Alma mater studiorum - Università di Bologna

DOTTORATO DI RICERCA IN FISICA: XXI ciclo

Settore scientifico disciplinare di afferenza: FIS/07

COMPUTATIONAL METHODS FOR THE ANALYSIS OF PROTEIN STRUCTURE AND FUNCTION

Presentata da:
Lisa Bartoli

Coordinatore Dottorato:
Chiar.mo Prof. Fabio Ortolani

Relatore:
Chiar.mo Prof. Franco Casali

Correlatori:
Chiar.ma Prof.ssa Rita Casadio
Dr. Piero Fariselli
Dr. Pier Luigi Martelli

Esame finale anno 2009

Contents

List of Figures 5
List of Tables 9

Introduction 11

1 Proteins, proteomes and genomes 13
1.1 From genome to proteome 13
1.1.1 Nucleic acids 13
1.1.2 The genetic code 14
1.1.3 Genome complexity 15
1.1.4 Genome databases 16
1.2 Proteins and proteomes 16
1.2.1 Protein structure 17
1.2.2 The protein archives 21

2 Analysis and prediction of protein structure and function . . 25
2.1 Protein structure prediction and modelling 27
2.2 Sequence-based protein function prediction 29
2.3 The evolutionary information 32
2.3.1 Protein sequence alignments 33
2.3.2 BLAST 34
2.3.3 Alignment profiles 35
2.3.4 PSI-BLAST 36

3 Methods 37
3.1 Graphical models 37
3.1.1 Introduction to graph theory 37
3.1.2 From random to small-world networks 38
3.1.3 Algorithms on graphs: some definitions 42
3.2 The probabilistic framework 44
3.2.1 The learning procedure 45
3.3 Markov chains and Hidden Markov Models 46
3.3.1 Parameters estimation (ML) 47
3.3.2 First order models 47
3.3.3 Higher order models 47

3.3.4 Hidden Markov Models 48
3.3.5 The three fundamental problems of HMMs 50
3.4 Dynamic programming algorithms for HMMs 53
3.4.1 The evaluation problem: the forward algorithm 53
3.4.2 The evaluation problem: the backward algorithm 55
3.4.3 The decoding problem: the Viterbi algorithm 56
3.4.4 The Posterior Viterbi decoding 58
3.4.5 The Expectation Maximization algorithm and the training problem 59

4 The effect of backbone on the small-world properties of protein contact maps 65
4.1 From protein structure to contact maps 65
4.2 Small-world and protein structures 68
4.3 The data set 72
4.4 Protein representation and threshold cut-off 73
4.5 The effect of the protein backbone 75
4.6 Discussion 78

5 Improving coiled-coil prediction with evolutionary information 79
5.1 Coiled-coils 79
5.2 The state of the art in coiled-coil prediction 83
5.3 The data set 84
5.4 The Hidden Markov Model (CCHMM_PROF) 86
5.5 Scoring the performance 87
5.5.1 Per-residue indices 87
5.5.2 Per-segment indices 87
5.5.3 Per-protein index 88
5.6 Results 90
5.6.1 Discriminating coiled-coil sequences 90
5.6.2 Locating coiled-coil regions in protein sequences 92
5.6.3 Comparative evaluation 93
5.7 Biological insight: targeting viruses 95
5.8 Discussion 96

6 Comparative large-scale genome analysis 97
6.1 Introduction 97
6.2 The data set 99
6.3 Sequence comparison and clustering 100
6.4 Mapping functions and structures on the clusters 101
6.5 Statistical evaluation of GO terms 102
6.6 Results 103
6.6.1 Clustering procedure and cluster content: clusters vs singletons 104
6.6.2 From sequence to function 105

6.6.3 From sequence to structure 106
6.6.4 From sequence to function and structure 108
6.6.5 Blind test on a data set of 201 genomes 109
6.7 Discussion 110

Conclusions 111

Bibliography 115

Publications 125

List of Figures

1.1 The standard genetic code. 15
1.2 General structure of an amino acid. 17
1.3 The 20 amino acids. 18
1.4 Formation and charactersitics of the peptide bond. 20

2.1 Growth in the content of PDB and SwissProt databases from 1986 to 2008. 25
2.2 Relationships among protein sequence, structure and function. . . 26
2.3 Variation of size of the protein core with the percentage of its identical residues [Chothia and Lesk, 1986]. 28
2.4 Variation of r.m.s deviation of the protein core with the percentage of its identical residues. The figure shows computed results for 32 pairs of homologous proteins of different structural types [Chothia and Lesk, 1986]. 28
2.5 Multi-domain proteins problem in three bacterial proteins: O31395 (Transcriptional activator protein irlR), P54662 (Transcriptional regulatory protein degU), O30919 (Transcriptional activator protein solR). 31
2.6 The BLOSUM62 matrix. 33
2.7 Example of a pairwise alignment performed by the BLAST program. 34
2.8 Building of sequence profile starting from a multiple sequence alignment. 35

3.1 Random rewiring procedure with different outputs: regular, small-world and random graphs [Watts and Strogatz, 1998]. 39
3.2 The degree distribution that results from the numerical simulation of a random graph (N is the total number of nodes, X_k are the nodes with degree k) [Barabasi and Albert, 2002]. 40
3.3 Characteristic path length $l(p)$ and clustering coefficient $C(p)$ for the Watts-Strogatz model [Watts and Strogatz, 1998]. 41
3.4 Degree distribution of the Watts-Strogatz model for $K = 3$ and various p [Barrat and Weigt, 2000]. 42
3.5 A pictorial example of a graph G1 with 5 nodes and 7 connections. 42
3.6 First order Markov model for DNA sequences. 48

3.7 Markov models of the first and second order for sequences of a two-characters alphabet. The second order model (on the left) can be reduced to a first order model with 4 states. It can be noticed that the number of transition parameters increases. 48

3.8 HMM for secondary structure prediction. It includes the constraints on the minimum sequence length of α-helix and β-strand. 51

3.9 Representation of the forward algorithm. Each arrow represents the product of an element of the matrix with the corresponding transition probability. In the inizialization, all the elements in position 0 are null, with the exception of the BEGIN. The definition of a matrix element requires the sum of all the quantities corresponding to the *in-arrows* $\rightarrow$, coming from the previous position in the sequence and the product for the emission probability. . . 54

3.10 The Viterbi algorithm. Each arrow represents the product of an element of the matrix with the corresponding transition probability. The definition of a matrix element requires the search of the maximum among all the quantities corresponding to the *in-arrows* $\rightarrow$, coming from the previous position in the sequence and its product for the emission probability. 57

3.11 The Viterbi algorithm. The matrix elements corresponding to the maximum *in-arrows* (in red) are stored in the pointers matrix, indicated in the figure with $\leftarrow$. Following the pointers starting from the END, the Viterbi path (coloured in blu) is obtained. If more than one pointer comes from a given position, as depicted in fugure, the Viterbi paths are more than one and all with the same probability. 57

4.1 Contact map of protein HSP-60 protein fragment (PDB id: 1kid). 66

4.2 Residue contact distribution $p(k)$. 70

4.3 Distribution of the characteristic path length (L) as a function of the clustering coefficient (C). Black circles represent the complete protein contact maps, grey triangles are protein long-range contacts while random networks are labeled with black squares. . 70

4.4 Clustering coefficient (C) as a function of the protein length (N). 71

4.5 L-C plot for all-α proteins (black) and all-β proteins (grey). The two crosses indicate the average values for the two groups. 72

4.6 Average Hamming distance as a function of the CA-threshold distance when the all-atom one is set to 5 Å. 74

4.7 Characteristic path length (L) and clustering coefficient (C) for the protein contact maps computed using CA-representation (threshold set to 7 Å, C=0.56±0.03, L=5.3±1.4) and all-atom representation (threshold set to 5 Å, C=0.56±0.04, L=5.3±1.4), with respect to the corresponding values obtained for random and regular graphs. 74

4.8 Characteristic path length (L) and clustering coefficient (C) as a function of the different threshold cut-offs (6-12 Å) for the CA representation. 75

4.9 Characteristic path length (L) and clustering coefficient (C) of the real contact maps (real have C=0.56±0.03, L=5.3±1.4) with respect to Random (randomizing all contacts), Shuffled (the randomly generated contact maps using complete uncorrelated shuffling) and RandomNN the randomly assigned locally correlated contacts (C=0.55±0.03, L=5.0±0.8). 76

4.10 An example of real protein contact map (PDB code 1amx) is shown in the upper triangle with respect to a randomly generated one computed by assuming a RandomNN procedure (rules 1-3) as detailed in the text (lower triangle). 78

5.1 On the left, X-ray diffraction pattern from a lock of Mozarts hair taken by W.T. Astbury in 1958 (from The Brotherton Collection, Leeds University Library, UK). On the right, X-ray pattern from α-keratin showing the very sharp 0.51 nm meridional arc and complex of equatorial and near-equatorial maxima at a lateral spacing of 0.98 nm. Near-equatorial layer lines having an axial spacing of about 7 nm are associated with the pitch length of the coiled-coil and are indicated by horizontal arrows [Parry *et al.*,2008]. 80

5.2 (a) Schematic representation of the positions around the two α-helices in a two-chain heptad coiled-coil of the amino acids in the a to g positions. (b) Two superimposed radial nets of the tropomyosin chain, which was the first fibrous protein sequence determined. Adjacent to the line of contact between the two chains (dashed line) there is a high concentration of apolar residues fitting together with knob-into-hole packing. (c) Representation of a heptad (left-handed) two-chain coiled-coil where each amino acid has been represented by a circle [Parry *et al.*,2008]. 81

5.3 Periodicities of coiled-coil proteins: parallel and antiparallel coiled-coil structures with heptad and non-heptad periodicities. 83

5.4 Automaton representation of the CCHMM_PROF Hidden Markov Model. The allowed transitions are indicated with the arrows. The states inside the two coiled-coil boxes (H, a, b, c, d, e, f, g) are fully-connected but for sake of clarity only the most probable transitions are indicated. 86

5.5 Receiver Operating Characteristic (ROC) curve of the results of CCHMM_PROF with different window sizes. 89

5.6 Receiver Operating Characteristic (ROC) curve of the results of the different predictors in the discrimination task. 91

6.1 Cumulative distributions of the Bonferroni corrected P-values. Dotted line: cumulative function of the observed P-values. Solid line: cumulative distribution of the average of the P-values of a 100 random benchmark set; error bars represent standard deviations.. 103

6.2 Barplot of the different degrees of annotation provided by the method for each standard deviation class (Table 6.1), computed on the S201 data set. 109

List of Tables

1.1 Total amount of genetic information in cells of different species. . 13

1.2 Main characteristics of 20 standard amino acids. 19

1.3 Classification of main protein structures. 23

5.1 Performance of the different predictors in the discrimination task. 91

5.2 Area under the ROC curve (AUC) computed for all the classifiers. 92

5.3 Comparison of the prediction efficiency of the HMM-based methods, based on per-residue indices. 93

5.4 Comparison of the prediction efficiency of the HMM-based methods, based on per-segment indices. 93

5.5 Comparison of the prediction efficiency of the HMM-based methods, based on per-protein indices. 93

5.6 Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-residue indices. 94

5.7 Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-segment indices. 94

5.8 Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-protein indices. 94

6.1 Non hierarchical clustering of S599 protein sequences. 104

6.2 From sequence to function: mapping of molecular function GO terms into the S599 clusters .*The total number of clusters of S599 is 187,594. The total sum of protein sequences of S599 is 2,624,555; the total number of singletons is 660,851. Direct annotation: sequences with at least a GO term. Inherited annotation: sequences that inherit GO term/s in the cluster. Changed annotation: sequences that after statistical validation, change functional annotation with respect to Uniprot. 105

6.3 From sequence to structure: mapping of PDB structures into the S599 clusters. *,° see Table 6.2; #The PDB structure is monodomain or multi-domain according to the SCOP classification. . . 106

6.4 Template coverage over the clusters. *The total number of clusters of S599 with at least a PDB template is 5,734 for a total sum of 596,411 protein sequences. Direct annotation: sequence with a PDB structure; Inherit annotation: sequences that inherit the structure in the cluster. +PDB: with at least a PDB template in the cluster; the PDB structure is mono-domain or multi-domain according to the SCOP classification. 107

6.5 From sequence to structure and function. *The total number of clusters of S599 is 187,594 for a total sum of 1,963,704 protein sequences. Direct annotation and inherited annotation is as in Table 2, 3 and 4, depending on the annotation type. Three different annotations are possible: GO and PDB; GO without PDB; PDB without GO. The rightmost bottom corner contains the number of clusters and sequences that is without annotation. 108

Introduction

The vast majority of known proteins have not yet been experimentally characterized and little is known about their function. The design and implementation of computational tools can provide insight into the function of proteins based on their sequence, their structure, their evolutionary history and their association with other proteins.

Knowledge of the three-dimensional (3D) structure of a protein can lead to a deep understanding of its mode of action and interaction, but currently the structures of <1% of sequences have been experimentally solved. For this reason, it became urgent to develop new methods that are able to computationally extract relevant information from protein sequence and structure. The starting point of my work has been the study of the properties of contacts between protein residues, since they constrain protein folding and characterize different protein structures. Prediction of residue contacts in proteins is an interesting problem whose solution may be useful in protein folding recognition and *de novo* design. The prediction of these contacts requires the study of the protein inter-residue distances related to the specific type of amino acid pair that are encoded in the so-called *contact map*. An interesting new way of analyzing those structures came out when network studies were introduced, with pivotal papers demonstrating that protein contact networks also exhibit small-world behavior. In order to highlight constraints for the prediction of protein contact maps and for applications in the field of protein structure prediction and/or reconstruction from experimentally determined contact maps, I studied to which extent the characteristic path length and clustering coefficient of the protein contacts network are values that reveal characteristic features of protein contact maps.

Provided that residue contacts are known for a protein sequence, the major features of its $3D$ structure could be deduced by combining this knowledge with correctly predicted motifs of secondary structure. In the second part of my work I focused on a particular protein structural motif, the coiled-coil, known to mediate a variety of fundamental biological interactions. Coiled-coils are found in a variety of structural forms and in a wide range of proteins including, for example, small units such as leucine zippers that drive the dimerization of many transcription factors or more complex structures such as the family of viral proteins responsible for virus-host membrane fusion. The coiled-coil structural motif is estimated to account for 5-10% of the protein sequences in the various genomes.

Given their biological importance, in my work I introduced a Hidden Markov Model (HMM) that exploits the evolutionary information derived from multiple sequence alignments, to predict coiled-coil regions and to discriminate coiled-coil sequences. The results indicate that the new HMM outperforms all the existing programs and can be adopted for the coiled-coil prediction and for large-scale genome annotation.

Genome annotation is a key issue in modern computational biology, being the starting point towards the understanding of the complex processes involved in biological networks. The rapid growth in the number of protein sequences and structures available poses new fundamental problems that still deserve an interpretation. Nevertheless, these data are at the basis of the design of new strategies for tackling problems such as the prediction of protein structure and function. Experimental determination of the functions of all these proteins would be a hugely time-consuming and costly task and, in most instances, has not been carried out. As an example, currently, approximately only 20% of annotated proteins in the *Homo sapiens* genome have been experimentally characterized. A commonly adopted procedure for annotating protein sequences relies on the “inheritance through homology” based on the notion that similar sequences share similar functions and structures. This procedure consists in the assignment of sequences to a specific group of functionally related sequences which had been grouped through clustering techniques. The clustering procedure is based on suitable similarity rules, since predicting protein structure and function from sequence largely depends on the value of sequence identity. However, additional levels of complexity are due to multi-domain proteins, to proteins that share common domains but that do not necessarily share the same function, to the finding that different combinations of shared domains can lead to different biological roles. In the last part of this study I developed and validate a system that contributes to sequence annotation by taking advantage of a validated transfer through inheritance procedure of the molecular functions and of the structural templates. After a cross-genome comparison with the BLAST program, clusters were built on the basis of two stringent constraints on sequence identity and coverage of the alignment. The adopted measure explicity answers to the problem of multi-domain proteins annotation and allows a fine grain division of the whole set of proteomes used, that ensures cluster homogeneity in terms of sequence length. A high level of coverage of structure templates on the length of protein sequences within clusters ensures that multi-domain proteins when present can be templates for sequences of similar length. This annotation procedure includes the possibility of reliably transferring statistically validated functions and structures to sequences considering information available in the present data bases of molecular functions and structures.

All the projects of this thesis have been developed in the Bologna Biocomputing Group under the direction of Prof. Rita Casadio and the supervision of Dr. Piero Fariselli and Dr. Pier Luigi Martelli.

Chapter 1

Proteins, proteomes and genomes

1.1 From genome to proteome

The genome of an organism is its whole hereditary information and it is encoded in the DNA. The total amount of genetic information per cell, namely the sequence of nucleotides of DNA, is nearly constant for all members of a species but it varies widely between species (Table 1.1). Different pattern of proteins also characterize different cells, therefore the amount of protein sequence information in a cell, cannot be estimated from the genome size.

1.1.1 Nucleic acids

Nucleotides are the subunits of DNA. A nucleotide consists of a nitrogenous heterocyclic base, which is either a purine or a pyrimidine, a pentose sugar and a phosphate group. A nucleoside is the group formed by the pentose sugar and the phosphate group. The sugar can be either ribose or deoxyribose and it carries one or more phosphate groups. Nucleotides containing ribose are known as ribonucleotides, and those containing deoxyribose as deoxyribonucleotides. The nitrogen-containing rings are generally referred to as bases for historical reasons:

Table 1.1. : Total amount of genetic information in cells of different species.

Organism	**Genome size**
	(base pairs)
Epstein-Barr virus	0.172×10^6
Bacterium (E.Coli)	4.6×10^6
Yeast (S. cerevisiae)	12.5×10^6
Nematode worm (C. elegans)	100.3×10^6
Thale cress (A. thaliana)	115.4×10^6
Fruit fly (D. melanogaster)	128.3×10^6
Human (H. sapiens)	$3,223 \times 10^6$

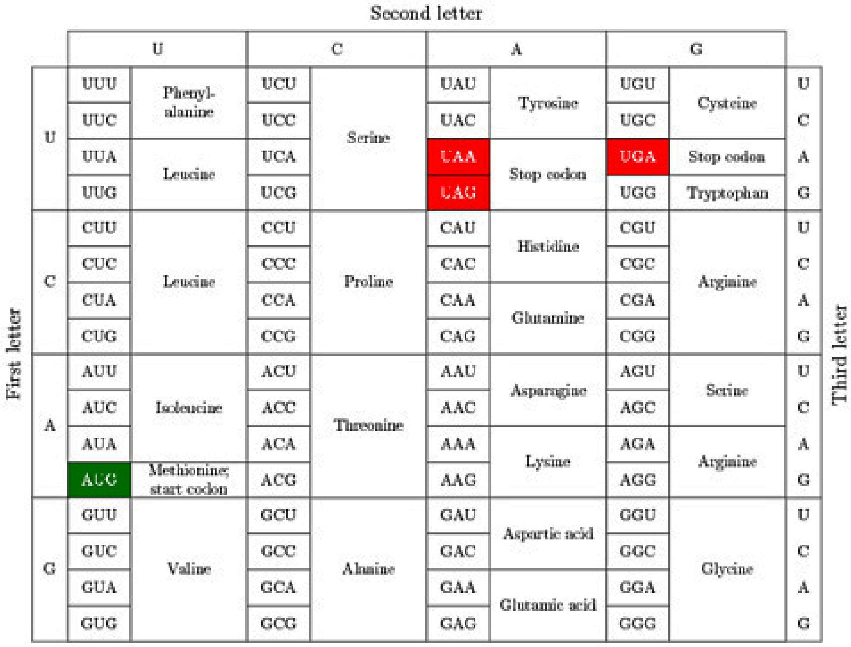
under acidic conditions they can each bind a proton H^+ and thereby increase the concentration of OH^- ions in aqueous solution. There is a strong family resemblance between the different bases. Cytosine (C), thymine (T), and uracil (U) are called pyrimidines because they all derive from a six-membered pyrimidine ring; guanine (G) and adenine (A) are purine compounds, and they have a second, five-membered ring fused to the six-membered ring.

Nucleotides can act as short-term carriers of chemical energy but their most fundamental role in the cell is in the storage and retrieval of biological information. Nucleic acid chains are synthesized from energy-rich nucleoside triphosphates by a condensation reaction that releases inorganic pyrophosphate during phosphodiester bond formation. There are two main types of nucleic acids, differing in the type of sugar in their sugar-phosphate backbone. Those based on the sugar ribose are known as ribonucleic acids, or RNA, and contain the bases A, G, C, and U. Those based on deoxyribose are known as deoxyribonucleic acids, or DNA, and contain the bases A, G, C, and T (T is chemically similar to the U in RNA, merely adding the methyl group on the pyrimidine ring). RNA usually occurs in cells in the form of a single polynucleotide chain, but DNA is virtually always in the form of a double-stranded molecule, the *DNA double-helix*, composed of two polynucleotide chains running antiparallel to each other and held together by hydrogen-bonding between the bases of the two chains.

The linear sequence of nucleotides in the DNA encodes the genetic information of the cell. The ability of the bases in different nucleic acid molecules to recognize and pair with each other by hydrogen-bonding (called base-pairing), namely G with C and A with either T or U, underlies all of heredity and evolution.

1.1.2 The genetic code

The DNA in genomes does not direct protein synthesis itself, but instead uses RNA as an intermediary molecule. When the cell needs a particular protein, the nucleotide sequence of the appropriate portion of the DNA molecule in a chromosome is first copied into RNA (a process called transcription). In particular, mRNA is the RNA that carries information from DNA to the ribosome sites of protein synthesis in the cell. These RNA copies of segments of the DNA are used directly as templates to direct the synthesis of the protein (a process called translation). The flow of genetic information in cells is therefore from DNA to RNA to protein. All cells, from bacteria to humans, express their genetic information in this way, a principle so fundamental that it is termed *the central dogma of molecular biology*. The correspondence between the basis of the DNA and the amino acids is named genetic code (Fig.1.1). Genetic code is universal, with very few exceptions. During protein synthesis, the genetic code interprets the information contained in the DNA to determine the sequence of the amino acids of the protein. The code defines a mapping between tri-nucleotide sequences, called *codons*, and amino acids. Every triplet of nucleotides in a nucleic acid sequence specifies a single amino acid.



		Second letter							
	U		C		A		G		
First letter U	UUU	Phenyl-alanine	UCU	Serine	UAU	Tyrosine	UGU	Cysteine	U Third letter
	UUC		UCC		UAC		UGC		C
	UUA	Leucine	UCA		UAA	Stop codon	UGA	Stop codon	A
	UUG		UCG		UAG		UGG	Tryptophan	G
C	CUU	Leucine	CCU	Proline	CAU	Histidine	CGU	Arginine	U
	CUC		CCC		CAC		CGC		C
	CUA		CCA		CAA	Glutamine	CGA		A
	CUG		CCG		CAG		CGG		G
A	AUU	Isoleucine	ACU	Threonine	AAU	Asparagine	AGU	Serine	U
	AUC		ACC		AAC		AGC		C
	AUA		ACA		AAA	Lysine	AGA	Arginine	A
	AUG	Methionine; start codon	ACG		AAG		AGG		G
G	GUU	Valine	GCU	Alanine	GAU	Aspartic acid	GGU	Glycine	U
	GUC		GCC		GAC		GGC		C
	GUA		GCA		GAA	Glutamic acid	GGA		A
	GUG		GCG		GAG		GGG		G

Fig. 1.1. : The standard genetic code.

Genetic code is degenerate since it is formed by 64 codons specifying 20 amino acids. Three codons do not specify any amino acid but act as termination sites (stop codons), signaling the end of the protein-coding sequence. One codon, AUG, acts both as an initiation codon, signaling the start of a protein-coding message, and also as the codon that specifies the amino acid methionine (Fig.1.1).

1.1.3 Genome complexity

A major open challenge in molecular biology is understanding what the genome contains and how the genome functions. A genome sequence is not an end in itself. Once a DNA sequence has been obtained, whether it is the sequence of a single cloned fragment or of an entire chromosome, then various methods can be employed to locate the genes that are present. These methods can be divided into those that involve the inspection of the sequence, manually or automated, to look for the special sequence features associated with genes, and those methods that locate genes by experimental analysis of the DNA sequence. Then, the question of the gene function has to be addressed. This is turning out to be an important area of genomics research, because completed sequencing projects have revealed that we know rather less than we thought about the content of individual genomes. Although comparative analysis of genomes reveals a great deal of information about the relationships between genes and organisms, it often does not provide immediate information about how these genes function, or what roles they have in the physiology of an organism.

Indeed, even if the organisms whose genomes have been sequenced share many cellular pathways and possess many proteins that are homologous in their amino acid sequences or structure, the functions of a very large number of newly identified proteins remain unknown. For most of the newly determined proteins no primary experimental evidence to annotate is available. Even for the best studied organism, *Escherichia Coli*, experimental information is available for no more than about 60% of the gene products. Furthermore, some 15-40% of the proteins encoded by these sequenced genomes do not resemble any other protein that has been characterized functionally.

1.1.4 Genome databases

Ensembl (`http://www.ensembl.org`) is the universal information source for selected eukaryotic genomes. Data collected in Ensembl include genes, mutations, repeats and homologies. The Ensembl database project provides a bioinformatics framework to organise biology around the sequences of large genomes [Hubbard *et al.*, 2002].

The National Center for Biotechnology Information (NCBI) has a genome resource (`http://www.ncbi.nlm.nih.gov/Genomes/`) for the eukaryotic, fungi, insects, mammals, plants and microbial genomes. At the NCBI is also hosted the Reference Sequence (RefSeq, [Pruitt *et al.*, 2007]) database, a non-redundant collection of richly annotated DNA, RNA and protein sequences from diverse taxa. The collection includes sequences from plasmids, organelles, viruses, archaea, bacteria, and eukaryotes.

These databases are mantained with the aim of providing a comprehensive and standard dataset that represents sequence information for the different species.

1.2 Proteins and proteomes

At the protein level, large-scale analysis of complete genomes has its counterpart in what it has become the *proteome* analysis. The proteome is the set of proteins of an organism. An organism's genome gives a complete but static set of specifications of the potential life of an individual. Nevertheless, the state of development of the organism, as its activity at the molecular level at any time, depend mainly on the amount and on the distribution of its proteins.

Proteins are by far the most structurally complex and functionally sophisticated molecules known. They are not only the building blocks from which cells are built but they also execute nearly all cell functions: enzymes promote many chemical reactions, other proteins carry messages from one cell to another, specialized proteins act as antibodies, toxins, hormones. In a polar solvent proteins spontaneousely assume a stable and active three-dimensional (3D) structure, called *native* structure. The process by which the unstructured string of amino acids acquires its correct three-dimensional structure to achieve the biologically active native state is called protein *folding*. The observation that the shape of a protein is specified by its amino acid sequence is at the basis of the Anfinsen thermodynamic hypothesis [Anfinsen, 1973] that states that for each natural amino acid

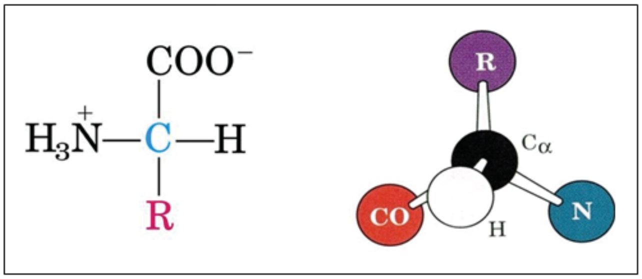


Fig. 1.2. : General structure of an amino acid.

sequence, there is a unique stable native state that, under proper conditions, is adopted spontaneously. Since it appears that nature has an algorithm for predicting protein structures from amino acid sequence, a major issue in molecular biology is then the determination of the three-dimensional structure of a protein, starting from its primary structure and to understand the basic mechanisms of the folding process.

1.2.1 Protein structure

Amino acids. From a chemical point of view proteins are linear hetero-polymers of simpler organic molecules, the amino acids. There are 20 types of amino acids in proteins, each with different chemical properties but with a common chemical structure: a central carbon atom (C_α) to which are attached a hydrogen atom, an amino group (NH_2) and a carboxyl group ($COOH$) (Fig.1.2). The peculiar feature of each amino acid is encoded in its residue (R) or side chain (Fig.1.2 and Fig.1.3) that determines the physico-chemical properties of the molecule. The only exception to the structure depicted in Fig.1.2 is the amino acid Proline, whose side chain is closed on the nitrogen of the amino group. The distinctive features of the 20 side chains are size, electrical charge, polarity and shape (Table 1.2). The smallest amino acid, glycine, consists of only a hydrogen atom, while one of the largest, phenylalanine, contains a benzene ring. Some side chains bear a net positive or negative charge at normal pH. Asp and Glu are negatively charged. Lys, Arg and often His are positively charged. Some side chains are polar, so that they can form hydrogen bonds to other polar side chains, or to the backbone, or to water. Other side chains are electrically neutral. Some of these contain chemical groups related to ordinary hydrocarbons such as methane or benzene. The overall shape of a side chain depends on its chemical structure and on its degrees of internal conformational freedom.

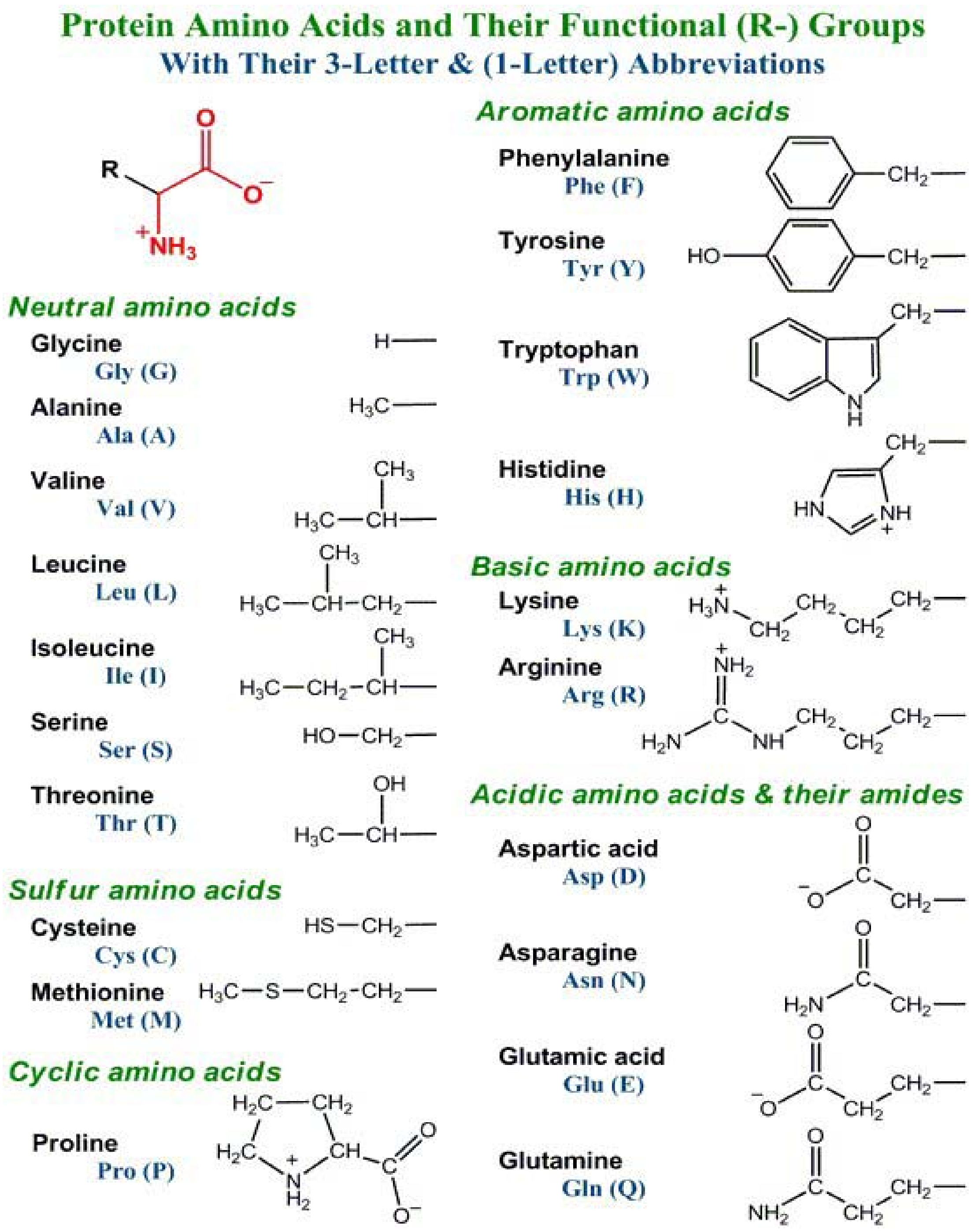


Fig. 1.3. : The 20 amino acids.

Table 1.2. : Main characteristics of 20 standard amino acids.

Amino acid	**One-letter code**	**Three-letters code**	**Polarity at pH 7**	**pK of side chain**
Alanine	A	Ala	Not polar	-
Cysteine	C	Cys	Polar	-
Aspartic acid	D	Asp	Charged (-)	3.9
Glutamic acid	E	Glu	Charged (-)	4.3
Phenylalanine	F	Phe	Not polar	-
Glycine	G	Gly	Not polar	-
Histidine	H	His	Polar	6.0
Isoleucine	I	Ile	Not polar	-
Lysine	K	Lys	Charged (+)	10.5
Leucine	L	Leu	Not polar	-
Methionine	M	Met	Not polar	-
Asparagine	N	Asn	Polar	-
Proline	P	Pro	Not polar	-
Glutamine	Q	Gln	Polar	-
Arginine	R	Arg	Charged (+)	12.5
Serine	S	Ser	Polar	-
Threonine	T	Thr	Polar	-
Valine	V	Val	Not polar	-
Tryptophan	W	Trp	Not polar	-
Tyrosine	Y	Tyr	Polar	10.1

Peptide bond and primary structure. Each amino acid is linked to its neighbour through a covalent peptide bond, where the carboxyl group of one amino acid condenses with the amino group of the next to eliminate water (Fig.1.4). This process is repeated as the chain elongates. One consequence is that the amino group of the first amino acid of a polypeptide chain and the carboxyl group of the last amino acid remain intact, and the chain is said to extend from its amino terminus (*N-terminus*) to its carboxy terminus (*C-terminus*). Proteins are therefore also known as *polypeptides.* The peptide bond is dipolar and has double bond character. The repeating sequence of C_α atoms along the core of the polypeptide chain is referred to as the protein *backbone*. The primary structure of the protein is direct expression of the information encoded in the DNA and is represented by a sequence of characters (a string) each one representing an amino acid. Accordingly to the Anfinsen's hypothesis, this sequence contains all the information about protein three-dimensional structure.

Secondary structure. The folding of a protein is constrained by many different *weak non-covalent bonds* that form between one part of the chain and

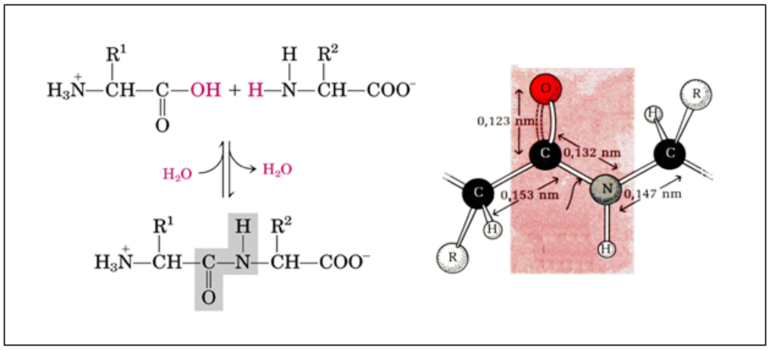


Fig. 1.4. : Formation and charactersitics of the peptide bond.

another. These involve atoms in the polypeptide backbone as well as atoms in the amino acid side chains. The weak bonds are of three types: hydrogen bonds, ionic bonds and van der Waals attractions. Individual non-covalent bonds are 30-300 times weaker than the typical covalent bonds that create biological molecules. But many weak bonds can act in parallel to hold two regions of a polypeptide chain tightly together.

