Machine Learning in Structural Genomics

Andrea Passerini and Alessandro Vullo

Dipartimento di Sistemi e Informatica, Università di Firenze
Via di Santa Marta 3, I-50139 Firenze, Italy
Email: {passerini,vullo}@dsi.unifi.it
Phone: +39 055 4796 362, Fax: +39 055 4796 363

1 Introduction

Proteins are polymer chains composed of twenty simpler molecules, called amino acids, that carry out most of the molecular functions in living organisms. Although a protein can be first characterized by its amino acid sequence, or primary sequence, most proteins fold into three-dimensional (3D) structures that determine their function in living organisms. After the completion of several genome-sequencing projects, well over a million protein sequences are known and the sequence-structure gap has dramatically increased. Experimental methods for structure determination, X-ray crystallography and NMR spectroscopy, are costly and time consuming and do not keep pace with sequencing speed: at the beginning of 2004, the ratio of known structures to known sequences is approaching 1:50. Prediction of structure and function of novel proteins thus represents a strategic research frontier: bridging the gap between sequence and structure would increase our understanding of biological processes and our ability to enhance the quality and span of our lives. The ability to provide effective computational tools for protein structure prediction is a key to overcome experimental problems and to guide part of the future scientific effort in molecular biology.

In spite of four decades of intensive research effort, reliable protein structure prediction from sequence is far from being achieved, mainly because no comprehensive theory of folding exists and a global search in the conformational space of proteins is inherently intractable.

Only recently, this task has been aided by the development of intelligent tools borrowed from machine learning approaches. These methods extract relevant pieces of information from databases of known structures in order to focus the search on more tractable problems. So far, research has mainly focused on intermediate local structural descriptions (e.g. secondary structure, relative solvent accessibility etc.). These are commonly referred to as one-dimensional (1D) features, as they can be represented with linear strings

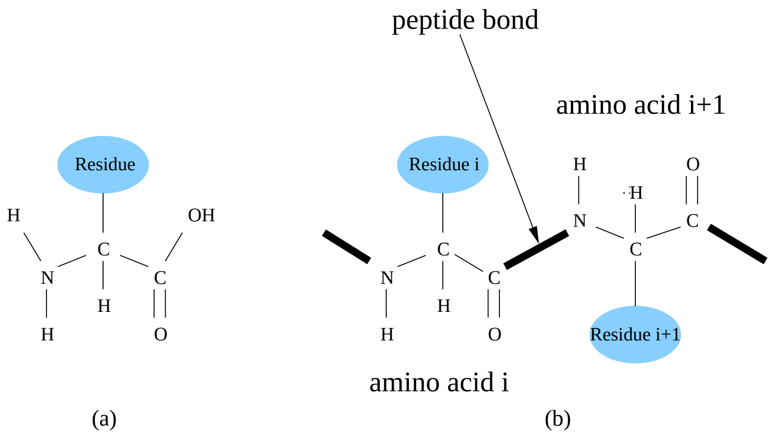


Figure 1: Amino Acids: (a) chemical structure; (b) peptide bond.

of symbols, and have proven to be a valuable aid for the prediction of full 3D structure. Machine learning has an intrinsic appeal because it represents the best theoretical and practical framework for building reliable statistical models, especially in presence of large amounts of data. Moreover, learning applications can be orders of magnitude faster than the alternatives.

This chapter is intended to give an overview of traditional and more recent approaches used for protein structure prediction. The focus is on applications that rely on the machine learning paradigm. The chapter is organized as follows. Section 2 and 3 provide basic elements of structural biology and a sketch of the important concept of alignment. These sections serve as an introduction to those computer scientists with no background in structural genomics and related computational tasks. In section 4 we elaborate on the concept of learning from examples. In order to understand the ideas forming the basis of this paradigm, we introduce to the field of statistical learning theory, the framework that studies the mathematical properties of learning machines. The section concludes with a discussion of connectionist (e.g. neural networks) and kernel-based methods, the most popular learning approaches used to solve structural biology problems. The folding problem is introduced in section 5 together with an outline of the context in which traditional prediction techniques (knowledge based methods and de novo algorithms) can and cannot be reliably applied. We conclude with a survey of applications of traditional and more recent learning algorithms related to important problems in structural genomics.

