3D-Coffee when only a few sequences in the alignment have available tertiary structures. This is due to the fact that the STRIKE evaluation works independently with any structure whereas 3D-Coffee needs at least two known or predicted structures to perform their superposition.

- Although the consuming time was not taken into account in this work, it can be notated that both MO-SAStrE and 3D-Coffee showed similar computational costs. Although these costs may be excessive in some cases, the genetic nature of MO-SAStrE will allow us to run this approach in parallel architectures like computer clusters in the future, thus reducing this time significantly.

- Finally, it must be also clarified that MO-SAStrE was still effective even when there were not known structures for the sequences. The multi-objective strategy gave a certain robustness to the procedure in case any of the objective was unavailable. Anyway, MO-SAStrE also applied for these cases an additional evaluation based on the PAM250 score in order to substitute the missing evaluation. Thus, it was confirmed that the MO-SAStrE optimization using this other evaluation was still acceptably accurate although the improvement may be reduced.

Thus, we may conclude that MO-SAStrE demonstrated a successful performance in order to optimize MSAs. Moreover, the usage of this optimizer was especially recommendable for alignments with evolutionary distant sequences, which are harder to align and where the structural data could be essential for the alignment optimization.

6.2 Future Work

Some other proposals are being currently developed with relation to the algorithms and systems presented in this dissertation. The main objective of these future works is to complete, to adapt and to improve the presented algorithms in order to fulfill the most recent challenges and problems in MSAs and other bioinformatic fields. In this regard, some of these issues that have been planned are introduced below:

- The PAcAlCI tool is being updated in order to extend the possible alignment tools including more recent methodologies, for instance, ClustalΩ or Promals3D. Together with these important tools, the main idea is to integrate in PAcAlCI additional optimizers like the MO-SAStrE presented in this dissertation. Thus, PAcAlCI could also determine when it may be worthwhile in terms of accuracy to run an time-consuming optimizer.

- A novel implementation for the multi-objective genetic algorithm MO-SAStrE is being developed. This novel design has four principal aims. Firstly, the codification process is being redesigned trying to avoid using the original alignment representation during all the optimization process. The main idea here is to provide a more compact representation, easy to handle for the operators and the fitness function. Secondly, the existing operators are being extended including other kind of crossover or mutation. The third objective is to optimize the computational cost of MO-SAStrE by running it in parallelization architectures like computer clusters. Finally, other possible evaluations are being considered for the multi-objective fitness function. The main purpose here is to integrate any of our own-design scoring schemes (Chapter 4) as the principal objective in the new genetic optimizer instead of or in addition to STRIKE.

- Taking advantage of the acquired knowledge in biological databases and feature management, a database and a software tool for the extraction of biological features related to protein sequences and alignments are being designed. It has been shown that these features have had a key role in the prediction algorithms presented here for MSAs, but they may be also helpful for users interested in other bioinformatics issues, for instance, prediction of PPIs or estimation of phylogenetic trees. These databases and software tools mainly seek to standardize the heterogeneous set of features presented in this dissertation for any set of sequences or proteins. Thus, the main objective is to calculate and retrieve this set of features when a query of sequences or proteins is carried out. Thus, several repositories will be consulted creating a new group of features which interrelate different aspects of the queried proteins or sequences. To the best of our knowledge, there are no databases today with this specific purpose where several features are integrated and can be retrieved to relate several protein, protein sequences or even protein alignments.

Conclusiones y Trabajo Futuro

Conclusiones

Esta sección recapitula las conclusiones generales y más relevantes de esta tesis doctoral, además de aquellas conclusiones más específicas que han sido enunciadas previamente al término de cada capítulo correspondiente.

Conclusiones respecto los alineamiento múltiple de secuencias

Esta tesis ha estado principalmente orientada al estudio, análisis y optimización de alineamientos múltiples de secuencia (MSAs). En este sentido, se han explicado algunas de las principales metodologías y estrategias para el alineamiento múltiple de secuencia que existen en la literatura (Capítulo 2) con el objetivo de facilitar la comprensión del estado actual de dicha temática así como los principales retos y problemas que se deben de afrontar.

A lo largo de este capítulo, se han seleccionado diez metodologías diferentes de las cuales se han determinado sus principales ventajas e inconvenientes. Esta comparación se ha llevado a cabo aplicando cada metodología en el alineamiento de 218 conjuntos de secuencias extraídas de BaliBase y las diferencias entre dichas metodologías se analizaron y validaron utilizando el test estadístico no paramétrico de Wilcoxon. De este estudio, en consecuencia, se obtuvieron las siguientes conclusiones:

- Al comparar las diferentes herramientas de alineamiento múltiple de secuencia, nuestros resultados demostraron que la precisión en dichos alineamientos distaba notablemente de la óptima, incluso en los casos en los que se utilizaban las herramientas más sofisticadas y recientes como 3D-coffee y Promals. Por este motivo podemos concluir que las técnicas de alineamiento múltiple de secuencia son susceptibles de mejora en términos de precisión y que constituyen un campo abierto de estudio en Bioinformática.

- Otra medida que se ha tenido en cuenta en relación con las herramientas de alineamiento múltiple de secuencias es el tiempo de computación. Aunque los métodos clásicos tales como ClustalW, Muscle, T-Coffee, etc, requieren un tiempo de computación aceptable, es lógico pensar que aquellos algoritmos más complejos como 3D-Coffee o Promals requerirán unos costes de computación mayores para mejorar la precisión del alineamiento. Sin embargo, hemos demostrado que sólo en unos casos determinados, en aquellos en los que era necesario alinear secuencias poco relacionadas, esta mejora era realmente significativa. Por lo tanto, podríamos concluir que la efectividad de cada método depende de manera muy estrecha de las características de las secuencias que van a ser alineadas. Por tanto, no es necesario llevar a cabo el alineamiento de secuencias usando los métodos más complejos en aquellos casos en los que las secuencias están íntimamente relacionadas, pudiendo obtener resultados muy precisos y estrategias más rápidas.

- Podemos afirmar que en general no existe un consenso claro entre los investigadores pertenecientes a este campo en cuanto a la metodología más adecuada para evaluar los alineamientos. Así, aunque las metodologías que han sido motivo de comparación utilizan sistemas clásicos de evaluación tales como BLOSUM, GONNET o PAM, las herramientas más recientes tienden a incorporar recursos adicionales además de las propias secuencias como son las homologías o las estructuras secundaria y terciaria. Por lo tanto es lógico pensar que un sistema de evaluación que incluya información heterogénea podría beneficiar enormemente las herramientas de alineamiento.

Los motivos y desafíos expuestos anteriormente motivaron los algoritmos y la metodología que han sido expuestos a lo largo de esta disertación de manera que nuestro objetivo principal ha sido afrontar y mejorar dichos retos.

Conclusiones acerca de PAcAlCI

El Capítulo 3 ha estado centrado en el problema de predecir la metodología más adecuada para alinear un grupo específico de secuencias. Tal y como se ha demostrado previamente, la precisión con la que cada herramienta trabaja depende directamente de las secuencias que van a ser alineadas. Por lo tanto a lo largo de dicho capítulo se propone una novedosa herramienta de predicción de precisión denominada ***P**rediction of **Ac**curacy **Al**ignment based on **C**omputational **I**ntelligence* o PAcAlCI. Esta herramienta está basada en la integración de un conjunto de características biológicas heterogéneas y en la implementación de un modelo de regresión para la predicción de la precisión de cada herramienta de alineamiento, de manera previa a realizar dicho alineamiento. Hemos propuesto un conjunto de 24 características relacionadas con las proteínas que son codificadas por las secuencias a alinear, entre las que se encuentran, por ejemplo, dominios, anotación molecular o estructuras secundaria y terciaria. Las características se han extraído por un lado de las más importantes y frecuentemente consultadas bases de datos denominadas Uniprot, Pfam, PDB y Gene Ontology y por otro de literatura especializada. El modelo de regresión planteado se diseñó utilizando un modelo LS-SVM.