A fourth weak force also has a central role in determining the shape of a protein: the distribution of its polar and non-polar amino acids. Hydrophobic molecules, including the non-polar side chains of particular amino acids, tend to be forced together in an aqueous environment in order to minimize their disruptive effect on the hydrogen-bonded network of the surrounding water molecules in the cell. In particular, hydrophobic side chains (belonging to Phe, Leu, Val and Trp, for example) tend to cluster in the interior of the molecule. This enables them to avoid contact with the water. In contrast, polar side chains (such as those belonging to Arg, Gln and His) tend to be exposed to the solvent, where they can form hydrogen bonds with water and with other polar molecules.

There is a major problem, however, with creating such a hydrophobic core from a protein chain. To bring the side chains into the core, the main chain must also fold into the interior. The main chain is highly polar and therefore hydrophilic, with one hydrogen bond donor, NH, and one hydrogen bond acceptor, C=O, for each peptide unit. In a hydrophobic environment, these main chain polar groups must be neutralized by the formation of hydrogen bonds. This problem is solved in a very elegant way by the formation of structural and local organization motifs. This is the regular secondary structure within the interior of the protein molecule. Such secondary structure is usually one of two types: α-helices or β-sheets and they constitute about the 50% of the structure of each protein. Most protein structures are built up from combinations of α-helices or β-sheets which are connected by loop regions of various lengths and irregular shape. Loop regions exposed to solvent are rich in charged and polar hydrophilic residues. Helices and sheets are characterized by hydrogen-bonding between the main chain

NH and C=O groups. The secondary structure elements, formed in this way and held together by the hydrophobic core, provide a rigid and stable framework. They exhibit relatively little flexibility with respect to each other, and they are the best defined parts of protein structures determined both by X-ray and NMR techniques.

Tertiary structure. The secondary structure elements are organized into a complex three-dimensional structure, called tertiary structure,which is stable and functional. The tertiary structure is the result of the folding process. Therefore, it depends only on the interactions among the amino acids and between the amino acids and the solvent, described above.

The evolution of living organisms selects polypeptide chains with the ability to acquire stable conformations in the aqueous or lipid environment where they perform their function. The universe of protein sequences can be compared in its entirety across species.

1.2.2 The protein archives

The knowledge about biological data is very far from complete. Nevertheless, it is of impressive size and it is constantly and rapidly growing. For this reason, information about biological molecules is generally collected into integrated databases publicy available through the World Wide Web.

Protein sequences databases. Since 2003, the Swiss Institute of Bioinformatics (SIB) and the Department of Bioinformatics and Structural Biology of the Geneva University, the European Bioinformatics Institute (EBI) and the Georgetown University Medical Center's Protein Information Resource (PIR) have coordinated their efforts in the UniProt consortium ([Apweiler *et al.*, 2004]). UniProt/Swiss-Prot, together with UniProt/TrEMBL, its computer-annotated supplement, constitutes the UniProt Knowledgebase (UniProtKB), a major project of the UniProt consortium. UniProt/Swiss-Prot and UniProt/TrEMBL give access to all the publicly available protein sequences. The UniProt/SwissProt database is accessible through the web site `http://www.expasy.org/sprot/`. The UniProt Knowledgebase contains proteins sequences and its peculiar characteristics are:

- *Annotation.* The protein sequence and the current information about each protein are manually checked and regularly updated. Each entry contains primary data (for example the sequence and the description of the biological source of the protein, together with the related literature) and annotation, which consists of the description, when available, of the function/s, the post-translational modifications, the domains and active sites (calcium binding regions, ATP-binding sites), the secondary and quaternary structure (such as homo-dimer, hetero-trimer), the similarities to other proteins and the sequence conflicts or variants of the protein sequence. A further annotation field concerns the indications of the diseases associated with deficiencies in the protein.

- *Minimal redundancy.* In order to have a minimal redundancy of the information all protein sequences encoded by a same gene are merged into a single UniProt entry. However, the differences between various sequencing reports are reported.
- *Integration.* External databases referencing the entry are available via cross-references to specialized data collections such as nucleotide sequence databases, 3D structure database, protein domain and family characterization databases. UniProtKB/Swiss-Prot is currently cross-referenced with about 60 different databases.

Databases of structures. The major database for biological macromolecular structures is the Protein Data Bank (PDB, [Berman HM *et al.*, 2000]), the result of the effort of a distributed organization called Research Collaboratory for Structural Bioinformatics (RCSB). It is freely accessible at `http://www.pdb.org/`. The PDB stores, annotates and distributes sets of atomic coordinates. The RCSB PDB together with the Molecular Structure Database at the European Bioinformatics Institute (EBI) and the Protein Data Bank Japan hosted at the Osaka University, is a member of the worldwide PDB (wwPDB, [Berman HM *et al.*, 2003]) that mainly integrates all the structures of proteins but also of nucleic acids and few carbohydrates known with atomic resolution. For proteins, each PDB entry contains not only atomic coordinates for all the atoms of the molecule but also related information, such as the protein description, the literature references, experimental details about the structure determination, the residues sequence, if additional molecules appear in the structure, the assignments of secondary structure. Furthermore, the PDBsum database has been developed ([Laskowski *et al.*, 1997], `http://www.ebi.ac.uk/pdbsum/`) to provide an overview of every protein structure deposited in the PDB.

Classification of protein structures. Classification of protein structures occupies a key position in bioinformatics, not least as a bridge between sequence and function. The most general classification of families of protein structures, presented in Table 1.3, is based on the secondary and tertiary structures of proteins. Several web sites offer hierarchical classifications of all proteins of known structure according to their folding patterns. Among them SCOP (Structural Classification of Proteins, `http://iris.physics.iisc.ernet.in/scop/`) [Murzin *et al.*, 1995] and CATH (Class, Architecture, Topology, Homologous superfamily, `http://www.cathdb.info/`) [Orengo *et al.*, 1997] are the most important.

SCOP is widely adopted because it is manually curated and because it organizes protein structures in a hierarchy according to evolutionary origin and structrural similarity. At its lowest level are individual *domains*,

Table 1.3. : Classification of main protein structures.

Class	**Characteristic**
α-helical	Secondary structure exclusively (or almost) α-helical
β-sheet	Secondary structure exclusively (or almost) β-sheet
$\alpha + \beta$	α-helices and β-sheets separated in different parts of the molecule
α/β	Helices and sheets assembled from $\beta - \alpha - \beta$ units
Proteins with little or no secondary structure	

extracted from the PDB entries. Domains are compact units within the folding pattern of a single protein chain, that look as if they should have independent stability. Sets of domains are grouoped into *families* of homologues, for which the similarities in structure, sequence and sometimes function imply a common evolutionary origin. Groups of families containing proteins of similar structure and function, but for which the evidence for an evolutionary relationship is suggestive but not compelling, form *superfamilies*. Superfamilies that share a common folding topology, for at least the central proportion of the structure, are grouped as *folds*. Finally, each fold group falls into one of the general *classes*. In Table 1.3 the most general classes of protein structures are detailed.

Classification of protein function. The Gene Ontology Consortium has produced a systematic classification of gene function, in the form of a dictionary of terms and their relationships (`http://www.geneontology.org`, [The Gene Ontology Consortium, 2000]). Organizing concepts of the Gene Ontology project include three categories:

- *Molecular function.* A function associated with what an individual protein does in itself. This is function from the biochemical point of view.
- *Biological process.* A component of the activities of a living system, mediated by a protein or RNA, possibly in a concerted action with other proteins or RNA molecules. This is function from the cell's point of view.
- *Cellular component.* The assignment of site of activity or partners.

Chapter 2

Analysis and prediction of protein structure and function

Up to now, the structure of about 56,000 proteins have been solved, while more than 7 millions are the protein sequences deposited in the UniProt database. The difference among the number of solved structures and the number of known sequences is even greater if we consider that a number of structures in the PDB belong to the same protein, solved in different environments and with different resolutions. Analyzing the growth of PDB and SwissProt entries in the years (Fig. 2.1), it can also be observed that their difference is still increasing.

The development of computational predictive tools has it basis in this difference: a protein functions because of its structure and all applications, from the drug design to the study of single point mutations, require structural infor-

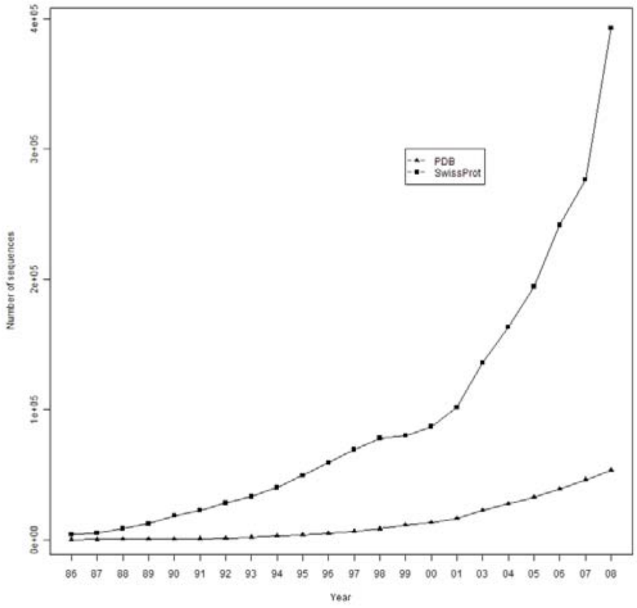


Fig. 2.1. : Growth in the content of PDB and SwissProt databases from 1986 to 2008.

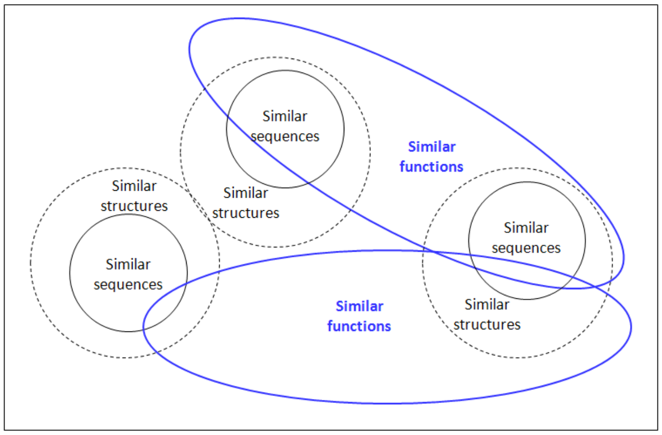


Fig. 2.2. : Relationships among protein sequence, structure and function.

mation. Since the vast majority of protein sequences produced by the various genome projects has not yet been experimentally characterized and since there is very little that is known about their function, these sequences need to be interpreted from a structural point of view (*structural genomics*) and from the point of view of the functions (*functional genomics*).

The cascade of inference should ideally flow from sequence to structure to function. However, although we can be confident that similar amino acids sequences will produce similar protein structures [Sander and Schneider, 1991], the relationship between structure and function is more complex (Fig.2.2):

- *Similar sequences produce similar protein structures*, with divergence in structure increasing progressively with the divergence in sequence.

- Conversely, *similar structures are often found with very different sequences.* In many cases the relationships in a family of proteins can be detected only in structures, the sequences having diverged beyond the point of our being able to detect the underlying common features.

- *Similar sequences and structures often produce proteins with similar functions*, but there are exceptions.

- At the contrary, *similar functions are often carried out by non-homologous proteins with dissimilar structures.*

It is in this perspective that computational analyis and studies offer tools that can provide insight into the function of proteins based on their sequence, their structure, their evolutionary history and their association with other proteins.

2.1 Protein structure prediction and modelling

Proteins live and function in three dimensions and therefore structural information is very helpful for predicting function. Structural information can come directly from the protein of interest but it can also be derived from a homologous protein via modelling. Unfortunately, as it is for sequences, two proteins having the same overall structural architecture, and even conserved functional residues can have unrelated functions. In addiction, two proteins can perform the same function while having radically different structures [Whisstock and Lesk, 2003].

The amino acids sequence of a protein dictates its three-dimensional structure. The functions of proteins depend on their adopting this native three-dimensional structure. This means that proteins have evolved so that one folding pattern of the main chain, the native structure, is thermodynamically significantly better than other conformations. Thus, if we could computationally analyze a large enough set of possible conformations to be sure of including the correct one, it would be possible to predict protein structures from amino acids sequences on the basis of *a priori* physico-chemical properties. There has been progress towards this goal but it is not yet achieved.

Systematic studies of the structural differences between pairs of related proteins have defined a quantitative relationship between the divergence of the amino acid sequences of the core of a family of structures and the divergence of their structure [Chothia and Lesk, 1986]. As the sequence diverges, there are progressively increasing distortions in the main chain conformation and the fraction of the residues in the core usually decreases. Until the fraction of identical residues in the sequence drops below about 40%-50% these effects are relatively modest. Almost all the structure remains in the core and the deformation of the main chain atoms is on average no more than 1.0 Å. When the sequence divergence increases, in most cases some regions entirely refold, the size of the core reduces and the distortions result in a greater effect. The variation of the size of the core with the percentage of identical residues is shown in Figure 2.3. In Figure 2.4 the changes in structure of the core, expressed as the root mean square deviation of the main chain atoms after optimal structural superimpostion are plotted against the sequence divergence, expressed as the percentage of conserved amino acids in the core after optimal sequence alignment.

Many of the most effective methods for protein structure prediction exploit known structures of homologous proteins. As detailed above, the level of sequence similarity between a protein of unknown structure and its nearest homologue with known structure limits the degree of information we can produce for the protein with unknown structure and poses constraints on the method to be adopted. Generally speaking:

1. If a protein of unknown structure has homologues of known structure with $\geq$40% identical residues in an optimal alignment, *homology modelling* methods are likely to produce a nearly complete structural model and the quality of the model is likely to be good enough to interpret the protein's function.

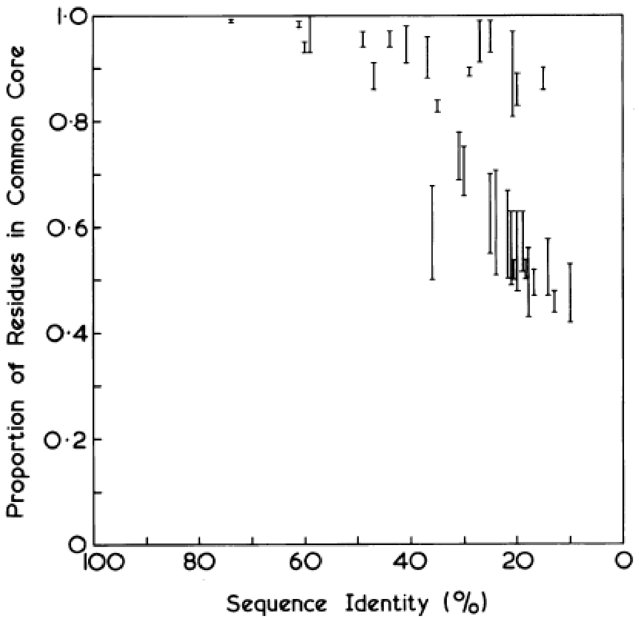


Fig. 2.3. : Variation of size of the protein core with the percentage of its identical residues [Chothia and Lesk, 1986].

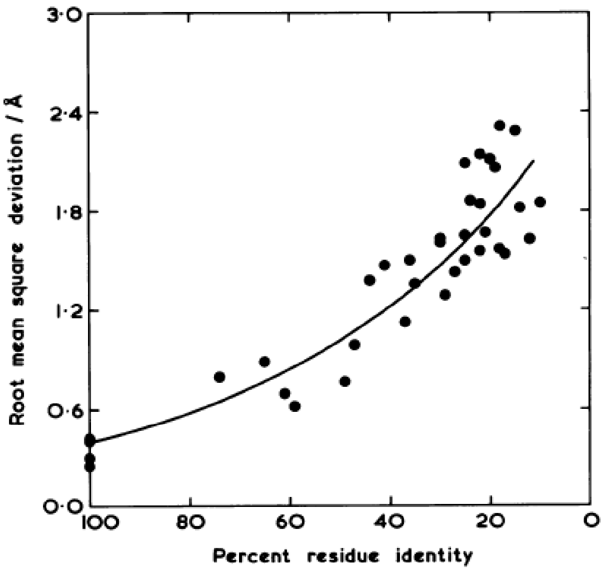


Fig. 2.4. : Variation of r.m.s deviation of the protein core with the percentage of its identical residues. The figure shows computed results for 32 pairs of homologous proteins of different structural types [Chothia and Lesk, 1986].

2. If no homologue with known structure has the sequence similarity to the unknown one with $\geq$40%, it may still be possible to assign a general folding pattern to the protein of unknown structure. It should be possible to predict its secondary structure with $\approx$ 70%-80% accuracy on a residue by residue basis.

3. If no homologue of known structure is recognizable from the sequences, the last solution is to use a prediction method general enough to handle novel folds. Such methods include both *a priori* and knowledge-based approaches.

Summing up, methods for the prediction of protein structure from amino acid sequence include:

- Attempts to predict secondary structure

- *Homology modelling*, the prediction of the three-dimensional structure of a protein from the known structures of one or more related proteins

- *Fold recognition*. Given a library of known structures, determine which of them shares a folding pattern with a query protein of known sequence but unknown structure.

- *Prediction of novel folds*, either by *a priori* and knowledge-based approaches.

Contacts between protein residues constrain protein folding and characterize different protein structures. Therefore, prediction of residue contacts in proteins, discussed in Chapter 4, is an interesting problem whose solution may be useful in protein folding recognition and *de novo* design. Furthermore, some of the most powerful sequence to structure predictors involve HMMs whose theory and applications to secondary structure prediction will be presented in Chapter 3 and 5, respectively.

2.2 Sequence-based protein function prediction

While the number of sequenced genomes continues to grow, experimentally verified functional annotation of whole genomes remains a callenging problem. There are now more than 800 completely sequenced genomes of cellular organisms, contributing to more than seven million unique protein sequences in the publicly accessible databases, such as UniProt. Experimental determination of the functions of all these proteins would be a hugely time-consuming and costly task and, in most instances, has not been carried out. Currently, approximately about 20%, 7%, 10% and 1% of annotated proteins in the *Homo sapiens*, *Mus musculus*, *Drosophila melanogaster* and *Caenorhabditis elegans* genomes, respectively, have been experimentally characterized. However, as the volume of data has increased, so too have the number and sophistication of computational methods for predicting function. Knowledge of the three-dimensional structure of a protein can also provide a crucial insight into its mode of action, but currently the structures of <1% of sequences have been experimentally solved.

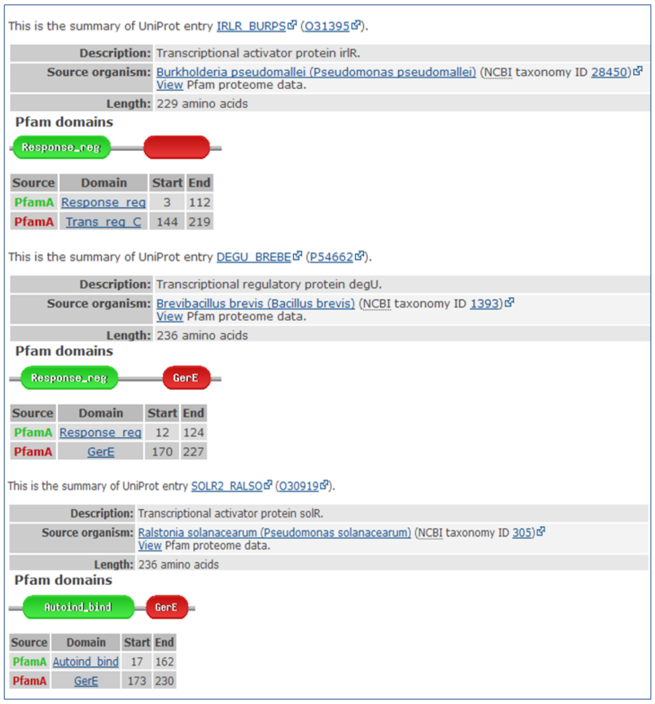
Computational methods that exploit sequence and structural data using computational means alone exist to predict protein function/s. The most common approach to function prediction is the *inheritance through homology* that is based on the observation that proteins with similar sequences frequently perform similar functions. Since there is no perfect and general rule that leads from a protein sequence to ist correct function, the design of computational methods that achieve high degree of accuracy in this task became very urgent, especially with the increase in the number of protein sequenced that still lack a functional and structural characterization.

A first step, common to a number of methods to predict the function of a given protein sequence is to use the information provided by the main publicly available databases, such as those of the National Center for Biotechnology Information (NCBI) and of the European Molecular Biology LaboratoryEuropean Bioinformatics Institute (EMBL-EBI). These resources include manually-curated and automated generated data, including protein, domain and family information and functional sites. One way to relate the databases annotation to a protein sequence is the alignment of the query sequence against the protein sequences from various databases. If an exact match is not found, the search usually identifies a similar sequence from which it may be possible to inherit annotations.

However, many of the incorrect annotations found in databases today are a consequence of the overly liberal application of inheritance through homology and this is compounded by the fact that the source of these annotations is often not given. Estimates of the error rate for the annotation of complete genomes vary from <5% to >40% depending on the types of function ([Brenner SE, 1999, Devos and Valencia, 2001]).

As in the protein structure prediction, there have been many studies aimed at establishing sequence similarity measures for safely transferring function between related proteins. However, genes evolve at different rates owing to both uneven selection pressure on their functions and the inherent mutation rate of different species, which means that it is difficult to establish a similarity measure that is reliable in all cases. As it will described in Chapter 6, many new family-based resources have emerged over the past ten years that group together protein sequences or individual protein domains into putative evolutionary families across many different sequenced genomes. These family resources make it easier to gauge the reliability of functional inheritance through homology.

Nevertheless, although higher sequence similarity increases confidence in function annotation transfer, there is no threshold that can be considered universally safe. An extreme case is represented by the so-called “moonlighting proteins”, which are proteins that perform multiple and, at times, significantly different functions. For example, $\mu-$crystallin is a protein that plays a structural role in the eye lens of several species, while working as an enzyme in other tissues. Homologs of these proteins may retain only some of the original functions. As a



This is the summary of UniProt entry IRLR_BURPS (O31395).

Description:	Transcriptional activator protein irlR.
Source organism:	Burkholderia pseudomallei (Pseudomonas pseudomallei) (NCBI taxonomy ID 28450) View Pfam proteome data.
Length:	229 amino acids

Pfam domains

Source	Domain	Start	End
PfamA	Response_reg	3	112
PfamA	Trans_reg_C	144	219

This is the summary of UniProt entry DEGU_BREBE (P54662).

Description:	Transcriptional regulatory protein degU.
Source organism:	Brevibacillus brevis (Bacillus brevis) (NCBI taxonomy ID 1393) View Pfam proteome data.
Length:	236 amino acids

Pfam domains

Source	Domain	Start	End
PfamA	Response_reg	12	124
PfamA	GerE	170	227

This is the summary of UniProt entry SOLR2_RALSO (O30919).

Description:	Transcriptional activator protein solR.
Source organism:	Ralstonia solanacearum (Pseudomonas solanacearum) (NCBI taxonomy ID 305) View Pfam proteome data.
Length:	236 amino acids

Pfam domains

Source	Domain	Start	End
PfamA	Autoind_bind	17	162
PfamA	GerE	173	230

Fig. 2.5. : Multi-domain proteins problem in three bacterial proteins: O31395 (Transcriptional activator protein irlR), P54662 (Transcriptional regulatory protein degU), O30919 (Transcriptional activator protein solR).

consequence, function annotation transfer may result in erroneous or incomplete assignments.

Furthermore, the multi-domain nature of many proteins can also be the cause of annotation transfer errors. In fact, in databases storing entire sequences (such as SwissProt), functional annotation of a protein may refer to any of its domains. If the analyzed protein does not align to that specific domain, annotation transfer is totally unjustified and will very likely result in a misannotation. While a number of databases and tools attempt to split proteins into domains based on sequence (such as Pfam, a database of protein families, each represented by multiple sequence alignments and Hidden Markov Models (HMMs), [Bateman A *et al.*, 2000]), the most reliable way to identify protein domains is by using, when possible, structural knowledge (by means of SCOP [Murzin *et al.*, 1995] or CATH [Orengo *et al.*, 1997]). Figure 2.5 shows that if the template is annotated based on the function of a domain that is not aligned to the query, annotation transfer is not possible. Coloured boxes represent Pfam domains: [Swiss-Prot:O31395] has a response regulator receiver domain (in green) and a C-terminal transcriptional regulatory domain (in red); [Swiss-Prot:P54662] has the same response regulator receiver domain of [Swiss-Prot:O31395] (in green) and a luxR family response regulator domain (in red); [Swiss-Prot:O30919] shows an autoinducer binding domain (in green) and a luxR family domain (in red), such as [Swiss-Prot:P54662]. The protein [Swiss-Prot:P54662] is similar to each of the two other proteins via a different domain, therefore careless use of transitivity might lead to the false conclusion that [Swiss-Prot:O31395] is similar to [Swiss-Prot:O30919].

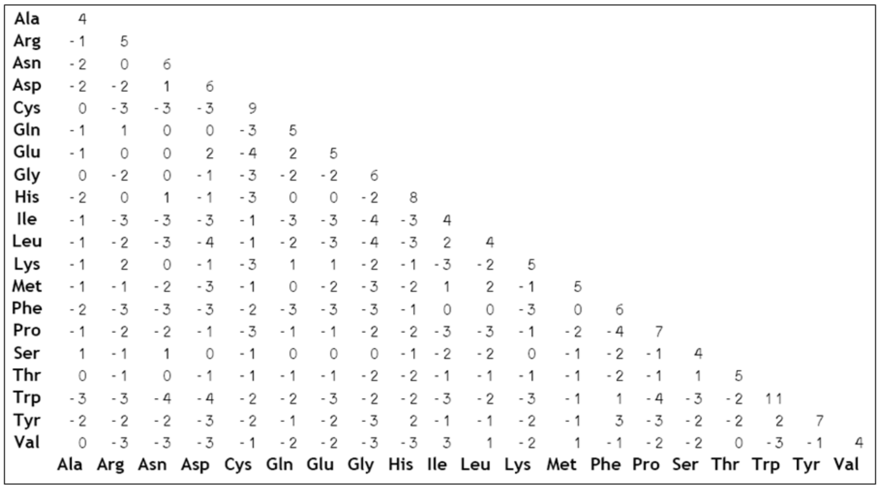
In conclusion, homology between two proteins does not guarantee that they have the same function, not even when sequence similarity is very high. On the positive side, the higher the sequence similarity the better the chance that homologous proteins in fact share functional features.

Although protein structure is more conserved than sequence, knowledge of the specific fold adopted by a given protein does not directly imply a function [Chothia and Lesk, 1986]. With the advent of the structural genomics initiatives, that are complementing the data on which computational methods rely by increasing the functional diversity of protein sequences for which the structure has been determined, an increasing number of protein structures are being experimentally determined while their function is still unknown. In these cases, function can sometimes be predicted by using the structure rather than the sequence of the protein. Nevertheless, the scarcity of experimentally solved protein structures means that most function prediction is carried out by comparing protein sequences, and the recent substantial growth in complete genome sequences is making these methods more powerful. In particular, family-based methods that exploit sequence clustering can be extremely valuable in providing information on the variation in functional properties across a family. For this reason, there is considerable activity today trying to bridge the gap between protein sequence, structure and function. The integration between these different aspects of the analysis of protein structure and function aims to develop better tools for protein function prediction.

2.3 The evolutionary information

The so-called evolutionary information has a key role in the prediction of protein structural and functional features. Some systems, such as the Hidden Markov Model that will be described in Chapter 5, are able to efficiently process this information, improving the predictive performances with respect to those obtained with the sequence alone.

This fact is based on the idea that all the sequences can be separated in classes, named *classes of homology* that group the sequences evolved from a common ancestral sequence. This ancestor sequence, after casual mutations and genetic rearrangements, gave origin to different sequences that structure in proteins with the same three-dimensional conformation and the same function, assuming that the evolution eliminated all the elements not able to assume a function or a native structure. It appears that lot of information can be retrieved by comparing a given sequence of interest against its similar sequences.



Ala	4																			
Arg	-1	5																		
Asn	-2	0	6																	
Asp	-2	-2	1	6																
Cys	0	-3	-3	-3	9															
Gln	-1	1	0	0	-3	5														
Glu	-1	0	0	2	-4	2	5													
Gly	0	-2	0	-1	-3	-2	-2	6												
His	-2	0	1	-1	-3	0	0	-2	8											
Ile	-1	-3	-3	-3	-1	-3	-3	-4	-3	4										
Leu	-1	-2	-3	-4	-1	-2	-3	-4	-3	2	4									
Lys	-1	2	0	-1	-3	1	1	-2	-1	-3	-2	5								
Met	-1	-1	-2	-3	-1	0	-2	-3	-2	1	2	-1	5							
Phe	-2	-3	-3	-3	-2	-3	-3	-3	-1	0	0	-3	0	6						
Pro	-1	-2	-2	-1	-3	-1	-1	-2	-2	-3	-3	-1	-2	-4	7					
Ser	1	-1	1	0	-1	0	0	0	-1	-2	-2	0	-1	-2	-1	4				
Thr	0	-1	0	-1	-1	-1	-1	-2	-2	-1	-1	-1	-1	-2	-1	1	5			
Trp	-3	-3	-4	-4	-2	-2	-3	-2	-2	-3	-2	-3	-1	1	-4	-3	-2	11		
Tyr	-2	-2	-2	-3	-2	-1	-2	-3	2	-1	-1	-2	-1	3	-3	-2	-2	2	7	
Val	0	-3	-3	-3	-1	-2	-2	-3	-3	3	1	-2	1	-1	-2	-2	0	-3	-1	4
	Ala	**Arg**	**Asn**	**Asp**	**Cys**	**Gln**	**Glu**	**Gly**	**His**	**Ile**	**Leu**	**Lys**	**Met**	**Phe**	**Pro**	**Ser**	**Thr**	**Trp**	**Tyr**	**Val**

Fig. 2.6. : The BLOSUM62 matrix.

2.3.1 Protein sequence alignments

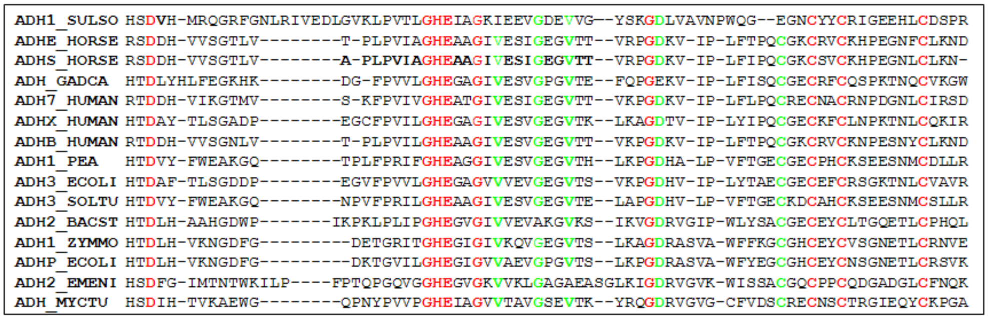
The common procedure to compare protein sequences is the alignment. Two sequences are aligned when they are superimposed such that their residues coincide or are similar in the great part of the positions. Sequence alignment is the identification of residue-residue correspondences. The similarity between two amino acids is quantified in the elements of the *substitution matrix*, symmetric 20×20 matrix that in each position (i, j) contains a score $\sigma(i, j)$ that is relative to the substitution of residue i with residue j. These scores are generally derived from analysis carried out by experts, on well-known protein families and they reflect the similarity of the physico-chemical characteristics of the amino acids. As an example, in Figure 2.6 one of the most used substitution matrix, the BLOSUM62 [Henikoff and Henikoff, 1992], is shown. To avoid overweighting closely related sequences, groups of proteins that have a sequence identity higher than a threshold (62% for BLOSUM62) are replaced either by a single representative or a weighted average.

To perform alignments several algorithms can be implemented. In general a *pairwise alignment* allows to find the superimposition between two sequences s_1 and s_2 that maximizes the global score $S(s_1, s_2)$ defined as the sum of the scores of the substitutions in all the positions:

$$S(s_1, s_2) = \sum_i \sigma(s_1^i, s_2^i) \tag{2.1}$$

An algorithm for *multiple sequence alignment* of N sequences allows to compute the correspondence between the residues of all the sequences that maximizes the score $S(s_1, s_2, ..., s_N)$ defined on the basis of the scores of pairwise alignments:

$$S(s_1, s_2, ..., s_n) = \sum_{k<l} S(s_k, s_l) \tag{2.2}$$



```
ADH1_SULSO HSDVH-MRQGRFGNLRIVEDLGVKLPVTLGHEIAGKIEEVGDEVVG--YSKGDLVAVNPWQG--EGNCYYCRIGEEHLCDSPR
ADHE_HORSE RSDDH-VVSGTLV--------T-PLPVIAGHEAAGIVESIGEGVTT--VRPGDKV-IP-LFTPQCGKCRVCKHPEGNFCLKND
ADHS_HORSE RSDDH-VVSGTLV--------A-PLPVIAGHEAAGIVESIGEGVTT--VRPGDKV-IP-LFIPQCGKCSVCKHPEGNLCLKN-
ADH_GADCA  HTDLYHLFEGKHK--------DG-FPVVLGHEGAGIVESVGPGVTE--FQPGEKV-IP-LFISQCGECRFCQSPKTNQCVKGW
ADH7_HUMAN RTDDH-VIKGTMV--------S-KFPVIVGHEATGIVESIGEGVTT--VKPGDKV-IP-LFLPQCRECNACRNPDGNLCIRSD
ADHX_HUMAN HTDAY-TLSGADP--------EGCFPVILGHEGAGIVESVGEGVTK--LKAGDTV-IP-LYIPQCGECKFCLNPKTNLCQKIR
ADHB_HUMAN RTDDH-VVSGNLV--------T-PLPVILGHEAAGIVESVGEGVTT--VKPGDKV-IP-LFTPQCGKCRVCKNPESNYCLKND
ADH1_PEA   HTDVY-FWEAKGQ--------TPLFPRIFGHEAGGIVESVGEGVTH--LKPGDHA-LP-VFTGECGECPHCKSEESNMCDLLR
ADH3_ECOLI HTDAF-TLSGDDP--------EGVFPVVLGHEGAGVVVEVGEGVTS--VKPGDHV-IP-LYTAECGECEFCRSGKTNLCVAVR
ADH3_SOLTU HTDVY-FWEAKGQ--------NPVFPRILGHEAAGIVESVGEGVTE--LAPGDHV-LP-VFTGECKDCAHCKSEESNMCSLLR
ADH2_BACST HTDLH-AAHGDWP-------IKPKLPLIPGHEGVGIVVEVAKGVKS--IKVGDRVGIP-WLYSACGECEYCLTGQETLCPHQL
ADH1_ZYMMO HTDLH-VKNGDFG---------DETGRITGHEGIGIVKQVGEGVTS--LKAGDRASVA-WFFKGCGHCEYCVSGNETLCRNVE
ADHP_ECOLI HTDLH-VKNGDFG---------DKTGVILGHEGIGVVAEVGPGVTS--LKPGDRASVA-WFYEGCGHCEYCNSGNETLCRSVK
ADH2_EMENI HSDFG-IMTNTWKILP----FPTQPGQVGGHEGVGKVVKLGAGAEASGLKIGDRVGVK-WISSACGQCPPCQDGADGLCFNQK
ADH_MYCTU  HSDIH-TVKAEWG--------QPNYPVVPGHEIAGVVTAVGSEVTK--YRQGDRVGVG-CFVDSCRECNSCTRGIEQYCKPGA
```

Fig. 2.7. : Example of a pairwise alignment performed by the BLAST program.