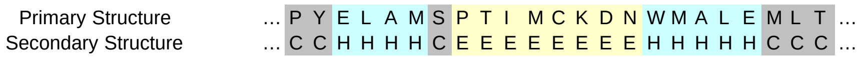
Glycine	Gly (G)	Lysine	Lys (K)
Alanine	Ala (A)	Arginine	Arg (R)
Valine	Val (V)	Tryptophan	Trp (W)
Leucine	Leu (L)	Serine	Ser (S)
Isoleucine	Ile (I)	Threonine	Thr (T)
Methionine	Met (M)	Cysteine	Cys (C)
Proline	Pro (P)	Tyrosine	Tyr (Y)
Phenylalanine	Phe (F)	Asparagine	Asn (N)
Aspartic acid	Asp (D)	Glutamine	Gln (Q)
Glutamic acid	Glu (E)	Histidine	His (H)

Table 1: The twenty different amino acids that occur in proteins.

2 Protein Structural Principles

Proteins are polymer chains composed of twenty simpler molecules, called *amino acids*, that carry out most of the molecular functions in living organisms. All amino acids share the same chemical structure as shown in Figure 1 (a). There is a central carbon atom (C_α) attached to an hydrogen atom, an amino group (NH_2), a carboxyl group (COOH) and a side chain or residue (R) that discriminate one amino acid from the others. The twenty possible residues that occur in proteins are listed in Table 1, together with their standard 3-letters and 1-letter code. The table groups the amino acids according to the chemical nature of their side chain. Belong to the class of the hydrophobic side chains the amino acids from Alanine to Phenylalanine. The class of charged residues goes from Aspartic Acid to Arginine, whereas the group from Tryptophan to Hystidine comprises those with polar side chains. The side chain of Glycine has simply a hydrogen atom and it is neither hydrophobic nor charged nor polar. During the synthesis process, proteins are sequentially assembled through the formation of *peptide bonds* (Figure 1 (b)), where the carboxyl group of one amino acid is joined with the amino group of another to release water. From a biochemical point of view, a protein is thus represented as a polypeptide chain formed by a *backbone* (the sequential repetition of the NH-C_αH-C=O basic unit) and a *side chain* (the sequence of residues attached to the backbone). Conventionally, the structure of proteins is hierarchically represented with three levels of description: primary, secondary and tertiary structure. The **primary structure** is the amino acid sequence of the polypeptide chain. Formally, it can be modeled as a string from a finite alphabet Σ_{aa} where $|\Sigma_{aa}| = 20$ (the one-letter codes of table 1). These strings form the basic information stored in biological data repositories, such as the Swissprot archive [BA00].

Higher level descriptions provide more information about the structure of a protein. This is mainly formed by a hydrophobic core and an hy-



Primary Structure	...	P	Y	E	L	A	M	S	P	T	I	M	C	K	D	N	W	M	A	L	E	M	L	T	...
Secondary Structure	...	C	C	H	H	H	H	C	E	E	E	E	E	E	E	E	H	H	H	H	H	C	C	C	...

Figure 2: A fragment of protein primary and secondary structure.

drophilic surface exposed to the solvent environment (mainly water). The hydrophobic core is formed packing hydrophobic side chains into the interior part of the molecule. This task is accomplished through the formation of a set of regular local structural patterns, the **secondary structure**, characterized by hydrogen-bonding between the main chain polar groups (NH and C=O) of different residues. There are two main patterns: *α-helix* (H) and *β-strand* (E). An α-helix is built up from one continuous region in the sequence through the formation of hydrogen bonds between C=O group of residue in position i and NH of residue $i+4$. The periodicity of the helix is 3.6 residues per turn. The π-helix and 3_{10} helix represent other types of observed helices, in which bonds are formed from residue i to residues $i+5$ and $i+3$, respectively. They are not energetically favorable and rarely occur. Similarly to helices, a β-strand is a fragment of consecutive residues, but it does not represent an isolated structural element: hydrogen-bonds are formed with one or more β-strands (that can be distant in sequence) to form a pleated sheet called *β-sheet*. Strands in a β-sheet are aligned adjacent to each other such that their amino acids have the same biochemical direction (parallel β-sheet) or alternating directions (anti parallel β-sheet). α-helices and β-strands are often connected by *loop regions* or *coils* (C), which can significantly vary in length and structure and have no fixed regular shape as the other two elements. For the majority of known protein structures, the hydrophobic core is built up from a combination of helices and strands and it provides the rigid stable structural framework. On the other hand, loop regions are generally found at the surface of the protein and usually act as functional (binding or active) sites. Every amino acid in the sequence belongs to one of the three structural types, and the secondary structure can be flattened to a string from an alphabet $\Sigma_{ss} = \{\text{H,E,C}\}$ and having the same length of the primary structure. Figure 2 is an example of representation of primary and secondary structure of a protein segment.