Para probar la utilidad de PAcAlCI, el conjunto de secuencias extraídas de BAliBASE fueron alineadas utilizando 10 metodologías de alineamiento estándares. Este conjunto de secuencias se empleó para la extracción de las características propuestas y para entrenar el modelo de regresión de LS-SVM. Finalmente, esta metodología fue estadísticamente validada empleando el método de validación cruzada de 10 iteraciones (*10-fold cross-validation*). De esta forma, los resultados obtenidos tras dicha validación nos permitieron determinar las siguientes importantes conclusiones:

- Los resultados obtenidos demostraron que la integración del conjunto de características heterogéneo propuesto junto con el modelo de regresión del LS-SVM fue eficiente, determinando de manera adecuada la precisión de cada alineador para un conjunto específico de secuencias. Dicha precisión se ratificó al comparar con el valor óptimo proporcionado por Baliscore.

- En relación con la conclusión anterior, también se determinó que la predicción de la precisión es útil para estimar el conjunto de metodologías que proporcionan los mejores alineamientos en cuanto a su precisión. Dependiendo del conjunto de secuencias que van a ser alineadas, se ha estimado que, en términos de precisión, varios métodos pueden ser equivalentes. De esta manera, aunque el tiempo de computación no es una variable considerada directamente en este estudio, PAcAlCI podría contribuir a reducir el tiempo de computación dado que esta herramienta propone los métodos alternativos que ofrecerían resultados similares en cuanto a precisión pero con menor coste computacional.

- Cabe destacar que no todas las características del conjunto son realmente necesarias. De hecho, se ha determinado que el subconjunto con las 10 características más relevantes es el óptimo para lograr el mínimo error en la predicción de la precisión. De estas características, se destacan tres propiedades o conjuntos de propiedades: el número de dominios hallados en las secuencias; características directamente relacionadas con el número y la longitud de las secuencias; y las características asociadas con las propiedades químicas de los aminoácidos (ácidos, polares o básicos). De las características no seleccionadas, es interesante destacar la desestimación de características importantes asociadas por ejemplo con la estructura terciaria o algunas ontologías GO.

- La capacidad de PAcAlCI para predecir la herramienta más adecuada para el alineamiento de una determinada secuencia fue comparada con un sistema similar denominado AlexSys, demostrando resultados muy similares. Sin embargo PAcAlCI incorpora una serie de ventajas que no han sido consideradas previamente: *(i)* PAcAlCI permite considerar un conjunto de alineadores mucho mayor y más heterogéneo de entre los cuales se elige la herramienta/s más adecuada/s para el alineamiento; *(ii)* PAcAlCI determina no sólo una única herramienta sino un conjunto de ellas las cuales son consideradas prácticamente equivalentes en términos de precisión; y finalmente *(iii)* PAcAlCI propone una estimación de la precisión para cada una de las técnicas seleccionadas.

Por tanto, hemos demostrado que PAcAlCI es una herramienta útil para aquellos investigadores que buscan la metodología más adecuada para alinear sus conjuntos de secuencias. PAcAlCI podría representar una excelente guía tanto para aquellos usuarios inexpertos que necesitan emplear una herramienta de alineamiento eficaz pero que desconocen el amplio abanico de posibles metodologías existentes como para aquellos investigadores experimentados que buscan la herramienta más potente que les permita determinar de manera rápida y eficaz qué alineador deben emplear.

Conclusiones respecto a los métodos avanzados para la evaluación de alineamientos

El Capítulo 4 esta orientado al desarrollo de varios esquemas de puntuación avanzados para la evaluación de alineamientos. Estos métodos de evaluación inteligentes fueron desarrollados mediante la implementación de diferentes modelos de regresión, concretamente procesos Gaussianos (GP), árboles de regresión, conjuntos ensamblados de árboles (*bagging*) y LS-SVMs. De forma similar a la herramienta PAcAlCI, los modelos de evaluación propuestos aprovechan la integración de un conjunto heterogéneo de características biológicas. En este caso, las características utilizadas fueron extraídas en relación con un conjunto de alineamientos. Adicionalmente, otros métodos de evaluación más estándar como PAM, BLOSUM, RBLOSUM, GONNET o STRIKE se integraron en nuestro conjunto de características con el fin de obtener una evaluación más precisa. Los valores de calidad proporcionados por BAliscore fueron considerados para entrenar los modelos de regresión propuestos.

La capacidad de los métodos de evaluación con características integradas para la evaluación de la calidad en alineamientos se validó comparando con otros métodos más estándar. Estas comparaciones fueron realizadas usando los conjuntos de alineamientos proporcionados por BAliBASE y OxBench así como diez diferentes herramientas de alineamientos. Alguna conclusiones importante son a continuación consideradas a partir de los resultados obtenidos en el capítulo 4 sobre estos métodos de evaluación avanzados:

- Los modelos de regresión son estrategias interesantes para predecir de la calidad de los alineamientos mediante la integración de diversas características y el aprendizaje supervisado a partir de los valores proporcionados por la herramienta Baliscore. Dado que BAliscore es capaz de proporcionar evaluaciones solo para aquellas secuencias cuyos alineamientos de referencia son conocidos (por ejemplo, secuencias proporcionadas por OxBench o BAliBASE), los modelos de regresión propuestos nos permiten aprender de esta herramienta para posteriormente proporcionar estimaciones de la calidad para otros alineamientos cuya referencia no esté disponible.
- Respecto al proceso de selección de características, hemos demostrado que un subconjunto de características relevantes puede ser suficiente para obtener unas aceptables predicciones de calidad, sin necesidad de aumentar excesivamente la complejidad de los sistemas propuestos. Específicamente, se ha mostrado como los alineamientos de los contactos en estructuras terciarias o similares estructuras secundarias están directamente relacionados con la calidad del alineamiento final. Adicionalmente, algunos métodos de evaluación como STRIKE y GONNET así como otras importantes medidas directamente asociadas a los alineamientos (porcentajes de identidades, huecos o columnas totalmente conservadas) fueron también necesarios para obtener resultados eficientes en los cuatro modelos de regresión. Así, las 7 características más relevantes fueron consideradas por la propuesta basada en LS-SVMs mientras que los procesos Gaussianos y los árboles de regresión y ensamblados (*bagging*) necesitaron un número mayor de ellas (10 características para los árboles ensamblados y 9 características para los procesos Gaussianos y árboles de regresión).

- Todos los modelos de regresión propuestos con sus correspondientes subconjuntos de características mejoraron notablemente los métodos de evaluación estándar cuando fueron comparados utilizando el conjunto de test. De acuerdo con los resultados de correlación , el método e evaluación basado en árboles ensamblados (*bagging*) mostró los valores de calidad más correlacionados con los valores reales proporcionados por BAliscore. De cualquier forma, es posible concluir que las cuatro propuestas lograron valores de correlación igualmente aceptables y todas ellas pueden ser aplicadas para la obtención de valores de calidad precisos.

- En relación con la anterior conclusión, los cuatro métodos de evaluación propuestos se validaron también en términos de su capacidad para determinar el mejor alineamiento entre dos (comparación de alineamientos por pares). De acuerdo con los resultados de esta comparación, los cuatro métodos de evaluación inteligentes fueron generalmente capaces de determinar adecuadamente el mejor alineamiento (>80% de acierto), mejorando así los resultados de otros métodos estándar. Los resultados más precisos fueron de nuevo obtenidos por el model basado en árboles ensamblados (*bagging*). Sin embargo, se ha mostrado que los resultados de esta comparativa no tienen por qué estar relacionados con los mostrados previamente por la correlación con BAliscore. Así, el modelo basado en árboles de regresión mostró en esta comparativa resultados mejores que los procesos Gaussianos, aunque su correlación con Baliscore fue claramente inferior.