An alignment, beside describing mutation events on single residues, that is the substitution of a residue by another during the evolution, also takes into account insertions, the insertion of amino acidic fragments in specific positions of the sequence and deletion, the removal of residue fragments in specific positions of the sequence. This is performed by the insertion of symbols of *gap* (indicated with the symbol "-" in Figure 2.7). Alignment algorithms deal with the presence of gaps by adding rules for assigning scores, typically unfavorable, to the gap opening and extention. Nevertheless, the possibility to introduce gaps significantly increases the complexity of the optimal alignment search and it makes thus necessary to adopt *dynamic programming* techniques for computing the pairwise alignments and sub-optimal algorithms to perform multiple alignments.

Pairwise and multiple sequence alignments can be further distinguished in *global* and *local* alignments, if entire sequences or only portions of them are aligned, respectively. Given a set of sequences, generally it exists only one global optimal alignment, while there can be different local alignments that involve different substrings and that can be sorted with respect to their score.

2.3.2 BLAST

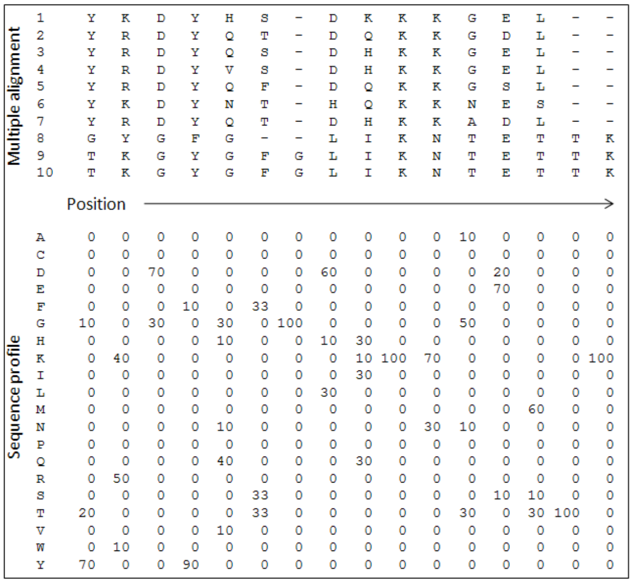
A particular class of systems based on local pairwise sequence alignments are the programs for searching similarities between a target sequence and a database of known sequences. The programs compare the best pairwise local alignments between the target sequence and all the sequences of the reference database.

Given the enormous amount of sequences deposited in the databases, it is impossible to perform these alignments with exact algorithms. For this reason heuristic algorithms have been implemented. Among them, the Basic Local Alignment Search Tool (BLAST) [Altschul *et al.*, 1990] is the most widely adopted. A typical BLAST output is showed in Figure 2.7, where positions that do not undergo mutations are highlighted in red while significantly conserved postions are in green. To measure the statistical significance of the match, namely to test if the found similarity is significant or could have arisen by chance, for each alignment BLAST computes a number of scores. The most important are the *sequence identity*, the length of the alignment and the *Expectation value* (E-value).

The sequence identity (SI) is computed by normalizing the number of identical residues I over the total length of the aligned region L_{aln}:

$$SI = \frac{I}{L_{aln}} \tag{2.3}$$

The E-value is the number of different alignments with scores equivalent to or better than the current one, that are expected to occur in a database search by chance. The lower the E-value, the more significant the match.


Multiple alignment
Position
Sequence profile

1	Y	K	D	Y	H	S	-	D	K	K	K	G	E	L	-	-
2	Y	R	D	Y	Q	T	-	D	Q	K	K	G	D	L	-	-
3	Y	R	D	Y	Q	S	-	D	H	K	K	G	E	L	-	-
4	Y	R	D	Y	V	S	-	D	H	K	K	G	E	L	-	-
5	Y	R	D	Y	Q	F	-	D	Q	K	K	G	S	L	-	-
6	Y	K	D	Y	N	T	-	H	Q	K	K	N	E	S	-	-
7	Y	R	D	Y	Q	T	-	D	H	K	K	A	D	L	-	-
8	G	Y	G	F	G	-	-	L	I	K	N	T	E	T	T	K
9	T	K	G	Y	G	F	G	L	I	K	N	T	E	T	T	K
10	T	K	G	Y	G	F	G	L	I	K	N	T	E	T	T	K

A	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	0
C	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
D	0	0	70	0	0	0	0	60	0	0	0	0	20	0	0	0
E	0	0	0	0	0	0	0	0	0	0	0	0	70	0	0	0
F	0	0	0	10	0	33	0	0	0	0	0	0	0	0	0	0
G	10	0	30	0	30	0	100	0	0	0	0	50	0	0	0	0
H	0	0	0	0	10	0	0	10	30	0	0	0	0	0	0	0
K	0	40	0	0	0	0	0	0	10	100	70	0	0	0	0	100
I	0	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0
L	0	0	0	0	0	0	0	30	0	0	0	0	0	0	0	0
M	0	0	0	0	0	0	0	0	0	0	0	0	0	60	0	0
N	0	0	0	0	10	0	0	0	0	0	30	10	0	0	0	0
P	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Q	0	0	0	0	40	0	0	0	30	0	0	0	0	0	0	0
R	0	50	0	0	0	0	0	0	0	0	0	0	0	0	0	0
S	0	0	0	0	0	33	0	0	0	0	0	0	10	10	0	0
T	20	0	0	0	0	33	0	0	0	0	0	30	0	30	100	0
V	0	0	0	0	10	0	0	0	0	0	0	0	0	0	0	0
W	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Y	70	0	0	90	0	0	0	0	0	0	0	0	0	0	0	0

Fig. 2.8. : Building of sequence profile starting from a multiple sequence alignment.

2.3.3 Alignment profiles

The evolutionary information contained in a N sequences alignment can be summarized in the *alignment profile*, that is a matrix $20 \times L$, where L is the alignment length. For each of the 20 positions, corresponding to the 20 types of amino acids, the profile stores the amino acidic composition on the aligned sequences. In Figure 2.8 the mapping between a multiple sequence alignment and a sequence profile is shown.

It is evident that the profile contains less information with respect to the multiple sequence alignment, because it is a vector that represents an average over the aligned sequences. Given a profile it is therefore not possible to reconstruct the alignment that generated it. However, the information regarding the degree of conservation of a given residue in a given position of the sequence or the related mutations, that still preserve the protein structure, can be directly read in the values of the profile. The most the value approaches 100 the more conserved is the corresponding residue.

2.3.4 PSI-BLAST

Position Specific Iterative BLAST (PSI-BLAST) is a feature of BLAST in which a profile is automatically constructed from the first set of BLAST alignments. PSI-BLAST was mainly developed to answer to speed, simplicity and automatic operation questions. The PSI-BLAST procedure can be summarized in five steps:

1. PSI-BLAST takes as an input a single protein sequence and compares it to a protein database, using the gapped BLAST program.

2. The program builds a multiple alignment, and then a profile, from any significant local alignment found. The original query sequence serves as a template for the multiple alignment and profile, whose lengths are identical to that of the query.

3. The profile is compared to the protein database, again seeking local alignments. After a few minor modifications, the BLAST algorithm can be used for this directly.

4. PSI-BLAST estimates the statistical significance of the local alignments found.

5. Finally, PSI-BLAST iterates, by returning to step (2), an arbitrary number of times or until convergence.

Profile-alignment statistics allow PSI-BLAST to proceed as a natural extension of BLAST; the results produced in iterative search steps are comparable to those produced from the first pass. PSI-BLAST reveals many protein relationships missed by single-pass database search methods and has identified relationships that were previously detectable only from information about the three-dimensional structure of the proteins.

Chapter 3

Methods

3.1 Graphical models

3.1.1 Introduction to graph theory

Complex structures are at the basis of a variety of systems. The cell, for example, is best described as a complex network of chemicals connected by chemical reactions. An interesting approach to the study of these systems is to investigate their topology. Physics has developed efficient tools for predicting the behaviour of a system as a whole from the properties of its constituents: how the spins contribute to the magnetic properties of matter, the behaviour of quantum particles in Bose-Einstein condensation or superfluidity. The key feature of these models is the way of representing interactions between the constituent elements: no ambiguity exists in detecting who interacts with what and the interaction strength is uniquely determined by the physical distance.

Traditionally, the study of complex networks has been the territory of graph theory. Recently, significant advances have concerned this field mainly due to the increasing amount of data produced, that led to the possibility to have large databases to study the topology of real networks. Thus, the domain of application of this approach extended to networks containing millions of nodes, exploring topics and relationships that could not be addressed before.

When analyzing the function of a cell, we can distinguish between two main levels of networks: the network of the interactions between the residues of a protein sequence that contain the rule to build its native structure and a physical network of protein-protein and protein-nucleic acid complexes interactions that are responsible of the cell metabolism, that is the flow of molecules and energy through pathways of chemical reactions. Graph theory can be successfully applied to the analysis of these networks, that are of special interest in systems biology.

In mathematical terms, an undirected graph is a pair of sets $G = (V, E)$, where V is a set of N nodes (or vertices) $\{V_1, V_2, ..., V_N\}$ and E is a set of edges (or links) that connect two elements of V. Among the concepts and measures that have been recently proposed to describe complex networks, three are the most investigated ones:

- *Small-world.* The small-world property describes the fact that in most networks, even in those of large size, there is a relatively short path between any two nodes. The distance between two nodes is defined as the number of edges along the shortest path connecting them. The measure to quantify this property, namely the *characteristic path length* (L) of the network, has been introduced by Watts and Strogatz in 1998 [Watts and Strogatz, 1998]. L is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices.

- *Clustering.* Watts and Strogatz also defined the *Clustering coefficient* (C) that measures the cliquishness of a typical neighbourhood of a given node. If a vertex i has k_i neighbours, then the maximum number of edges that can form between them is $k_i(k_i - 1)/2$ (when every neighbour of i is connected to every other neighbour of i). If C_i denotes the fraction of actual edges, C is the average of C_i over the total number of nodes N.

- *Degree distribution.* Not all the E nodes of a network have the same number of connections. The spread of the distribution $P(k)$ of the node degrees gives the probability that a randomly selected node has exactly k edges. The distribution function $P(k)$ gives the probability that a randomly selected node has exactly k edges. Since in a random graph the edges are placed randomly, the majority of nodes have approximately the same degree, close to the average degree $\langle k \rangle$ of the network. The degree distribution of a random graph is a Poisson distribution with a peak at $P(\langle k \rangle)$. Nevertheless, for most large networks the degree distribution has a power-law tail:

  $$P(k) \approx k^{-\gamma} \tag{3.1}$$

  Such networks are called *scale-free* [Barabasi and Albert, 1999]. While some networks display an exponential tail, often the functional form of $P(k)$ still deviates significantly from the Poisson distribution expected for a random graph.

3.1.2 From random to small-world networks

In their pivotal work on random graphs, Erdös and Rényi define a random graph as N labeled nodes connected by n edges, which are randomly chosen from the set of $N(N - 1)/2$ possible edges [Erdös and Rényi, 1960]. In total there are $C^n_{N(N-1)/2}$ graphs with N nodes and n edges, forming a probability space in which every realization is equiprobable. Random graph's theory studies the properties of the probability space associated with graphs with N nodes as $N \to \infty$.

The first studied property of random graphs has been the appearance of *subgraphs.* A graph $G1$ consisting of a set V_1 of nodes and a set E_1 of edges is a subgraph of a graph $G = (V, E)$ if all the nodes in V_1 are also nodes of V and all edges in E_1 are also edges of E. Examples of subgraphs are complete subgraphs or *connected components.* A graph is complete if every pair of vertices is connected

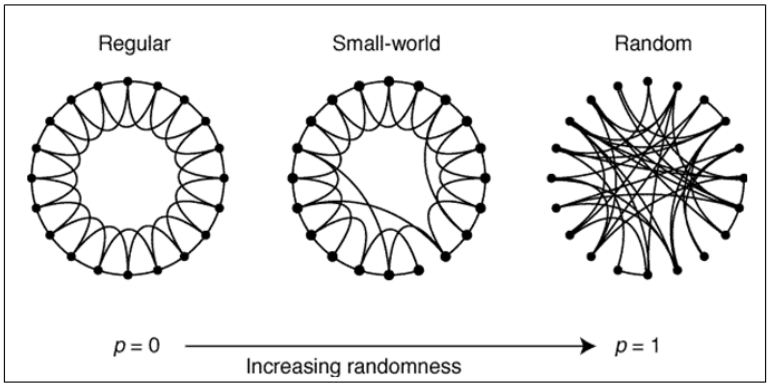


Fig. 3.1. : Random rewiring procedure with different outputs: regular, small-world and random graphs [Watts and Strogatz, 1998].

by an edge. Complete subgraphs of order k contain k nodes and all the possible $k(k-1)/2$ edges, in other words, they are completely connected.

Many properties of such random graphs can be determined using probabilistic arguments. In the mathematical literature the construction of a random graph is often called an evolution (Fig.3.1): starting with a ring of N vertices, each connected to its n nearest neighbours by undirected edges, with probability p, each edge is reconnected to a vertex chosen uniformly at random over the entire ring, with duplicate edges forbidden. The procedure continues until each edge in the original lattice has been considered once. In Fig.3.1 three realizations of this process are shown, for different values of p: for $p \rightarrow 0$, the original ring is unchanged, as p increases, the graph becomes increasingly disordered until for $p \rightarrow 1$, all edges are rewired randomly. For intermediate values of p, the graph is a small-world network whose peculiar characteristics are highly clustering like a regular graph and small characteristic path length, like a random graph. It can also been observed that the average degree of a random graph, defined as:

$$\langle k \rangle = 2n/N = p(N-1) \cong pN \tag{3.2}$$

has a value that is independent of the system size.

The diameter of a graph is the maximal distance between any pair of its nodes. Random graphs tend to have small diameters, provided p is not too small. The reason for this is that a random graph is likely to spread: with large probability the number of nodes at a distance L from a given node is not much smaller than $\langle k^l \rangle$. Equating $\langle k^l \rangle$ with N we find that the diameter is proportional to $ln(N)/ln(\langle k \rangle)$; thus it depends only logarithmically on the number of nodes. For almost values of p, almost all random graphs with the same N and p have precisely the same diameter. This means that when considering all graphs with N nodes and connection probability p, the average path length scales with the number of nodes:

$$l_{rand} = \frac{ln(N)}{ln(pN)} = \frac{ln(N)}{ln(\langle k \rangle)} \tag{3.3}$$

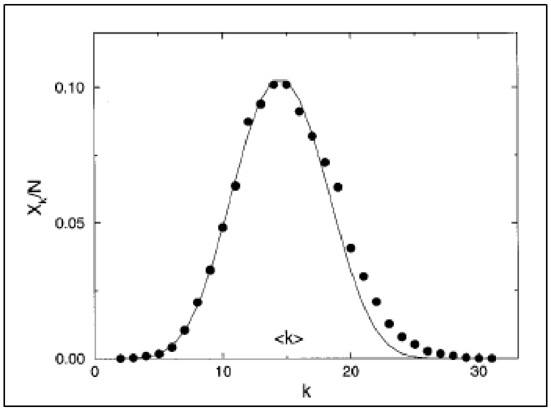


Fig. 3.2. : The degree distribution that results from the numerical simulation of a random graph (N is the total number of nodes, X_k are the nodes with degree k) [Barabasi and Albert, 2002].

Furthermore, by considering a node in a random graph and its nearest neighbours, the probability that two of these neighbours are connected is equal to the probability that two randomly selected nodes are connected, that is:

$$C_{rand} = p = \frac{\langle k \rangle}{N} \tag{3.4}$$

In a random graph with connection probability p the degree k_i of a node i follows a binomial distribution (Fig.3.2) with parameters $N-1$ and p:

$$P(k_i, k) = C^k_{N-1} p^k (1-p)^{N-1-k} \tag{3.5}$$

$P(k_i, k)$ is the number of ways in which k edges can be drawn from a certain node: the probability of k edges is p^k, the probability of the absence of additional edges is $(1-p)^{N-1-k}$, and there are C^k_{N-1} equivalent ways of selecting the k end points for these edges. For large N, the binomial can be replaced with a Poisson distribution:

$$P(k) \cong e^{-pN} \frac{(pN)^k}{k!} = e^{\langle -k \rangle} \frac{\langle k \rangle^k}{k!} \tag{3.6}$$

Real-world networks have a small-world character like random graphs, but they have unusually large clustering coefficients. Furthermore, the clustering coefficient appears to be independent of the network size, a property that is characteristic of ordered lattices, whose clustering coefficient is size independent and depends only on the coordination number. For example, in a one-dimensional lattice with periodic boundary conditions, most of the immediate neighbours of any site are also neighbours of one another. For such a lattice the clustering coefficient is:

$$C = \frac{3(K-2)}{4(K-1)} \tag{3.7}$$

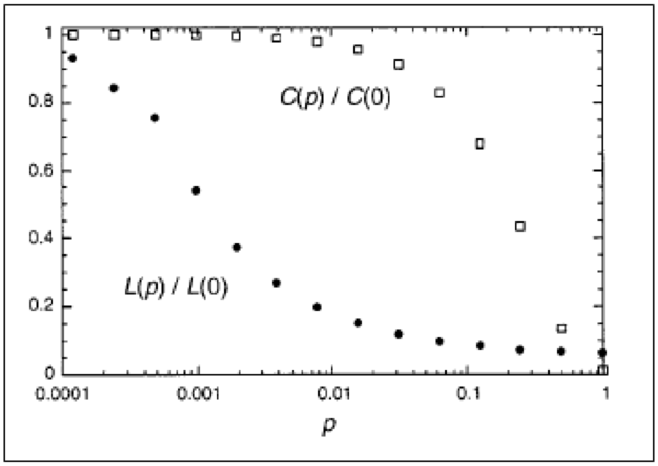


Fig. 3.3. : Characteristic path length $l(p)$ and clustering coefficient $C(p)$ for the Watts-Strogatz model [Watts and Strogatz, 1998].

which converges to 3/4 in the limit of large K. Such low-dimensional regular lattices, however, do not have short path lengths: for a $d-$dimensional hypercubic lattice the average node-node distance scales as $N^{1/d}$, which increases much faster with N than the logarithmic increase observed for random and real graphs.

In 1998 Watts and Strogatz [Watts and Strogatz, 1998] proposed a one parameter model that interpolates between an ordered finite-dimensional lattice and a random graph (Fig.3.3). The algorithm is the following:

- *Start with order.* Start with a ring lattice with N nodes in which every node is connected to its first K neighbours. In order to have a sparse but connected network at all times, consider $N \gg K \gg ln(N) \gg 1$.
- *Randomize.* Randomly rewire each edge of the lattice with probability p such that self-connections and duplicate edges are excluded.

To understand the coexistence of small path length and clustering, the behaviour of the clustering coefficient $C(p)$ and the average path length $l(p)$ as a function of the rewiring probability p, have to be analyzed.

For a ring lattice $l(0) \cong N/2K \gg 1$ and $C(0) \cong 3/4$. For this reason l scales linearly with the system size, and the clustering coefficient is large. On the other hand, for $p \to 1$ the model converges to a random graph for which $l(1) \approx ln(N)/ln(K)$ and $C(1) \approx K/N$. In this case, l scales logarithmically with N and C decreases with N. These observations suggest that large C is always associated with large l and small C with small l. On the contrary, Watts and Strogatz found that there is a broad interval of p over which $l(p)$ is close to $l(1)$ but $C(p) \gg C(1)$. These small-world networks result from the immediate drop in $l(p)$ caused by the introduction of a few long-range edges. For small p, each short cut has a highly nonlinear effect on l, contracting the distance not just between the pair of vertices that it connects, but between their immediate neighbourhoods, the neighbourhoods of neighbourhoods and so on (Fig.3.3).

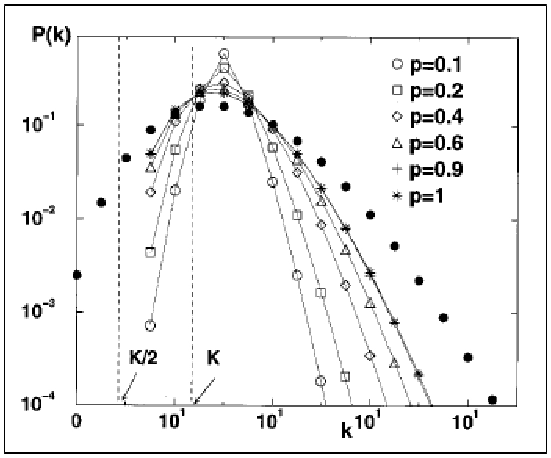


Fig. 3.4. : Degree distribution of the Watts-Strogatz model for $K = 3$ and various p [Barrat and Weigt, 2000].

In the Watts and Strogatz model for $p = 0$ each node has the same degree K. Thus the degree distribution is a delta function centered at K. A nonzero p introduces disorder in the network, broadening the degree distribution while maintaining the average degree equal to K. The shape of the degree distribution is similar to that of a random graph. It has a pronounced peak at K and decays exponentially for large k (Fig.3.4). Thus the topology of the network is relatively homogeneous, all nodes having approximately the same number of edges. This coexistence of small L and large C is in excellent agreement with the characteristics of real networks.

3.1.3 Algorithms on graphs: some definitions

A *walk* in a graph $G = (V, E)$ is a sequence of vertices $\{v_0, v_2, ..., v_n\}$ such that for all $0 \leq i < n$, (v_i, v_{i+1}) is an edge in G. The length of the walk $\{v_1, v_2, ..., v_n\}$ is the number n. A *path* is a walk in which no vertex is repeated. As introduced above, a graph G is connected if there is a path between all pairs of vertices i and j of $V(G)$. The diameter of a connected graph is the least integer D such that for all vertices i and j in G, $d(i, j) \leq D$, where $d(i, j)$ denotes the distance from i to j in G, that is, the length of a shortest path between i and j.

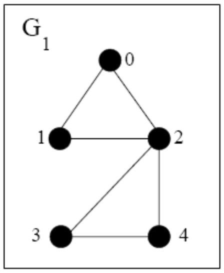


Fig. 3.5. : A pictorial example of a graph G1 with 5 nodes and 7 connections.

For example, for computing the diameter of graph $G1$ in Figure 3.5: $d(0,1) = 1$, $d(0,2) = 1$, $d(0,3) = 2$, $d(0,4) = 2$, $d(1,2) = 1$, $d(1,3) = 2$, $d(1,4) = 2$, $d(2,3) = 1$, $d(2,4) = 1$ and $d(3,4) = 1$. Furhthermore, for graphs $d(x,y) = d(y,x)$, for all vertices x and y.

There are two common computational representations for graphs called *adjacency matrices* and *adjacency lists*. For a graph G of order n, an adjacency matrix representation is a boolean matrix (often encoded with $0's$ and $1's$) of dimension n such that entry (i,j) is true if and only if the edge (i,j) is in $E(G)$. For a graph G of order n an adjacency lists representation is composed of n lists such that the i-th list contains a sequence of the neighbours of vertex i of G. Adjacency matrix for graph G1 is:

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

while an adjacency list is:
$0 : 1\ 2$
$1 : 0\ 2$
$2 : 0\ 1\ 3\ 4$
$3 : 2\ 4$
$4 : 2\ 3$

An empty list can occur. For a graph with n vertices and m edges, the adjacency matrix representation requires $O(n^2)$ storage while the adjacency lists representation requires $O(m)$ storage. So for sparse graphs the latter is preferable.

3.1.3.1 Graph searching

Two standard algorithms called *breadth first search* (BFS) and *depth first search* (DFS) guide the search over the vertices of the graph.

In BFS we start at a vertex v and then go to the neighbours N_1 of v, then to the neighbours N_2 of N_1 that have not been visited and so on. The idea is to process vertices N_i of distance i before processing the vertices of distance $i+1$ or greater. The breadth search is repeated until all vertices which are reachable from vertex v have been visited. In DFS we start at a vertex v but this time we deeply search as far away from vertex v until we cannot visit any new vertices. We then backtrack and try other neighbours missed along the search paths until we have tried all possible routes. The DFS yields a rooted subtree of the graph with the nice property that all edges not traversed in the search tree go up the tree.

Two algorithms for computing shortest paths bewteen the edges have been implemented by Dijkstra and Floyd, respectively. The shortest path problem is the problem of finding a path between two vertices of a graph such that the total weight of the edges on the path is minimized. These algorithms can also be applied to an unweighted graph to find the path of minimum length by simply treating it as a weighted graph with all edge weights equal (all set to 1). The *single-source shortest path problem* is the problem of finding the shortest path to all other vertices (or to 1 particular destination vertex) in the graph from a given origin vertex. On an unweighted graph, this problem can be solved using breadth-first search. Dijkstra's algorithm is a more sophisticated search that accounts for the edge weights as it traverses the graph (still visiting each node only once). To do this, for each node, a variable that tracks the distance of the node from the origin along the shortest path found to it so far has to be memorized, in addition to a pointer to keep track of the path. The initial assumtpion is that each node is infinitely far away, then the algorithm moves to the next closest node and update the estimates from that point. The priorities of the vertices are changing as the algorithm runs. Dijkstra's algorithm solves this single-source shortest paths problem in $O(V^2)$ time.

A variant of the single-source shortest path problem is the *all-pairs shortest paths problem*: find a shortest path from u to v for every pair of vertices u and v. This problem can be addressed with a dynamic programming formulation resulting in the Floyd-Warshall algorithm. The algortihm considers the intermediate vertices of a shortest path, where an intermediate vertex of a simple path $p = v\{v_1, v_2, ..., v_l\}$ is any vertex of p other than v_1 or v_l, that is, any vertex in the set $\{v_2, v_3, ..., v_{l-1}\}$. Floyd's algorithm calculates the costs of the shortest path between each pair of vertices in $O(V^3)$ time. Running Dijkstra's single source algorithm V times with each vertex as the source in turn also finds all shortest paths in $O(V^3)$ time but Floyd's algorithm is more direct. For a more detailed description of these algorithms see [Cormen *et al.*, 2001].

3.2 The probabilistic framework

Probabilistic models are suitable tools for handling with the diversity and the complexity of biological data (protein sequences, DNA or RNA sequences). In computational molecular biology there is large amount of data but little theory and the goal is to extract useful information from a set of data D by building good probabilistic models. The idea behind machine learning techniques is to automate this process as much as possible, often by using very flexible models characterized by large numbers of parameters, and to let the machine "take care" of the rest. While available sequence data are rapidly increasing, our current knowledge of biology constitutes only a small fraction of what remains to be discovered. Thus, in computational biology in particular, one must reason in the presence of a high degree of uncertainty and problems are induction or *inference* problems: building models for available data.

Many problems in biological sequence analysis have the same structure: based on sequences, defined as strings of symbols from an alphabeth with definite cardinality (20 for proteins and 4 for nucleic acids), find out what the sequence represents. For example, we tipically want to discover what protein family a given sequence belongs to. The Bayesian framework provides a strong underlying foundation that unifies the different machine learning techniques. The Bayesian approach assigns a degree of plausibility to any hypothesis or model.

In a general description, a probabilistic model M is an object able to generate each sequence s with a probability $P(s|M)$ which, for definition, satisfies the usual conditions of positivity and normalisation:

$$0 \leq P(s|M) \leq (1) \tag{3.8}$$

$$\sum_{s} P(s|M) = 1 \tag{3.9}$$

The distribution of that probability, over the space of all the possible sequences, determines the model specificity: for a specific class, an ideal model should generate with a high probability all and only the sequences of that class, excluding the others. For a probabilistic model to be adopted for problem solving in computational biology, an operative definition of the rules that are necessary to calculate the $P(s|M)$ value for each sequence should be given. In that sense, a model is an object that associates to each sequence a real number. Furthermore, the application of a probabilistic model is limited to the class of models for which algorithms for the estimation of parameters, starting from a set of known sequences used as examples, are available. A probabilistic model trained on a particular class of sequences, for example, is able to search in an overall proteome of a specific organism, the sequences that are most likely to belong to that class.

Another problem for which probabilistic models are suitable is the assignment of a given sequence s to a class. In case of different available models for the same sequence, the most appropriate for describing the sequence has to be choosen. This particular task requires the estimation of the so-called *posterior* probability $P(M|s)$ that, by means of the Bayes theorem, can be expressed as:

$$P(M|s) = \frac{P(s|M) \cdot P(M)}{P(s)} = \frac{P(s|M) \cdot P(M)}{\sum_{M'} P(s|M') \cdot P(M)} \tag{3.10}$$

Thus, this latter problem requires an estimation of the *a priori* probability $P(M^{'})$ of the different models $M^{'}$, under the assumtpion that the set of models is complete. In practice this can be done by computing the frequency of occurrence of the sequences in each class described by the set of models $M^{'}$. Once estimated the probability of Eq.3.10, the discrimination problem can be solved choosing the model with the highest probability.

3.2.1 The learning procedure

In a Bayesian framework, sequence models must be probabilistic. The learning procedure aims to infer the parameters $\theta = \theta_i$ of a model M that better describe a set of data D. The most commonly adopted strategy is to maximize the likelihood of the parameters with respect to the considered data, namely the probability $P(D|\theta, M)$, as a function of θ. This is commonly known as *Maximum Likelihood* (ML) rule and it can be formally written as:

$$\theta^{ML} = argmax_{\theta} P(D|\theta, M) \tag{3.11}$$

where θ^{ML} is the optimal set of parameters for ML. The parameters obtained for ML are consistent. This means that if the set D is generated from a model with parameters θ^0, and if that data set is big enough, the parameters estimated for ML tend to be equal to θ^0. When the data set of examples is not sufficiently populated, this procedure can lead to a wrong estimation of parameters. In this case, it is possible to take into consideration assumptions and knowledge on the *a priori* distribution of the parameters, $P(\theta|M)$, by estimating, with the Bayes theorem, the *posterior* probability of the parameters:

$$P(\theta|D, M) = \frac{P(D|\theta, M) \cdot P(\theta, M)}{P(D|M)} \tag{3.12}$$

The maximization of the probability in Eq.3.12 is the *Maximum (probability) A Posteriori* (MAP):

$$\theta^{MAP} = argmax_{\theta} P(D|\theta, M) \cdot P(\theta|M) \tag{3.13}$$

The Bayesian approach is essentially concerned only with assessing the value of models with respect to the available knowledge and data. It is not dircetly concerned with the creative process of generating new hypothesis and models. However, this assessment procedure is at the basis of the design of new models.

3.3 Markov chains and Hidden Markov Models

Markov models are a class of probabilistic graphical models that are widely adopted because of their simple rules. In these models, each element of a sequence is generated with a probability that depends only on a finite number of elements that come before that element in the considered sequence. This number is called the *order* of the models and it determines the degree of approximation of the model upon the sequences space. Formally, in a Markov model of order n, the probability of generating the character in position t of the sequence s depends on the string $s^{t-n}s^{t-n+1}...s^{t-1}$. Thus, the parameters of the model are the variables

$$a(\gamma^{-n}\gamma^{-n+1}...\gamma^{-1} \rightarrow \gamma) = P(\gamma|\gamma^{-1}\gamma^{-2}...\gamma^{-n}), \quad \gamma, \gamma^{-1}, \gamma^{-2}, ..., \gamma^{-n} \in \mathbf{A} \tag{3.14}$$

where $\mathbf{A}$ is the alphabet of characters that compose the sequence, with two extra characters labeled as *BEGIN* and *END*. The following rules still hold:

$$0 \leq a(\gamma^{-n}\gamma^{-n+1}...\gamma^{-1} \rightarrow \gamma) \leq 1 \tag{3.15}$$

$$\sum_{\gamma} a(\gamma^{-n}\gamma^{-n+1}...\gamma^{-1} \rightarrow \gamma) = 1 \tag{3.16}$$

A Markov model of order 0, in which the probability of generating a character depends only on the character itself, can describe only the global composition of a set of sequences; instead, a model of the first order can deal with the statistic of pairs of consecutive symbols, one of the second order with the groups of three consecutive characters and so on. The highest is the order of the model, the greater is the quantity of information that it can process. Nevertheless, this growth in the information analyzed also corresponds to the rapid increase in the number of parameters: if the alphabet has M symbols, the number of parameters of the model is equal to the sum of all the possible combinations of $n + 1$ characters, that is M^{n+1}.

3.3.1 Parameters estimation (ML)

As introduced above, the Maximum Likelihood estimation of the parameters of a Markov model consists in the computation, starting from the training set, of the frequency of occurrence of each of the characters with each one of the possible combinations of n characters before it.

$$a(\gamma^{-n}\gamma^{-n+1}...\gamma^{-1} \rightarrow \gamma) = \frac{N(\gamma, \gamma^{-1}, \gamma^{-2}, ..., \gamma^{-n})}{N(\gamma^{-1}, \gamma^{-2}, ..., \gamma^{-n})} \tag{3.17}$$

where $N(\gamma)$ is the number of substrings γ in the training set. Therefore, the growth in the order dimensionality is limited by the number of sequences available for training and by the problems due to the statistical counting of the rare n-tuples.

3.3.2 First order models

The simplest and non trivial Markov models are that of the first order and they can be described as a set of states connected with the probability of transition a_{ij} between state i and state j. Each state has univoquely associated a character from the alphabet: a path among the states generates a sequence. In Fig.3.6 it is schematically represented a model for the description of DNA sequences. It is composed of four states, corresponding to the four bases, completely connected between them and of two extra states, BEGIN and END. Each arrow indicates a transition probability.

3.3.3 Higher order models

In general, a model of order n on an alphabet $\mathbf{A}$ is equivalent to a model of the first order on the alphabet $\mathbf{A}^n$ of ordered n-tuples.

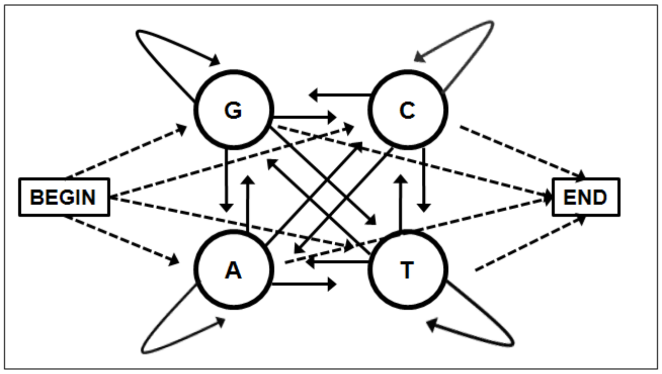


Fig. 3.6. : First order Markov model for DNA sequences.

This equivalence holds because:

$$P(s^t|s^{t-1}s^{t-2}...s^{t-n}) = P(s^t s^{t-1}s^{t-2}...s^{t-n+1}|s^{t-1}s^{t-2}...s^{t-n}) \tag{3.18}$$

Let us consider an alphabet of only two symbols $\mathbf{A} = \{a, b\}$ and a model of the second order. Each sequence of characters from the alphabet $\mathbf{A}$ can be translated in a sequence of characters of the alphabet $\mathbf{A} \times \mathbf{A}$. For example:
$a - a - b - b - a - b - a \rightarrow aa - ab - bb - ba - ab - ba$
Thus, the model of the second order can be mapped onto a model of the first order of Fig.3.7. In the latter some transitions cannot occur beacuse they have no meaning under the point of view of the grammar that underlies the generation of the sequences of $\mathbf{A} \times \mathbf{A}$ from the sequences of $\mathbf{A}$.