It has been observed that simple geometrical arrangements of few secondary structure elements occur in most of the proteins. They act as basic structural or functional building blocks and are called *motifs*. These motifs are formed by packing side chains from adjacent α-helices or β-strands close to each other. Two or more motifs combine to form compact globular structures, called *domains* [BT99]. One polypeptide chain can be arranged into one or several domains and a domain might act independently or in combination with other domains to define a functional unit.

The term **tertiary structure** refers to the three-dimensional arrange-

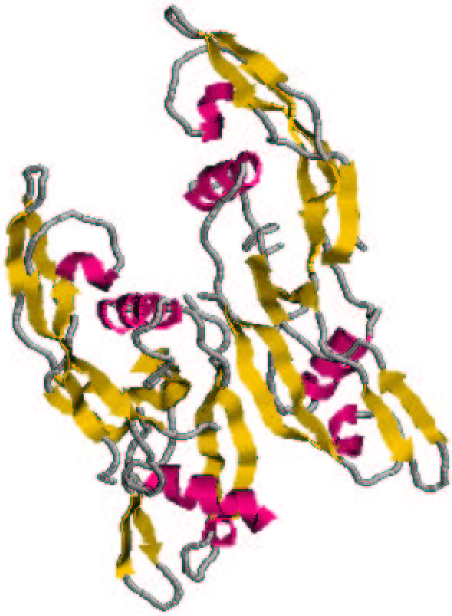


Figure 3: 3D structure of neurotrophic growth factor from rat, PDB code 1agq. Primary structure composed of four identical chains of 135 amino acids.

ment of backbone and side chain atoms of a polypeptide chain composed of one or several domains. It is the result of the combination of secondary structure elements due to interactions between the amino acids and the solvent environment. The protein universe can be roughly partitioned into single-chain (monomeric) proteins and molecules composed of two or more distinct polypeptide chains (multimeric proteins). Chains of a multimeric protein are often called protein sub-units. The **quaternary structure** is the term adopted to describe the complex spatial conformation of multimeric proteins. Figure 3 shows an example of schematic representation of tertiary and quaternary protein structure. The protein is composed of two identical molecules, each one being formed by two independent amino acid chains. Secondary structure is composed of eight α-helices (H, in purple) and several β-sheets formed by pairing two β-strands (E, in yellow). Loop regions connect consecutive strands and helices.

3 Algorithmic Processing of Evolution

Evolution at molecular level is commonly modeled as a process in which currently observed sequences have been selected from a common ancestral sequence. This process is guided by casual and deterministic events: random mutations preserving structural and functional features occur on sequences; those products that present an environmental advantage are then selected. Chains can be roughly grouped into homology classes, where *homology* is defined as similarity of structure, physiology, development and evolution of organisms based upon common genetic factors [BT99]. Usually, two proteins are considered to be homologous when they have identical amino acid residues in a significant number of of sequential positions along the polypeptide chains. However, it is frequently found that two proteins with sequence identity below the level of statistical significance have similar functions and structures. Sequence similarity can be inferred with the so called *alignment* algorithms. These algorithms usually employ dynamic programming techniques to compute string matching between two or more sequences. Similarity of the sequences is inferred when the level of sequence identity is above some manually derived threshold. We can distinguish between pairwise and multiple sequence alignment algorithms depending whether two or more sequences are super posed for the match. If we compute matching between entire sequences or only between substrings, the alignment is said to be global or local, respectively.

Computational molecular biology programs are often designed to elaborate *evolutionary information.* Concerning proteins structure prediction, it is known that protein structure is more conserved than sequence and similar sequences share similar structure [BB01, Ros96]. Suppose we are given a sequence s with unknown structure. To predict its structure, we can directly exploit the information contained into s. However, if we can find a set of sequences that present high similarity once aligned with s, we can think that this set contains more structural information than s itself. The success of the most effective predictive systems is largely based upon this empirical argument and then on their ability to process the information provided by multiple alignments.

The evolutionary information contained into a multiple alignment of say N sequences and L positions is usually compressed to a *profile*, that is a $20 \times L$ matrix P (PSSM, Position Specific Scoring Matrix), where for each position $l = 1, \ldots, L$ the column vector P_l contains the frequency of each amino acid in that position. This representation is more suitable for automatic numerical processing and still contains valuable information. For instance, in every position of the profile we can immediately recognize the degree of conservation of each residue or the mutations that are compatible with a correct structure.