- Considerando los resultados de correlación y de la comparativa de alineamientos por pares, podemos concluir que los cuatro métodos de evaluación propuestos son capaces de evaluar alineamientos de forma adecuada. Por tanto, es importante destacar que estos modelos podrían resultar muy útiles para su aplicación en la optimización o construcción de nuevos alineamientos. Así, estos métodos de evaluación avanzados ayudan a obtener alineamientos con más significado biológico ya que serán evaluados considerando un amplio rango de propiedades biológicas.

Conclusiones acerca de MO-SAStrE

Esta parte de la tesis estuvo enfocada al problema de mejorar la precisión de los alineamientos múltiples de secuencia (MSAs) gracias a la incorporación de información adicional. En este caso, se propuso un algoritmo genético multiobjetivo denominado ***M**ultiobjective **O**ptimizer for **S**equence **A**lignments based on **Str**uctural **E**valuation* (MO-SAStrE) con el objetivo de optimizar los alineamientos previamente obtenidos usando otras ocho herramientas. De manera más concreta, la estrategia multiobjetivo NSGA-II se desarrolló basada en tres funciones fitness diferentes: el valor de STRIKE, el porcentaje de no-huecos y el porcentaje de columnas totalmente conservadas.

El valor de STRIKE, el cual se considera el objetivo principal del optimizador, evalúa los alineamientos incorporando información sobre la estructura terciaria de la secuencia. De esta manera podemos determinar relaciones filogenéticas o evolutivas más distantes entre las secuencias lo que nos permite generar alineamientos más precisos. Además, se implementó una novedosa pseudo-codificación para los alineamientos que permitía el diseño de operadores de mutación y de crossover eficientes.

La utilidad de MO-SAStrE se corroboró al comparar con los alineamientos de entrada, usados en el proceso de optimización. Adicionalmente se llevaron a cabo comparaciones con otros optimizadores basados en algoritmos genéticos así como con herramientas que incluían información estructural (3D-Coffee). De los resultados de dichas comparaciones se pueden destacar varias conclusiones importantes:

- La metodología multi-objetivo resultó apropiada para la optimización de los alineamientos debido principalmente a que, aunque se han propuesto un gran abanico de sistemas de evaluación de alineamientos, existe una discordancia entre ellos acerca de la determinación de la precisión del alineamiento. De este modo, los tres sistemas de evaluación que propone MO-SAStre alcanzan un compromiso para optimizar de manera conjunta la precisión de los alineamientos.

- La codificación propuesta produjo una representación adecuada del alineamiento basada en la posición de los aminoácidos, lo cual fue esencial para el posterior diseño de operadores eficiente. Esta representación fue definida como una pseudo-codificación dado que se requería el alineamiento original en alguns de las etapas de la optimización, por ejemplo en la evaluación del fitness. En cualquier caso, la aplicación alternante tanto de la pseudo-codificación como del alineamiento original no influyó de manera significativa en la precisión o en el coste de computación del proceso de optimización.

- El algoritmo genético propuesto demostró una capacidad de optimización de los alineamientos de entrada superior a cualquiera de las demás herramientas gracias a la incorporación de un sistema de evaluación multiobjetivo relacionado con la información estructural. Así, una optimización muy significativa fue determinada para todo el conjunto de secuencias de BAliBASE. Estas mejoras se debieron principalmente a la implementación de los operadores de mutación y de cruce. Específicamente, el operador de cruce consiguió el ensamblaje de las mejores regiones de cada alineamiento de partida mediante el cruce progresivo de determinadas regiones de dichos alineamientos. Además, el operador de mutación redistribuyó los huecos existentes en el alineamiento con el objetivo de encontrar alineamientos alternativos que no habían sido considerados hasta el momento.

- Dados los resultados obtenidos, es necesario destacar que la eficiencia de MO-SAStrE es incluso más significativa en aquellos alineamientos con secuencias más distantes. Esta conclusión es clave debido a que estos alineamientos son los más complejos de obtener y, tal y como se ha descrito a lo largo de esta disertación, las herramientas de las que se dispone en la actualidad no son capaces de obtener alineamientos con una precisión aceptable.

- La comparación con las tres estrategias genéticas MSA-GA, RBT-GA y VDGA así como con otras herramientas clásicas no genéticas nos permitió confirmar que MO-SAStre mejoraba de manera muy notable estas herramientas, proporcionando alineamientos más precisos de acuerdo a la medida de calidad BAliscore. La significación estadística de esta notable mejora se comprobó aplicando el test no paramétrico de Wilcoxon.

- El optimizador MO-SAStrE proporcionó resultados ligeramente mejores (aunque no estadísticamente significativos) que otras herramientas utilizando información similar como 3D-Coffee. Además, hemos demostrado que MO-SAStrE superaría a esta última herramienta en el caso en que sólo algunas de las secuencias que van a ser alineadas disponen de estructuras terciarias conocidas. Esto es debido a que el sistema de evaluación STRIKE es capaz de trabajar independientemente con cualquier estructura mientras que 3D-Coffee necesita al menos dos estructuras conocidas o predichas para llevar a cabo su superposición.

- Aunque el tiempo de computación no ha sido considerado en este trabajo, es de destacar que tanto MO-SAStrE como 3D-Coffee tienen unos costes de computación similares. Si bien estos costes de computación pueden ser considerados como excesivos en algunos casos, la naturaleza genética de MO-SAStrE permite el uso de esta herramienta en arquitecturas paralelas tales como clusters y supercomputadores lo que nos permitiría en un futuro, reducir estos tiempos de manera significativa.

- Finalmente, es importante señalar que MO-SAStrE trabajó de manera efectiva incluso en aquellas circunstancias en las que se desconocía la estructura de las secuencias. La estrategia multi-objetivo proporcionó solidez al proceso en los casos en los que no existía disponibilidad de algunos de los objetivos. En estos casos, MO-SAStrE aplica una evaluación adicional basada en el sistema de evaluación de PAM250, el cual sustituye al evaluador que no esté disponible. De esta manera también hemos confirmado que la optimización de MO-SAStrE usando este último evaluador alcanzaba unos niveles de precisión significativos aunque la mejora podría verse reducida.

De esta manera podemos concluir que MO-SAStrE es capaz de optimizar los alineamientos múltiples de secuencia de forma exitosa. Además, es especialmente recomendable el uso de este optimizador en los casos de alineamientos con secuencias evolutivamente distantes, las cuales son muy complejas de alinear y en las que la información estructural es esencial para la optimización del alineamiento.

Trabajo Futuro

En la actualidad se están desarrollando nuevas propuestas relacionadas con los algoritmos y sistemas presentados a lo largo de esta tesis doctoral. El principal objetivo de estos trabajos futuros es completar, adaptar y mejorar estos algoritmos para hacer frente a retos más actuales y problemas todavía sin resolver relacionados con el alineamiento múltiple de secuencias y con otros campos de la Bioinformática. A este respecto se resumen a continuación algunos de los objetivos que se han planificado para un trabajo futuro:

- La herramienta PAcAlCI está siendo actualizada con el objetivo de incorporar metodologías más recientes como por ejemplo ClustalΩ o Promals3D. Junto con estas herramientas, la idea principal sería integrar en PAcAlCI el optimizador MO-SAStrE, desarrollado en esta tesis doctoral. De esta manera la herramienta PAcAlCI podría además determinar cuándo sería adecuado en término de precisión, usar un optimizador que requiriera un tiempo de computación mayor como MO-SAStrE.
- Una nueva implementación del algoritmo multiobjetivo genético MO-SAStrE esta actualmente en desarrollo. Este nuevo diseño consta de cuatro objetivos principales. En primer lugar, el proceso de codificación está siendo rediseñado para intentar evitar recurrir al alineamiento original durante el proceso de optimización. La principal idea es proporcionar una representación más compacta, fácil de manejar por los operadores y la función

de fitness. En segundo lugar, los operadores existentes se están extendiendo para incluir otros tipos de operadores de cruce y de mutación. El tercer objetivo será optimizar el tiempo de computación de MO-SAStrE gracias al empleo de arquitecturas paralelas. Finalmente se están considerando otros sistemas de evaluación para la función fitness multiobjetivo. Así, se pretende integrar alguno de los sistemas de evaluación diseñados en nuestro grupo (Capítulo 4) como el principal objetivo del genético multiobjectivo, en lugar de STRIKE o además de él.