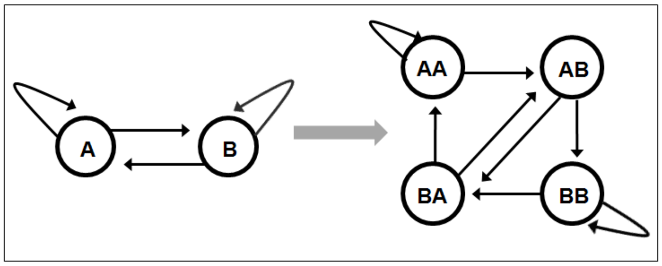


Fig. 3.7. : Markov models of the first and second order for sequences of a two-characters alphabet. The second order model (on the left) can be reduced to a first order model with 4 states. It can be noticed that the number of transition parameters increases.

3.3.4 Hidden Markov Models

The most commonly adopted probabilistic models in biological sequence analysis are the Hidden Markov Models (HMMs). These models have been introduced during the '70s and they have been extensively exploited in the field of "speech recognition" and of the signals reconstruction.

HMMs are probabilistic models in which sequences are generated from two stochastic processes that cohexist. The first one is a Markov model that, considering the observations made in the previous section and without any lack of generality can be considered of the first order. The second is the emission of one character of the alphabet **A** from each state, following a probability distribution that only depends on the state. A sequence is therefore generated by this latter process together with a path among the states of the Markov model. In general, what only the sequence can be observed, while the path that generated it remains hidden. HMMs are Hidden models because they allow to interpret the observed sequence as the result of a unknown Markov process.

3.3.4.1 Formal definition

If s is a generic sequence generated from the path π, an HMM is defined by:

- a set of N states;
- a set of probabilities of transition between the states, $\{a_{ij}\}$, of cardinality N^2:

$$a_{ij} = P(\pi^t = j|\pi^{t-1} = i); \tag{3.19}$$

- a set of probabilities of starting the Markov process from state i, $\{a_{0i}\}$, of cardinality N:

$$a_{0i} = P(\pi^t = i|\pi^{t-1} = BEGIN); \tag{3.20}$$

- a set of probabilities of ending the Markov process after state i, $\{a_{i0}\}$, of cardinality N:

$$a_{i0} = P(\pi^t = END|\pi^{t-1} = i); \tag{3.21}$$

- an alphabet **A** of M characters;
- a set of probabilities of emission of the characters from each state, $\{e_k(c)\}$, of cardinality $M \cdot N$:

$$e_k(c) = P(s^t = c|\pi^t = k). \tag{3.22}$$

On the basis of the given defintions it is possible to calculate, knowing a sequence s of length L and the corresponding path π, their joint probability:

$$P(s, \pi|M) = a_{0\pi^1} \cdot \prod_{t-1}^{L} a_{\pi^t\pi^{t+1}} \cdot e_{\pi^t}(s^t) \tag{3.23}$$

3.3.5 The three fundamental problems of HMMs

The goal of the HMM theory is to solve three main problems, the evaluation problem, the decoding problem and the problem of the training, whose solutions have important applications.

3.3.5.1 The evaluation

Since the paths associated to the sequences are usually hidden, the problem of finding a way to calculate the emission probability of the only sequence s from the model M, $P(s|M)$, is posed. Formally:

$$P(s|M) = \sum_{\pi} P(s, \pi|M) \tag{3.24}$$

The solution to this problem requires, in theory, the evaluation of the joint probability by means of Eq.3.23.

Given a model with N states completely connected and given a sequence of length L, there are N^L possible paths that can have generated it. This number makes impossible the application of Eq.3.24 for the computation of $P(s|M)$, with the exception of short sequences. Thus, the evaluation problem consists in the search of an efficient algorithm whose execution time grows only polinomially with the sequence length. Without such an algorithm HMMs will be mathematical tools without any practical utility. In the following a dynamic programming technique that allows the formulation of suitable algorithms will be described.

3.3.5.2 The decoding

The second problem of HMMs concerns the search of the hidden part of the model, that is, given a sequence, the path that generated it. Indeed, although it is no longer possible to tell what state the system is in by looking at the corresponding symbol, it is often the sequence of underlying states that we are interested in. To find out what the observation sequence “means” by considering the underlying states is called decoding in the jargon of speech recognition. This task, with the exception of few cases, can be addressed only in probabilistic terms.

The most common approach is the Viterbi algorithm, a dynamic programming algorithm that looks for the path π^* (*Viterbi path*) that has the highest probability of having generated the sequence:

$$\pi^* = argmax_{\pi} P(\pi|s, M) \tag{3.25}$$

Following the above definition, also in this case Eq.3.24 should be computed over all the possible paths. The dynamic programming implements an efficient algorithm also for dealing with the decoding problem, whose solution is at the basis of the HMM design.

Indeed, if each state k of a model is associated to a label $l(k)$, by defining a *semantic*, the decoding of the Viterbi path relative to a give sequence, corresponds to an association between the sequence and a string of labels [Krogh A, 1994].

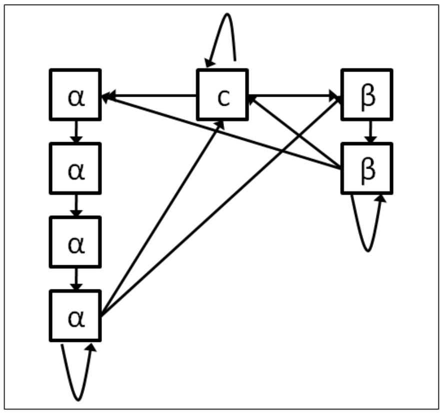


Fig. 3.8. : HMM for secondary structure prediction. It includes the constraints on the minimum sequence length of α-helix and β-strand.

In this way HMM performs a mapping. In Fig.3.8 is depicted a simple model for the prediction of the secondary structure of a protein sequence. Each state is associated to a secondary structure motif (α-helix, β-strand and coil) and the Viterbi algorithm computes a probability (makes a prediction) for each residue of the sequence. A fundamental property of the model, as can be seen from the model in Fig.3.8, is that it includes, in its allowed transitions, a *syntax* that reflects some prior knowledge on the system. This poses several constraints on the predictions: any predicted α-helix can be shorter than 4 residues, any β-strand shorter than 2.

The Viterbi algorithm is neither the only possible decoding algorithm nor the recommended procedure when many different paths have almost tha same probability as the most probable one (the Viterbi path).

In these latter cases, the *posterior decoding* can be adopted. Here, the path is the set of states $\underline{\pi}^t$ defined as:

$$\underline{\pi}^t = argmax_k P(\pi^t = k|s, M) \tag{3.26}$$

The posterior state sequence is the set of states that, position by position, have the highest probability. Nevertheless, since the computation is independent for each position, $\underline{\pi}^t$ can contain transitions that are not permitted and that violate the syntax of the model.

Another similar decoding approach arises when it is not the state sequence itself which is of interest, but some othe property derived from it. This decoding associates at each position the label with the highest probability $\underline{\lambda}^t$:

$$\underline{\lambda}^t = argmax_\lambda \sum_{l^k=\lambda} P(\pi^t = k|s, M) \tag{3.27}$$

Even in this case the respect of the syntax is not guaranteed.

3.3.5.3 The parameter estimation

Probably, the most difficult problem faced when using HMMs is that of specifying the model in the first place. There are two parts of this: the design of the structure, that is what states there are and how they are connected, and the assignment of the transition and emission probabilities and of the set of parameter values $\theta = \{a_{ij}, e_k(c)\}$ of the model M that better describe the sequences of a training set D. Just as it was easier to write down the probability of a sequence when the path was known, so it is easier to estimate the probability parameters when the paths are known for all the examples. When all the paths are known we can count the number of times each particular transition or emission is used among the sequence-path pairs of the training set. Let A_{ij} be the number of transitions between states i and j and $E_k(c)$ be the number of emissions of the character c from tha state k. Then, the ML estimators are:

$$a_{ij}^{ML} = \frac{A_{ij}}{\sum_{j'} A_{ij'}} \tag{3.28}$$

$$e_k^{ML}(c) = \frac{E_k(c)}{\sum_{c'} E_k(c')} \tag{3.29}$$

The estimation in Eq.3.28 is exactly the same that for a simple Markov chain. As always, ML estimators are vulnerable to overfitting if there are insufficient data. Indeed if there is a state k that is never used in the training set, the estimation equations are undefined for that state, because both the numerator and denominator will have zero value. To avoid such problems it is preferable to add predetermined pseudo-counts to A_{ij} and $E_k(c)$ before using Eq.3.28 and Eq.3.29.

$$A_{ij} = number\ of\ transitions\ i\ to\ j\ in\ training\ data\ + r_{ij} \tag{3.30}$$

$$E_k(c) = number\ of\ emissions\ c\ from\ k\ in\ training\ data\ + r_k(c) \tag{3.31}$$

The pseudo-counts r_{ij} and $r_k(c)$ should reflect the prior biases about the probability values.

When paths are unknown for the training sequences, there is no longer a direct closed-form equation for the estimated parameter values, and some form of iterative procedure must be used. All the standard algorithms for optimization of continuous functions can be used. However, there is a particular iteration method that is standardly used, known as the Baum-Welch algorithm [Baum LE, 1972]. This is a particular case of a more general algorithm, known as Expectation Maximization, that has a natural probabilistic interpretation and that will be described in the following.

3.4 Dynamic programming algorithms for HMMs

The solution to biological sequence analysis problems requires a search in the space of all the possible solutions, whose dimension grows exponentially with the length of the sequence studied. In some cases, it is possible to divide this problems in smaller ones that are easier to solve and that can be recursively concatenated to reach a global solution, in a time that grows only as a polinomial function of the sequence length. This is possible thanks to a technique called *dynamic programming* that is largely adopted in computational biology: for example to the search of the optimal global alignment between two sequences ([Needleman and Wunsch, 1970], [Smith and Waterman, 1981], [Gotoh O, 1982]), to the reconstruction of DNA sequences starting from the fragments obtained with experimental sequencing tecnhiques [Anson and Myers, 1997], to the solution of the evaluation problem and also of the decoding one detailed in the next paragraph.

3.4.1 The evaluation problem: the forward algorithm

The evaluation problem, as described above, consists in the efficient computation of the probability of emission of a sequence s from a model M, $P(s|M)$. In the dynamic programming approach, the evaluation of $P(s|M)$ is the computation, for each of the L positions (t) of the sequence and for each of the N states (k), of the quantity $f_k(t)$ defined as the probability of generating the partial observation $s^1s^2...s^t$, ending in state k:

$$f_k(i) = P(s^1s^2...s^t, \pi^t = k|M) \tag{3.32}$$

This probability can be computed recursively:

$$\begin{aligned} P(s^1s^2...s^t, \pi^t = k|M) &= \sum_l P(s^1s^2...s^{t-1}, \pi^{t-1} = l|M) \cdot \\ &\cdot \quad P(\pi^t = k|\pi^{t-1} = l) \cdot P(s^t|\pi^t = k) \end{aligned} \tag{3.33}$$

From this expression is appears that $f_k(t)$ can be calculated using the $f_l(t-1)$ values, corresponding to the previous position $(t-1)$ in the sequence, multiplied by the values a_{lk} and $e_k(s^t)$.

To take advantage of this recursive rule, an initialization condition should be established, that comes from the fact that each sequence has to start from the $BEGIN$ state (labeled with 0). The evaluation problem is thus solved when all the $f_k(L)$ quantities are known for the last position in sequence:

$$P(s|M) = \sum_l P(s^1s^2...s^L, \pi^L = l|M) \cdot P(\pi^{L+1} = END|\pi^L = l) \tag{3.34}$$

The second member of Eq.3.34 is the sum of all $f_k(L)$ multiplied by the relative probabilities of ending the path after the state l.

The algorithm can be summed up in three main steps:

- *Inizialization*

$$\begin{aligned} f_0(0) &= 1 \\ f_k(0) &= 0 \text{ , } \forall k \neq 0 \end{aligned} \tag{3.35}$$

- *Recursion* $(t = 1, 2, ..., L)$

$$f_k(t) = e_k(s^t) \cdot \sum_l f_l(t-1) \cdot a_{lk} \tag{3.36}$$

- *Termination*

$$P(s|M) = \sum_l f_l(L) \cdot a_{l0} \tag{3.37}$$

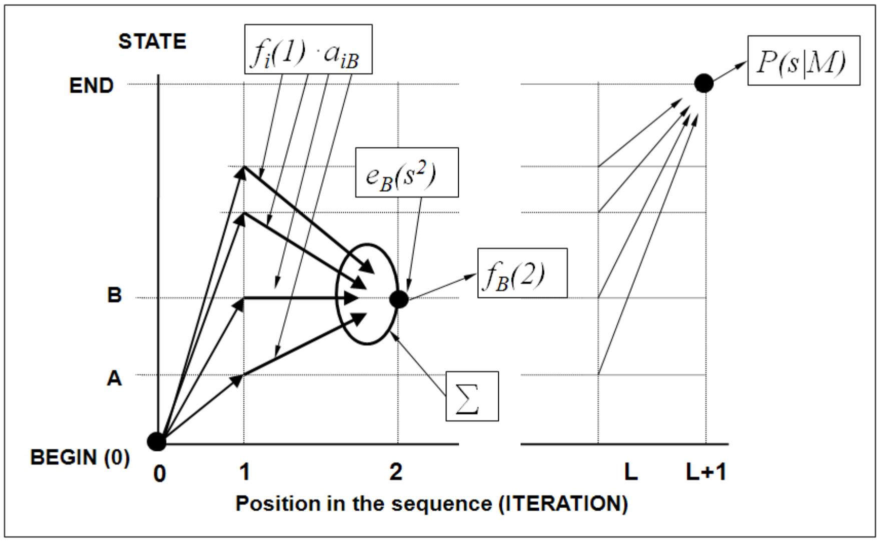


Fig. 3.9. : Representation of the forward algorithm. Each arrow represents the product of an element of the matrix with the corresponding transition probability. In the inizialization, all the elements in position 0 are null, with the exception of the BEGIN. The definition of a matrix element requires the sum of all the quantities corresponding to the *in-arrows* →, coming from the previous position in the sequence and the product for the emission probability.

The algorithm is schematically represented in Fig.3.9, where it is highlighted how the definition of each element in position i of the matrix f depends only on the elements that correspond to the position $(i-1)$ in the sequence. This algorithm is called *forward*, since the recurrence rule proceeds from the first to the last element of the sequence. The matrix f has $N \times L$ elements, each one requires about $2N + 1$ computations (sums and products). Therefore, the computational complexity of the forward algorithm only increases linearly with the sequence length.

3.4.2 The evaluation problem: the backward algorithm

Together with the forward, another algorithm is able to efficiently calculate the probability $P(s|M)$. This is called *backward* algorithm, because, in this case, the recursion rule proceeds from the end to the beginning of the sequence. The quantity that defines the sub-problems in which the evaluation problem is divided is $b_k(t)$, defined as the probability of generating the terminal subsequence $s^{t+1}s^{t+2}...s^L$ once visited the state k.

$$b_k(t) = P(s^{t+1}s^{t+2}...s^L|\pi^t = k, M) \tag{3.38}$$

The inizialization condition is given by the probability of transition to the END state.

The recursion derives from the expression:

$$\begin{aligned} P(s^{t+1}s^{t+2}...s^L|\pi^t = k, M) &= \sum_l P(s^{t+2}s^{t+3}...s^L|\pi^{t+1} = l, M) \cdot \\ &\cdot \quad P(\pi^{t+1} = l|\pi^t = k) \cdot P(s^{t+1}|\pi^{t+1} = l) \end{aligned} \tag{3.39}$$

The backward algorithm is described in the next three steps:

- *Inizialization* $(t = L)$
$$b_k(L) = a_{k0}\ , \forall k \tag{3.40}$$
- *Recursion* $(t = L-1, L-2, ..., 1)$
$$b_k = \sum_l b_l(t+1) \cdot a_{kl} \cdot e_l(s^{t+1}) \tag{3.41}$$
- *Termination*
$$P(s|M) = \sum_l b_l(l) \cdot a_{0l} \cdot e_l(s^l) \tag{3.42}$$

Also the computational time of the backward increases linearly with the sequence length.

For instance, we can be interested what the most probable state is for a given position s. More generally, we may want the probability that observation s came from state k, given the observed sequence, that is $P(\pi^t = k|s, M)$. This is the *posterior* probability of state k at time s when the emitted sequence is known. Taking advantage from the computations of the matrices f and b:

$$\begin{aligned} P(s, \pi^t = k|M) &= P(s^1s^2...s^t, \pi^t = k|M) \cdot P(s^{t+1}s^{t+2}...s^L|\pi^t = k, M) \\ &= f_k(t) \cdot b_k(t) \end{aligned} \tag{3.43}$$

As a result:

$$P(\pi^t = k|s, M) = \frac{f_k(t) \cdot b_k(t)}{P(s|M)} \tag{3.44}$$

The posterior path is thus reconstructed by choosing, for each position, the state with the highest probability.

3.4.3 The decoding problem: the Viterbi algorithm

Also the most probable path can be found recusively with dynamic programming techniques. For each position, $v_k(t)$ is the probability of the most probable path to generate the starting substring $s^1 s^2 ... s^t$, ending in the state k. This path can only be the result of one of the N most probable paths that produce the previous substring $s^1 s^2 ... s^{t-1}$, each ending in a different state l. More precisely:

$$v_k(t) = max_l(v_l(t-1) \cdot P(\pi^t = k|\pi^{t-1} = l) \cdot P(s^t|\pi^t = k)) \tag{3.45}$$

The transition between the state l^* that maximizes Eq.3.45 and the state k determines the last step of the optimal path that generates $s^1 s^2 ... s^t$ and ends in k. l^* is stored in a variable, $ptr_k(t)$, called pointer and the pointers matrix allows, after the computation of the overall v matrix, to reconstruct the Viterbi path π^*. The inizialization, identical to the case of the forward and the other steps of the Viterbi algorithm are the following:

- *Inizialization* $(t = 0)$

$$\begin{aligned} v_0(0) &= 1 \\ v_k(0) &= 0 \text{ }, \forall k \neq 0 \end{aligned} \tag{3.46}$$

- *Recursion* $(t = 1, 2, ..., L)$

$$v_k(t) = e_k(s^t) \cdot max_l(v_l(t-1) \cdot a_{lk}) \tag{3.47}$$

$$ptr_k(t) = argmax_l(v_l(t-1) \cdot a_{lk}) \tag{3.48}$$

- *Termination*

$$P(s, \pi^*|M) = max_l(v_l(L) \cdot a_{l0}) \tag{3.49}$$

$$\pi^{*L} = argmax_l(v_l(L) \cdot a_{l0}) \tag{3.50}$$

- *Reconstruction* $(t = L-1, L-2, ..., 1)$

$$\pi^{*t-1} = ptr_{\pi^{*t}}(t) \tag{3.51}$$

The algorithm is represented in Figure 3.10 and Figure 3.11.

The computation of the matrix v is similar to that of matrix f in the forward algorithm, with the substitution of the sum with the *max* function. The final iteration, by definition, allows the computation of the joint probability of sequence and path for the emission on the Viterbi path. The *ptr* matrix is also built, that allows to reconstruct backwards the Viterbi path, or better the Viterbi paths, since more than one can exist with the same probability (see Fig.3.11)

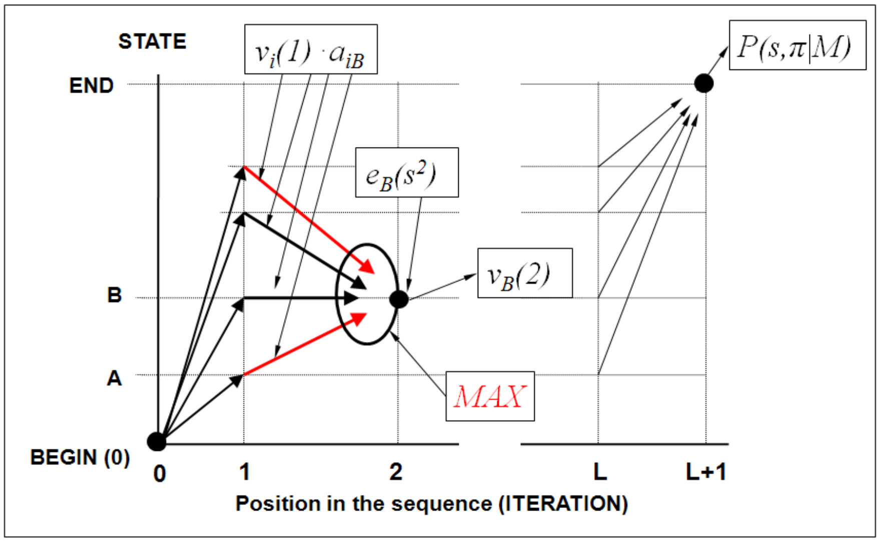


Fig. 3.10. : The Viterbi algorithm. Each arrow represents the product of an element of the matrix with the corresponding transition probability. The definition of a matrix element requires the search of the maximum among all the quantities corresponding to the *in-arrows* →, coming from the previous position in the sequence and its product for the emission probability.

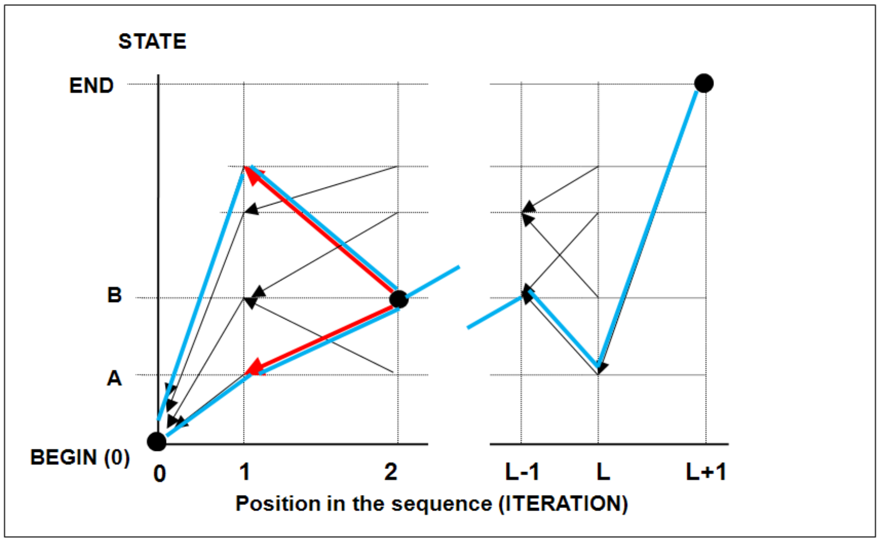


Fig. 3.11. : The Viterbi algorithm. The matrix elements corresponding to the maximum *in-arrows* (in red) are stored in the pointers matrix, indicated in the figure with ←. Following the pointers starting from the END, the Viterbi path (coloured in blu) is obtained. If more than one pointer comes from a given position, as depicted in fugure, the Viterbi paths are more than one and all with the same probability.

3.4.4 The Posterior Viterbi decoding

Posterior-Viterbi (PV) decoding is based on the combination of the Viterbi and posterior algorithms. After having computed the posterior probabilities, a Viterbi algorithm is used to find the best allowed posterior path through the model. A related idea, specific for pairwise alignments, was introduced to improve the sequence alignment accuracy [Holmes and Durbin, 1998]. In the PV algorithm, the basic idea is to compute the path:

$$\pi^{PV} = argmax_{\{\pi \in A_p\}} \sum_{i=1}^{L} P(\pi_i|O, M) \tag{3.52}$$

where A_p is the set of the allowed paths through the model and $P(\pi_i|O, M)$ is the posterior probability of the state assigned by the path π at position i (as computed in the posterior algorithm case).

Defining a function $\delta^*(s, t)$ equal to 1 if $s \rightarrow t$ is an allowed transition of the model M, 0 otherwise; $v_k(i)$ as the probability of the most probable allowed posterior path ending at state k having observed the partial $O1, ..., Oi$ and p_i as the traceback pointer, the best path π^{PV} can be computed using the Viterbi algorithm:

- *Inizialization*

$$v_B(0) = 1 \; v_k(0) = 0, \; for \; k \neq B$$

- *Recursion*

$$v_k(i) = max_s[v_s(i-1)\delta^*(s, k)]P(\pi_i = k|O, M)$$

$$p_i(k) = argmax_{\{s\}}[v_s(i-1)\delta^*(s, k)]$$

- *Termination*

$$P(\pi^{PV}|M, O) = max_s[v_s(L)\delta^*(s, END)]$$

$$\pi_L^{PV} = argmax_{\{s\}}[v_s(L)\delta^*(s, END)]$$

- *Traceback*

$$\pi_{i-1}^{PV} = p_i(\pi_i^{PV}), \; for \; i = L, ..., 1$$

- *Label assignment*

$$\Lambda_i = \lambda(\pi_i^{PV}), \; for \; i = 1, ..., L$$

3.4.5 The Expectation Maximization algorithm and the training problem

The Expectation Maximization (EM) algorithm is the most efficient one for the otpimal parameters estimation of a probabilistic model, when the data available for training offer only an incomplete representation of the considered stochastic problem [Dempster AP *et al.*, 1977]. To demonstrate the EM algorithm utility to HMMs, s is the sequence of observations and π is the data that the process lacks. In the next section the Jensen theorem on convex functions, that is necessary to demonstrate the results at the basis of the EM algorithm, is introduced.

A function $f(x)$ is defined as convex if:

$$f(ax_0 + (1-a)x_1) \leq a \cdot f(x_0) + (1-a) \cdot f(x_1) \ , \forall \ 0 \leq a \leq 1 \tag{3.53}$$

that is, if given two points x_0 and x_1 in the function domain, the values of the function in the intermediate points lie always under the segment that joins x_0 and x_1. From the definition in Eq.3.53, considering a stochastic variable x that can assume the two values x_0 and x_1 and indicating with a the probability of the point x_0, it can be obtained that the value of the function, evaluated on the expectation value x is always less or equal to the expected value of the values of the function computed on the stochastic variable. This result can be generalized for complete induction for any number of points, so that, if $E[y]$ is the expectation value of a discrete stochastic variable y distributed following a function $p(y)$, for all the convex functions f the Jensen theorem holds:

$$f(E[x]) \leq E[f(x)] \tag{3.54}$$

$$f(\sum_x p(x)x) \leq \sum_x p(x) \cdot f(x) \tag{3.55}$$

A fundamental result on the logarithm function can be derived from this expression. Since the function $-log$ is convex, for each function $q(x)$:

$$log(\sum_x p(x)q(x)) \geq \sum_x p(x) \cdot \log(q(x)) \tag{3.56}$$

In particular, if $q(x)$ is the ratio between two probability distributions $p^{'}$ and p:

$$-\sum_x p(x) \cdot \log \frac{p^{'}(x)}{p(x)} \geq 0 \tag{3.57}$$

where the first member is a quantity called, in information theory, relative *entropy* of the distributions.

The goal of the EM algorithm is to increase, and possibly maximize, the likelihood of the parameters of a probabilistic model M with respect to a set of data s, results of a stochastic process that involves an unknown process π. If $\{\theta^0\}$ are the current parameters of the model, we want to obtain a new set of parameters $\{\theta\}$ such as:

$$\log P(s|\theta, M) - \log P(s|\theta^0, M) \geq 0 \tag{3.58}$$

Introducing the hidden variables:

$$P(s|\theta, M) = \frac{P(s, \pi|\theta, M)}{P(\pi|s, \theta, M)} \tag{3.59}$$

And, passing to the logarithm representation:

$$\log P(s|\theta, M) = \log P(s, \pi|\theta, M) - \log P(\pi|s, \theta, M) \tag{3.60}$$

Multiplying by the probability distribution of the hidden variable given tha actual parameters, $P(\pi|s, \theta^0, M)$ and summing up over all the values that the hidden variable can assume:

$$\log P(s|\theta, M) = \sum_{\pi} P(\pi|s, \theta^0, M) \cdot (\log P(s, \pi|\theta, M) - \log P(\pi|s, \theta, M)) \tag{3.61}$$

If we define an auxiliary function $Q(\theta|\theta^0)$, as the expectation value of the logarithm of the joint probability of s and π over the possible values of the hidden variable:

$$Q(\theta|\theta^0) = \sum_{\pi} P(\pi|s, \theta^0, M) \cdot \log P(s, \pi|\theta, M) \tag{3.62}$$

The expression to maximize is:

$$\begin{aligned} \log P(s|\theta, M) - \log P(s|\theta^0, M) &= Q(\theta|\theta^0) - Q(\theta^0|\theta^0) - \qquad (3.63) \\ &- \sum_{\pi} P(\pi|s, \theta^0, M) \cdot \log \frac{P(\pi|s, \theta, M)}{P(\pi|s, \theta^0, M)} \end{aligned}$$

The third term of the second member of the previous expression is the relative entropy of the distributions $P(\pi|s, \theta, M)$ and $P(\pi|s, \theta^0, M)$ that is always positive.

As a consequence:

$$\log P(s|\theta, M) - \log P(s|\theta^0, M) \geq Q(\theta|\theta^0) - Q(\theta^0|\theta^0) \tag{3.64}$$

This inequality represents the core of the EM algorithm. Indeed, if it is possible to calculate a set of parameters $\{\theta^0\}$ such that the difference between the Q functions is positive, this will increase the likelihood of the model with respect to the data. In particular, the attention is on the values $\{\theta^{MAX}\}$ that maximize that difference:

$$\theta^{MAX} = argmax_{\theta} Q(\theta|\theta^0) \tag{3.65}$$

The algorithm has two main phases:

- The computation of the expectation value $Q(\theta|\theta^0)$, staring from the parameters of the actual model
- The maximization of $Q(\theta|\theta^0)$ as a function of $\{\theta\}$ and the update of the model

Starting from an initial hyothesis on the parameters of the model, these two steps are iteratively applied till the convergence, when updating the parameters do not further increase the likelihood. The algorithm, at each iteration, ensures the achievement of a local maximum, instead of the global maximum likelihood. Moreover, in some cases it is not possible to compute exactly the maximization step, or at least to compute it in an efficient and not time consuming way. In these cases, the requirement of the maximization is no longer valid and we look for a set of parameters that simply make positive the second member of Eq.3.64. This latter conditions describes the algorithms generally known as generalized EM algorithms ([Dempster AP *et al.*, 1977], [Neal and Hinton , 1998]).

The EM algorithm is the most suitable one for training HMMs on a set of sequences, if there is a lack of knowledge on the paths that generated them. This paths are the hidden variables of the problem. In this case, the implementation of the algorithm is known as *Baum-Welch algorithm* [Baum LE, 1972]. As already described, the first step consists in the computation of the expected value of the logarithm of the joint probability of sequence and path over all the possible paths, given the actual parameters of the model (Eq.3.62):

$$Q(\theta|\theta^0) = \sum_{\pi} P(\pi|s, \theta^0, M) \cdot \log(a_{0\pi^1} \cdot \prod_{t=1}^{L} a_{\pi^t\pi^{t+1}} \cdot e_{\pi^t}(s^t)) \tag{3.66}$$

For simplicity, the sum over all the sequences of the training set is omitted by considering only one sequence. Given a path, it contains a certain number of transitions between the states i and j and a defined number of emissions of a character c from a state k. I indicate this numbers with $A_{ij}(\pi, s)$ and $E_k(c, \pi, s)$ respectively.

The argument of the logarithm in Eq.3.66 can be rewritten as:

$$\begin{aligned} Q(\theta|\theta^0) &= \sum_{\pi} P(\pi|s, \theta^0, M) \cdot \qquad (3.67) \\ &\cdot (\sum_{i=0}^{N}\sum_{j=1}^{N} A_{ij}(\pi, s) \cdot \log a_{ij} + \sum_{k=1}^{N}\sum_{c\in\mathbf{A}} E_k(c, \pi, s) \cdot \log e_k(c)) \end{aligned}$$

where, usually, N is the number of states and $\mathbf{A}$ the alphabet of characters. The sum over all the possible paths involves, in the last expression, only $A_{ij}(\pi, s)$ and $E_k(c, \pi, s)$, thus their expectation values can be defined:

$$A_{ij}(s) = \sum_{\pi} P(\pi|s, \theta^0, M) \cdot A_{ij}(\pi, s) \tag{3.68}$$

$$E_k(c,s) = \sum_{\pi} P(\pi|s,\theta^0,M) \cdot E_k(c,\pi,s) \tag{3.69}$$

Again, rewriting Eq.3.67:

$$Q(\theta|\theta^0) = \sum_{i=0}^{N}\sum_{j=1}^{N} A_{ij}(s) \cdot \log a_{ij} + \sum_{k=1}^{N}\sum_{c\in\mathbf{A}} E_k(c,s) \cdot \log e_k(c)) \tag{3.70}$$

Summing up over all the sequences in the training set:

$$Q(\theta|\theta^0) = \sum_{i=0}^{N}\sum_{j=1}^{N} A_{ij} \cdot \log a_{ij} + \sum_{k=1}^{N}\sum_{c\in\mathbf{A}} E_k(c) \cdot \log e_k(c)) \tag{3.71}$$

where A_{ij} and $E_k(c)$ are the sums of $A_{ij}(s)$ and $E_k(c,s)$ over all the sequences.