4 Machine Learning

In its most general (and difficult) formulation, protein fold prediction from sequence amounts at predicting the three dimensional structure of a protein given its primary structure. More formally, we can see this task as a mapping problem in which the input is given by the sequence of amino acids and the output is represented by the sequence of coordinates triplets (one triplet for each amino acid). Other simplified problems in structural genomics admit similar formulations of the predictive task: given a set of instances the task consists in predicting the unknown characteristics (the output) of each instance given the known ones (the input).

In the machine learning framework, such problem is addressed with the development of algorithms capable of approximating the true (and unknown) mapping on the basis of experience provided by a set of training examples. In *supervised* learning, an algorithm is trained with a set of instances represented by input-output pairs, while in *unsupervised* learning only the input is made available to the algorithm. We will focus on the former approach, as it is the most common situation in protein structure prediction. In order to clarify the theoretical framework underlying the development of machine learning algorithms, we will start by introducing the basic principles of statistical learning theory, and will then present some of the most popular algorithms highlighting their characteristics within such framework.

4.1 Statistical Learning Theory

Learning from examples amounts at approximating an unknown function given a finite number of (possibly noisy) input-output pairs. A learning algorithm is characterized by the set of candidate functions, also called the hypothesis space, and the search strategy within this space. The sparseness and finiteness of training data poses the problem of generalization over unseen instances, e.g. the ability of the learned function to predict the correct output for an unseen input. A learning algorithm which simply outputs the function that best fits training data, this function being chosen from the set of *all* possible functions from the input to the output space, would simply memorize training examples without really developing a model of the underlying target function. In other words, the learner fails to generalize to unseen cases. Moreover, the problem is ill-posed, as there is no unique solution. The problem of avoiding *overfitting* [MP92] training data is typically addressed by restricting the set of allowed candidate functions, either by directly reducing the hypothesis space, by acting on the search strategy, or both, and is termed in different ways, such as bias-variance dilemma [GBD92], inductive bias [Mit97] or capacity control [GVB^{+}92]. In regularization theory, turning an ill-posed problem into a well-posed one is done by adding a regularization term to the objective function [Tik63]. In the learning framework this corresponds to modifying the search strategy by trading off between fitting of training data and limiting the complexity of the learned function. We will now give a more formal representation of these concepts.

Loss Function and Risk Minimization

Let $D_m = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^m$ be a training set, whose data are independently drawn and identically distributed with respect to an unknown probability distribution $P(x, y)$. Suppose we have a set of functions $\mathcal{F}_\alpha$ with parameters α, such that for each value of α we have a function $f_\alpha : \mathcal{X} \to \mathcal{Y}$. The set $\mathcal{F}_\alpha$ is called *hypothesis space* and is the space of candidate hypotheses. This space will be searched by the learning algorithm looking for the best hypothesis according to its search strategy. Let us give a formal definition of the loss incurred by the function f_α at example (x, y), that is the penalty the function should pay for predicting $f_\alpha(x)$ instead of y.

Definition 4.1 (Loss Function) *Given a triplet $(x, y, f_\alpha(x))$ containing a pattern x , its observation y and a prediction $f_\alpha(x)$, we define loss function any map $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty]$ such that $\ell(x, y, y) = 0$ for all $x \in \mathcal{X}$ and $y \in \mathcal{Y}$.*

The choice of the loss function typically depends on the type of learning task addressed. For binary classification problems, where $\mathcal{Y} = \{-1, 1\}$, common choices are the misclassification error:

$$\ell(x, y, f_\alpha(x)) = \begin{cases} 0 & \text{if } y = f_\alpha(x) \\ 1 & \text{otherwise} \end{cases} \tag{1}$$

and the *soft margin* loss function [BM92] (see fig.4(a)):

$$\ell(x, y, f_\alpha(x)) = |1 - yf_\alpha(x)|_+ = \begin{cases} 0 & \text{if } yf_\alpha(x) \geq 1 \\ 1 - yf_\alpha(x) & \text{otherwise} \end{cases} \tag{2}$$

which takes into account the confidence of the prediction. In the case that the function f_α outputs a class conditional probability, that is $f_\alpha(x) = p_\alpha(Y = 1|X = x)$, a common choice for the loss function is the *cross entropy*, typically employed in the log form:

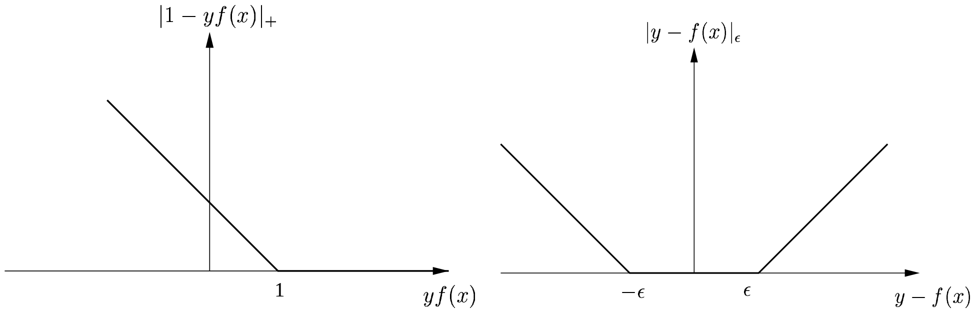
$$\ell(x, y, f_\alpha(x)) = -\left(\frac{1-y}{2}\log(1 - f_\alpha(x)) + \frac{1+y}{2}\log(f_\alpha(x))\right). \tag{3}$$

For regression tasks ($\mathcal{Y} = \mathbb{R}$), common losses are the square error:

$$\ell(x, y, f_\alpha(x)) = (y - f_\alpha(x))^2 \tag{4}$$

and the extension of soft margin loss called *ϵ-insensitive* loss (see fig.4(b)):

$$\ell(x, y, f_\alpha(x)) = |y - f_\alpha(x)|_\epsilon = \begin{cases} 0 & \text{if } |y - f_\alpha(x)| \leq \epsilon \\ |y - f_\alpha(x)| - \epsilon & \text{otherwise} \end{cases} \tag{5}$$



(a) Soft margin loss

(b) Epsilon insensitive loss

Figure 4: Confidence-based losses for binary classification (a) and regression (b).

which does not penalize deviations up to ϵ from the target value, and gives a linear penalty to further deviations. Note that all these losses only depend on x by $f_\alpha(x)$, whereas definition 4.1 is more general.

Given a loss function weighting errors on individual patterns, we can define the expectation of the test error for a trained function on the entire set of possible patterns.

Definition 4.2 (Expected Risk) *Given a probability distribution $P(x, y)$ of patterns and observations, a trained function $f_\alpha : \mathcal{X} \to \mathcal{Y}$ and a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, the expected risk for f_α is defined as*

$$R[f_\alpha] = \int_{\mathcal{X}\times\mathcal{Y}} \ell(x, y, f_\alpha(x))dP(x, y). \tag{6}$$

The expected risk is also known as generalization error or true error. An ideal learning algorithm should be able to choose the hypothesis which minimizes such value. However, we cannot directly minimize the expected risk, as the probability distribution $P(x, y)$ is unknown. The only error we can actually measure is the mean error rate on the training set D_l, also called the empirical risk.

Definition 4.3 (Empirical Risk) *Given a training set $D_m = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^m$ of patterns and observations a trained function $f_\alpha : \mathcal{X} \to \mathcal{Y}$ and a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \to [0, \infty)$, the empirical risk for f_α is defined as*

$$R_{emp}[f_\alpha] = \frac{1}{m}\sum_{i=1}^{m} \ell(x_i, y_i, f_\alpha(x_i)). \tag{7}$$

Minimizing this risk alone, however, does not give any guarantee on the value of the expected risk itself, especially if the size of the training set is small compared to the dimension of the hypothesis space. If we choose the set of functions $\mathcal{F}_\alpha$ to be the set of all functions from $\mathcal{X}$ to $\mathcal{Y}$, we can always find a function which has zero empirical error, but large generalization error. For instance, a function mapping each training pattern x_i to its observation y_i, and that maps every other pattern $x_j, j > m$ to a fixed value, does not achieve any learning at all.

In order to generalize to unseen patterns, we have to restrict the set of possible learning functions, taking into account the *complexity* or *capacity* of such set with respect to the learning task and the number of training examples available.

4.2 Examples of learning algorithms

Given a particular predictive task (i.e. binary classification), a learning algorithm is characterized by its hypothesis space, that is the set of all possible functions it can implement, and the search strategy within this space. The search strategy typically consists in minimizing a loss function over the set of training examples, while the problem of generalization can be addressed by reducing the hypothesis space and/or by modifying the search strategy in various ways in order to prefer simpler hypotheses as well as more suitable hypotheses given the prior knowledge available. In the following we will introduce two of the most popular learning algorithms, focusing on their characteristics in terms of hypothesis space, search strategy and generalization capabilities.