- Aprovechando el conocimiento adquirido sobre bases de datos biológicas y manejo de características, se quiere diseñar una nueva base de datos y una herramienta que permita la extracción de las características biológicas relacionadas con las secuencias proteicas y los alineamientos. Se ha visto que estas características representan un papel esencial en los algoritmos de predicción que se han presentado en esta tesis para el alineamiento múltiple de secuencias pero además podrían ser de enorme ayuda para usuarios interesados en otros campos de la Bioinformática como por ejemplo la predicción de interacciones proteína-proteína o la estimación de árboles filogenéticos. Esta base de datos y herramientas adicionales pretenden estandarizar el conjunto heterogéneo de características presentado a lo largo de esta tesis para su aplicación a cualquier conjunto de secuencias o proteínas. Así, el principal objetivo es calcular y extraer este conjunto de características cuando se lleva a cabo una determinada consulta. De esta manera, a través de la consulta de numerosos repositorios se creará un nuevo grupo de características que interrelacionarán diferentes aspectos de las proteínas o secuencias que se desea consultar. Hasta donde nuestro conocimiento alcanza, en la actualidad no existen bases de datos que cumplan este propósito especifico y en las que se integren y se puedan recuperar tal diversidad de características relacionadas con proteínas, secuencias de proteínas o alineamientos.

Bibliography

Ahola, V., Aittokallio, T., Vihinen, M., and Uusipaikka, E. (2008). Model-based prediction of sequence alignment quality. *Bioinformatics*, 24(19):2165–2171.

Akaike, H. (1974). A new look at the statistical model identification. *Automatic Control, IEEE Transactions on*, 19(6):716–723.

Allahviranloo, M. and Recker, W. (2013). Daily activity pattern recognition by using support vector machines with multiple classes. *Transportation Research Part B: Methodological*, 58:16–43.

Altschul, S. F., Boguski, M. S., Gish, W., Wootton, J. C., et al. (1994). Issues in searching molecular sequence databases. *Nature Genetics*, 6(2):119–129.

Altschul, S. F., Gish, W., Miller, W., Myers, E. W., and Lipman, D. J. (1990). Basic local alignment search tool. *Journal of Molecular Biology*, 215(3):403–410.

Altschul, S. F., Madden, T. L., Schäffer, A. A., Zhang, J., Zhang, Z., Miller, W., and Lipman, D. J. (1997). Gapped blast and psi-blast: a new generation of protein database search programs. *Nucleic Acids Research*, 25(17):3389–3402.

Anderson, C., Strope, C., and Moriyama, E. (2011). Suitemsa: visual tools for multiple sequence alignment comparison and molecular sequence simulation. *BMC Bioinformatics*, 12:184.

Aniba, M., Poch, O., Marchler-Bauer, A., and Thompson, J. (2010). Alexsys: a knowledge-based expert system for multiple sequence alignment construction and analysis. *Nucleic Acids Research*, 38:6338–6349.

Anisimova, M., Cannarozzi, G., and Liberles, D. A. (2010). Finding the balance between the mathematical and biological optima in multiple sequence alignment. *Trends in Evolutionary Biology*, 2(1):e7.

Armougom, F., Moretti, S., Poirot, O., Audic, S., Dumas, P., Schaeli, B., Keduas, V., and Notredame, C. (2006). Expresso: automatic incorporation of structural information in multiple sequence alignments using 3d-coffee. *Nucleic Acids Research*, 34(suppl 2):W604–W608.

Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T., et al. (2000). Gene ontology: tool for the unification of biology. *Nature Genetics*, 25(1):25–29.

Attwood, T. and Parry-Smith, D. (2003). Multiple sequence analysis. In *Introduction to Bioinformatics*, Cell and molecular biology in action series. Prentice Hall.

Babich, G. and Camps, O. (1996). Weighted parzen windows for pattern classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18:567–570.

Barrett, T., Wilhite, S. E., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., Marshall, K. A., Phillippy, K. H., Sherman, P. M., Holko, M., Yefanov, A., Lee, H., Zhang, N., Robertson, C. L., Serova, N., Davis, S., and Soboleva, A. (2013). Ncbi geo: archive for functional genomics data setsupdate. *Nucleic Acids Research*, 41(D1):D991–D995.

Barton, G. J. and Sternberg, M. J. E. (1987). A strategy for the rapid multiple alignment of protein sequences - confidence levels from tertiary structure comparisons. *Journal of Molecular Biology*, 198(2):327–337.

Bellman, R. (1956). Dynamic programming and lagrange multipliers. *Proceedings of the National Academy of Sciences of the United States of America*, 42(10):767.

Benson, D. A., Cavanaugh, M., Clark, K., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J., and Sayers, E. W. (2013). Genbank. *Nucleic Acids Research*, 41(D1):D36–D42.

Berman, H., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T., Weissig, H., Shindyalov, I., and Bourne, P. (2000). The protein data bank. *Nucleic Acids Research*, 28(1):235–242.

Binns, D., Dimmer, E., Huntley, R., Barrell, D., O'Donovan, C., and Apweiler, R. (2009). Quickgo: a web-based tool for gene ontology searching. *Bioinformatics*, 25(22):3045–3046.

Bins, J. and Draper, B. (2001). Feature selection from huge feature sets. In *8th IEEE International Conference on Computer Vision, 2001. ICCV 2001. Proceedings*, volume 2, pages 159–165.

Blackburne, B. P. and Whelan, S. (2012). Measuring the distance between multiple sequence alignments. *Bioinformatics*, 28(4):495–502.

Blackburne, B. P. and Whelan, S. (2013). Class of multiple sequence alignment algorithm affects genomic analysis. *Molecular Biology and Evolution*, 30(3):642–653.

Boratyn, G. M., Schaffer, A., Agarwala, R., Altschul, S. F., Lipman, D. J., and Madden, T. L. (2012). Domain enhanced lookup time accelerated blast. *Biology Direct*, 7(1):12.

Bradley, R. K., Roberts, A., Smoot, M., Juvekar, S., Do, J., Dewey, C., Holmes, I., and Pachter, L. (2009). Fast statistical alignment. *PLoS Computational Biology*, 5(5):e1000392.

Breiman, L. (1993). *Classification and regression trees*. CRC press.

Breiman, L. (1996). Bagging predictors. *Machine Learning*, 26(2):123–140. cited By (since 1996)28.

Collins, F. S., Morgan, M., and Patrinos, A. (2003). The human genome project: lessons from large-scale biology. *Science*, 300(5617):286–290.

Connolly, M. L. (1983). Solvent-accessible surfaces of proteins and nucleic acids. *Science*, 221(4612):709–713.

Conover, W. J. (1999). *Practical nonparametric statistics*. Wiley, New York, 3er edition edition.

Cooper, G. M. and Hausman, R. E. (2000). The complexity of eukaryotic genomes. In *The cell: A molecular Approach*. Sinauer Associates Sunderland.

Cover, T. and Thomas, J. (2006). Elements of Information Theory. Wiley-Interscience, New York, NY, USA.

Dasgupta, D., Hernandez, G., Romero, A., Garrett, D., Kaushal, A., and Simien, J. (2009). On The Use of Informed Initialization and Extreme Solutions Subpopulation in Multiobjective Evolutionary Algorithms. In *MCDM: 2009 IEEE Symposium on Computational Intelligence in Multi-criteria Decision-making*, pages 58–65.