A_{ij} and $E_k(c)$ can be efficiently computed starting from the parameters of the model. In this case, the maximization step can be achieved exactly. Operatively, the second member in Eq.3.71 should be maximized, as a function of a_{ij} and $e_k(c)$, following the constraints imposed on them. The computation is possible thanks to the Lagrange multipliers technique, in which $2N$ new variables are introduced, λ_k and μ_i, to satisfy the constraints. Thus, the function to maximize now is:

$$\begin{aligned} f(a_{ij}, e_k(c), \lambda_k, \mu_i) &= \sum_{i=0}^{N}\sum_{j=1}^{N} A_{ij} \cdot \log a_{ij} + \sum_{k=1}^{N}\sum_{c\in\mathbf{A}} E_k(c) \cdot \log e_k(c) - \\ &- \lambda_k \cdot (\sum_{c\in\mathbf{A}} e_k(c) - 1) - \mu_i \cdot (\sum_{j=1}^{N} a_{ij} - 1) \end{aligned} \tag{3.72}$$

Imposing the partial derivatives to be equal to zero, we obtain the following equations system:

$$\begin{aligned} \frac{\partial f}{\partial a_{ij}} &= \frac{A_{ij}}{a_{ij}} - \mu_i = 0 && \forall\ i,j \in \{states\} \\ \frac{\partial f}{\partial e_k(c)} &= \frac{E_k(c)}{e_k(c)} - \lambda_k = 0 && \forall\ k \in \{states\}, \forall\ c \in \mathbf{A} \\ \frac{\partial f}{\partial \mu_i} &= \sum_{j=1}^{N} a_{ij} - 1 = 0 && \forall\ i \in \{states\} \\ \frac{\partial f}{\partial \lambda_k} &= \sum_{c\in\mathbf{A}} e_k(c) - 1 = 0 && \forall\ k \in \{states\} \end{aligned} \tag{3.73}$$

For a N states HMM that emits on a alphabet $\mathbf{A}$, this system of $N \cdot (N + A + 2)$ equations is solved by:

$$a_{ij} = \frac{A_{ij}}{\sum_{j=1}^{N} A_{ij}} \tag{3.74}$$

$$e_k(c) = \frac{E_k(c)}{\sum_{c \in \mathbf{A}} E_k(c)} \tag{3.75}$$

The computed parameters are positive and normalized. The expected values for the number of transitions (A_{ij}) and for the number of emissions ($E_k(c)$) can be computed using the values of the matrices f and b adopted in the forward and backward algorithms. Indeed, the expected number of transitions between the states i and j for the emission of the character in position t of a sequence s is:

$$\begin{aligned} P(\pi^t = j, \pi^{t-1} = i | s, \theta, M) &= \frac{1}{P(s|\theta, M)} \cdot P(s^1 s^2 ... s^{t-1}, \pi^{t-1} = i | \theta, M) \cdot \\ &\cdot \; P(\pi^t = j | \pi^{t-1} = i, \theta, M) \cdot \\ &\cdot \; P(s^t | \pi^t = j, \theta, M) \cdot P(s^{t+1} s^{t+2} ... s^L | \pi^t = j, \theta, M) \\ &= \frac{1}{P(s|\theta, M)} f_i(t-1) \cdot a_{ij} \cdot e_j(s^t) \cdot b_j(t) \end{aligned} \tag{3.76}$$

Summing over all the positions and over all the sequences of the training set:

$$A_{ij} = \sum_{s \in D} \frac{1}{P(s|\theta, M)} \cdot \sum_{t=1}^{L} f_i(t-1) \cdot a_{ij} \cdot e_j(s^t) \cdot b_j(t) \tag{3.77}$$

$E_k(c)$ can be computed with the same procedure:

$$\begin{aligned} P(s^t = c, \pi^t = k | s, \theta, M) &= \frac{1}{P(s|\theta, M)} \cdot P(s^1 s^2 ... s^t, \pi^t = k | \theta, M) \cdot \\ &\cdot \; P(s^{t+1} s^{t+2} ... s^L | \pi^t = k, \theta, M) \cdot \delta(s^t, c) \cdot \\ &= \frac{1}{P(s|\theta, M)} f_k(t) \cdot b_k(t) \cdot \delta(s^t, c) \end{aligned} \tag{3.78}$$

where $\delta(s^t, c)$ is a function that assumes the value 1 only when s^t is equal to c. Thus,

$$E_k(c) = \sum_{s \in D} \frac{1}{P(s|\theta, M)} \sum_{t=1}^{L} f_k(t) \cdot b_k(t) \cdot \delta(s^t, c) \tag{3.79}$$

As explained when discussing the three main problems of HMMs, the association of a label $l(k)$ at each state k of an HMM allows to design predictors that map a sequence $s^1 s^2 ... s^L$ to a string of labels $l^1 l^2 ... l^L$. These predictors are trained on pairs $s - l$ of known associations. The training procedure substantially remains that described in the previous sections, taking into consideration that not all the paths between the states of a model are allowed to generate a sequence of the training set, but only the paths coherent with the string corresponding to the label of the sequence. For this reason, all the expected values are computed only over the possible paths. The easiest way to do this is by computing the matrices of the forward and backward algorithms, considering only these paths. This means to fix at zero, in the recursion expressions, the $f_k(t)$ and $b_k(t)$ values when l^t is different from $l(k)$ (the label of the position t is not the same of the state k).

Chapter 4

The effect of backbone on the small-world properties of protein contact maps

4.1 From protein structure to contact maps

Protein structures are described by the coordinates (CO-representation) of the atoms that concur to constitute the macromolecule. For a protein with n atoms we need $3n$ numbers (x,y and z coordinates for each atom) to specify its three-dimensional (3D) structure. An alternative view is to consider the *distance matrix* (DM). The distance matrix is a symmetric matrix that contains in its cells the Euclidean distance between each pair of atoms. If the number of atoms is n we need n^2 elements. Since the matrix is symmetric (the distance between atoms i and j is the same of that between j and i) the real number of needed elements is only $n(n-1)/2$. Both representations, namely the coordinates and the distance matrix, are equivalent: one representation can be converted into the other. DM can be computed from the CO-representation simply by evaluating the Euclidean distance between each pair of atoms: values stored in the appropriate DM cell uniquely identify the pair i and j. Conversely, to go from DM to CO is not so trivial. There exists a Lagrange theorem [Havel TF, 1998] that states that once that DM is diagonalized, the three eigenvectors that correspond to the three highest eigenvalues are the coordinate axes (x, y and z). By projecting the DM values on these three eigenvectors we obtain back the atom coordinates. Actually, there are two solutions, but the chirality of the molecule routinely can help in selecting the correct one [Havel TF, 1998].

The main advantage of adopting a DM representation, that has far more elements than the coordinate-based representation, arises when only a part of the data is known, for example in low resolution NMR experiments. Another advantage of DM is that the protein is represented in a framework that automatically incorporates translational and rotational invariance and this, in principle, is more suitable for learning approaches. Quite often, in order to simplify the protein representation, not all protein atoms are taken into account and residues are considered as unique entities. In this case the distance matrix has a number of rows (and columns) equal to the residue numbers.

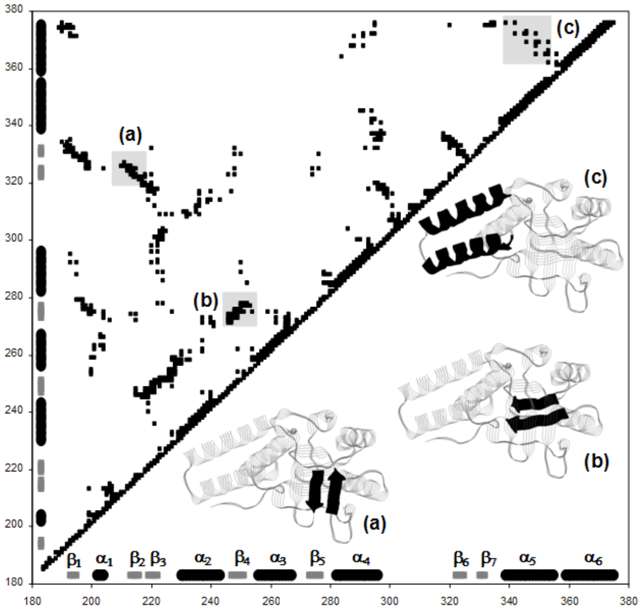


Fig. 4.1. : Contact map of protein HSP-60 protein fragment (PDB id: 1kid).

Each DM entry is then the distance between residue i and j. The distance between two residues can be defined in different ways:

- The distance between a specific pair of atoms (such as CA-CA or CB-CB)
- The shortest distance among the atoms belonging to residue i and those belonging to residue j
- The distance between the centres of mass of the two residues.

These choices provide enough information to build the protein backbone. Starting from the protein DM and selecting an arbitrary distance cut-off, a further simplified representation can be obtained: the *protein contact map* (CM). CMs are binary symmetric matrices whose elements different from 0 (and set to 1) represent the contacts between residues (black dots in the upper triangle of Fig.4.1). In more details, given a DM and a defined threshold T the corresponding CM can be computed as:

$$CM[i,j] = 1\ if DM[i,j] < T \tag{4.1}$$

$$CM[i,j] = 0\ if DM[i,j] \geq T \tag{4.2}$$

While the problem of reconstructing the protein coordinates from the DM has a well known solution, there are not analogous theorems for CMs. However, some empirical applications have been built to address this issue. The obtained results indicate that (at least for the tested proteins) it is possible to reconstruct the CO-representations from CMs [Vendruscolo and Domany, 1999, Bohr *et al.*, 1993, Fariselli *et al.*, 2001, Galaktionov and Marshall, 1994].

CMs are more suited than DMs for learning problems, because their binary nature can be regarded as a classical problem of a two-state classification and this has been thoroughly studied. There are several machine learning methods available to address the problem of the prediction of CM from the protein residue sequence [Baldi and Brunak, 2001]. Moreover, it has been shown that the empirical reconstruction algorithms are quite insensitive to high levels of random noise in CMs, so that for reconstructing the 3D structure of the protein it is not necessary to correctly predict all contacts [Vendruscolo and Domany, 1999, Bohr *et al.*, 1993, Fariselli *et al.*, 2001].

On the other hand, there is no theory on CMs that can help to define the limits and the strength of this representation. For instance, the effect of the contact threshold on the information content is not theoretically assessable. For this reason different researchers adopt different protein representations and contact thresholds. Moreover, CM prediction is an intrinsically not local problem. Also, this is a very difficult problem to deal with, since a contact between two residues poses constraints on the feasibilities of all other contacts. The implemented algorithms to reconstruct protein structure starting from CM prove that for a wide range of distance cut-offs the contact map is a good representation of the protein backbone conformation. It is possible to reconstruct the structure in the best cases with a deviation of less than 3 Å.

After it was shown that it is possible to reconstruct protein structures from their CMs, several researchers have been predicting contact maps starting from protein sequence information. A possible approach is to learn the correlation between sequence and CM using machine learning tools.

In this respect several methods have been introduced: neural networks that exploit multiple sequence alignments [Fariselli *et al.*, 2001], [Punta and Rost, 2005], Hidden Markov Models [Shao and Bystroff, 2003], support vector machines - [Zhao and Karypis, 2003], genetic programming [MacCallum, 2004] and recurrent neural networks [Pollastri and Baldi, 2002].

In the following I will present a study on protein contact maps regarded not only as symmetric matrices but as graphs. Actually, CM is the representation of a graph as an adjacent matrix, where the contacts are the edges and the residues are the nodes. It is also useful to distinguish between short-range and long-range contacts. The distinction between short-range and long-range contacts is not due to the type of the involved interaction, or to the spatial distance, but it is referred to the relative sequence separation. Contacts between residues that are separated less than a given number of residues $S(S = |i - j|)$ are said to be short-range. Conversely if the sequence separation is greater than S they are said to be long-range. The choice of S is arbitrary, but it is commonly accepted that $S \leq 7 - 10$ represents short-range contacts, while $S > 7 - 10$ represents long-range ones.

For applications in the field of protein structure prediction and/or reconstruction from experimentally determined contact maps, it is very important to verify whether the characteristic path length (L) and clustering coefficient (C) of the graph are indeed values that reveal characteristic features of protein contact maps. This may be particularly relevant in order to highlight constraints for the prediction of protein contact maps.

4.2 Small-world and protein structures

As described in Chapter 2, small-world topology is an alternative graph topology lying between random and regular network topologies. To describe their characteristics, Watts and Strogatz [Watts and Strogatz, 1998, Watts DJ, 1999] introduced two measures: the characteristic path length (or average path length) L and the clustering coefficient C. L is defined as the number of edges in the shortest path between two vertices, averaged over all pairs of vertices:

$$L = \frac{2}{N(N-1)} \sum_{N-1}^{i=1} \sum_{j=i+1}^{N} L_{ij} \tag{4.3}$$

where L_{ij} is the shortest path length between vertices i and j. L is a measure of the network dimension. C in turn provides a measure of the average fraction of neighbours of a given node which are also neighbours to each other. Formally C is computed as:

$$C = \frac{1}{N} \sum_{k=1}^{N} \frac{n_k}{N_k(N_k-1)/2} \tag{4.4}$$

where for the k-th node N_k is the number of its neighbours while n_k is the number of contacts among them. The normalization factor $N_k(N_k-1)/2$ defines the maximum number of possible connections among the N_k nodes. Regular networks have large values of both L and C, while random ones have small L and small C. Small-world graphs have small L but large C. L measures the typical separation between two vertices in the graph (a global property) and C is a measure of local clustering or cliquishness of a typical neighbourhood (a local property) [Watts and Strogatz, 1998].

In recent years several network studies on proteins have emphasized the relevance of the graph topology to shed light on protein function and dynamics. In a seminal paper Vendruscolo *et al.* [Vendruscolo *et al.*, 2002] showed for the first time the small-world behaviour of protein structure networks. The authors considered two residues as connected if the distance between their CA atoms is less than a threshold distance fixed at 8.5 Å. By analyzing a data set of 978 representative proteins it was found that the average value of L is 4.1±0.9 and that of C is 0.58±0.04. These values were compared with those obtained for random and regular graphs. By assuming that K is the average number of links in the graph (the average number of contacts in a protein) and N is the number of vertices, then $L_{random} \approx lnN/lnK$ and $C_{random} \approx K/N$, $L_{regular}$ is $\approx N(N+K-2)/2K(N-1)$ and $C_{regular}$ is $\approx 3(K-2)/4(K-1)$ [Dorogovtsev and Mendes, 2000]. Values of 2.4±0.3 and 0.08±0.06 were reported for L_{random} and C_{random} respectively. $L_{regular}$ and $C_{regular}$ were 10.4±7.0 and 0.67±0.04, respectively.

The small-world paradigm was adopted also for homopolymers obtained with a contact map dynamics [Vendruscolo and Domany, 1999] and for atomic clusters obtained with Lennard-Jones interactions with a Monte Carlo method [Andricioaei *et al.*, 2001]. In both cases the values of C and L were found similar to those of proteins, indicating a small-world topology also for these systems. It was therefore concluded that protein chain connectivity plays a minor role in the small-world behaviour and that for a globular protein the small-world character would mainly arise from the overall geometry (surface to volume ratio) [Vendruscolo *et al.*, 2002].

Greene and Higman [Greene and Higman, 2003] made a distinction between long-range and short-range interaction graphs in protein structures. They adopted an all-atom representation of the proteins instead of the less informative CA simplified representation and allowed a contact between two residues when at least one pair of their atoms was within 5 Å from each other. By this, multiple links between residues are allowed. The small-world property was analysed on a set of 65 non redundant proteins divided in nine highly populated fold types representing the four SCOP protein classes: all-α, all-β, α/β, $\alpha+\beta$. Interestingly, these authors found a difference of the behaviour between what they called networks of short-range and long-range contacts. Interactions are considered short-range or long range if they occur between residues they have a separation < 10 residues or ≥ 10 residues, respectively. Long-range interaction networks do not show a small-world topology and they are not scale-free but single-scale, with fast decaying exponential tails in the degree distribution.

Scale-free networks are small world but small-world networks are not necessarily scale-free [Barabasi and Albert, 1999]. In the protein world, contact maps are not scale-free networks. A scale-free connectivity follows a power law $p(k) \approx k^{-\gamma}$ (where k is the number of links of a node and p is the probability of a node to have k links). In a typical scale-free network $2 \leq \gamma \leq 3$. The distribution of both long and short-range contacts reveals a tendency to a bell-shaped Poisson curve which is typical of random networks and not of scale-free ones [Greene and Higman, 2003]. In Figure 4.2 I plotted the result of a study on my data set of contact maps (described in the next section), confirming the non-scale-free behaviour of contact distribution.

By following the short and long-range contact distinction, I also computed C and L values for my protein data set, apllying the Floyd-Warshall algorithm (described in Chapter 3) to the shortest paths computation. The results are shown in Figure 4.3, confirming that long-range contacts can be modelled by a random graph and that small world properties emerge only when the whole contact map is considered.

Atilgan *et al.* [Atilgan *et al.*, 2004], studying the small-world topology of a significant set of proteins, demonstrated that the core residues have the same local packing irrespective of the protein length and provided an interesting link with protein dynamics [Atilgan *et al.*, 2007]. I found the same trend on my data set (Fig.4.4).

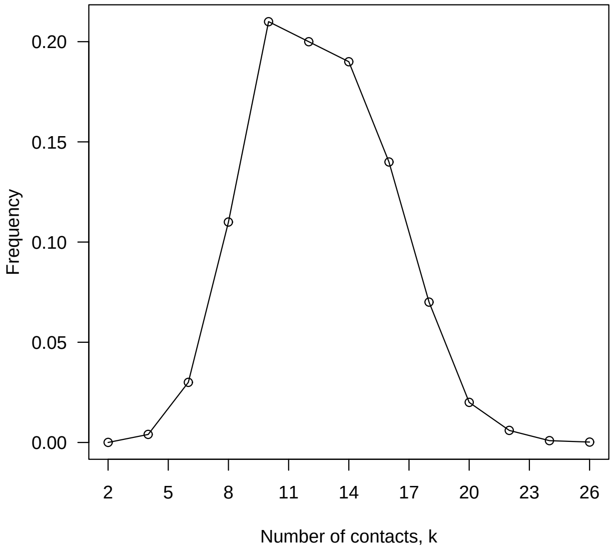


Fig. 4.2. : Residue contact distribution $p(k)$.

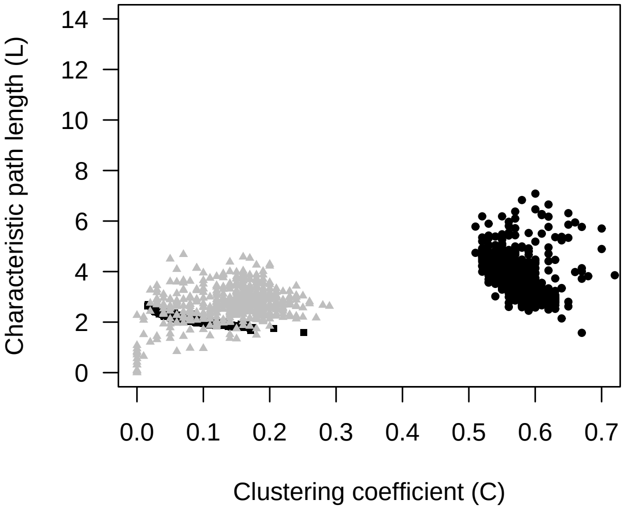


Fig. 4.3. : Distribution of the characteristic path length (L) as a function of the clustering coefficient (C). Black circles represent the complete protein contact maps, grey triangles are protein long-range contacts while random networks are labeled with black squares.

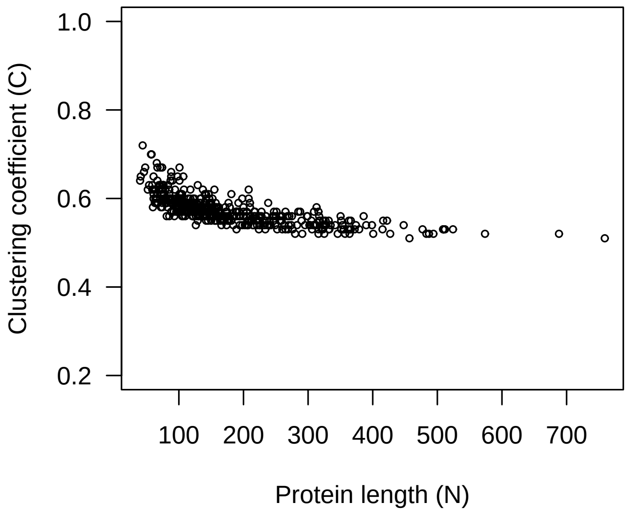


Fig. 4.4. : Clustering coefficient (C) as a function of the protein length (N).

According to the results of Bagler and Sinha [Bagler and Sinha, 2005], protein structure networks exhibit small-world behaviour regardless of their structural class (namely SCOP classification [Murzin *et al.*, 1995]), and tertiary structure, even if all-α and all-β classes slightly differ. A thorough investigation study reveals a marginal but consistent difference in the C index value of all-α and all-β proteins. I show my results in Figure 4.5. When considering the average C values, I found that they are 0.597 for all-α and 0.551 for all-β proteins, respectively. These values confirm the difference previously reported. This difference may be due to the larger geometrical compactness of α-helices as compared to β-sheets. The data set that we have analyzed contains 113 all-α proteins and 110 all-β proteins.

All these researches were performed using different protein representations, such as all-atom distances between residues, or distances between the CA-atoms as residue representatives, and weighted or un-weighted links ([Atilgan *et al.*, 2004, Böde *et al.*, 2007, Atilgan *et al.*, 2007]).

However in spite of several efforts it is still unclear whether the L and C values typical of a protein set are due to the physical complexity of real proteins or may be obtained also by other graphs. I verified that L and C values, computed on random graphs and generated imposing constraints similar to those due to collapsed necklaces, are nearly indistinguishable from those computed using the real protein contact maps. This result supports the idea that the particular L and C values of the proteins are due to the local geometrical interactions imposed by the protein backbone in collapsed structures, rather independently of the protein residue composition and secondary structure types. Furthermore, I will show that these results are independent of the adopted protein representation.

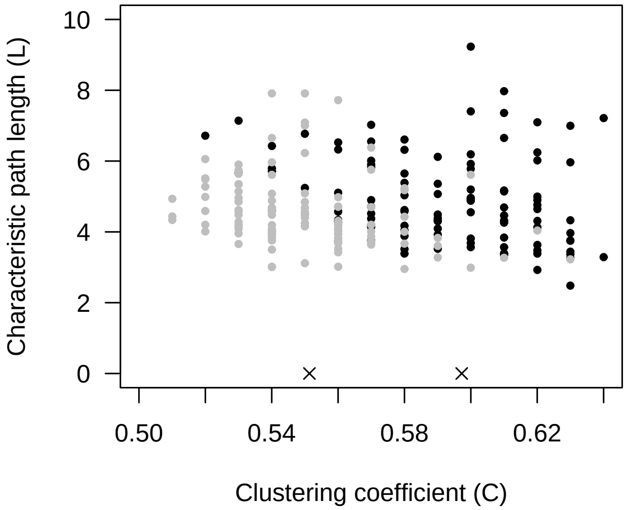


Fig. 4.5. : *L*-*C* plot for all-α proteins (black) and all-β proteins (grey). The two crosses indicate the average values for the two groups.

4.3 The data set

To test the proposed hypothesis I used a significant and non-redundant data set of protein structures. In particular, I extracted from the PDB a subset of protein structures that fulfil the following criteria:

- High-resolution structures (resolution <2.5 Å obtained with X-ray experiments)

- Sequence length $\geq$ 40 residues

- No sequence redundancy (no sequence of the data set has a sequence identity $\geq$ 25% to any other sequence of the data set)

- No holes in the protein chains (I removed all the proteins whose structures have incomplete coordinates)

After this procedure I ended up with 1753 protein chains that are included in my data set.

4.4 Protein representation and threshold cut-off

Among the different protein representations the most widely adopted are those based on carbon-alpha trace (CA-trace), carbon-beta (CB) and all-atom (without taking into account the hydrogen atoms since they are not detected by X-ray experiments). Of course the all-atom representation is the most accurate, since it captures the essence of the physical contacts. Moreover it has been suggested that for this representation the natural contact threshold is in the range of 4.5-5 Å, since this is the largest distance that does not allow insertion of water molecules between two residues [Hinds and Levitt, 1992]. On the opposite side, the CA-trace is the coarsest representation, since it is based only on the "cartoon-like" information indicative of general protein structure shape. The strategy in the following is therefore to compare these two extreme and different protein representations when measuring the C and L values (Eq. 4.3 and Eq. 4.4).

The most relevant point is to select the contact thresholds for comparing the two representations. Considering the all-atom as the "master" representation (with its natural threshold 4.5-5 Å), the selection of the corresponding CA threshold can be obtained by computing the *average Hamming distance* (AHD) between the contact maps calculated using the two representations on my data set. The Hamming distance between two strings of equal length is the number of positions with mismatching characters. Here, AHD is formally defined as:

$$AHD = \frac{1}{N}\sum_{i=1}^{N}\frac{Hd(CM_i^{all-atom}(T_{AA}), CM_i^{CA}(T_{CA}))}{M_i} \tag{4.5}$$

where the index i runs from 1 to N (the number of proteins in the data set), $CM_i^R(T_R)$ is the contact map computed using the protein representation R (AA = all-atom or CA) with the threshold distance T_R. $Hd(x, y)$ is the Hamming distance between x and y maps and M_i is the number of elements of the i-th contact map. In Figure 4.6 the resulting AHD as function of the CA-threshold distance when the all-atom threshold is set to 5 Åis reported. I found that AHD reaches its minimum for a value corresponding to the CA-threshold of 7 Å. This threshold is still the minimum AHD value for each all-atom thresholds set in the natural range of 4.5-5 Å.

Using the all-atom threshold of 5 Åand the one that minimizes the AHD for the CAs (7 Å), we computed the L and C values for both representations, together with the corresponding regular and random graphs. In Figure 4.7 the results show that with this choice of the CA threshold the two average values are exactly the same (CA: C=0.56±0.03, L=5.3±1.4; all-atom: C=0.56±0.04, L=5.3±1.4). The random L and C values were obtained with a theoretical model [Dorogovtsev and Mendes, 2000], $C_{rand} \approx K/N$, $L_{rand} \approx lnN/lnK$, where N is the number of vertices and K is the average number of neighbours of the graph. The regular C and L values were obtained with the following approximation $C_{reg} \approx 3(K-2)/[4(K-1)]$, $Lreg \approx N(N+K-2)/[2K(N-1)]$. ($C_{rand}$=0.05±0.03, L_{rand}=2.6±0.3, C_{reg}=0.64±0.01, L_{reg}=14.9±8.7).

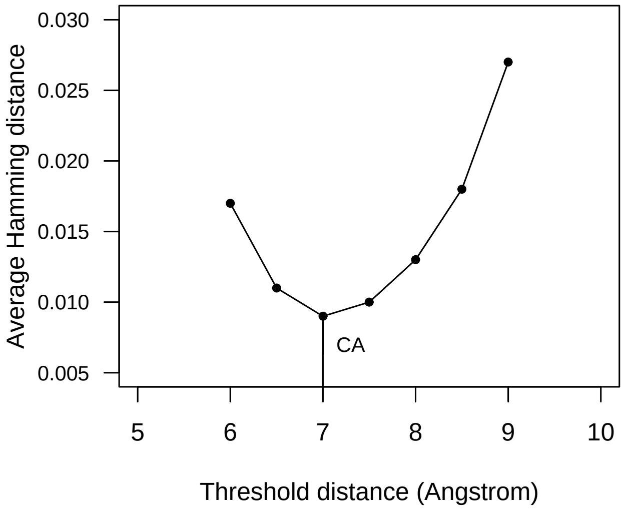


Fig. 4.6. : Average Hamming distance as a function of the CA-threshold distance when the all-atom one is set to 5 Å.

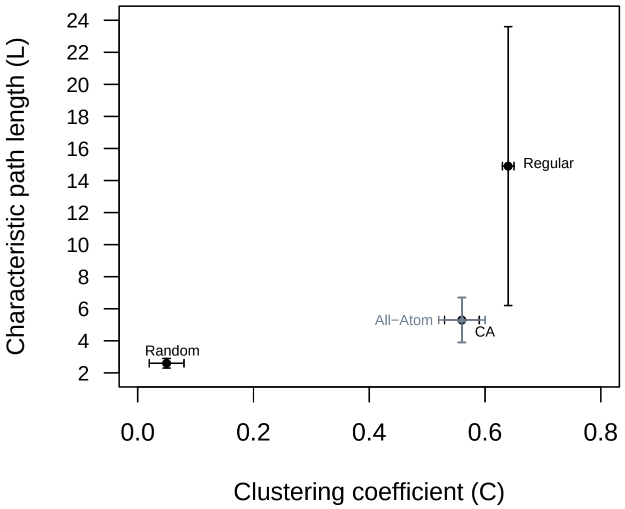


Fig. 4.7. : Characteristic path length (L) and clustering coefficient (C) for the protein contact maps computed using CA-representation (threshold set to 7 Å, C=0.56±0.03, L=5.3±1.4) and all-atom representation (threshold set to 5 Å, C=0.56±0.04, L=5.3±1.4), with respect to the corresponding values obtained for random and regular graphs.

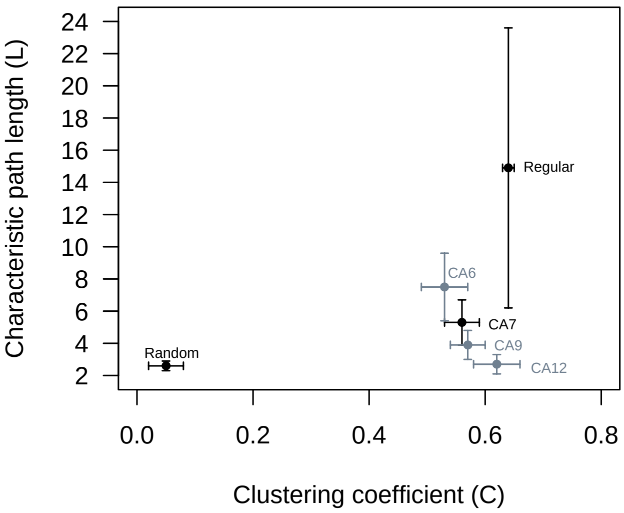


Fig. 4.8. : Characteristic path length (L) and clustering coefficient (C) as a function of the different threshold cut-offs (6-12 Å) for the CA representation.

This finding indicates that, at least for the small-world graph properties, the different representations are equivalent, provided that the contact threshold that minimizes the AHD is selected. On the other hand computing C and L values using different thresholds on the same protein representation (CA in Fig.4.8), there are visible differences, indicating that the choice of the contact threshold may lead to different L and C values.

4.5 The effect of the protein backbone

From the results shown in Figure 4.7 and Figure 4.8 and from the literature it is clear that, for any undirected graph, it is possible to determine whether the computed L and C values fall in the small-world region, like real proteins. It is then relevant to investigate whether it is possible to build graphs that are randomly generated and at the same time are protein-like in terms of L and C numbers.

A major characteristic of a protein is its *backbone connectivity*. Previous studies have been shown that a great portion of the contribution to the contacts in the first coordination shell is actually due to the chain connectivity [Atilgan *et al.*, 2004]. In my representation, giving the chosen threshold, preserving this property means to preserve the contacts included in the first two diagonals of the matrix, namely $\{i, i+1\}$ and $\{i, i+2\}$. The resulting graph describes a two-dimensional lattice, with L and C values clustering in the regular region. Obviously neither real nor randomized protein graph has this property.

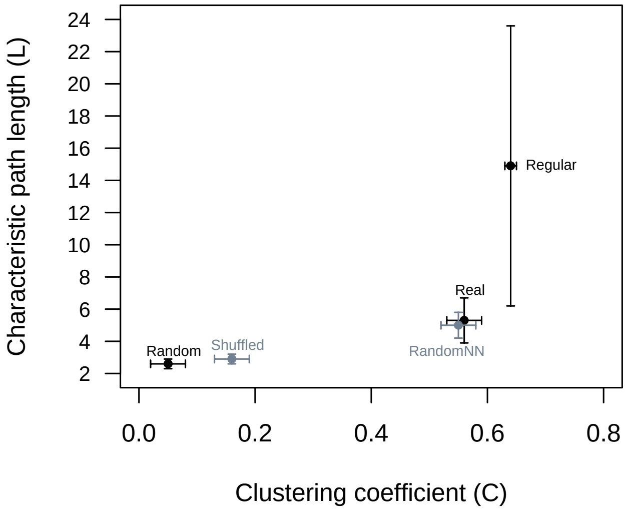


Fig. 4.9. : Characteristic path length (L) and clustering coefficient (C) of the real contact maps (real have C=0.56±0.03, L=5.3±1.4) with respect to Random (randomizing all contacts), Shuffled (the randomly generated contact maps using complete uncorrelated shuffling) and RandomNN the randomly assigned locally correlated contacts (C=0.55±0.03, L=5.0±0.8).

There are many ways of generating random contact maps starting from the real contact map of a protein [Atilgan *et al.*, 2004]. I will show that three different randomization procedures affect to different extents the final L and C values of the obtained randomized sets.

- A first very simple way to generate random contact maps from the real ones is just by randomly shuffling the original contacts. However by adopting this procedure also all the contacts of the protein backbone are destroyed and, as expected, the final L and C values cluster in the typical random region (*Random* in Fig.4.9).

- A second naive way is to keep the backbone contacts while randomizing the edges. This can be done maintaining the contacts generated by the backbone covalent structure (defined by residues in the diagonals $\{i, i+1\}$ and $\{i, i+2\}$ of the contact matrix) and shuffling all the remaining contacts in the map. This random-shuffling procedure is indicated with *Shuffled* in Fig.4.9).

- However, even this second randomization strategy does not take into account the backbone-induced correlation. Think of a necklace: bringing together two pearls i and j, then the connecting thread brings also into proximity the pearls that are close in sequence to i and j. This is also true for collapsed protein structures and it leads to a correlation between nearest neighbours with a general higher probability of forming contacts among residues that are near in sequence than those far apart. This randomization procedure that takes into consideration nearest neighbours is labeled *RandomNN* (in Fig.4.9). With these constraints, for each real protein contact map a large number of random contact maps has been generated, by keeping the number of vertices of each real protein contact map and by:

1. assigning contacts to the first two diagonals (this are simply the backbone contacts equal for any protein folding type, this ensures the backbone connectivity)

2. randomly selecting a pair of residues i and j with a probability that linearly decreases with their sequence distance (defined as $|i-j|$)

3. taking as contacts all the 9 residue pairs generated by the first nearest neighbours (which are the Cartesian product of $\{i-1, i, i+1\} \times \{j-1, j, j+1\}$).

This last procedure is iterated until the number of contacts in the random graph is close (±8 edges) to those of the corresponding real protein. The RandomNN procedure generates random graphs that take into consideration the basic constraints due to the backbone connectivity. It has been already demonstrated that random contact maps are unlikely to represent physical protein structures [Vendruscolo *et al.*, 1999], and this should hold also for the RandomNN graphs.

Furthermore, I also tested if the RandomNN contact maps represent physical objects by submitting them to a recently developed method that reconstructs protein three-dimensional structures starting from contact maps [Vassura *et al.*, 2007]. None of the tested RandomNN contact maps was reconstructed without errors, as opposed to a real protein. In Figure 4.10, an example of a randomly generated contact map with respect to a real contact map of the same dimension and with the same number of contacts is reported. Even though the two maps in the Figure are clearly different, they are indistinguishable in terms of computed L and C values.

By adopting the described RandomNN procedure, I built over 3000 undirected graphs from the real protein contact maps (1753) and I measured the average L and C for the two groups. The results are reported in Figure 4.9. It is evident that real contact maps are similar to the RandomNN ones, supporting the hypothesis that backbone constraints play a fundamental role in determining the small-world properties. This is in agreement with the fact that the shortest characteristic path length computed from reduced networks by screening long-range contacts in a hierarchical manner, maintains the protein-like qualities for a large number of broken contacts [Atilgan *et al.*, 2007].

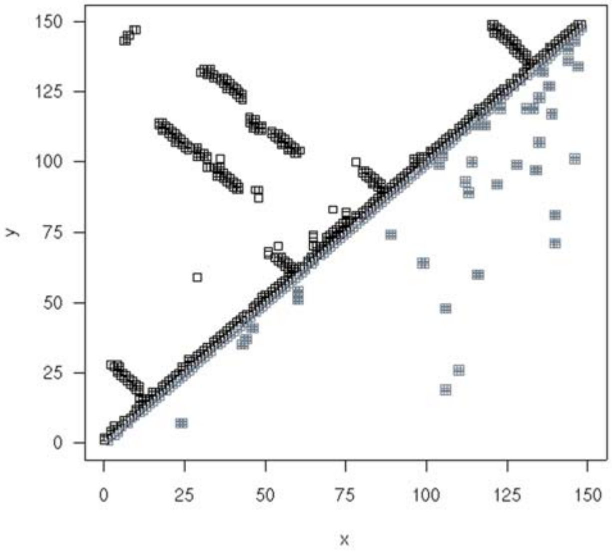


Fig. 4.10. : An example of real protein contact map (PDB code 1amx) is shown in the upper triangle with respect to a randomly generated one computed by assuming a RandomNN procedure (rules 1-3) as detailed in the text (lower triangle).

4.6 Discussion

I demonstrated that the small-world behavior of inter-residue contact graphs is conditioned by the backbone connectivity. In particular, it appears that the characteristic path length (L) and clustering coefficient (C) are not useful quantities for "protein fingerprinting", since L and C values computed for RandomNN-generated contact maps are indistinguishable from those of real proteins. This finding can also explain that:

i) when the short-range contacts (sequence distance $<$ 10 residues) are removed, the corresponding L and C values became indistinguishable from those of random graphs [Greene and Higman, 2003]

ii) when the long-range contacts of proteins are screened in a hierarchical manner, the shortest path lengths of the reduced networks are preserved for a large number of broken contacts [Atilgan *et al.*, 2007]

iii) collapsed structures of homopolymers have C and L values similar to those of real protein structures [Vendruscolo *et al.*, 2002].