Kernel Machines

The concepts of statistical learning theory were investigated by Vapnik in the late Seventies [Vap79] and led to the development of the Support Vector Machine [Vap95] learning algorithm, later generalized to Kernel Machines. In the case of classification tasks, Kernel Machine algorithms learn a decision function which separates examples with a large margin, possibly accounting for training errors, thus actually trading off between function complexity and fitting the training data. Versions of the algorithm have been developed for tasks different from classification, such as regression [Vap95], clustering [BHHSV01] and ranking [FSS98, CD02], and have been successfully applied to a vast range of learning tasks, from handwritten digit recognition [CV95] to text categorization [Joa98].Many tutorials and books have been written on Kernel Machines (see for example [Bur98, SS02, CST00]).

Kernel Machines can be viewed in the framework of statistical learning theory as the problem of minimizing a regularized risk functional. Capacity control is thus implemented by modifying the search strategy, adding to the empirical risk a regularization term which penalizes more complex hypotheses.

Recall the problem of empirical risk minimization (see section 4.1), where we have a training set $D_m = \{(x_i, y_i) \in \mathcal{X} \times \mathcal{Y}\}_{i=1}^m$, a set of functions $\mathcal{F}_\alpha$ from $\mathcal{X}$ to $\mathcal{Y}$, and a loss function $\ell : \mathcal{X} \times \mathcal{Y} \times \mathcal{Y} \rightarrow [0, \infty]$. Assume for simplicity that $\mathcal{Y} \equiv \mathbb{R}$. Assume also that $R_{emp}[f_\alpha]$ is continuous in f_α [1]. Regularization theory deals with the problem of restricting $\mathcal{F}_\alpha$ in order to make it a compact set, thus obtaining a well-posed minimization problem [Tik63, Vap98]. Instead of directly specifying a compact set for $\mathcal{F}_\alpha$, which would cast the problem into a complex constrained optimization task, we add a regularization term $\Omega[f_\alpha]$ to the objective functional, such that $\Omega[f_\alpha] \geq 0$ for all f_α, and the sets

$$\mathcal{F}_{\alpha,c} = \{f_\alpha : \Omega[f_\alpha] \leq c\}, \quad c \geq 0,$$

are all compact. This results in a regularized risk functional

$$R_{reg}[f_\alpha] = R_{emp}[f_\alpha] + \lambda\Omega[f_\alpha]. \tag{8}$$

giving a well-posed minimization problem [Tik63, Vap98]. Here the regularization parameter $\lambda > 0$ trades the effect of training errors with the complexity of the function, thus providing a mean to control overfitting. By choosing Ω to be convex, and provided $R_{emp}[f_\alpha]$ is also convex, the problem has a unique global minimum.

When $\mathcal{F}_\alpha$ is a reproducing kernel Hilbert space $\mathcal{H}$ [Aro50, BCR84] associated to a kernel k, the *representer theorem* [KW71] gives an explicit form of the minimizers of $R_{reg}[f_\alpha]$:

$$f_\alpha(x) = \sum_{i=1}^{m} \alpha_i k(x_i, x). \tag{9}$$

The theorem states that regardless of the dimension of the RKHS $\mathcal{H}$, the solution lies on the span of the m kernels centered on the training points. Different choices of the loss function ℓ and the regularization functional Ω give rise to different Kernel achines. Support Vector Machines for binary classification [CV95, Vap95, Vap98], for example, employ the soft margin loss function (eq. 2), which as a cumulative loss becomes

$$\ell((x_1, y_1, f_\alpha(x_1)), \ldots, (x_m, y_m, f_\alpha(x_m))) = \frac{1}{m}\sum_{i=1}^{m} |1 - y_i f_\alpha(x_i)|_+, \tag{10}$$

and a regularizer of the form $\lambda\Omega[f_\alpha] = \frac{\lambda}{2}||f_\alpha||^2$. For $\lambda \rightarrow 0$ we have hard margin SVM where all training patterns have to be correctly classified. Note that the decision function for SVM is actually $sign(f_\alpha)$.

[1] This doesn't hold for the misclassification loss (eq. (1)), which should be replaced by a continuous approximation such as the soft margin loss (eq. (2)).