Dayhoff, M. O., Schwartz, R. M., and Orcutt, B. C. (1979). A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure 5*, 3:345352.

De Brabanter, K., Karsmakers, P., Ojeda, F., C., A., De Brabanter, J., Pelckmans, K., De Moor, B., Vandewalle, J., and Suykens, J. (2011). Ls-svmlab: a matlab toolbox for least squares support vector machines (v1.8).

de Juan, D., Pazos, F., and Valencia, A. (2013). Emerging methods in protein co-evolution. *Nature Reviews Genetics*, 14(4):249–261.

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(2):182–197.

Devereux, J., Haeberli, P., and Smithies, O. (1984). A comprehensive set of sequence-analysis programs for the vax. *Nucleic Acids Research*, 12(1):387–395.

Di Tommaso, P., Moretti, S., Xenarios, I., Orobitg, M., Montanyola, A., Chang, J.-M., Taly, J.-F., and Notredame, C. (2011). T-coffee: a web server for the multiple sequence alignment of protein and rna sequences using structural information and homology extension. *Nucleic acids research*, 39(suppl 2):W13–W17.

Do, C., Mahabhashyam, M., Brudno, M., and Batzoglou, S. (2005). Probcons: Probabilistic consistency-based multiple sequence alignment. *Genome Research*, 15(2):330–340.

Doolittle, R. F. (1981). Similar amino acid sequences: chance or common ancestry? *Science*, 214(4517):149–159.

Dua, S. and Chowriappa, P. (2012). *Data Mining for Bioinformatics*. An auerbach book. Taylor & Francis.

Eddy, S. R. (1995). Multiple alignment using hidden markov models. In *ISMB*, pages 114–120.

Edgar, R. (2004). Muscle: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Research*, 32:1792–1797.

Edgar, R. (2009). Bench. http://www.drive5.com/bench.

Edgar, R., Domrachev, M., and Lash, A. E. (2002). Gene expression omnibus: Ncbi gene expression and hybridization array data repository. *Nucleic Acids Research*, 30(1):207–210.

Eiben, A. E. and Smith, J. E. (2008). *Introduction to Evolutionary Computing (Natural Computing Series)*. Springer.

Espejo, P. G., Ventura, S., and Herrera, F. (2010). A survey on the application of genetic programming to classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 40(2):121–144.

Estevez, P., Tesmer, M., Perez, C., and Zurada, J. (2009). Normalized mutual information feature selection. *IEEE Transactions on Neural Networks*, 20(2):189–201.

Feng, D. and Doolittle, R. (1987). Progressive sequence alignment as a prerequisite correct phylogenetic trees. *Journal of Molecular Evolution*, 25(4):351–360.

Finn, R. D., Bateman, A., Clements, J., Coggill, P., Eberhardt, R. Y., Eddy, S. R., Heger, A., Hetherington, K., Holm, L., Mistry, J., et al. (2014). Pfam: the protein families database. *Nucleic acids research*, 42(D1):D222–D230.

Fitch, W. M. (1966). An improved method of testing for evolutionary homology. *Journal of Molecular Biology*, 16(1):9–16.

Fonseca, C. M., Guerreiro, A. P., López-Ibáñez, M., and Paquete, L. (2011). On the computation of the empirical attainment function. In *Evolutionary Multi-Criterion Optimization*, pages 106–120. Springer.

Gacto, M. J., Alcalá, R., and Herrera, F. (2009). Adaptation and application of multi-objective evolutionary algorithms for rule reduction and parameter tuning of fuzzy rule-based systems. *Soft Computing*, 13(5):419–436.

Gelly, J., Joseph, A., Srinivasan, N., and de Brevern, A. (2011). ipba: a tool for protein structure comparison using sequence alignment strategies. *Nucleic Acids Research*, 39:W18–W23.

Gondro, C. and Kinghorn, B. P. (2007). A simple genetic algorithm for multiple sequence alignment. *Genetics and Molecular Research*, 6(4):964–982.

Gonnet, G. H., Cohen, M. A., and Benner, S. A. (1992). Exhaustive matching of the entire protein sequence database. *Science*, 256(5062):1443–1445.

Gotoh, O. (1990). Consistency of optimal sequence alignments. *Bulletin of Mathematical Biology*, 52(4):509–525.

Gotoh, O. (1996). Significant improvement in accuracy of multiple protein sequence alignments by iterative refinement as assessed by reference to structural alignments. *Journal of Molecular Biology*, 264(4):823–838.

Guex, N. and Peitsch, M. C. (1997). Swiss-model and the swiss-pdb viewer: an environment for comparative protein modeling. *Electrophoresis*, 18(15):2714–2723.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. H. (2009). The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18.

Henikoff, S. and Henikoff, J. G. (1992). Amino-acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America*, 89(22):10915–10919.

Hestenes, M. and Stiefel, E. (1952). Methods of conjugate gradients for solving linear systems. *Journal of research of the National Bureau of Standards*, 49:409–436.

Hicks, S., Wheeler, D., Plon, S., and Kimmel, M. (2011). Prediction of missense mutation functionality depends on both the algorithm and sequence alignment employed. *Human Mutation*, 32:661–668.

Hogeweg, P. and Hesper, B. (1984). The alignment of sets of sequences and the construction of phyletic trees: An integrated method. *Journal of Molecular Evolution*, 20:175–186.

Huang, X. and Miller, W. (1991). Lalign-find the best local alignments between two sequences. *Advances in Applied Mathematics*, 12:373–381.

Huerta, M., Downing, G., Haseltine, F., Seto, B., and Liu, Y. (2000). Nih working definition of bioinformatics and computational biology. *US National Institute of Health*.

James Watson, Francis Crick et al. (1953). Molecular structure of nucleic acids. *Nature*, 171(4356):737–738.

John, G., Kohavi, R., and Pfleger, K. (1994). Irrelevant features and the subset selection problem. In *International Conference on Machine Learning*, pages 121–129.

Jones, D. T. (1999). Protein secondary structure prediction based on position-specific scoring matrices. *Journal of molecular biology*, 292(2):195–202.

Joosten, R. P., Te Beek, T. A., Krieger, E., Hekkelman, M. L., Hooft, R. W., Schneider, R., Sander, C., and Vriend, G. (2011). A series of pdb related databases for everyday needs. *Nucleic Acids Research*, 39(suppl 1):D411–D419.

Kabsch, W. and Sander, C. (1983). Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22(12):2577–2637.

Katoh, K., Misawa, K., Kuma, K., and Miyata, T. (2002). Mafft: a novel method for rapid multiple sequence alignment based on fast fourier transform. *Nucleic Acids Research*, 30:3059–3066.

Katoh, K. and Toh, H. (2008). Recent developments in the mafft multiple sequence alignment program. *Briefings in Bioinformatics*, 9(4):286–298.

Kececioglu, J. and DeBlasio, D. (2013). Accuracy estimation and parameter advising for protein multiple sequence alignment. *Journal of Computational Biology*, 20(4):259–279.

Kececioglu, J., Kim, E., and Wheeler, T. (2010). Aligning protein sequences with predicted secondary structure. *Journal of Computational Biology*, 17(3):561–580.

Kemena, C. and Notredame, C. (2009). Upcoming challenges for multiple sequence alignment methods in the high-throughput era. *Bioinformatics*, 25:2455–2465.

Kemena, C., Taly, J. F., Kleinjung, J., and Notredame, C. (2011). Strike: evaluation of protein msas using a single 3d structure. *Bioinformatics*, 27(24):3385–3391.

Kerrien, S., Orchard, S., Montecchi-Palazzi, L., Aranda, B., Quinn, A. F., Vinod, N., Bader, G. D., Xenarios, I., Wojcik, J., Sherman, D., et al. (2007). Broadening the horizon–level 2.5 of the hupo-psi format for molecular interactions. *BMC biology*, 5(1):44.