Finally, I also showed that my results are independent of the protein representation adopted, since I demonstrated that two very different protein representations (such as CA-trace or all-atom) are indistinguishable in terms of network properties, providing that correct corresponding thresholds are selected.

Chapter 5

Improving coiled-coil prediction with evolutionary information

5.1 Coiled-coils

The coiled-coil is a common protein structural motif [Lupas A, 1996] known to mediate the oligomerization of a large number of proteins [Parry *et al.*,2008]. Its structure is characterized by two or more α-helices that wrap around each other to form a rope-like structure [Gruber and Lupas, 2003]. Their history is as long as the DNA history, since in 1953 both Francis Crick and Linus Pauling proposed the first model of supercoiled helices. Nevertheless, although these observations occurred about 60-80 years ago their impact have only been appreciated in quite recent times.

From an experimental point of view, the most famous X-ray diffraction measurements were made on fibrous proteins that became known as the *keratin-myosin-epidermin-fibrinogen (k-m-e-f)* class. The measures were performed by William Astbury, a student of Sir William Bragg at the University of Leeds (UK). Each diffraction pattern displayed similar features: a meridional reflection with a spacing of 0.515 nm and a group of equatorial/near equatorial reflections at a radial spacing of about 0.98 nm (Fig.5.1). These features specify what became known as the "α-pattern". Later, a third meridional reflection with a spacing of 0.15 nm was added by Perutz.

This pattern, named *α-form*, became the focus of particular attention after a similar diffraction pattern was obtained for the two globular proteins myoglobin and hemoglobin. This finding showed that this α-form generally occurred in proteins, which raised hopes that it would provide the key to a general model for the polypeptide chain.

After exploring possible systematic ways in which a protein chain could fold Pauling proposed two structures, one of which was termed the α-helix. This was predicted to have 3.6 residues per turn and an axial rise per residue of 0.15 nm. Much of the rigidity of the α-helix, now known to be right-handed, arose from the bracing provided by three distinct strands of hydrogen bonds of the type C=O$\cdots$H-N formed between main chain atoms. Each of these hydrogen bonds lies approximately parallel to the axis of the helix.

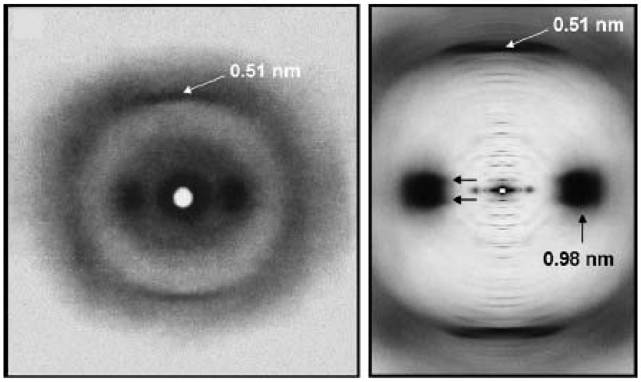


Fig. 5.1. : On the left, X-ray diffraction pattern from a lock of Mozarts hair taken by W.T. Astbury in 1958 (from The Brotherton Collection, Leeds University Library, UK). On the right, X-ray pattern from α-keratin showing the very sharp 0.51 nm meridional arc and complex of equatorial and near-equatorial maxima at a lateral spacing of 0.98 nm. Near-equatorial layer lines having an axial spacing of about 7 nm are associated with the pitch length of the coiled-coil and are indicated by horizontal arrows [Parry *et al.*,2008].

Since the α-helix was generally believed to be only marginally stable in water, it was thought that assembly into larger groupings would be favoured.

The calculated X-ray diffraction pattern of the α-helix contained many of the features that were observed in the α-form from the *k-m-e-f* family. In particular, the 0.15 nm meridional reflection corresponding to the axial rise per residue was expected as was a 0.98 nm maximum on the equator arising from the separation of the α-helices. What was not accounted for, however, was the observation of a 0.515 nm meridional reflection and a non-observed but nonetheless predicted off-meridional reflection on a layer line with a spacing of 0.54 nm. In several papers Crick showed quantitatively that all of the experimental observations were compatible with an α-helical structure provided that several α-helices assembled together in such a manner that each was tilted and coiled around one another with an opposite hand to that of the individual α-helices. This generated a multi-stranded structure now known as the α-helical coiled-coil (Fig.5.2).

The Crick's model is characterized by a *knobs-into-holes* packing of the side-chains and requires a repeating pattern of seven residues over two helical turns (7/2) with an antiparallel supercoil of the helices. This model became the most popular one because of its complete mathematical description [Gruber and Lupas, 2003]. The canonical heptad repeat is routinely represented in the form $[abcdefg]_n$, where about 70–75% of the residues in a and d positions are hydrophobic residues (such as leucine, isolecucine and valine) [Conway and Parry, 1990], resulting in a hydrophobic stripe that spans each helix. Despite of the simplicity of its conformation, the coiled-coil motif contributes to a broad range of functions arising from its peculiar three-dimensional arrangement.

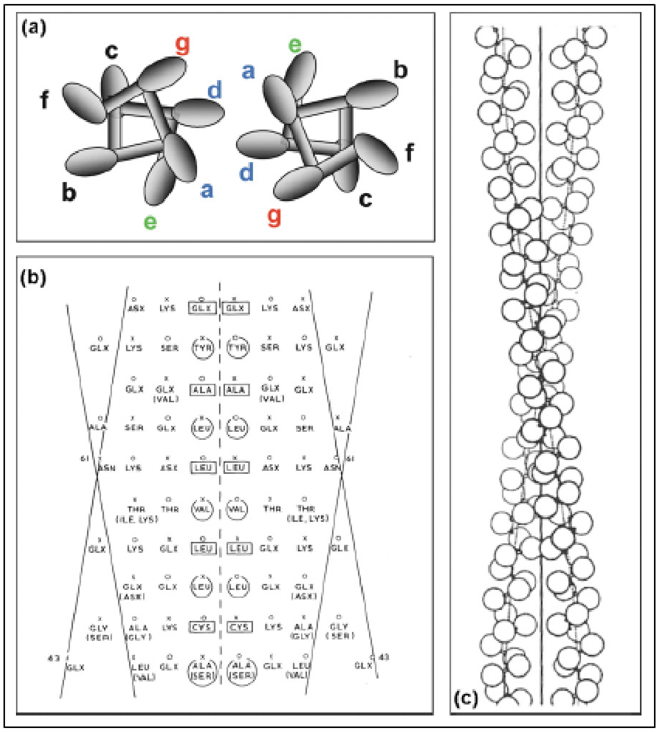

(a)
c
g
e
f
d
a
b
b
a
e
d
f
g
c
(b)
ASX LYS GLX GLX LYS ASX
GLX LYS SER TYR TYR SER LYS GLX
GLX GLX (VAL) ALA ALA GLX (VAL) GLX
ALA SER GLX LEU LEU GLX SER ALA
61 ASN LYS ASX LEU LEU ASX LYS ASN 61
THR (ILE, LYS) THR VAL VAL THR THR (ILE, LYS)
GLX LYS GLX LEU LEU GLX LYS GLX
GLX (ASX) GLX LEU LEU GLX GLX (ASX)
GLY (SER) ALA (GLY) LYS CYS CYS LYS ALA (GLY) GLY (SER)
43 GLX LEU (VAL) GLX ALA (SER) ALA (SER) GLX LEU (VAL) GLX 43
(c)

Fig. 5.2. : (a) Schematic representation of the positions around the two α-helices in a two-chain heptad coiled-coil of the amino acids in the a to g positions. (b) Two superimposed radial nets of the tropomyosin chain, which was the first fibrous protein sequence determined. Adjacent to the line of contact between the two chains (dashed line) there is a high concentration of apolar residues fitting together with knob-into-hole packing. (c) Representation of a heptad (left-handed) two-chain coiled-coil where each amino acid has been represented by a circle [Parry *et al.*,2008].

Whereas Crick presented a rigorous model based on one periodicity (7/2) and backed by equations, Pauling envisaged a broader set of sequence periodicities (4/1, 7/2 and 15/4), leading to supercoils with senses of twist both the same and opposite to those of the constituent helices. His supercoils also included a structurally bundle of six helices coiling around a straight seventh helix. None of his arguments were converted into quantitative parameters and side-chain packing played no part in his considerations. Indeed, in his model, supercoiling was not a result of side-chain packing but of coiled-coil sequences being formed by exact sequence repeats, which caused periodic fluctuations in backbone hydrogen-bond lengths. Unfortunately, his work did not achieve the impact of Cricks one and its contents were mostly forgotten with time. However in 1960, the first high-resolution structure of the protein myoglobin provided the final confirmation of the α-helix, but the helices were arranged without any obvious regularity. In addition, the packing of their side-chains did not follow the knobs-into-holes model, but a less regular packing that became known as “ridges-into-grooves”. Direct confirmation of the coiled-coil model had to wait until the determination of the tropomyosin sequence in 1974, which displayed the hydrophobic heptad-repeat pattern unbroken from N to C terminus, and the structure of influenza hemagglutinin in 1981, which proved the knobs-into-holes packing model. At this point, Cricks model of left-handed coiled-coils built on heptad repeats became canonical.

It was gradually recognized that the Cricks equations were describing an idealized situation. As more coiled-coil proteins were sequenced, the unbroken heptad repeat pattern of tropomyosin became an exception: other sequences were less regular and contained various discontinuities. Two common discontinuities, *skips* (insertions of one residue into the heptad pattern) and *stutters* (insertions of four residues), were identified, but their prevalence in coiled-coils, together with a third discontinuity called *stammer* (insertion of three residues) was only shown much later. Also, coiled-coils were discovered in which the basic periodicity differed globally. Efforts to interpret discontinuities in structural terms (i.e. backbone conformation and side-chain packing) subsequently led to the recognition that some coiled-coil sequences were more suitably described by patterns other than the heptad repeat.

Stutters and stammers break the heptad periodicity disrupting the knobs-into-holes packing. In a canonical coiled-coil the hydrophobic core is formed by the regular interlocking of residues in positions *a* and *d* (*a* layers and *d* layers) in a parallel structure and by the interlocking of these layers in an antiparallel structure (Fig.5.2). The effect of stutters is to shift residues in position a towards the center of the core, which results in a geometry called an *x layer*, while they shift residues in position *d* out of the core and the residues that follow towards position *a*, resulting in a ring of interacting residues around a central cavity (*da layer*). The situation for stammers is analogous, except that residues in position *d* yield the *x* layers and the *da* layers are formed by residues in positions *d* and *a*. In both cases, the knobs-into-holes packing is transformed locally into a knobs-to-knobs interaction.

Generally speaking, sequences that combine hydrophobic patterns of three and four residues are likely to be compatible with the basic coiled-coil structure, while sequences alternating patterns (three then four residues) yield knobs-into-holes packing and succeeding repeating patterns (three then three or four then four residues) knobs-to-knobs packing. Combinations of these basic patterns might lead to an astounding structural diversity, even in closely related proteins (Fig.5.3).

Presently, several examples of coiled-coil structures are known with atomic resolution and are deposited in the PDB [Berman HM *et al.*, 2000]. Furthermore, thanks to Murzin and co-workers' efforts, the annotation of the SCOP database [Murzin *et al.*, 1995] actually represents a useful source of information. SCOP contains manually annotated coiled-coil domains that are labelled with the specific class identifier *h*.

Another relevant step of the coiled-coils structural annotation was the development of SOCKET [Walshaw and Woolfson, 2001], an algorithm that recognises, starting from the protein structure, the canonical knobs-into-holes side-chain

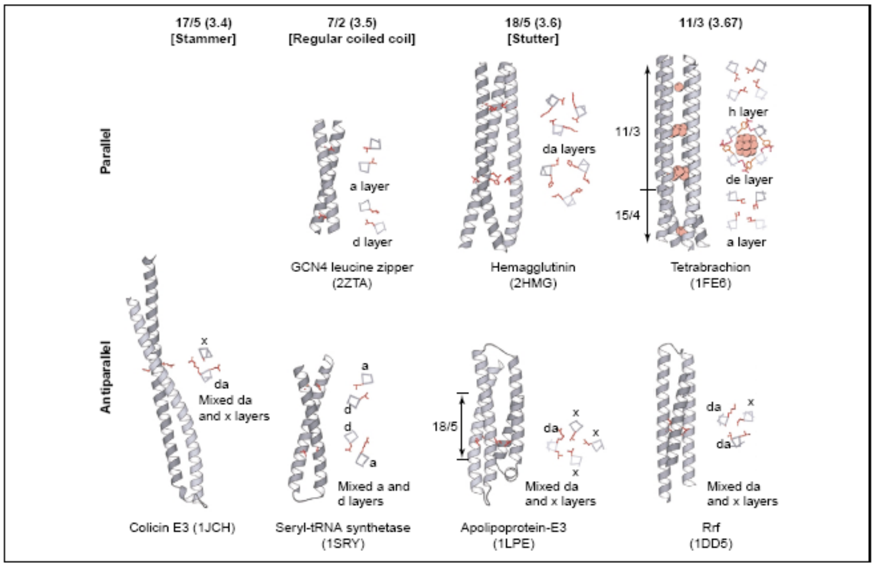


Fig. 5.3. : Periodicities of coiled-coil proteins: parallel and antiparallel coiled-coil structures with heptad and non-heptad periodicities.

packing motif and that is able to distinguish coiled-coils from the great majority of helix-helix packing motifs observed in globular domains. To achieve this, all residues are represented by a centre of mass. A side-chain is classed as a knob if it contacts four or more side-chain centres within a specified packing cut-off. The nearest four side-chains were taken as the corresponding hole. Packing cut-offs were determined empirically by analyzing several classical coiled-coils of different oligomer states and orientations and also on some non-coiled-coil α-helical domains. 7.0 Å and and 7.4 Å are therefore used as cut-offs for the evaluation of the PDB. SOCKET has also been recently adopted to define a periodic table of coiled-coil structures ([Moutevelis and Woolfson, 2009], [Testa *et al.*, 2009]).

5.2 The state of the art in coiled-coil prediction

So far, a number of computational methods have been implemented to identify coiled-coil sequences and to predict coiled-coil regions. Since the heptad repeat is the most informative constraint [Parry *et al.*,2008], all the methods were parameterized on the basis of the heptad module.

The first and widely-used COILS ([Lupas *et al.*, 1991], [Lupas A, 1996]) exploits the residue frequencies computed on the heptads of the experimentally-determined structures known at that time. PAIRCOIL [Berger *et al.*, 1995] and the retrained version PAIRCOIL2 [McDonnell *et al.*, 2006] are based on pairwise residue correlations. MULTICOIL extends PAIRCOIL to the identification of three-stranded coiled-coils too [Wolf *et al.*, 1997]. All these methods are substantially based on the amino acid propensities stored in the Position Specific Scoring Matrix (PSSM). The PSSM stores the seven position (corresponding to the heptad) specific propensities for the 20 amino acids. Every propensity is given by the ratio of the frequency in a given heptad position to the background frequency of the same amino acid. Two scoring matrices, MTK and MTIK are widely used [Lupas *et al.*, 1991, Lupas A, 1996].

HMMs have become a standard technique in sequence analysis. In Chapter 3, the HMMs consistent probabilistic framework and the different good algorithms known for their application [Durbin *et al.*, 1998] were described. Single-sequence based Hidden Markov Models were also developed to address the coiled-coil prediction: MARCOIL [Delorenzi and Speed, 2002] and CCHMM [Fariselli *et al.*, 2007].

It is very well known that evolutionary information in the form of sequence profile routinely increases the overall accuracy of a predictive method [Rost and Sander, 2003]. However, it is interesting to note that only the COILS method was so far modified to exploit evolutionary information. The profile-based version PCOILS [Gruber *et al.*, 2005] substitutes sequence-profile comparisons with profile-profile comparisons.

In this part of my work I introduced, for the first time in this field, a Hidden Markov Model that exploits evolutionary information for discriminating coiled-coil sequences and for locating coiled-coil residues within sequences. The first development of a sequence-profile-based HMM was successfully designed [Martelli *et al.*, 2002] for predicting and discriminating β-barrel membrane proteins. Furthermore, I expanded a recent comparative analysis of coiled-coil prediction methods [Gruber *et al.*, 2006] by testing the available methods on a new blind structurally-determined data set and by scoring them on the basis of per-residue, per-segment and per-protein indices.

5.3 The data set

The only annotated data set publicly available created for developing a predictor is the data set of protein sequences selected for the MARCOIL implementation. However, the same MARCOIL authors stated that the coiled-coil annotations in their database are not reliable [Delorenzi and Speed, 2002].

For this reason, I followed the prescription suggested by Lupas and co-workers [Gruber *et al.*, 2006] by adopting as a safer and more impartial set with respect to those used in literature the intersection between SCOP and SOCKET. I generated my data set of experimentally-determined coiled-coil structures following this suggestion and considering only the intersection between the SCOP coiled-coil class and the output of the SOCKET program. Thus, I selected the protein structures on the basis of the following steps:

1. I downloaded SCOP (release 1.69) and I selected all the structures classified as belonging to coiled-coil class (class h)

2. Each structure selected at point 1, was then processed with the SOCKET program that automatically identifies coiled-coil motifs.

To identify the contacts a packing cutoff of 7.4 Å has been chosen. If no coiled-coil segments were predicted by SOCKET the structure was discarded. The annotation was thus obtained using the sequence segments labelled by SOCKET as coiled-coil regions. When SOCKET detected overlapping segments in a given position of a sequence (due to the multiple contacts of coiled-coils in three-dimesional structure),the coiled-coil domain has been defined as the union of all the coiled-coil segments.

Furthermore, I also excluded:

- protein structures with holes in the coordinates
- protein structures with sequence length shorter than 30 residues
- protein structures with coiled-coil domains shorter than 9 residues.

Following this procedure I collected a structurally annotated data set of 558 protein chains (S558).

In order to test the different methods on a blind set, I selected from S558 a subset of 239 protein chains (S239) with sequence identity <30% with respect to the sequences of the MARCOIL data set. With the exception of PAIRCOIL2, all the other methods were trained before MARCOIL was implemented, and therefore the S239 data set can be considered a reliable blind testing set also for the previous methods. The S239 data set contains 23,998 residues, among which 6,851 belong to coiled-coil regions (about 30%). The complete S558 data set contains 63,860 residues with 16,974 coiled-coil residues (about 27% of the overall data set).

Finally, a data set that do not contains coiled-coil domains according to SCOP and SOCKET has also been selected starting from the Astral SCOP (release 1.69) which contains sequences with less than 40% identity. The selected sequences have been processed with SOCKET (7.4 Å packing cut-off) and all the sequences for which the program detected at least one coiled-coil residue were removed from the data set. All the sequences similar (<25% identity) to one of the MARCOIL negative set were also filtered out. I also checked the corresponding entries of the PDB in order to further remove all the structures annotated as coiled-coil or coiled-coil related. Finally, the remaining sequences were clustered fixing the sequence identity threshold to 25% and a representative for each cluster was chosen. The negative data set consists of 1,139 proteins sequences (S1139).

In the following I will refer with S1378 to the data set composed of the S239 and the S1139 data sets that has been adopted for evaluating the performance of the method in discriminating coiled-coil sequences from sequences that do not contain them.

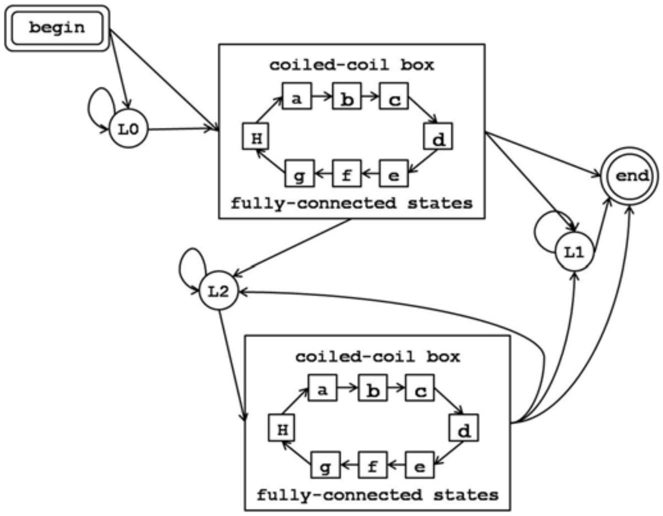


Fig. 5.4. : Automaton representation of the CCHMM_PROF Hidden Markov Model. The allowed transitions are indicated with the arrows. The states inside the two coiled-coil boxes (H, a, b, c, d, e, f, g) are fully-connected but for sake of clarity only the most probable transitions are indicated.

5.4 The Hidden Markov Model (CCHMM_PROF)

The model I designed is depicted in Figure 5.4. CCHMM_PROF consists of three background states, labelled with L0, L1 and L2, which model the connections between coiled-coil segments. These three states share the same emission probabilities and they are therefore *tied*. Moreover, the HMM has two coiled-coil boxes in order to consider different transition probabilities for sequences that contain both one and two or more coiled-coil segments. Each box has a background state *H* that accounts for the non-heptad coiled-coil periodicities, such as skips, stutters and stammers [Gruber and Lupas, 2003, Lupas and Gruber, 2005].

The box includes eight coiled-coil states which are fully connected and whose transition probabilities are initialized so that the heptad order is favoured: the probability to follow this order is close to one while the other transitions have a probability close to zero. The states within the two boxes that correspond to the same repeat type are also tied. A similar model, taking however single protein sequence as input, was first proposed by Fariselli *et al.* [Fariselli *et al.*, 2007]. In my work, for the first time in relation to the coiled-coils prediction problem, the sequence profile is allowed to be adopted as input encoding to a specifically implemented HMM model (CCHMM_PROF). The sequence profile is computed with an automated procedure after a PSI-BLAST [Altschul *et al.*, 1997] sequence alignment of each sequence against the Uniref90 database. Uniref90 is a non-redundant subset of the Uniprot database (The Uniprot Consortium, 2008) that contains no pair of sequences with >90% sequence identity. The HMM model is trained with a labelled Baum-Welch algorithm [Durbin *et al.*, 1998] and its accuracy is tested with the posterior-Viterbi decoding [Fariselli *et al.*, 2005], both described in Chapter 3.

5.5 Scoring the performance

5.5.1 Per-residue indices

The results of the different methods were evaluated using the following definitions. The overall accuracy ($Q2$), namely the number of correctly predicted residues is:

$$Q2 = p/N \tag{5.1}$$

where p is the number of correctly predicted residues and N is the total number of residues. The correlation coefficient (C) for a given class s is defined as:

$$C(s) = [p(s)n(s) - o(s)u(s)]/d(s) \tag{5.2}$$

where $d(s)$ is the normalization factor

$$d(s) = [((p(s) + u(s))(p(s) + o(s))(n(s) + u(s))(n(s) + o(s)))]^2 \tag{5.3}$$

$p(s)$ and $n(s)$ are respectively the true positive and true negative predictions for class s, while $o(s)$ and $u(s)$ are the numbers of false positive and false negative predictions. The sensitivity (Sn) for each class s is defined as:

$$Sn(s) = p(s)/[p(s) + u(s)] \tag{5.4}$$

and it accounts for the coverage of the prediction for each class, positive and negative. The specificity (Sp) is the probability of correct predictions and it is defined as follows:

$$Sp(s) = p(s)/[p(s) + o(s)] \tag{5.5}$$

All the scores that will be reported in the following tables are averaged over each protein sequence.

5.5.2 Per-segment indices

So far, coiled-coil predictors were evaluated using the above per-residue indices, such as sensitivity and specificity. However, the per-segment index SOV (Segment OVerlap) was defined both to account for the different segment distributions and to evaluate secondary structure segments rather than individual residues [Zemla *et al.*, 1999]. If (s_1, s_2) is a pair of overlapping segments, $S(i)$ is defined as the set of all the overlapping pairs in state i:

$$S(i) = \{(s_1, s_2) : s_1 \cap s_2 \neq \oslash, s_1\ and\ s_2\ in\ conformation\ i\} \tag{5.6}$$

while $S'(i)$ is the set of all segments s1 for which there is no overlapping segment s_2 in state i.

$$S'(i) = \{(s_1, s_2) : s_1 \cap s_2 = \oslash, s_1 \; and \; s_2 \; in \; conformation \; i\} \tag{5.7}$$

For state i the segment overlap (SOV) is defined as:

$$SOV(i) = 100 \times \frac{1}{N} \sum_{S(i)} \left[\frac{minov(s_1, s_2) + \delta(s_1, s_2)}{maxov(s_1, s_2)} \times len(s_1) \right] \tag{5.8}$$

with the normalization factor N_i defined as:

$$N_i = \sum_{S(i)} len(s_1) + \sum_{S'(i)} len(s_1) \tag{5.9}$$

The sums over $S(i)$ run over the segment pairs in state i which overlap by at least one residue. The other sum in the second equation runs over the remaining segments in state i. $len(s1)$ and $len(s2)$ are the lengths of segments s_1 and s_2 respectively, $minov(s_1, s_2)$ is the length of the overlap between s_1 and s_2, $maxov(s_1, s_2)$ is the total extent for which either of the segments has a residue labelled with i and $\delta(s_1, s_2)$ is defined as:

$$\delta(s_1, s_2) = min\{(maxov(s_1, s_2) - minov(s_1, s_2)); minov(s_1, s_2); \tag{5.10}$$

$$; int(len(s_1)/2); int(len(s_2)/2)\} \tag{5.11}$$

In particular, the segment overlap accuracy was computed both for the coiled-coil regions (SOV(CC)) and for the non coiled-coil regions (SOV(N)).

5.5.3 Per-protein index

In addition to the described measures, I introduced another scoring index in order to compare and validate the predicting methods at the protein level. For each protein, if the number of predicted coiled-coil segments (N_p) and the number of observed coiled-coil ones (N_o) is different, the prediction P is considered as wrong. Formally,

$$if \; N_p \neq N_o \Rightarrow P = 0 \tag{5.12}$$

If the number of predicted and of observed coiled-coil segments is the same and if the intersection between the two corresponding segments (namely the predicted segment p_i with the corresponding observed segment o_i, for $i = 1, ..., N_p$=N_o) is above a fixed threshold the prediction P is counted as a correct prediction:

$$if \; (N_p = N_o \; and \; p_i \cap o_i \geq th, \; \forall \; i = j) \Rightarrow P = 1 \tag{5.13}$$

Equation 5.13 defines the new Protein OVerlap (POV) index.

I tested two thresholds for the overlap. The first one is the minimum between the half lengths of the segments:

$$th = min(L_p/2, L_o/2) \tag{5.14}$$

where L_p is the length of the predicted coiled-coil segment and L_o is the length of the corresponding observed segment. A second and more strict threshold is the mean of the half lengths of the segments:

$$th = (L_p/2, L_o/2)/2 \tag{5.15}$$

For a set of proteins the average of all POVs over the total number of proteins N is:

$$POV = \frac{\sum_{i=1}^{N} P_i}{N} \tag{5.16}$$

All the per-residue, per-segment and per-protein final scores are obtained by averaging over the whole set the values computed for each protein. This measure is usually more stringent than summing up all the predictions and computing the indices at the end, since in this last case the scores of completely misclassified proteins can be absorbed by other predictions. For this reason, it may happen that both Sn and Sp can be lower than the corresponding $Q2$.

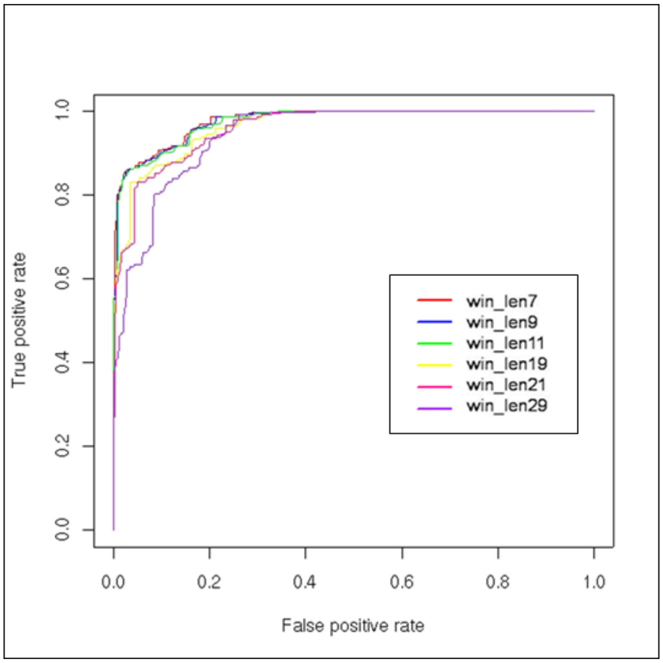


Fig. 5.5. : Receiver Operating Characteristic (ROC) curve of the results of CCHMM_PROF with different window sizes.

5.6 Results

5.6.1 Discriminating coiled-coil sequences

As detailed in Chapter 2, in the protein annotation field, an important problem is the structural classification of protein sequences. One of the purposes of CCHM_PROF is to filter out proteins that contain coiled-coil segments from the entire proteome. The method is trained on coiled-coil segments by adopting evolutionary information as input and it is tested on a blind set that contains both coiled-coil proteins (never seen before by the system) and protein chains that do not have coiled-coil segments. During this *discrimination* task, each protein is labelled with a single score and this number is used to classify proteins into two classes:

1. Proteins containing coiled-coils
2. Proteins not containing coiled-coils

Since the decoding algorithm assigns a label with posterior probability at each residue position, I introduced a global score for the entire protein of length N defined as:

$$maxav = max_{i=1}^{N} \frac{\sum_{j=1}^{W} prob_j}{W} \tag{5.17}$$

For each residue of the given sequence, the average of the scores (which in my case are the output posterior probabilities of the HMM) over a window of different size W (number of sequence neighbours of the predicted residue) was computed and I took as the protein score ($maxav$) the maximum of these average values, as detailed in Equation 5.17. Then, fixing different decision thresholds for the $maxav$, I evaluated the scoring indices. In particular, I monitored the False Positive Rate as a function of the sensitivity by computing a Receiver Operating Characteristic (ROC) curve.

From the curves in Figure 5.5, one can observe that by varying the amplitude of the window W from 7 to 29 residues, the best results were reached with windows of length 7, 9 and 11. Thus, since considering larger windows does not significantly improve predictions, all the evaluations in the following refer to the 7 residue window size. The values reported on the ROC curve are the True Positive Rate (namely $Sn(CC)$) and the False Positive Rate, defined as $1 - Sn(N)$, evaluated over the S1378 data set.

To assess whether the predictor was accurate in the recognition of coiled-coil sequences, I further compared my results with the results of the other available methods. With the exception of PAIRCOIL2, the score for each protein was computed following Equation 5.17, where *prob* is the output of the predictors relative to a given residue. For scoring PAIRCOIL2, I computed the average values for each residue of each sequence but instead of considering their maximum value, the minimum of the averages has been associated to each protein, since

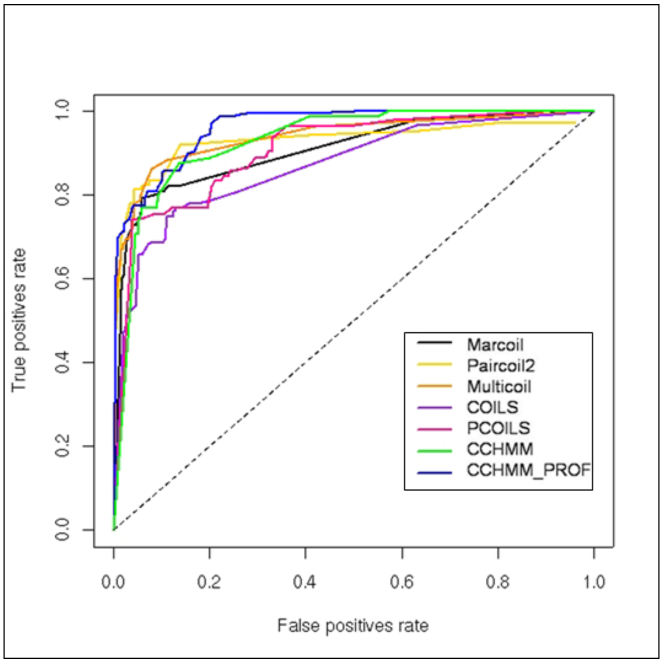


Fig. 5.6. : Receiver Operating Characteristic (ROC) curve of the results of the different predictors in the discrimination task.

Table 5.1. : Performance of the different predictors in the discrimination task.

Method	**Q2**	**Sn(CC)**	**Sn(N)**	**Sp(CC)**	**Sp(N)**	**C**
MARCOIL	0.91	0.55	0.99	0.89	0.91	0.65
PAIRCOIL2	0.92	0.63	0.99	0.90	0.93	0.72
MULTICOIL	0.92	0.58	0.99	0.93	0.92	0.70
COILS	0.89	0.47	0.98	0.82	0.90	0.57
PCOILS	0.91	0.67	0.96	0.79	0.93	0.68
CCHMM	0.91	0.71	0.96	0.75	0.94	0.68
CCHMM_PROF	0.94	0.71	0.99	0.93	0.94	0.78

PAIRCOIL2 classifies as coiled-coil the residues below the 0.025 threshold score. The ROC curve of Figure 5.6 shows the behaviour of the different classifiers in the discrimination task.

From the ROC curve it can be observed that CCHMM_PROF scores with a value of sensitivity for the positive class ($Sn(CC)$) equal to 71% when the False Positive rate is only 1%. The other methods score below CCHMM_PROF, as it can be seen also from the results reported in Table 5.1. In particular, MARCOIL, PAIRCOIL2 and MULTICOIL achieve the same lower level of False Positive Rate (1%) but with a True Positive Rate of 55%, 63% and 58% respectively. Moreover, it should be noted that all the sequences in our negative set are contained or similar to those used as a negative learning set to retrain PAIRCOIL2.

Table 5.2. : Area under the ROC curve (AUC) computed for all the classifiers.

Method	AUC
MARCOIL	0.92
PAIRCOIL2	0.94
MULTICOIL	0.94
COILS	0.87
PCOILS	0.91
CCHMM	0.94
CCHMM_PROF	0.96

To compare different classifiers, it is useful to compute a single scalar value for representing the performance. With this purpose I computed the area under the ROC curve (AUC) which is equal to the value of the Wilcoxon-Mann-Whitney test [Bradley AP, 1997]. Since a random classifier produces the diagonal line in a ROC plot, with an area under the line of 0.5, we expect that all the classifiers have an AUC greater than 0.5. The higher is the AUC the better is the method performance. In Table 5.2 all the AUCs for all the tested predictors are reported. The results further confirm that CCHMM_PROF is the best performing predictor (it has the highest AUC value equal to 0.96).

5.6.2 Locating coiled-coil regions in protein sequences

Given the relevance of the coiled-coil structural motifs in a number of biological processes, prediction methods aim to compute their modelling. For this reason, a fundamental problem in protein structure prediction is the location of coiled-coil segments in proteins. So far, all the available methods (MARCOIL, PAIRCOIL2, MULTICOIL, COILS, PCOILS) have been proved to be very accurate in the prediction of manually-annotated coiled-coil domains. Therefore, as stated in [McDonnell *et al.*, 2006], their behaviour is less accurate when predicting structurally-defined coiled-coil regions.

The evolutionary-based HMM model presented in this work is also suitable for locating coiled-coil regions in structurally-determined coiled-coil proteins. I trained my model on the S319 data set which is the complement of the initial S558 data set after removing the 239 sequences with <30% sequence identity with the sequences of the MARCOIL data set (that I adopted for testing). This means that any sequence within the S239 data set has less than 30% sequence identity with any of the sequences of the S319 data set. In Table 5.3, Table 5.4 and Table 5.5 the improvement achieved with the introduction of the evolutionary information is highlighted by comparing the performance of PCOILS with respect to COILS and of CCHMM_PROF when compared to the single-sequence based CCHMM. All the results are computed adopting a 0.5 decision threshold. As it can be seen from the scoring indices reported in Tables, PCOILS clearly outperforms COILS.