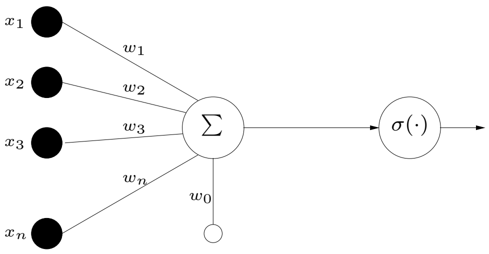


Figure 5: A schematic representation of the kind of processing performed by a neural network unit.

Neural Networks

Neural networks represent a popular framework for learning arbitrarily complex functions from examples. They are among the most effective learning methods currently used for structural biology applications (see section 5.2). Briefly, neural networks are built out of a set of interconnected simple units, where each unit processes a number of real-valued input (which may be the output of other units) and outputs a single value, which may be the input to other units. Figure 5 represents the computation performed by a generic simple unit. Given a vector $\bar{x} = (x_1, x_2, \ldots, x_n)$ of real-valued inputs, the unit computes a weighted linear combination of these inputs and outputs the function $\sigma(\bar{w} \cdot \bar{x} + w_0)$, where $\bar{w} = (w_1, \ldots, w_n)$ is the vector of weights of the linear combination and plays a central role in the learning procedure. $\sigma(\cdot)$ is called the activation function and can be of different types: linear, $\sigma(x) = x$, perceptron, $\sigma(x) = sign(x)$, sigmoid, $\sigma(x) = 1/(1 + e^{-\alpha x})$ or hyperbolic tangent, $\sigma(x) = tanh(x)$.

Typically, individual units are interconnected in layers that form a directed acyclic graph. Connections run from every unit in one layer to every unit in the next layer. The edge connecting unit i in layer $k-1$ to unit j in layer k is labeled by the weight w_{ij} which represents the strength of the corresponding connection. Figure 6 shows an example of a feed-forward network having two layers of weights. Networks of this form are called multilayered perceptrons (MLP). Intermediate units between input and output units are called hidden units and their corresponding layer is called hidden layer. The network of Fig. 6 has n inputs and m outputs and can be seen as a representation of a multivariate mapping between a set of n input variables and a set of m output variables. Depending of the type of unit activation functions, networks of this form can approximate arbitrarily complex functions. A three-layer network with threshold activation functions (i.e. perceptron) can represent an arbitrary decision boundary to arbitrary accuracy. Essentially any continuous functional mapping can be represented to arbitrary accuracy by a network having two layers of weights

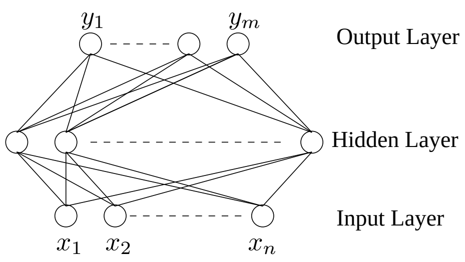


Figure 6: A feed-forward network having two layers of adaptive weights. The bias parameters (i.e. weights w_0 as in Fig.5) for hidden and output units are not shown.

with sigmoidal hidden units.

In MLPs, model parameters (i.e. the weights) are adaptive and can be adjusted to adapt the behavior of the model according to the training data. The key idea of network training is to use gradient descent to search the hypothesis space of possible weight vectors to find the weights that best fit the training examples. Gradient descent provides the basis for the popular *backpropagation* algorithm, which can learn networks with many interconnected units. Training is achieved by minimizing the empirical risk over the training set. For regression estimation, the criterion usually chosen is the Mean Squared Error (see loss in eq. 4)

$$R_{emp}[f] = \frac{1}{m}\sum_{i=1}^{m}(y_i - f(x_i))^2$$

and for classification the Cross-Entropy (see loss in eq. 3)

$$R_{emp}[f] = -\frac{1}{m}\sum_{i=1}^{m}\left(\frac{1-y_i}{2}\log(1-f(x_i)) + \frac{1+y_i}{2}\log(f(x_i))\right).$$

Controlling the generalization ability of MLPs (i.e. avoiding overfitting) is usually achieved in two different ways. The first and more principled method consists in the minimization of the regularized risk functional of eq. (8). A popular approach is known as the *weight decay* strategy, where the regularization term is a convex function of model parameters (i.e. the weights)

$$\Omega[f] = \frac{\|w\|^2}{2}$$

that simply corresponds at penalizing solutions with high values of the weights. The underlying idea is that if the weights are kept small, the unit activation functions are almost linear, hence with low complexity or capacity.