Ku, C.-S. and Roukos, D. H. (2013). From next-generation sequencing to nanopore sequencing technology: paving the way to personalized genomic medicine. *Expert Review of Medical Devices*, 10(1):1–6.

Kukic, P., Mirabello, C., Tradigo, G., Walsh, I., Veltri, P., and Pollastri, G. (2014). Toward an accurate prediction of inter-residue distances in proteins using 2d recursive neural networks. *BMC Bioinformatics*, 15(1):6.

Kullback, S. (1997). *Information Theory and Statistics*. Courier Dover Publications, New York, NY, USA.

Lassmann, T. and Sonnhammer, E. (2005). Kalign - an accurate and fast multiple sequence alignment algorithm. *BMC Bioinformatics*, 6:298.

Leinonen, R., Diez, F. G., Binns, D., Fleischmann, W., Lopez, R., and Apweiler, R. (2004). Uniprot archive. *Bioinformatics*, 20(17):3236–3237.

Li, A., Marz, M., Qin, J., and Reidys, C. (2011). Rna-rna interaction prediction based on multiple sequence alignments. *Bioinformatics*, 27:456–463.

Li, H. and Homer, N. (2010). A survey of sequence alignment algorithms for next-generation sequencing. *Briefings in Bioinformatics*, 11(5):473–483.

Li, J. and Fang, H. (2012). Substitution transformation of score matrix for improving alignment quality of local sequence of distantly related proteins. *Current Bioinformatics*, 7(1):35–42.

Lin, K., Kleinjung, J., Taylor, W. R., and Heringa, J. (2003). Testing homology with contact accepted mutation (cao): a contact-based markov model of protein evolution. *Computational Biology and Chemistry*, 27(2):93–102.

Liu, K., Raghavan, S., Nelesen, S., Linder, C. R., and Warnow, T. (2009). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science*, 324(5934):1561–1564.

Liu, K. and Warnow, T. (2014). Large-scale multiple sequence alignment and tree estimation using saté. In *Multiple Sequence Alignment Methods*, pages 219–244. Springer.

MacKay, D. J. (1998). Introduction to gaussian processes. *NATO ASI Series F Computer and Systems Sciences*, 168:133–166.

Mann, H. B., Whitney, D. R., et al. (1947). On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1):50–60.

Martinez-Muñoz, G., Hernandez-Lobato, D., and Suarez, A. (2009). An analysis of ensemble pruning techniques based on ordered aggregation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(2):245–259.

Mathews, C., Holde, K., and Ahern, K. (2000). Biochemistry. Benjamin Cummings Publishing, Redwood City, CA, USA.

Mirarab, S. and Warnow, T. (2011). Fastsp: linear time calculation of alignment accuracy. *Bioinformatics*, 27(23):3250–3258.

Mizuguchi, K., Deane, C. M., Blundell, T. L., and Overington, J. P. (1998). Homstrad: a database of protein structure alignments for homologous families. *Protein Science*, 7(11):2469–2471.

Morgenstern, B., Dress, A., and Werner, T. (1996). Multiple dna and protein sequence alignment based on segment-to-segment comparison. *Proceedings of the National Academy of Sciences of the United States of America*, 93(22):12098–12103.

Muller, J., Creevey, C. J., Thompson, J. D., Arendt, D., and Bork, P. (2010). Aqua: automated quality improvement for multiple sequence alignments. *Bioinformatics*, 26(2):263–265.

Nakamura, Y., Cochrane, G., and Karsch-Mizrachi, I. (2013). The international nucleotide sequence database collaboration. *Nucleic Acids Research*, 41(D1):D21–D24.

National Center for Biotechnology Information (NCBI) (2013). The ncbi handbook. http://www.ncbi.nlm.nih.gov/books/NBK143764/.

National Human Genome Research Institute (NHGRI) (2014). Dna sequencing costs. http://genome.gov/sequencingcosts.

National Human Genome Research Institute (NHGRI) (2014). Talking glossary of genetic terms. http://genome.gov/Glossary.

Naznin, F., Sarker, R., and Essam, D. (2011). Vertical decomposition with genetic algorithm for multiple sequence alignment. *BMC Bioinformatics*, 12.

Needleman, S. B. and Wunsch, C. D. (1970). A general method applicable to the search for similarities in the amino acid sequence of two proteins. *Journal of Molecular Biology*, 48(3):443–453.

Nelder, J. A. and Mead, R. (1965). A simplex method for function minimization. *Computer Journal*, 7(4):308–313.

Notredame, C., Higgins, D., and Heringa, J. (2000). T-coffee: A novel method for fast and accurate multiple sequence alignment. *Journal of Molecular Biology*, 302(1):205–217.

Notredame, C. and Higgins, D. G. (1996). Saga: sequence alignment by genetic algorithm. *Nucleic Acids Research*, 24(8):1515–24.

Nozaki, Y. and Bellgard, M. (2005). Statistical evaluation and comparison of a pairwise alignment algorithm that a priori assigns the number of gaps rather than employing gap penalties. *Bioinformatics*, 21(8):1421–1428.

Nuin, P., Wang, Z., and Elisabeth, R. (2006). The accuracy of several multiple sequence alignment programs for proteins. *BMC Bioinformatics*, 7:471.

Ombuki, B., Ross, B. J., and Hanshar, F. (2006). Multi-objective genetic algorithms for vehicle routing problem with time windows. *Applied Intelligence*, 24(1):17–30.

Orchard, S., Ammari, M., Aranda, B., Breuza, L., Briganti, L., Broackes-Carter, F., Campbell, N. H., Chavali, G., Chen, C., del Toro, N., et al. (2014). The mintact projectintact as a common curation platform for 11 molecular interaction databases. *Nucleic acids research*, 42(D1):D358–D363.

Orobitg, M., Cores, F., Guirado, F., Roig, C., and Notredame, C. (2013). Improving multiple sequence alignment biological accuracy through genetic algorithms. *The Journal of Supercomputing*, 65(3):1076–1088.

Ortuño, F., Florido, J. P., Urquiza, J. M., Pomares, H., Prieto, A., and Rojas, I. (2012). Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on nsga-ii. In *Evolutionary Computation (CEC), 2012 IEEE Congress on*, pages 1–8. IEEE.

Ortuño, F., Rojas, I., Pomares, H., Urquiza, J., and Florido, J. (2011). Emerging methodologies in multiple sequence alignment using high throughput data. In *5th International Conference on Practical Applications of Computational Biology and Bioinformatics (PACBB 2011)*, volume 93 of *Advances in Intelligent and Soft Computing*, pages 183–190. Springer Berlin Heidelberg.

Ortuño, F. M., Valenzuela, O., Pomares, H., Rojas, F., Florido, J. P., Urquiza, J. M., and Rojas, I. (2013). Predicting the accuracy of multiple sequence alignment algorithms by using computational intelligent techniques. *Nucleic acids research*, 41(1):e26–e26.

O'Sullivan, O., Suhre, K., Abergel, C., Higgins, D., and Notredame, C. (2004). 3dcoffee: Combining protein sequences and structures within multiple sequence alignments. *Journal of Molecular Biology*, 340(2):385–395.

Pei, J. and Grishin, N. V. (2007). Promals: towards accurate multiple sequence alignments of distantly related proteins. *Bioinformatics*, 23(7):802–808.

Pei, J., Kim, B.-H., and Grishin, N. V. (2008). Promals3d: a tool for multiple protein sequence and structure alignments. *Nucleic Acids Research*, 36(7):2295–2300.

Peng, H., Long, F., and Ding, C. (2005). Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27:1226–1238.

Pietrokovski, S., Henikoff, J. G., and Henikoff, S. (1996). The blocks databasea system for protein classification. *Nucleic Acids Research*, 24(1):197–200.