Table 5.3. : Comparison of the prediction efficiency of the HMM-based methods, based on per-residue indices.

Method	**Per-residue**					
	Q2	**Sn(CC)**	**Sn(N)**	**Sp(CC)**	**Sp(N)**	**C**
COILS	0.68	0.41	0.88	0.43	0.63	0.28
PCOILS	0.72	0.69	0.73	0.53	0.76	0.41
CCHMM	0.81	0.72	0.80	0.74	0.82	0.54
CCHMM_PROF	0.81	0.96	0.66	0.72	0.95	0.62

Table 5.4. : Comparison of the prediction efficiency of the HMM-based methods, based on per-segment indices.

Method	**Per-segment**	
	SOV(CC)	**SOV(N)**
COILS	0.45	0.60
PCOILS	0.62	0.59
CCHMM	0.75	0.74
CCHMM_PROF	0.81	0.70

Table 5.5. : Comparison of the prediction efficiency of the HMM-based methods, based on per-protein indices.

Method	**Per-protein**	
	POV_min	**POV_av**
COILS	0.45	0.40
PCOILS	0.63	0.56
CCHMM	0.75	0.62
CCHMM_PROF	0.86	0.71

Furthermore, CCHMM_PROF not only scores higher than PCOILS, but it also reaches better results than CCHMM, in particular for what concerns the values of segment overlap index of the coiled-coil regions and of the per-protein score (0.86 instead of 0.75). This finding corroborates the high global quality of the predictor.

5.6.3 Comparative evaluation

In Table 5.6, Table 5.7 and Table 5.8 the comparison of the results of my new model with the results of the other available methods is reported. All the methods were tested on the S239 data set using their default thresholds.

Table 5.6. : Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-residue indices.

Method	**Per-residue**					
	Q2	**Sn(CC)**	**Sn(N)**	**Sp(CC)**	**Sp(N)**	**C**
CCHMM_PROF	0.81	0.96	0.66	0.72	0.95	0.62
CCHMM	0.81	0.72	0.80	0.74	0.82	0.54
MARCOIL	0.70	0.66	0.71	0.50	0.78	0.38
PAIRCOIL2	0.72	0.58	0.63	0.46	0.49	0.19
MULTICOIL	0.66	0.54	0.75	0.39	0.62	0.26
COILS	0.68	0.41	0.88	0.43	0.63	0.28
PCOILS	0.72	0.69	0.73	0.53	0.76	0.41

Table 5.7. : Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-segment indices.

Method	**Per-segment**	
	SOV(CC)	**SOV(N)**
CCHMM_PROF	0.81	0.70
CCHMM	0.75	0.74
MARCOIL	0.58	0.56
PAIRCOIL2	0.55	0.45
MULTICOIL	0.48	0.51
COILS	0.45	0.60
PCOILS	0.62	0.59

Table 5.8. : Comparative evaluation of the different methods in the location of the coiled-coil regions, based on per-protein indices.

Method	**Per-protein**	
	POV_min	**POV_av**
CCHMM_PROF	0.86	0.71
CCHMM	0.75	0.62
MARCOIL	0.63	0.53
PAIRCOIL2	0.55	0.47
MULTICOIL	0.52	0.40
COILS	0.45	0.40
PCOILS	0.63	0.56

PAIRCOIL2 was scored with the decision threshold set to 0.025 and considering the 21 residues-long window. I also tested the 28 residues-long window; since the results did not change significantly, here I show only the best performing window. It is worth noticing that CCHMM_PROF not only outperforms the other HMM-based predictors but also all the other methods. CCHMM_PROF achieves the best per-residue accuracy (81%) and the highest correlation coefficient (62%). Also the sensitivity (96%) and the specificity (72%) for the coiled-coil class significantly raise with respect to the other methods. Furthermore, the global indices referring to the best per-segment and per-protein efficiencies (81% and 86%, respectively), are about 20 percentage points higher than the best ones reached by the methods developed so far (not considering that the sequence-based CCHMM results were already better than the results of the other reported methods).

5.7 Biological insight: targeting viruses

To further assess the performances of CCHMM_PROF the predictions of the different classifiers were evaluated in more detail. My results show that all the correct predictions made by the single methods are also correctly assigned by CCHMM_PROF. However, I was also able to select a subset of coiled-coil structures predicted only by CCHMM_PROF. Among them, there are interesting targeting virus proteins, such as: the Human Respiratory Syncytial Virus (HRSV) protein fusion core, the ectodomain of the Simian Immunodeficiency Virus (SIV) glycoprotein 41 and the core of glycoprotein 2 from Ebola Virus.

Viral fusion or transmembrane glycoproteins are the major responsible of the entry of a virus in the host cell. It is known that most of these structures are antiparallel trimeric coiled-coil heterodimers ([Matthews *et al.*, 2000, Lu *et al.*, 1995, Caffrey *et al.*, 1998, Malashkevich *et al.*, 1998]). The fusion glycoproteins have a common mechanism of activation: they are synthesized as a precursor peptide that is processed with a proteolytic cleavage to produce two disulfide-linked fragments. In HRSV the two fragments are indicated with F1 and F2. The fusion peptide is located at the N-terminal of the F1 segment, while the transmembrane region is close to its C-terminal. Analogous to the HRSV F1 and F2 subunits are the GP1 and GP2 regions from Ebola virus and the surface subunit and transmembrane subunit (TM) from retroviruses, including gp120 and gp41 from SIV. In each of the reported examples, the first subunit is responsible for binding to cell-surface receptors, while the second subunit mediates membrane fusion [Zhao *et al.*, 2000]. Two 4,3-hydrophobic heptad repeats are located adjacent to both the fusion peptide and the transmembrane region. These regions, that we denote with HR-C and HR-N respectively, form trimeric hairpin-like structures with the HR-C segment that packs in an antiparallel direction around the inner coiled-coil formed by the HR-N region [Malashkevich *et al.*, 1998, Malashkevich *et al.*, 1999, Chan and Kim, 2000]. In Table 5 the annotated and predicted coiled-coil segments (with CCHMM_PROF) in HRSV fusion protein core, in the gp41 ectodomain of SIV and in the core structure of gp2 from Ebola virus are reported. For each of the studied structures, the Uniprot annotation, if available and the annotation provided by the SOCKET program were significantly overlapping (overlap $> min(l_1/2, l_2/2)$, where l_1 and l_2 are the length of the coiled-coil segments as annotated in the two ways). My method correctly predicts the two coiled-coil regions of the HRSV protein fusion core. The method is also able to correctly identify the two coiled-coil regions of the SIV protein core, which are not yet reported on the Uniprot database. CCHMM_PROF also correctly recognizes one of the two regions of the core structure of the gp2 of Ebola virus.

Several biochemical studies suggested that viral fusion proteins change their conformation upon activation. At the beginning, the fusion glycoprotein has a native structure in which the fusion and the HR-N regions are not accessible. Then, during the activation process, the coiled-coil region is exposed and the fusion peptide is exposed on the target cell membrane. The last change requires that the two heptad regions HR-N and HR-C associate and bring the cell and viral membranes together to promote fusion [Chan and Kim, 2000]. So far, no effective treatment is available for any of the described viruses. For this reason, accurate computational methods able to locate functionally important sequence segments, such as coiled-coils, are of fundamental importance. A better understanding of the virus infection biology, related in particular to the fusion protein, can lead to the experimental evidence or de-novo design of new targets for antiviral therapies.

5.8 Discussion

The prediction of coiled-coil domains is a twofold issue: a predictive method should be able both to identify the proteins that contain coiled-coil segments in a given set of protein sequences (or in proteomes) and to predict the number and the location of coiled-coil domains in a protein chain. In my work I addressed both problems and I built a HMM that takes the evolutionary information obtained from multiple sequence alignments as input. As shown from the reported results, the introduction of the sequence profile significantly improves the method accuracy (see PCOILS versus COILS and CCHMM_PROF versus CCHMM). Indeed, CCHMM_PROF scores better than all the methods developed so far. Furthermore, I provided a new structurally-annotated and freely-available benchmark data set of coiled-coil structures that can be used to reliably train and test computational methods. I also suggested a more robust evaluation of the method performances, by introducing a new scoring frame that takes into account not only the residue accuracy but also the accuracy at the segment and at the protein levels. The development of accurate computational methods for coiled-coil prediction can drive experiments towards the *de-novo* design of *ad-hoc* coiled-coil structures. Indeed, despite the fact that decades of theoretical and practical advances constantly added to the understanding of the coiled-coil structure and function, the recognition of this structural motif with confidence in genome-wide annotation processes is still a challenging issue.

Chapter 6

Comparative large-scale genome analysis

6.1 Introduction

Functional annotation of protein sequences is a major requirement in the post-genomic era. This is particularly true given the large number of ongoing genome sequencing projects which requires a fast and efficient automatic annotation process. Historically, function prediction was based on the relationship between protein sequence and structure. By this, the annotation procedure includes the functional information enclosed in the protein structure. This procedure goes back to early observations [Chothia and Lesk, 1986] indicating that the ability of predicting protein structure and function from sequence largely depends on the value of sequence identity. A protein structure can be a close and general model for other proteins, provided that sequence homology is >40-50%. In general, proteins with a high structural similarity over the entire sequence length are likely to share at least a similar function. This fact is at the basis of a number of methods that were developed to functionally annotate proteins starting from their structure (DALI [Holm and Sander, 1993], CE [Shindyalov and Bourne, 1998], and the most recent CATHEDRAL [Redfern *et al.*, 2007], EVEREST [Portugaly *et al.*, 2006]).

In automatic protein structure classification however it is not trivial to assess the level of structure similarity in order to infer functional similarity. Greene and co-workers [Greene *et al.*, 2007], analyzing the SCOP database, observed that although most domains with a common fold show a similar function, some "superfolds" (such as the Rossmann fold) can correspond to about 50 different functions.

A sequence similarity search is a common initial step of all the annotations methods described in literature including RefSeq [Pruitt *et al.*, 2007], VEGA [Wilming *et al.*, 2008] and PEDANT [Walter *et al.*, 2009]. However, in the absence of a golden rule for correlating function to sequence, different constraints were ruled out in setting out to which extent functional and structural information can be transferred between pairs of related protein sequences at various levels of sequence similarity. Depending on how the function conservation is defined, the level of sequence similarity allowed to functionally annotate a sequence protein may vary from 30-40% down to 20-25%, provided that folding is also conserved [Wilson *et al.*, 2000]. A pairwise sequence identity higher than 40% was suggested as a confident threshold to transfer the first three digits of an EC number [Rost B, 2002, Tian and Skolnick, 2003].

As a general trend, similarity search allows clustering procedures by which sequences are collected into sets of similarity. Clustering is therefore a basic task in automatic processing of large data sets of protein sequences, and annotation methods following this procedure are classified as hierarchical and non-hierarchical ones, depending on the clustering procedure. Hierarchical clustering methods aim to categorize data items into a tree-structured hierarchical organization. This is a possible approach to dealing with different levels of similarity required to annotate different protein families. Earlier attempts of hierarchical clustering include SYSTERS [Krause *et al.*, 2002], Picasso [Hedger and Holm, 2001], and iProClass [Wu *et al.*, 2001]. CluSTr [Kriventseva *et al.*, 2001] and ProtoNet [Kaplan *et al.*, 2005, Loewenstein *et al.*, 2008] are presently the hierachical algorithms that take advantage of the largest number of fully sequenced genomes for protein sequence classification into families. CluSTr (that includes the UniProt Knowledgebase, all the International Protein Index (IPI) and all the completely sequenced genomes retrieved from Integr8) clusters proteins starting from a similarity matrix of all-against-all protein sequence alignments computed with the Smith-Waterman algorithm. Z-scores for evaluating the similarity between potentially related proteins are then computed with a Monte Carlo simulation and proteins are grouped using a single-linkage algorithm by choosing different levels of protein similarities. ProtoNet is an automatic and unsupervised agglomerative clustering system that builds a hierarchical tree of proteins based only on sequence similarity by means of an Unweighted Pair Group Method with Arithmetic mean (UPGMA). The latest data set of ProtoNet consists of all the protein sequences taken from UniProt Knowledgebase (release 8.1). Both these methods use hierarchical algorithms to group sequences into clusters, ultimately relying on the sequence identity of the aligned proteins as taken into account by selecting different E-value thresholds (E-values set to 100 and to 1e-40 by CluSTr and ProtoNet, respectively).

Non-hierarchical clustering procedures catch functional relationships among proteins by providing some additional constraints. The TribeMCL algorithm [Enright *et al.*, 2002] allows a fast annotation that is rather independent of the presence of multi-domain proteins, promiscuous domains and fragmented proteins. This method includes an all-against-all BLAST comparison with an E-value threshold of 1e-10. The results (represented with a binary matrix) cluster into protein families after successive rounds of Smith-Waterman dynamic programming alignments.

Summing up, the most common and widely exploited approach of automatically annotating unknown sequences relies on the so-called “inheritance through homology” based on the notion that similar sequences share similar functions and structures.

In automatic annotations processes, however, additional levels of complexity are due to:

i) proteins that contain multiple domains;

ii) proteins that share common domains and that do not necessarily share the same function;

iii) the finding that different combinations of shared domains can lead to different biological roles.

It has been discussed that multi-domain proteins are less functionally conserved than single-domain ones, with the exception of those proteins in which structures show an overlapping combination of domain folds. While in single domain proteins the level of confidence of function transfer can be as high as 67%, in multi-domain proteins it goes down to 35%. The probability that two multi-domain proteins share the same function increases up to 80% when the two proteins have the same combination of two structural superfamilies and it increases up to 90% only when the coverage of the alignment reaches the full length of both proteins [Hegyi and Gerstein, 2001]. Methods that only take into account sequence similarity parameters, particularly when dealing with putative multi-domain proteins, may run into the risk of pulling sequences in the same cluster and erroneously transferring functions. Therefore the coverage of the alignment on top of high sequence identity is a fundamental parameter to consider when applying sequence comparison techniques to problems of protein sequence annotation. Up to now, none of the available methods explicitly faces this problem. Only ProtoNet refers to an external database (EVEREST) containing protein domains.

In the following I will describe a fast and reliable automatic method for bridging the gap between protein sequence, structure and function. First I will present the annotation performance of the method after non-hierarchical clustering on 599 genomes and the statistical validation on this set of the inheritance of functional and structural annotation, when possible, within the clusters. Another 201 genomes are then annotated by adopting the proposed clustering procedure. The method explicitly constrains coverage of the sequence alignment within clusters that, when possible, are also labeled with statistically validated GO terms, structures and their SCOP classification.

6.2 The data set

The data set for generating property-specific clusters includes 599 completely sequenced genomes (S599), among which 551 are from prokaryotic organisms (18 Archaea and 533 Bacteria) and 48 are from Eukaryotes (2 Protozoa, 9 Fungi, 4 Plants and 33 Animals). The data set contains 2,624,555 protein sequences: 65% (1,713,574 sequences) from prokaryotes and 35% (910,981 sequences) from eukaryotes. Another 201 genomes, comprising 4 eukaryotes and 197 prokaryotes, were adopted as a blind test for the annotation procedure (S201), including 724,854 protein sequences (11% of which are from Eukaryotes and 89% from Prokaryotes). The bacterial genomes were downloaded from the NCBI genome resource database (`ftp://ftp.ncbi.nih.gov/genomes/Bacteria`) while the eukaryotic ones were taken both from the RefSeq at NCBI (`ftp://ftp.ncbi.nih.gov/refseq/release/`) and from The Ensembl Genome Project (`ftp://ftp.ensembl.org/pub/`).

6.3 Sequence comparison and clustering

An all-against-all pairwise comparison of the 2,624,555 (S599) protein sequences was performed with the BLAST program [Altschul *et al.*, 1990]. The Grid middleware has been a suitable tool for handling the data size problem and for accessing, sharing and processing the retrieved and computed data. Therefore, all the BLAST comparisons, which represent the data production step of the method, were carried out in parallel on the Grid. In order to consider only high-scored alignments, the E-value was set to 1e-10, a high restrictive threshold. In addition, in order to obtain reproducible results, the BLAST database size was kept constant for each independent run. Both input files (Fasta format files containing the protein sequences) and output files (BLAST tabular results) were stored and replicated on two different Storage Elements (essentially disk servers) in order to properly weight the traffic of data on the Grid and to ensure fault tolerance. The job submission was controlled by an automated procedure with 4 different specific devices distributed all around Europe that are responsible for the job management and for the resource selection (Resource Brokers).

The protein space, encoded in the computed alignments, was represented by means of an undirected graph structure. The nodes of the graph are the protein sequences and an edge is established between two nodes only when the two corresponding proteins share BLAST hits that simultaneously satisfy the following constraints:

$$\begin{aligned} Sequence\ Identity &\geq 40\% \\ Coverage &\geq 90\% \end{aligned} \tag{6.1}$$

Given two protein sequences that share a BLAST hit, the coverage of the match is defined as:

$$Coverage = I/U \tag{6.2}$$

where I is the length of the intersection of the aligned regions on the two sequences and U is the overall length of the alignment (namely the sum of the lengths of the two sequences minus the alignment length). After building the graph, proteins were clustered by computing the connected components of the graph with a transitive closure algorithm [Cormen *et al.*, 2001]. These components are by definition disjointed, so to say that no protein sequence is present in two different clusters. With the clustering procedure, each connected component includes all the pairs of sequences satisfying Eq.6.1 and more importantly, it includes also chains that are not directly linked but are connected through a path of proteins which undergo the imposed criteria. For this reason, a cluster can also contain sequences that do not share any sequence similarity when measured with a global sequence alignment but that are distantly related to the initial seed. A cluster is by definition a connected component of the graph whose dimension is ≥ 2. When sequences are found alone after clustering, they are called singletons.

6.4 Mapping functions and structures on the clusters

To investigate the functional and the structural information enclosed in the computed clusters we mapped their content over three major databases: the Uniprot knowledgebase [Apweiler *et al.*, 2004](release 12.6), the Protein Data Bank [Berman HM *et al.*, 2000] (PDB, March 2008) and the Gene Ontology [The Gene Ontology Consortium, 2000] (GO, March 2008). To establish a correspondence between a sequence from my database and an entry of one of the selected databases, each protein sequence within the data set and each sequence in the Uniprot database were uniquely identified by computing their CRC64 checksum. An association is established between the two sequences when they correspond to the same checksum. The Uniprot database is the most complete resource of annotated sequences that provides relations with other databases, including GO terms. I initially mapped all the Uniprot sequences on the overall set of computed clusters (187,594). 1,985,132 proteins of S599 have an associated Uniprot accession (76% of S599), and out of these, 1,490,135 sequences spread over the obtained clusters; the remaining are to be found in singletons. As a result, about 40% of the entire Uniprot data set was mapped on the clustered protein sequences (so to say that 2,066,511 Uniprot accession numbers matched at least one sequence belonging to our clusters).

For exploring the distribution of the protein domains in the clusters, I took advantage of the PDBsum database [Laskowski *et al.*, 1997], since it provides the correspondence between the protein chains contained in the PDB and the Uniprot accession identifiers. In this way, an association between the protein sequences of the analyzed data set and the PDB protein structures was established.

Gene Ontology [The Gene Ontology Consortium, 2000] is the standard language adopted by the scientific community for annotating genomes. To determine the functions associated to the sequences, we make use of the GO terms, by specifically considering the molecular function ontology, which describes the molecular activities carried out by proteins or protein complexes and actually contains 8,351 molecular function terms. Particularly, I considered the molecular function tree, since it captures the biochemical function of the protein families. A biological process can be the combined effect of one or more molecular functions.

When considering the GO terms, it is also necessary to take into account all their parent terms (the data structure underlying the GO ontology is a Directed Acyclic Graph (DAG) that allows a child to have multiple parents). To handle with the DAG structure, I devised a two step procedure. First, when available, each protein sequence within a cluster was associated to the corresponding GO term/s given by the Uniprot annotation. Then, for each sequence, all the possible branches of the GO hierarchy were extended by recursively walking along the parent branches of the molecular function GO tree.

6.5 Statistical evaluation of GO terms

To assess whether a GO term is significant for a cluster, I performed a statistical test by computing the P-values. If N is the number of sequences in the cluster which correspond to the same specific GO term, the P-value is the probability of finding N or more proteins that have a given annotation by chance, given the dimension of the cluster, the dimension of the database and the overall number of sequences in the database with the given annotation. For each GO term within a cluster the corresponding P-value is evaluated as:

$$Pvalue(GO) = \sum_{i=N}^{min(K,P)} \left(\frac{\binom{K}{i}\binom{D-K}{P-i}}{\binom{D}{P}} \right) \tag{6.3}$$

where D is the dimension of the database (total number of sequences with at least one associated GO term), P is the dimension of the cluster (total number of sequences of the cluster with at least one associated GO term), K is the number of sequences in all the database which have associated the same specific GO term and N is defined above. In order to compute the binomial expressions at the numerator in Eq.6.3, the Stanica approximation has been adopted [Stanica P, 2001]. To account for the multiplicity of the GO terms in each cluster, the Bonferroni correction was applied to the computed P-values [Moore and McCabe, 2006].

Given the high dimensionality of the problem, a careful statistical analysis is required to investigate whether the GO terms present in each cluster can be considered statistically significant and therefore whether the associated molecular function can be considered specific for the given cluster. To address this problem, a statistical evaluation by computing P-values and adopting a bootstrapping procedure was carried out. For this, I considered which range of values would be computed for the P-value, when the GO terms were randomly distributed among the clusters (the base line random distribution) and I compared the observed distribution of the P-values with the baseline random distribution. The random distribution is a function of the data set composition, being computed by preserving the total number of clusters, the total number of GO terms and the cluster dimensions.

To estimate a P-value threshold the cumulative distributions of the real and random computed P-values were compared (Fig.6.1). By this, a GO term is sta-

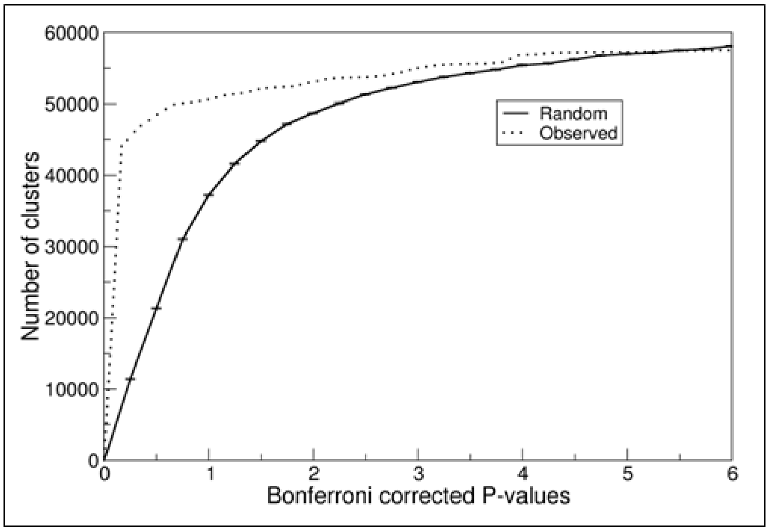


Fig. 6.1. : Cumulative distributions of the Bonferroni corrected P-values. Dotted line: cumulative function of the observed P-values. Solid line: cumulative distribution of the average of the P-values of a 100 random benchmark set; error bars represent standard deviations..

tistically significant for a cluster (i.e. that GO term is cluster-specific) if its associated P-value is smaller than the threshold value for which the real and the random curves are significantly different (Fig.6.1). An optimal threshold of the P-value can be considered the one for which the observed curve significantly differs from the random ones. In particular, given the described clustering procedure, the value for the observed cumulative distribution corresponding to a P-value of 0.001 is 31,606 (namely, the number of clusters with at least a GO term with P-value $\leq$0.001), while the mean of the random cumulative distributions is 25 (namely, the number of randomly generated clusters having a minimum P-value $\leq$0.001). For a P-value threshold of 0.001 we are expecting 8 un-correctly assigned GO terms out of 10,000. On these bases we adopted 0.001 as a suitable threshold of P-value in order to guarantee statistical significance to the GO terms/clusters association, which is peculiar of the described data set.

6.6 Results

The main feature of the designed non-hierarchical clustering procedure consists in the organization of the protein sequences into clusters according to a very strict criterion that considers constraints based simultaneously on high sequence similarity ($\geq$40%) and high sequence coverage ($\geq$90%). PDB structures, their SCOP classification and GO terms, when available, are also included in the clusters, in order to exploit annotations in terms of sequence to structure, to function, and to structure and function. It is also important to notice that the annotation process is statistically validated by computing the level of confidence at which the GO annotation transfer process is performed. To exploit the above described added values of this new method in the following I will describe:

1. the clustering procedure and its statistical validation performed with an initial set comprising 599 genomes (S599);

2. a blind test on a set comprising some other 201 genomes (S201).

6.6.1 Clustering procedure and cluster content: clusters vs singletons

Considering the initial S599 set, the clustering procedure produced 187,594 clusters, which include a total of 1,963,704 protein sequences (75% of S599). Given the cluster procedure described above, each protein chain belongs only to one cluster and that all the clusters are disjointed. The number of sequences in the clusters ranges from a minimum of 2 (minimum cluster dimension allowed) to 15,875 (dimension of the biggest connected component). The remaining 25% of S599 (for a total of 660,851 sequences) are the so-called singletons, namely sequences without any BLAST match satisfying our restrictive constraints (Eq.6.1). Among the clusters, 122,686 (65%) contain sequences deriving from only prokaryotic organisms while 63,230 (34%) are specific for Eukaryotes. Only 1,678 clusters (1%) contain sequences from both prokaryotic and eukaryotic organisms. In Table 6.1, clusters are grouped according to the standard deviation of the protein sequence length distribution of each cluster. These statistics highlights that the clusters are quite homogenous in protein length: the sequence length variability of most of the clusters (over 90%, containing some 80% of the total number of sequences) is ≤40 residues that is the minimum number of residues for a structural domain according to the SCOP structure classification. The table also lists the percentage of clusters and sequences that are annotated with the procedure described in the following.

Table 6.1. : Non hierarchical clustering of S599 protein sequences.

St_dev (residues)	**Clusters**	**Annotated clusters(%)**	**Sequences**	**Annotated sequences(%)**
A* ≤5	109,058 (58%)	19	498,054 (26%)	12
B >5-≤10	29,583 (16%)	8	380,352 (19%)	14
C >10-≤20	25,233 (13%)	6	490,517 (25%)	20
D >20-≤30	10,043 (5%)	2	218,085 (11%)	8
E >30-≤40	5,191 (3%)	1	120,987 (6%)	5
F >40-≤50	2,852 (2%)	0.7	73,868 (4%)	3
G >50	5,634 (3%)	1.3	181,661 (9%)	7

		GO	GO (P-value≤0.0001)	GO (P-value>0.0001)	**-GO**
Clusters	clusters*	70,508 (38%)	31,606 (17%)	38,902 (21%)	117,086 (62%)
	sequences*	1,317,026 (50%) direct: 919,895 (35%) inherited: 397,131 (15%)	1,078,180 (41%) direct: 824,407 (31.4%) inherited: 250,329 (9.9%) changed: 3,444 (0.1%)	238,846 (9%) direct: 92,044 (3.5%) inherited: 146,802 (5.5%)	646,678 (25%)

Table 6.2. : From sequence to function: mapping of molecular function GO terms into the S599 clusters .*The total number of clusters of S599 is 187,594. The total sum of protein sequences of S599 is 2,624,555; the total number of singletons is 660,851. Direct annotation: sequences with at least a GO term. Inherited annotation: sequences that inherit GO term/s in the cluster. Changed annotation: sequences that after statistical validation, change functional annotation with respect to Uniprot.

6.6.2 From sequence to function

A sequence is defined to be functionally annotated when it is endowed in Uniprot with a GO term of the molecular function ontology, being this term more specific for protein biochemical functions. In Table 6.2 it is shown how the GO annotated sequences distribute among the clusters. 40% of the sequences of the S599 have at least one associated GO term in Uniprot (35% populate 38% of the clusters and 5% fall into 21% of the singletons) and according to our definition are therefore functionally annotated. This is reported as “direct” annotated in Table 6.2. After clustering, another 15% of S599 “inherit” annotation, 9.5% of which with a cluster specific GO term/s (P-value ≤0.001). Interestingly, after statistical validation a small percentage (0.1%) of sequences changed their associated GO term with respect to Uniprot, confirming that automatic annotation may also lead to unappropriate functional transfer. The important results out of this effort over 599 genomes are:

i) a statistical validation of the GO-Uniprot annotated sequences (directly annotated sequences);

ii) a more rigorous annotation of some GO-Uniprot annotated sequences (those sequences that change annotation);

iii) the statistical validation of the annotation of 397,131 uncharacterized sequences (15% of S599 sequences).

		PDB	SCOP (Mono-domain)$^\#$	SCOP (Multi-domain)$^\#$	**-PDB°**
Clusters	clusters*	5,734 (3%)	3,064 (2%)	897 (0.5%)	181,860 (97%)
	sequences*	596,411 (23%) direct: 18,357 (7%) inherited: 578,054 (16%)	334,647 (2%) direct: 9,754 (3.7%) inherited: 324,893 (9.3%)	167,779 (6%) direct: 3,341 (1.2%) inherited: 164,438 (4.8%)	1,367,293 (52%)

Table 6.3. : From sequence to structure: mapping of PDB structures into the S599 clusters. *,$^\circ$ see Table 6.2; $^\#$The PDB structure is mono-domain or multi-domain according to the SCOP classification.

6.6.3 From sequence to structure

The procedure also aims to give a structural template to previously uncharacterized protein sequences. To achieve this task the information derived from PDB is also included. Roughly 50% of the PDB structures can be presently included in the computed clusters: a total of 18,357 protein sequences of our data set correspond to 23,050 PDB files for a total of 28,978 PDB chains. These chains are distributed over 5,734 clusters (about 3% of the total number of clusters). Only 151 singletons (0.02% of the total number of singletons) are endowed with the correspondent PDB chains (Table 6.3), for a total of 559 structures. In order to evaluate the structural congruency within the clusters, each PDB chain was then linked to its corresponding SCOP [Murzin *et al.*, 1995] identifier/s (a SCOP identifier is a four digit code that allows classification of protein domains; the same SCOP identifier guarantees high structural similarity among protein domains). 21,090 PDB chains are endowed with SCOP identifiers in 3,961 clusters.

I found that 3,064 (of the total 5,734) clusters contain a unique SCOP identifier (single domain identifier) for a total of 14,340 PDB chains (with an associated SCOP id) and 334,647 protein sequences (second and third columns in Table 6.3). 897 clusters are endowed with multiple SCOP identifiers (ranging from to 2 to 6 SCOP identifiers per cluster), containing 6,750 PDB chains and 167,779 sequences.

When transferring a template structure to a given sequence a major problem is the relative coverage of the target and template: the higher the coverage, the higher the probability of obtaining a template covering the whole protein sequence. In the computed clusters, given the clustering procedure, and as shown above (Table 6.1), sequences of similar length tend to be associated in the same cluster. To measure whether the structural template of a cluster can be adopted as a template for the protein sequences within the same cluster, I computed the coverage of the template over the protein sequences of the cluster (*StructCov*).

		PDB$^+$	SCOP$^+$ (Mono-domain)$^\#$	SCOP$^+$ (Multi-domain)$^\#$
>0.90	clusters*	5,686 (99%)	3,031 (53%)	887 (0.5%)
	sequences*	588,344 (99%) direct: 18,290 (3%) inherited: 570,054 (96%)	328,345 (55%) direct: 9,690 (2%) inherited: 318,655 (53%)	158,442 (26.6%) direct: 3,325 (0.6%) inherited: 155,117 (26%)
>0.80–≤0.90	clusters*	47 (0.8%)	32 (0.6%)	9 (0.15%)
	sequences*	7,963 (1.3%) direct: 65 inherited: 7,898	6,198 (1%) direct: 62 inherited: 6,136	1,423 (0.2%) direct: 13 inherited: 1,410
≤0.80	clusters*	47 (0.034%)	1 (0.017%)	1 (0.017%)
	sequences*	8,018 (1.34%) direct: 5 inherited: 8,013	104 (1%) direct: 2 inherited: 102	7,914 (1.33%) direct: 3 inherited: 7,911

Table 6.4. : Template coverage over the clusters. *The total number of clusters of S599 with at least a PDB template is 5,734 for a total sum of 596,411 protein sequences. Direct annotation: sequence with a PDB structure; Inherit annotation: sequences that inherit the structure in the cluster. +PDB: with at least a PDB template in the cluster; the PDB structure is mono-domain or multi-domain according to the SCOP classification.

For each cluster, the coverage is the average of the ratios:

$$StructCov = \frac{l}{l_{PDB}} \tag{6.4}$$

where l is the length of each protein sequence and l_{PDB} is:

- the length of the longest protein with a PDB structure (second column)
- the length of the longest protein with at least one domain annotated with SCOP.

The coverage is computed for all the clusters with a PDB structure (5,734 clusters) and for the clusters with a mono-domain (3,064 clusters) and multi-domain (897 clusters) structural template, separately.

Consequently, when PDB/SCOP are mapped into the clusters, a very high coverage is detected. Most of the clusters endowed with a PDB/SCOP label are endowed with a coverage ≥0.90 (Table 6.4).

It should also be noticed that the coverage is high also in the case of SCOP multi-domain proteins, indicating that even in this case template inheritance allows transfer to the whole protein target. After the PDB/SCOP mapping of the clusters (considering also that a small fraction of PDBs is lacking a SCOP label) a structural template can be safely transfered to about 20% of the total number of S599 sequences, including SCOP mono and multi-domain structures.

		GO	GO P-value≤0.0001	GO P-value>0.0001	**-GO**
PDB	clusters*	4,233 (2.2%)	3,233 (1.7%)	1,000 (0.5%)	1,501 (0.8%)
	sequences*	554,192 (28%) direct: 11,484 (0.5%) inherited: 542,708 (27.5%)	519,756 (26%) direct: 10,997 (0.48%) inherited: 508,759 (25.52%)	34,436 (2%) direct: 487 (0.02%) inherited: 33,949 (1.98%)	42,219 (2%) direct: 2,723 (0.1%) inherited: 39,496 (1.9%)
SCOP **Mono-domain**	clusters*	2,245 (1.2%)	1,745 (0.9%)	500 (0.3%)	819 (0.44%)
	sequences*	309,744 (16%) direct: 6,099 (0.3%) inherited: 303,645 (15.7%)	290,907 (15%) direct: 5,855 (0.25%) inherited: 285,052 (14.75%)	18,837 (1%) direct: 244 (0.05%) inherited: 18,593 (0.95%)	24,903 (1.3%) direct: 1,493 (0.1%) inherited: 23,410 (1.2%)
SCOP **Multi-domain**	clusters*	816 (0.43%)	%688 (0.37%)	128 (0.06%)	81 (0.04%)
	sequences*	165,125 (8.4%) direct: 2,524 (0.13%) inherited: 162,601 (8.27%)	160,516 (8.2%) direct: 2,471 (0.12%) inherited: 158,045 (8.08%)	4,609 (0.2%) direct: 53 (0.01%) inherited: 4,556 (0.19%)	2,654 (0.14%) direct: 148 (0.02%) inherited: 2,506 (0.12%)
-PDB	clusters*	66,275 (35%)	28,373 (15%)	37,902 (20%)	115,585 (62%)
	sequences*	762,834 (39%) direct: 525,753 (27%) inherited: 237,081 (12%)	558,424 (28%) direct: 433,268 (22%) inherited: 125,156 (6%)	204,410 (11%) direct: 92,485 (5%) inherited: 111,925 (6%)	604,459 (31%)

Table 6.5. : From sequence to structure and function. *The total number of clusters of S599 is 187,594 for a total sum of 1,963,704 protein sequences. Direct annotation and inherited annotation is as in Table 2, 3 and 4, depending on the annotation type. Three different annotations are possible: GO and PDB; GO without PDB; PDB without GO. The rightmost bottom corner contains the number of clusters and sequences that is without annotation.

6.6.4 From sequence to function and structure

As a final result, the described annotation procedure can produce three main categories of annotation (Table 6.5): PDB and GO; PDB without GO; GO without PDB, and no annotation. These categories, with the numbers of clusters and sequences in the clusters are listed in the foremost left and right corners of Table 6.5. After SCOP labeling and GO statistically validation, the richest annotation (PDB and GO) give rise to six different types of annotation GO without PDB is subdivided in 2 more categories while PDB without GO is in turn splitted in three categories. About 21% of the S599 protein sequences inherits GO and PDB labels, with some 19% with a cluster specific GO term (P-value≤0.001); another

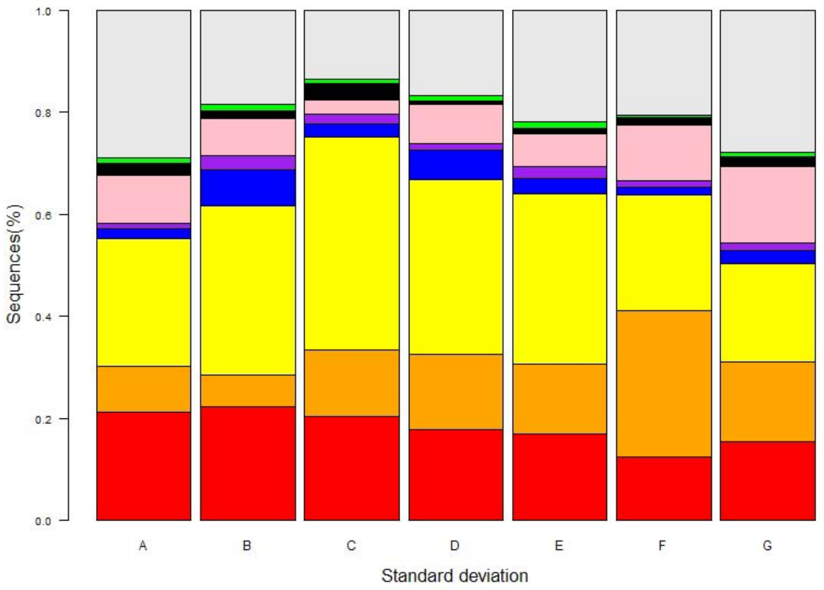


Fig. 6.2. : Barplot of the different degrees of annotation provided by the method for each standard deviation class (Table 6.1), computed on the S201 data set.

21% of S599 has statistically significant associated GO terms, and a remaining 8% has GO with a P-value>0.001. In total 50% of S599 is included in this annotation process, while the direct Uniprot annotation covers only 21% of S599. These results suggest that the method allows a more rigorous and safer annotation of S599, with the advantage of statistically validating and transferring the associated GO terms, and the template/s structures (Table 6.4).

6.6.5 Blind test on a data set of 201 genomes

In order to further test the predictive power of the method a blind test on 201 genomes (S201) not included in the clustering procedure was performed. The never-seen-before sequences of S201 were aligned with those contained in the clusters with the same constraints as before: a sequence ended up in a pre-computed cluster provided that both constraints of Eq.6.1 were simultaneously satisfied with respect to one of the S599 sequences in the clusters. 46% of the 724,854 protein sequences of S201 is assigned to property-specific clusters (characterized by GO terms with P-value $\leq$0.001) and for over 50% of these, a structure template is also provided. Another 6% is endowed with a non-cluster specific GO term/s. By this some 50% of S201 is annotated with our procedure. In this experiment, the Uniprot annotation is adopted only for comparison. The method recovers/predicts most of the Uniprot annotation (about 13% of S201) and allows through “inheritance” the annotation of 37% more sequences of S201. In Figure 6.2 the results on the S201 data set are shown. For each of the standard deviation classes (from A to G) defined in Table 6.1, the percentage of sequences of S201 data set annotated with the 8 most informative annotations in ascending order: statistically-validated GO term and mono-domain SCOP structural template (in red), statistically-validated GO term and SCOP multi-domain structural template (in orange), statistically-validated GO term without structural template (in yellow), GO term with P-value >0.001 with a mono-domain structural template (in blue), GO term with P-value>0.001 and a multi-domain structural template (in purple), GO term with P-value>0.001 and without structural template (in pink), without GO term but a mono-domain structural template (in black), without GO term with a multi-domain structure (in green), without GO term and PDB structure (in grey).

6.7 Discussion

Each cluster is characterized by a number of attributes among which are the GO terms and the PDB/SCOP structures that have been proved to be cluster-specific and that represent the annotation of the sequences within the cluster. With the described procedure, an uncharacterized sequence which falls into one of the computed clusters can be assigned to a function (GO term/s) and also to a structural template (PDB/SCOP), when available. As an example, the CDK2_HUMAN protein sequence (P24941) belongs to a cluster of 900 sequences. The cluster have 249 associated structures (corresponding to 27 proteins) and it corresponds to 24 validated GO molecular function terms. All the retrieved structures correspond to the same SCOP family, the catalytic subunit of protein kinases, which is an α+β domain (SCOP identifier d.144.1.7). The same protein, classified using the ProtoNet server, falls into a bigger cluster of 6,056 protein sequences, among which 80 are associated to 406 structures. ProtoNet includes in the cluster 37 keywords of GO molecular function. About 90% of the cluster structures correspond to the catalytic subunit of protein kinases but the remaining 10% of the structures is distributed over other 44 SCOP superfamilies keywords, that comprise not only α+β domains but also domains belonging to all other SCOP structural classes, mainly all-α (95 structures) and all-β (49) structures. The ClusTR server places the cyclin dependent kinase in a hierarchy of clusters with protein kinase activity but with no GO terms nor protein structures associated. This example highlights the fact that our clustering procedure ends with groups of functionally related proteins that are very specific for that type of function and that it can also give a reliable structural template to the cluster members.

Conclusions

Since the vast majority of protein sequences produced by the various genome projects has not yet been experimentally characterized and since there is very little that is known about their function, these sequences need to be interpreted from a structural point of view and from the point of view of the functions. The aim of this work has been the analysis, by means of computational tools, of the paths that connect protein structure to function and protein sequence to structure and function. With the advent of the structural genomics initiatives, that are complementing the data on which computational methods rely by increasing the functional diversity of protein sequences for which the structure has been determined, an increasing number of protein structures are being experimentally determined while their function is still unknown. In these cases, function can sometimes be predicted by using the structure rather than the sequence of the protein. Nevertheless, the scarcity of experimentally solved protein structures means that most function prediction is carried out by comparing protein sequences, and the recent substantial growth in complete genome sequences is making these methods more powerful. In particular, family-based methods that exploit sequence clustering can be extremely valuable in providing information on the variation in functional properties across a family. For this reason, there is considerable activity today trying to bridge the gap between protein sequence, structure and function. The integration between these different aspects of the analysis of protein structure and function aims to develop better tools for function prediction.

Contacts between protein residues constrain protein folding and characterize different protein structures. Therefore, prediction of residue contacts in proteins is an interesting problem whose solution may be useful in protein folding recognition and *de novo* design. In the last years, small-world behavior has been extensively described for proteins, when they are represented by the undirected graph defined by the inter-residue protein contacts. By adopting this representation it was possible to compute the average clustering coefficient (C) and characteristic path length (L) of protein structures, and their values were found to be similar to those of graphs characterized by small-world topology. Analyzing a large set of non-redundant protein structures, I showed that the small-world behaviour of inter-residue contact graphs is conditioned by the backbone connectivity. Indeed, by randomly mimicking the protein collapse, the covalent structure of the protein chain significantly contributes to the small-world behaviour of the inter-residue contact graphs. When protein graphs are generated, imposing constraints similar to those induced by the backbone connectivity, their characteristic path lengths and clustering coefficients are indistinguishable from those computed using the real contact maps showing that L and C values cannot be used for "protein fingerprinting". Moreover I verified that these results are independent of the selected protein representations, residue composition and protein secondary structures.

In a second part of my work I focused on a particular class of protein structures, coiled-coils. The coiled-coil is a widespread protein structural motif known to mediate a variety of fundamental biological interactions. For this reason, recognizing a coiled-coil sequence and locating its coiled-coil domains is a key step towards the determination of the protein structure and function. The prediction of coiled-coil domains is a twofold issue: a predictive method should be able both to identify the proteins that contain coiled-coil segments in a given set of protein sequences (or in proteomes) and to predict the number and the location of coiled-coil domains in a protein chain. With this aim I developed a specific Hidden Markov Model (CCHMM_PROF) that, starting from the evolutionary information derived from multiple sequence alignments, is able to recognize coiled-coil proteins and to locate the coiled-coil segments on the sequences. This new method discriminates coiled-coil sequences with accuracy of 94% and achieves a True Positive Rate of 71% with only 1% of False Positives. Furthermore when analyzing the localization of coiled-coil segments in protein sequences, the method reaches an accuracy value at the residue level of 81% and a best per-segment and per-protein efficiency of 81% and 86%, respectively. The per-segment and per-protein indices were proposed as a part of a more complete scoring framework in order to have a robust evaluation of the method performances. The development of accurate computational methods for coiled-coil prediction can drive experiments towards the *de-novo* design of *ad-hoc* coiled-coil structures and my results indicate that the CCHMM_PROF outperforms all the existing programs and that it can be adopted for large-scale genome annotation.

Protein sequence annotation is a major challenge in the post-genomic era and thanks to the availability of complete genomes and proteomes, protein annotation has recently taken invaluable advantage from large-scale genome comparisons. In the last part of my work, I devised a new non-hierarchical clustering procedure characterized by a metric which ensures a reliable transfer of function between related proteins even in the case of multi-domain and distantly related proteins. The method takes advantage of the comparative analysis of 599 completely sequenced genomes, both from prokaryotes and eukaryotes and of a GO and PDB/SCOP mapping over the computed clusters. The statistical validation of the method demonstrated that the proposed clustering technique captures the essential information shared between homologous and distantly related protein sequences. By this, uncharacterized proteins can be safely annotated by inheriting the annotation of the cluster and I showed that some 50% of the considered protein sequences can safely inherit a validated function. For further validate the procedure, the annotation of some other 201 genomes was blindly tested, aligning their sequences against the computed clusters. Most of the Uniprot annotation, that include 13% of the set, is retrieved. The proposed system increases the amount of annotated sequences of another 37%, indicating that the method fully exploits the sequence to function and to structure information, as it is recovered from the presently available reference data bases.

Bibliography

[Altschul *et al.*, 1990] Altschul SF, Gish W, Miller W, Myers EW and Lipman DJ. Basic local alignment search tool, J Mol Biol. 215:403-110 (1990).

[Altschul *et al.*, 1997] Altschul SF, Madden TL, Schaffer AA, Zhang J, Zhang Z, Miller W and Lipman DJ. (1997) Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. Nucleic Acids Res. 25(17):3389-3402 (1997).

[Andricioaei *et al.*, 2001] Andricioaei I, Voter AF and Straub JE. Smart Darting Monte Carlo. Journal of Chemical Physics 114:6994-7000 (2001).

[Anfinsen, 1973] Anfinsen CB. Priciples that govern the folding of proteins chain, Science 181, pp. 223-230 (1973).

[Anson and Myers, 1997] Anson EL and Myers GW. Realigner: a program for refining DNA sequence multi-alignments. J Comput Biol 1997, 4:369-383.

[Apweiler *et al.*, 2004] Apweiler R, Bairoch A, Wu CH, Barker WC, Boeckmann B, Ferro S, Gasteiger E, Huang H, Lopez R, Magrane M, Martin MJ, Natale DA, O'Donovan C, Redaschi N and Yeh LS. UniProt: the Universal Protein knowledgebase. Nucleic Acids Res 32:D115-119 (2004).

[Atilgan *et al.*, 2004] Atilgan AR, Akan P and Baysal. Small-world communication of residues and significance for protein dynamics Biophys J. 86:8591 (2004).

[Atilgan *et al.*, 2007] Atilgan AR, Turgut D and Atilgan C. Screened nonbonded interactions in native proteins manipulate optimal paths for robust residue communication Biophys J. 92:305262 (2007).

[Bagler and Sinha, 2005] Bagler G and Sinha S. Network properties of protein structures. Physica A 346:2733 (2005).

[Baldi and Brunak, 2001] Baldi P and Brunak S. BIOINFORMATICS: The machine learning approach. Second edition. A Bradford Book. The MIT Press (2001).

[Barabasi and Albert, 1999] Barabasi AL and Albert R. Emergence of scaling in random networks. Science 286:509 (1999).

[Barabasi and Albert, 2002] Barabasi AL and Albert R. Statistical mechanics of complex networks. Review of modern physics. 74:47-97 (2002).

[Barrat and Weigt, 2000] Barrat A and Weigt M. On the properties of small-world network models. Europhys Lett. 13:574 (2000).

[Bateman A *et al.*, 2000] Bateman A, Birney E, Durbin R, Eddy SR, Howe KL and Sonnhammer ELL. The Pfam Protein Families Database. Nucleic Acids Res. 28:263-266 (2000).

[Baum LE, 1972] Baum LE. An equality and associated maximization technique in statistical estimation for probabilistic functions of Markov processes. Inequalities, 3:1-8 (1972).

[Berger *et al.*, 1995] Berger B, Wilson DB, Wolf E, Tonchev T, Milla M and Kim PS. Predicting coiled coils by use of pairwise residue correlations. Proc Natl Acad Sci USA. 92:8259-8263 (1995).

[Berman HM *et al.*, 2000] Berman HM, Westbrook J, Feng Z, Gilliland G, Bhat TN, Weissig H, Shindyalov IN and Bourne PE. The Protein Data Bank. Nucleic Acids Res. 28:235-242 (2000).

[Berman HM *et al.*, 2003] Berman HM, Henrick K and Nakamura H. Announcing the worldwide Protein Data Bank. Nature Structural Biology 10 (12), 980 (2003).

[Böde *et al.*, 2007] Böde C, Kovacs IA, Szalay MS, Palotai R, Korcsmaros T and Csermely P. Network analysis of protein dynamics. FEBS Lett. 581:277682 (2007).

[Bohr *et al.*, 1993] Bohr J, Bohr H, Brunak S, Cotterill R M, Fredholm H, Lautrup B and Petersen S B. Protein structures from distance inequalities. J Mol Biol. 231: 861-869 (1993).

[Bradley AP, 1997] Bradley AP. The use of the area under the ROC curve in the evaluation of machine learning algorithms. Pattern recognition 30:1145-1159 (1997).

[Brenner SE, 1999] Brenner SE. Errors in genome annotation. Trends Genet. 15: 132133 (1999).

[Caffrey *et al.*, 1998] Caffrey M, Cai M, Kaufman J, Stahl J, Wingfiled SJ, Covell DG, Gronenborn AM and Clore GM. Three-dimensional solution structure of the 44 kDa ectodomain of SIV gp41. EMBO J. 17: 4572-4584 (1998).

[Chan and Kim, 2000] Chan DC and Kim PS. HIV entry and its inhibition. Cell 93(5):681-684 (2000).

[Chothia and Lesk, 1986] Chothia C and Lesk A M. The relation between the divergence of sequence and structure in proteins. EMBO J. 5:823826 (1986).

[Conway and Parry, 1990] Conway JF and Parry DA. Structural features in the heptad substructure and longer range repeats of two-stranded alpha-fibrous proteins. Int J Biol Macromol. 12(5):328-34 (1990).

[Cormen *et al.*, 2001] Cormen TH, Leiserson CE, Rivest RL and Stein. Introduction to Algorithms 2nd Edition, MIT press (2001).

[Delorenzi and Speed, 2002] Delorenzi M and Speed T. An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. Bioinformatics 18:617-625 (2002).

[Dempster AP *et al.*, 1977] Dempster AP, Laird NM and Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J Royal Stat Soc B, 39:1-38 (1977).

[Devos and Valencia, 2000] Devos D and Valencia A. Practical limits of function prediction. Proteins, 41:98-107 (2000).

[Devos and Valencia, 2001] Devos D and Valencia A. Intrinsic errors in genome annotation. Trends Genet. 17:429431 (2001).

[Dorogovtsev and Mendes, 2000] Dorogovtsev SN and Mendes JF. Exactly solvable small-world network. Europhys. Lett. 50:1 (2000).

[Dokholyan *et al.*, 2002] Dokholyan NV, Li L and Shakhnovich EI. Topological determinants of protein folding. Proc Natl Acad Sci. 99:863741 (2002).

[Durbin *et al.*, 1998] Durbin R, Eddy S, Krogh A and Mitchinson G. Biological sequence analysis: probabilistic models of proteins and nucleic acids. Cambridge University Press (1998).

[Enright *et al.*, 2002] Enright AJ, Van Dongen S and Ouzounis CA. An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. 30:1575-1584 (2002).

[Erdös and Rényi, 1960] Erdös P and Rényi A. On the strength of connectedness of a random graph. Publ Math Inst Hung Acad Sci. 5:17-61 (1960).

[Fariselli and Casadio, 1999] Fariselli P and Casadio R. A neural network based predictor of residue contacts in proteins. Protein Engineering 12:15-21 (1999).

[Fariselli *et al.*, 2001] Fariselli P, Olmea O, Valencia A and Casadio R. Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. Proteins Suppl 5:157-162 (2001).

[Fariselli *et al.*, 2001] Fariselli P, Olmea O, Valencia A and Casadio R. Prediction of contact maps with neural networks and correlated mutations. Protein Engineering 14:835-843 (2001).

[Fariselli *et al.*, 2005] Fariselli P, Martelli P and Casadio R. A new decoding algorithm for hidden Markov models improves the prediction of the topology of all-beta membrane proteins. BMC Bioinformatics 6 Suppl 4:S12 (2005).

[Fariselli *et al.*, 2007] Fariselli P, Molinini D, Casadio R and Krogh A. Prediction of structurally-determined coiled-coil domains with Hidden Markov Models. Lecture Notes in Computer Science 4414:292-302 (2007).

[Galaktionov and Marshall, 1994] Galaktionov SG and Marshall GR. 27th Annual Hawaii International Conference on System Sciences (HICSS-27), Maui, Hawaii (1994).

[Gotoh O, 1982] Gotoh O. An improved algorithm for matching biological sequences. J Mol Biol, 162:705-708 (1982).

[Greene and Higman, 2003] Greene LH and Higman VA. Uncovering network systems within protein structures. J Mol Biol. 334:78191 (2003).

[Greene *et al.*, 2007] Greene LH, Lewis TE, Addou S, Cuff A, Dallman T, Dibley M, Redfern O, Pearl F, Nambudiry R, Reid A, Sillitoe I, Yeats C, Thornton JM and Orengo CA. The CATH domain structure data base: new protocols and classification levels give a more comprehensive resource for exploring evolution. Nucleic Acids Res. D291-297 (2007).

[Gruber and Lupas, 2003] Gruber M and Lupas AN. Historical review: another 50th anniversary–new periodicities in coiled coils. Trends Biochem Sci. 28:679-685 (2003).

[Gruber *et al.*, 2005] Gruber M, Sding J and Lupas AN. REPPER-repeats and their periodicities in fibrous proteins. Nucleic Acids Res. 33:239-243 (2005).

[Gruber *et al.*, 2006] Gruber M, Sding J and Lupas AN Comparative analysis of coiled-coil prediction methods. J Struct Biol. 155(2):140-145 (2006).

[Havel TF, 1998] Havel TF. Distance geometry: Theory, algorithms, and chemical applications. Encyclopedia of Computational Chemistry. John Wiley & Sons.

[Hegyi and Gerstein, 2001] Hegyi H and Gerstein M. Annotation transfer for genomics: measuring functional divergence in multi-domain proteins. Genome Research. 11:1632:1640 (2001).

[Henikoff and Henikoff, 1992] Henikoff S and Henikoff JG. Amino acid substitution matrices from protein blocks. Proc Natl Acad Sci USA. 89:10915-10919 (1992).

[Hinds and Levitt, 1992] Hinds DA and Levitt MA A lattice model for protein structure prediction at low resolution. Proc Natl Acad Sci USA. 89:253640 (1992).

[Holm and Sander, 1993] Holm L and Sander C. Protein structure comparison by alignment of distance matrices. J Mol Biol. 233: 123-138 (1993).

[Holmes and Durbin, 1998] Holmes I and Durbin R. Dynamic programming alignment accuracy. J Comput Biol. 5:493-504 (1998).

[Hubbard *et al.*, 2002] Hubbard T, Barker D, Birney E, Cameron G, Chen Y, Clark L, Cox T, Cuff J, Curwen V, Down T, Durbin R, Eyras E, Gilbert J, Hammond H, Huminiecki L, Kasprzyk A, Lehvaslaiho H, Lijnzaad P, Melsopp C, Mongin E, Pettett R, Pocock M, Potter S, Rust A, Schmidt E, Searle S, Slater G, Smith J, Spooner W, Stabenau A, Stalker J, Stupka E, Ureta-Vidal A, Vastrik I and Clamp M. The Ensembl genome database project. Nucleic Acids Res. 30:38-41 (2002).

[Kaplan *et al.*, 2005] Kaplan N, Sasson O, Inbar U, Friedlich M, Fromer M, Fleischer H, Portugaly E, Linial N and Linial M. ProtoNet 4.0: a hierarchical classification of one million protein sequences. Nucleic Acids Res. 33:D216-D218 (2005).

[Krause *et al.*, 2002] Krause A, Stoye J and Vingron M. The SYSTERS protein sequence cluster set. Nucleic Acids Res. 28:270-272 (2002).

[Kriventseva *et al.*, 2001] Kriventseva EV, Fleischmann W, Zdobnov EM and Apweiler R. CluSTr: a database of clusters of SWISS-PROT+TrEMBL proteins. Nucleic Acids Res. 29:33-36 (2001).

[Krogh A, 1994] Krogh A. Hidden Markov models for labeled sequences. Proceedings 12th International conference on Pattern Recognition IAPR 1994; 12:140-144, IEEE Computer Soc Press (1994).

[Krogh *et al.*, 1994] Krogh A, Brown M, Mian S, Sjlander K and Haussler D. Hidden Markov models in computational biology: applications to protein modeling. J Mol Biol, 235:1501-1531 (1994).

[Hedger and Holm, 2001] Hedger A and Holm L. Picasso: generating a covering set of protein family profiles. Bioinformatics. 17:272-279 (2001).

[Laskowski *et al.*, 1997] Laskowski RA, Hutchinson EG, Michie AD, Wallace AC, Jones ML and Thornton JM. PDBsum: A Web-based database of summaries and analyses of all PDB structures. Trends Biochem Sci., 22:488-490 (1997).

[Lesk AM, 2008] Introduction to Bioinformatics (Third edition). Oxford University Press Inc, New York.

[Loewenstein *et al.*, 2008] Loewenstein Y, Portugaly E, Fromer M and Linial M. Efficient algorithms for accurate hierarchical clustering of huge data sets: tackling the entire protein space. Bioinformatics. 24: i41-i49 (2008).

[Lu *et al.*, 1995] Lu M, Blacklow SC and Kim PS. A trimeric structural domain of the HIV-1 transmembrane glycoprotein. Nat Struct Biol. 2:1075-1082 (1995).

[Lupas *et al.*, 1991] Lupas A, Van Dyke M and Stock J. Predicting coiled coils from protein sequences. Science 252:1162-1164 (1991).

[Lupas A, 1996] Lupas A. Prediction and analysis of coiled-coil structures. Methods Enzymol. 266:L513-525 (1996).

[Lupas and Gruber, 2005] Lupas AN and Gruber M. The structure of alpha-helical coiled coils. Adv Protein Chem. 70:37-78 (2005).

[MacCallum, 2004] MacCallum RM. Striped sheets and protein contact prediction. Bioinformatics 20 Suppl 1:I224-I231 (2004).

[Malashkevich *et al.*, 1998] Malashkevich VN, Chan DC, Chutkowski CT, Kim PS. Crystal structure of the simian immunodeficiency virus (SIV) gp41 core: conserved helical interactions underlie the broad inhibitory activity of gp41 peptides. Proc Natl Acad Sci USA 95(16):9134-9139 (1998).

[Malashkevich *et al.*, 1999] Malashkevich VN, Schneider BJ, McNally ML, Milhollen MA, Pang JX and Kim PS. Core structure of the envelope glycoprotein GP2 from Ebola virus at 1.9- resolution. PNAS 96(6):2662-267 (1999).

[Martelli *et al.*, 2002] Martelli PL, Fariselli P, Krogh A and Casadio R. A sequence-profile-based HMM for predicting and discriminating b-barrel membrane proteins. Bioinformatics 18:S46-S53 (2002).

[Mashaghi *et al.*, 2004] Mashaghi AR, Ramezanpour A and Karimipour V. Investigation of a protein complex network. Eur Phys J B. 41:11321 (2004).

[Matthews *et al.*, 2000] Matthews JM, Young TF, Tucker SP, Mackay JP. The core of the respiratory syncytial virus fusion protein is a trimeric coiled coil. Journal of Virology 74:5911-5920 (2000).

[McDonnell *et al.*, 2006] McDonnell AV, Jiang T, Keating AE, Berger B. Paircoil2: improved prediction of coiled coils from sequence. Bioinformatics 22:356-358 (2006).

[Moore and McCabe, 2006] Moore DS and McCabe GP. Introduction to the practice of statistics. 5th ed; W.H. Freeman and Company: New York (2006).

[Moutevelis and Woolfson, 2009] Moutevelis E, Woolfson DN. A periodic table of coiled-coil protein structures. J Mol Biol. 2009 Jan 23;385(3):726-32 (2009).

[Murzin *et al.*, 1995] Murzin AG, Brenner SE, Hubbard T and Chothia C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. J Mol Biol., 247:536-540 (1995).

[Neal and Hinton , 1998] Neal RM and Hinton GE. A new view of the EM algorithm that justifies incremental and other variants. Learning in graphical models. Editor: Michael Irwin Jordan, Edition, 1998, MIT press.

[Needleman and Wunsch, 1970] Needleman SB and Wunsch CD. A general method applicable to the search for similarities in the amino acid sequence of two proteins. J Mol Biol, 48:443-453 (1970).

[Orengo *et al.*, 1997] Orengo CA, Michie AD, Jones DT, Swindells MB, and Thornton JM. CATH: A Hierarchic Classification of Protein Domain Structures. Orengo,C.A., Michie,A.D., Jones,D.T., Swindells,M.B., Thornton,J.M. Structure, 5:1093-1108 (1997).

[Parry *et al.*,2008] Parry DA, Fraser RD, Squire JM. Fifty years of coiled-coils and alpha-helical bundles: A close relationship between sequence and structure. J Struct Biol. 163(3):258-69 (2008).

[Petryszak *et al.*,2005] Petryszak R, Kretschmann E, Wieser D and Apweiler R. The predictive power of the CluSTr database. Bioinformatics. 3604-3609 (2005).

[Pollastri and Baldi, 2002] Pollastri G and Baldi P. Prediction of contact maps by GIOHMMs and recurrent neural networks using lateral propagation from all four cardinal corners. Bioinformatics 18 Suppl 1:S62-S70 (2002).

[Portugaly *et al.*, 2006] Portugaly E, Harel A, Linial N and Linial M. EVEREST: automatic identification and classification of protein domains in all protein sequences. BMC Bioinformatics. 7: 277 (2006).

[Pruitt *et al.*, 2007] Pruitt KD, Tatusova T, Maglott DR. NCBI Reference Sequence (RefSeq): a curated non-redundant sequence database of genomes, transcripts and proteins. Nucleic Acids Res. 35(Database issue):D61-5 (2007).

[Punta and Rost, 2005] Punta M. and Rost B. PROFcon: novel prediction of long-range contacts. Bioinformatics 21:2960-2968 (2005).

[Redfern *et al.*, 2007] Redfern O, Harrison A, Dallman T, Pearl FMG and Orengo C. CATHEDRAL: a fast and effective algorithm to predict folds and domain boundaries from multidomain protein structures. PLOS Computationl Biology. 3:e232 (2007).

[Rost B, 2002] Rost B. Enzyme function less conserved than anticipated. J Mol Biol. 318:595-608 (2002).

[Rost and Sander, 2003] Rost B and Sander C. Improved prediction of protein secondary structure by use of sequence profiles and neural networks. Proc Natl Acad Sci USA. 90:7558-62 (2003).

[Sander and Schneider, 1991] Sander C and Schneider R. Database of homology-derived protein structures and the structural meaning of sequence alignment. Proteins 9:56-68 (1991).

[Scala A *et al.*, 2001] Scala A, Nunes Amaral LA and Barthélémy M. Small-world networks and the conformation space of a short lattice polymer. Europhys Lett. 55:594 (2001).

[Shao and Bystroff, 2003] Shao Y and Bystroff C. Predicting interresidue contacts using templates and pathways. Proteins 53 Suppl 6: 497-502 (2003).

[Shindyalov and Bourne, 1998] Shindyalov IN and Bourne PE. Protein structure alignment by incremental combinatorial extension (CE) of the optimal path. Protein Eng. 11:739-747 (1998).

[Smith and Waterman, 1981] Smith TF and Waterman MS. Identification of common molecular subsequences. J Mol Biol, 147:195-197 (1981).

[Stanica P, 2001] Stanica P. Good Lower and Upper Bounds on Binomial Coefficients. JIPAM. Volume 2, Issue 3, Article 30 (2001).

[Testa *et al.*, 2009] Testa OD, Moutevelis E and Woolfson DN CC+: a relational database of coiled-coil structures. Nucleic Acids Res. 37(Database issue):D315-22 (2009).

[The Gene Ontology Consortium, 2000] The Gene Ontology Consortium. Gene Ontology: tool for the unification of biology. Nature Genet. 25: 25-29 (2000).

[Tian and Skolnick, 2003] Tian W and Skolnick J. How well is enzyme function conserved as a function of pairwise sequence identity? J Mol Biol. 333:863-882 (2002).

[Vassura *et al.*, 2007] Vassura M, Margara L, Di Lena P, Medri F, Fariselli P and Casadio R. Reconstruction of 3D structures from protein contact maps. Proc. 3rd Int. Symp. on Bioinformatics Research and Applications: (ISBRA 2007) (Atlanta) (Berlin: Springer), Lecture Notes in Computer Science vol 4463:57889 (2007).

[Vendruscolo and Domany, 1999] Vendruscolo M and Domany E. Protein folding using contact maps. arXiv cond-mat/, 9901215 (1999).

[Vendruscolo *et al.*, 1999] Vendruscolo M, Subramanian B, Kanter I, Domany E and Lebowitz J. Statistical properties of contact maps Phys Rev E. 59:97784 (1999).

[Vendruscolo *et al.*, 2002] Vendruscolo M, Dokholyan NV, Paci E and Karplus K. Small-world view of the amino acids that play a key role in protein folding. Phys Rev E. 65:061910 (2002).

[Walshaw and Woolfson, 2001] Walshaw J and Woolfson DN. Socket: a program for identifying and analysing coiled-coil motifs within protein structures. J Mol Biol. 307:1427-1450 (2001).

[Walter *et al.*, 2009] Walter MC, Rattei T, Arnold R, Gldener U, Mnsterktter M, Nenova K, Kastenmller G, Tischler P, Wlling A, Volz A, Pongratz N, Jost R, Mewes HW and Frishman D. PEDANT covers all complete RefSeq genomes. Nucleic Acids Res. 37 (Database issue):D408-11 (2009).

[Watts and Strogatz, 1998] Watts DJ and Strogatz SH. Collective dynamics of "small-world" networks. Nature, 393:440-442 (1998).

[Watts DJ, 1999] Watts DJ. Small Worlds. The Dynamics of Networks Between Order and Randomness. Princeton, NJ. Princeton University Press (1999).

[Whisstock and Lesk, 2003] Whisstock JC and Lesk AM. Prediction of protein function from protein sequence and structure. Q Rev Biophys. 36: 307340 (2003).

[Wilming *et al.*, 2008] Wilming LG, Gilbert JGR, Howe K, Trevanion S, Hubbard T and Harrow JL. The vertebrate genome annotation (Vega) database. Nucleic Acids Res. 2008. doi:10.1093/nar/gkm987 (2008).

[Wilson *et al.*, 2000] Wilson CA, Kreychman J and Gerstein M. Assessing annotation transfer for genomics: quantifying the relations between protein sequence, structure and function through traditional and probabilistic scores. J Mol Biol. 297:233-249 (2000).

[Wu *et al.*, 2001] Wu CH, Huang H, Nikolskaya A, Hu Z and Barker WC. The iProClass integrated database for protein functional analysis. Nucleic Acids Res. 29:52-54 (2001).

[Wolf *et al.*, 1997] Wolf E, Kim PS and Berger B. MultiCoil: a program for predicting two- and three-stranded coiled coils. Protein Sci. 6:1179-1189 (1997).

[Zemla *et al.*, 1999] Zemla A, Venclovas C, Fidelis K and Rost B. A modified definition of Sov, a segment-based measure for protein secondary structure prediction assessment. Proteins 34:220-223 (1999).

[Zhao *et al.*, 2000] Zhao X, Sinh M, Malashkevich VN and Kim PS. Structural characterization of the human respiratory syncytial virus fusion protein core. PNAS 97(26):14172-14177 (2000).

[Zhao and Karypis, 2003] Zhao Y and Karypis G. 3rd IEEE International Conference on Bioinformatics and Bioengineering (BIBE)(2003).

Publications

- Bartoli L, Montanucci L, Fronza R, Martelli PL, Fariselli P, Carota L, Donvito G, Maggi GP and Casadio R. BAR: a non-hierarchical method for the functional and structural annotation of protein sequences relying on a comparative large-scale proteome analysis. Research article *Submitted* to Journal Of Proteome Research.

- Bartoli L, Fariselli P, Krogh A and Casadio R. CCHMM_PROF: Improving Coiled-Coil Prediction with Evolutionary Information. Research article *Submitted* to 17th annual International Conference on Intelligent Systems for molecular Biology (ISMB) and 8th European Conference on Computational Biology (ECCB).

- Ezkurdia I, Bartoli L, Fariselli P, Casadio R, Valencia A and Tress ML. Progress and challenges in predicting protein-protein interaction sites. Review *Submitted* to Briefings in Bioinformatics.

- Bartoli L, Fariselli P and Casadio R. (2008) The effect of backbone on the small-world properties of protein contact maps. *Physical Biology* Jan 8;4(4):L1-5.

- Carota L, Bartoli L, Fariselli P, Martelli PL, Montanucci L, Maggi G and Casadio R. (2008) High Throughput Comparison of Prokaryotic Genomes. Proceedings of PPAM Conference (Gdansk, Poland, 9-12 September 2007), PPAM 2007, Lecture Notes in Computer Science 4967, pp.1200-1209, Wyrzykowski et al. (Eds.), Springer-Verlag Berlin Heidelberg.

- Bartoli L, Capriotti E, Fariselli P, Martelli PL and Casadio R. (2007) The pros and cons of predicting protein contact maps. Protein structure prediction - Edited by Chris Bystroff and Mohammed Zaki, The Humana Press, Inc).

- Bartoli L, Calabrese R, Fariselli P, Mita DG and Casadio R. (2007) A computational approach for detecting peptidases and their specific inhibitors at the genome level. *BMC Bioinformatics*, 8 Suppl 1: S3.