Raghava, G., Searle, S. M., Audley, P. C., Barber, J. D., and Barton, G. J. (2003). Oxbench: a benchmark for evaluation of protein multiple sequence alignment accuracy. *BMC Bioinformatics*, 4(1):47.

Rasmussen, C. E. (2006). Gaussian processes for machine learning.

Rognes, T. (2011). Faster smith-waterman database searches with inter-sequence simd parallelisation. *BMC bioinformatics*, 12(1):221.

Rokach, L. (2008). *Data mining with decision trees: theory and applications*, volume 69. World scientific.

Rokach, L. and Maimon, O. (2005). Top-down induction of decision trees classifiers - a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 35(4):476–487.

Ronquist, F., Teslenko, M., van der Mark, P., Ayres, D. L., Darling, A., Höhna, S., Larget, B., Liu, L., Suchard, M. A., and Huelsenbeck, J. P. (2012). Mrbayes 3.2: efficient bayesian phylogenetic inference and model choice across a large model space. *Systematic Biology*, 61(3):539–542.

Roslan, R., Othman, R., Shah, Z., Kasim, S., Asmuni, H., Taliba, J., Hassan, R., and Zakaria, Z. (2010). Utilizing shared interacting domain patterns and gene ontology information to improve protein-protein interaction prediction. *Computers in Biology and Medicine*, 40:555–564.

Rossi, F., Lendasse, A., Francois, D., Wertz, V., and Verleysen, M. (2006). Mutual information for the selection of relevant variables in spectrometric nonlinear modelling. *Chemometrics and Intelligent Laboratory Systems*, 80:215–226.

Rothberg, J. M., Hinz, W., Rearick, T. M., Schultz, J., Mileski, W., Davey, M., Leamon, J. H., Johnson, K., Milgrew, M. J., Edwards, M., et al. (2011). An integrated semiconductor device enabling non-optical genome sequencing. *Nature*, 475(7356):348–352.

Rusk, N. (2013). Disruptive nanopores. *Nature Methods*, 10(1):35–35.

Saitou, N. and Nei, M. (1987). The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Molecular Biology and Evolution*, 4(4):406–425.

Sanger, F., Nicklen, S., and Coulson, A. R. (1977). Dna sequencing with chain-terminating inhibitors. *Proceedings of the National Academy of Sciences*, 74(12):5463–5467.

Schwartz, A. S. and Pachter, L. (2007). Multiple alignment by sequence annealing. *Bioinformatics*, 23(2):e24–e29.

Shi, J., Blundell, T. L., and Mizuguchi, K. (2001). Fugue: sequence-structure homology recognition using environment-specific substitution tables and structure-dependent gap penalties. *Journal of Molecular Biology*, 310(1):243–257.

Siddiqui, A. S., Dengler, U., and Barton, G. J. (2001). 3dee: a database of protein structural domains. *Bioinformatics*, 17(2):200–201.

Sierk, M. L., Smoot, M. E., Bass, E. J., and Pearson, W. R. (2010). Improving pairwise sequence alignment accuracy using near-optimal protein sequence alignments. *BMC Bioinformatics*, 11.

Sievers, F., Wilm, A., Dineen, D., Gibson, T. J., Karplus, K., Li, W., Lopez, R., McWilliam, H., Remmert, M., Söding, J., et al. (2011). Fast, scalable generation of high-quality protein multiple sequence alignments using clustal omega. *Molecular Systems Biology*, 7(1).

Smith, R. F. and Smith, T. F. (1992). Pattern-induced multi-sequence alignment (pima) algorithm employing secondary structure-dependent gap penalties for use in comparative protein modeling. *Protein Engineering*, 5(1):35–41.

Smith, T. and Waterman, M. (1981). Identification of common molecular subsequences. *Journal of Molecular Biology*, 147(1):195–197.

Soh, J., Gordon, P., and Sensen, C. (2012). Functional annotation. In *Genome Annotation*, Chapman & Hall CRC Mathematical & Computational Biology. Taylor & Francis.

Sokal, R. R. (1958). A statistical method for evaluating systematic relationships. *The University of Kansas Science Bulletin*, 38:1409–1438.

Sonnhammer, E. L., Eddy, S. R., Birney, E., Bateman, A., and Durbin, R. (1998). Pfam: multiple sequence alignments and hmm-profiles of protein domains. *Nucleic Acids Research*, 26(1):320–322.

Spears, W. M., Jong, K. A. D., Bck, T., Fogel, D. B., and De Garis, H. (1993). An overview of evolutionary computation. *Lecture Notes in Computer Science*, 667(1):442–459.

Styczynski, M. P., Jensen, K. L., Rigoutsos, I., and Stephanopoulos, G. (2008). BLOSUM62 miscalculations improve search performance. *Nature biotechnology*, 26(3):274–275.

Suykens, J., Van Gestel, T., De Brabanter, J., De Moor, B., and Vandewalle, J. (2003). *Least Squares Support Vector Machines*. World Scientific Pub. Co. Inc., Singapore.

Suzek, B. E., Huang, H., McGarvey, P., Mazumder, R., and Wu, C. H. (2007). Uniref: comprehensive and non-redundant uniprot reference clusters. *Bioinformatics*, 23(10):1282–1288.

Szabo, A., Novak, A., Miklos, I., and Hein, J. (2010). Reticular alignment: A progressive corner-cutting method for multiple sequence alignment. *BMC Bioinformatics*, 11:570.

Taheri, J. and Zomaya, A. Y. (2009). Rbt-ga: a novel metaheuristic for solving the multiple sequence alignment problem. *BMC Genomics*, 10 Suppl 1:S10.

Taylor, W. R. and Orengo, C. A. (1989). Protein structure alignment. *Journal of Molecular Biology*, 208(1):1–22.

The UniProt Consortium (2014). Activities at the universal protein resource (uniprot). *Nucleic Acids Research*, 42(D1):D191–D198.

Thompson, J., Koehl, P., Ripp, R., and Poch, O. (2005). Balibase 3.0: Latest developments of the multiple sequence alignment benchmark. *Proteins-Structure Function and Bioinformatics*, 61(1):127–136.

Thompson, J., Muller, A., Waterhouse, A., Procter, J., Barton, G., Plewniak, F., and Poch, O. (2006). Macsims: multiple alignment of complete sequences information management system. *BMC Bioinformatics*, 7:318.

Thompson, J. D., Higgins, D. G., and Gibson, T. J. (1994). Clustalw: Improving the sensivity of progressive multiple sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Research*, 22:4673–4680.

Thompson, J. D., Plewniak, F., and Poch, O. (1999). Balibase: a benchmark alignment database for the evaluation of multiple alignment programs. *Bioinformatics*, 15(1):87–88.

Tsujimoto, Y., Hitotsuyanagi, Y., Nojima, Y., and Ishibuchi, H. (2009). Effects of Including Single-Objective Optimal Solutions in an Initial Population on Evolutionary Multiobjective Optimization. In *2009 International Conference of Soft Computing and Pattern Recognition*, pages 352–357.

U.S. National Library of Medicine (NLM) (2014). Handbook: Help me understand genetics. http://ghr.nlm.nih.gov/handbook.

Wallace, I. M., O'Sullivan, O., Higgins, D. G., and Notredame, C. (2006). M-coffee: combining multiple sequence alignment methods with t-coffee. *Nucleic Acids Research*, 34(6):1692–1699.

Watson, J. D. (1990). The human genome project: past, present, and future. *Science*, 248(4951):44–49.

Westesson, O., Barquist, L., and Holmes, I. (2012). Handalign: Bayesian multiple sequence alignment, phylogeny and ancestral reconstruction. *Bioinformatics*, 28(8):1170–1171.

Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics Bulletin*, 1(6):80–83.

Wong, K. M., Suchard, M. A., and Huelsenbeck, J. P. (2008). Alignment uncertainty and genomic analysis. *Science*, 319(5862):473–476.

Wu, S. and Manber, U. (1992). Fast text searching: allowing errors. *Communications of the ACM*, 35(10):83–91.

Wu, X., Zhu, L., Guo, J., Zhang, D., and Lin, K. (2006). Prediction of yeast protein-protein interaction network: insights from the gene ontology and annotations. *Nucleic Acids Research*, 34:2137–2150.

wwPDB (2008). Protein data bank contents guide: Atomic coordinate entry format description.

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

Zitzler, E., Brockhoff, D., and Thiele, L. (2007). The hypervolume indicator revisited: On the design of pareto-compliant indicators via weighted integration. *Evolutionary Multi-Criterion Optimization, Proceedings*, 4403:862–876.

Zitzler, E., Knowles, J., and Thiele, L. (2008). Quality assessment of pareto set approximations. *Multiobjective Optimization: Interactive and Evolutionary Approaches*, 5252:373–404.

Zitzler, E., Laumanns, M., and Thiele, L. (2002). Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. *Evolutionary Methods for Design Optimisation and Control with Application to Industrial Problems EUROGEN 2001*, pages 95–100.

Curriculum Vitae

Personal Information

Francisco Manuel Ortuño Guzmán
June 28, 1985
Montefrío, Granada (Spain)

Education

2010 - 2014	PhD studies at CASIP group, Department of Computer Architecture and Computer Technology, University of Granada, Spain
2010 - 2011	Master's Degree in Computer and Network Engineering (Postgraduate Course), University of Granada, Spain
2009 - 2009	Teaching Certificate (CAP), University of Granada, Spain
2003 - 2008	M.Sc. Telecommunication Engineering at E. T. S. Ingenieras Informática y de Telecomunicación, University of Granada, Spain
1997 - 2003	Secondary and High School at IES Hiponova, Montefrío, Granada, Spain
1991 - 1997	Primary School at CEIP La Paz, Montefrío, Granada, Spain

Research/Work Experience

2010 - Today	PhD Position (Project of Excellence, P09-TIC-175476) at Department of Computer Architecture and Computer Technology, University of Granada, Spain
2011 - 2011	Visiting Research at Computational Biology and Data Mining group, Max Delbruck Center for Molecular Medicine, Berlin, Germany.
2009 - 2010	Research Assistant, Project ”Embedded Systems For Critical Infrastructures” (SEPIC) at Department of Computer Architecture and Computer Technology in collaboration with Telvent Energy S.A., University of Granada, Spain
2008 - 2009	Electronic Engineer, Development and Innovation Department, Mafero Electronics S.L.U.
2007 - 2008	Engineering Internship, Development and Innovation Department, Mafero Electronics S.L.U.

Editorial and Conference Positions

2014	Guest Editor, *Theoretical Biology and Medical Modelling*, IWBBIO 2013 Special Issue.
2014	Editor, *Proceedings of the* 2^{nd} *International Work-Conference of Bioinformatics and Biomedical Engineering* (IWBBIO 2014).
2014	Local Organizer, 2^{nd} *International Work-Conference of Bioinformatics and Biomedical Engineering* (IWBBIO 2014).
2013	Editor, *Proceedings of the* 1^{st} *International Work-Conference of Bioinformatics and Biomedical Engineering* (IWBBIO 2013).
2013	Local Organizer, 1^{st} *International Work-Conference of Bioinformatics and Biomedical Engineering* (IWBBIO 2013).

List of Publications

International Journals(SCI-Indexed)

Torres, C., Perales, S., Alejandre, M.J., Iglesias, J., Palomino, R., Caba, O., Prados, J.C., Aranega, A., Delgado, J.R., Irigoyen, A., Ortuño, F.M., Rojas, I., Linares, A. **Serum Cytokine Profile in Patients with Pancreatic Cancer**. *Pancreas* (Accepted 01/04/2014).

Ortuño, F.M., Rojas, I.: **Advances in bioinformatics and biomedical engineering - special issue of IWBBIO 2013.** *Theoretical Biology and Medical Modelling*, 11(Suppl 1):I1 (2014).

Ortuño, F.M., Valenzuela, O., Rojas, F., Pomares, H., Florido, J.P., Urquiza, J.M., Rojas, I.: **Optimizing multiple sequence alignments using a genetic algorithm based on three objectives: structural information, non-gaps percentage and totally conserved columns.** *Bioinformatics* 29, 2112-2121 (2013).

Ortuño, F.M., Rojas, I., Andrade-Navarro, M.A., Fontaine, J.-F.: **Using cited references to improve the retrieval of related biomedical documents.** *BMC Bioinformatics* 14(1), 113 (2013).

Ortuño, F.M., Valenzuela, O., Pomares, H., Rojas, F., Florido, J.P., Urquiza, J.M., Rojas, I.: **Predicting the accuracy of multiple sequence alignment algorithms by using computational intelligent techniques.** *Nucleic Acids Research* 41, e26 (2013).

Florido, J.P., Pomares, H., Rojas, H., Guillen, A., Ortuño, F.M., Urquiza, J.M.: **An effective, practical and low computational cost framework for the integration of heterogeneous data to predict functional associations between proteins by means of Artificial Neural Networks.** *Neurocomputing*. 121 (2013): 64-78.

International Conferences

Ortuño, F.M., Valenzuela, O., Pomares, H., Rojas, I.: **Evaluating multiple sequence alignments using a LS-SVM approach with a heterogeneous set of biological features.** *Advances in Computational Intelligence (IWANN 2013)*, vol. 7903, 150-158 (2013).

Valenzuela, O., Ortuño, F.M., Rojas, F., Pomares, H., Bernier, J.L., Herrera, L.J., Guillen, A.: **Using intelligent system for medical decision-making to magnetic resonance imaging.** *Proceedings of 1st International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO)*, 1, 205-214 (2013).

Ortuño, F.M., Valenzuela, O., Pomares, H., Rojas, I.: **Determining the most suitable multiple sequence alignment methodology by using a set of heterogeneous biological features.** *Proceedings of 1st International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO)*, 1, 283-288 (2013).

Valenzuela, O., Pasadas, M., Ortuño, F.M., Rojas, I.: **Optimal knots allocation in smoothing splines using intelligent system. Application in biomedical signal processing.** *Proceedings of 1st International Work-Conference on Bioinformatics and Biomedical Engineering (IWBBIO)*, 1, 289-295 (2013).

Ortuño, F.M., Florido, J.P., Urquiza, J.M., Pomares, H., Prieto, A., Rojas, I.: **Optimization of multiple sequence alignment methodologies using a multiobjective evolutionary algorithm based on NSGA-II.** *IEEE Congress on Evolutionary Computation (CEC)*, 1-8 (2012).

Torres, C., Romero, S.P., Morales, R.P., Perez, M.J.A., Gomez, J.I., Caba, O., Prados, J.C., Aranega, A., Delgado, J.R., Irigoyen, A., Ortuño, F.M., Gil, A.L.: **Cytokines profile expression in pancreatic cancer patients.** *FEBS Conference* 279, 228-228 (2012).

Florido, J.P., Pomares, H., Rojas, I., Urquiza, J.M., Ortuño, F.M.: **Prediction of functional associations between proteins by means of a cost-sensitive artificial neural network.** *Advances in Computational Intelligence (IWANN 2011)*, vol. 6692, 194-201 (2011).

Ortuño, F.M., Rojas, I., Pomares, H., Urquiza, J.M., Florido, J.P.: **Emerging methodologies in multiple sequence alignment using high throughput data.** *5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACCB 2011)* 93, 183-190 (2011).

Urquiza, J.M., Rojas, I., Pomares, H., Herrera, L.J., Florido, J.P., Ortuño, F.M.: **Using machine learning techniques and genomic/proteomic information from known databases for PPI prediction.** *5th International Conference on Practical Applications of Computational Biology & Bioinformatics (PACCB 2011) 93, 373-380 (2011).*