Differently from Kernel Machines, the problem of minimizing $R_{reg}[f]$ usually does not have a unique global optimum, because the network in general can represent an arbitrarily complex and possibly non-convex functional.

The second method is the use of *early stopping*, that is stopping training before reaching a local optimum, usually according to an estimate of generalization error on a independent "validation" set. Early stopping prevents tuning the weights to fit noise in the training data or unrepresentative characteristics of the particular training sample. Clearly, it can be applied only when enough data is available to provide an extra validation set.

5 Landscape of Protein Fold Prediction

After completion of the synthesis of its sequence, a protein assumes spatial structure with a process called *protein folding.* This process transforms the one-dimensional (linear) structure of the polypeptide chain into a three-dimensional (native) conformation which determines the biological function of the protein. According to the experimentally confirmed Anfinsen's hypothesis [Anf73], the amino acid sequence together with the solution environment contain all the information that is necessary to determine the native fold of a protein. This gave rise to the **protein folding problem**, the prediction of protein's tertiary structure from its amino acid sequence.

Finding a solution to the folding problem is one of the most difficult and challenging open problem in structural proteomics. Despite many decades of intensive research efforts, the problem has not a general solution yet. An evidence for this claim is the rapidly increasing *sequence-structure gap.* The number of proteins for which the sequences are known is well over a million [BA00], whereas the number of protein structures deposited in public databases is slightly more than twenty thousand [B$^+$77]. Excluding experimental difficulties, the reason for this impressive difference is largely due to the lack of a comprehensive theory of folding. At present, we only know a few reliable facts about the folding mechanism. Thus, it is still not possible to simulate protein folding (in a reasonable amount of time) [2] and generate the three-dimensional structure of any protein from its amino acid sequence.

In the absence of any useful theory of folding and automated high throughput structure determination projects, structure prediction tools play an increasing significant role. The development of new reliable and powerful prediction techniques represent one of most active research area in the field. In the following, we survey techniques and applications employed for protein

[2] In Molecular Dynamics, simulations compute folding trajectories using the physical laws of motion in appropriately devised potential fields. Unfortunately, protein folding can be simulated only for a negligible amount of time (nano or micro seconds). In *vivo* or in *vitro*, the process goes on for seconds or minutes.

fold prediction. Tools are mainly distinguished depending if the task is to predict directly 3D structure or instead simplified lower dimensional (1D or 2D) representations.

5.1 Prediction in three-dimensional space

The most accurate tools used to predict protein structure are the so called knowledge-based methods, i.e. comparative (homology) modelling and threading. These methods rely on the observation that many different sequences have similar folds, and in this case we say that they are homologous [BT99]. We are aware that the precise definition prescribes that two sequences are homologues if they have a common progenitor. Here we use the term homology in the context of structural similarity. Assuming that an input sequence with unknown structure is homologue to a protein with known fold, the target fold can be modelled using the structure of the homologue as a template. The existence and detection of homologous proteins with known structure is a necessary requirement for the application of these techniques.

Comparative (Homology) modelling

Empirical findings suggest that proteins whose percentage of sequence identity is more than 25-30% have similar 3D structures [MR03, Ros98, AB97]. Comparative modelling techniques such as [SS97] start from this argument to infer structural homology. Moreover, it is generally assumed that the unknown target and the template folds have identical backbones and very similar core regions[3] built up from a combination of secondary structure patterns. By this, model building is the problem of predicting conformation of the side-chain and the structure of loop regions, after having adjusted the input sequence to match the core of the template fold. Loop regions of the unknown structure are modelled using a database of side-chain orientations from proteins of known structure (rotamer libraries). Finally, conformation of the side-chain is predicted by energy refinement of the model with molecular dynamics simulations.

The accuracy of homology modelling clearly depends on the amount of sequence identity of the target and template structures. With high levels of sequence identity ($\geq 60\%$), homology-derived models are as accurate as experimental structure methods. When similarity is between 40% and 60%, energy refinement is successfully applied to find higher quality models only if it starts from an almost correct structure. An additional problem is the large computational time due to the increasing number of loop regions that have to be modeled. When the levels of sequence similarity are in the range

[3] An analysis of the relation between sequence similarity and 3D structure of the core region of homologous proteins revealed that proteins with high sequence identity are almost identical in structure (RMSD ≤ 2Å) [CL86].

25-30%, homology modeling can only find coarse-grained solutions. The main limitation of comparative modelling is its range of application. The requirement that sequence identity needs to be $\geq$ 25-30% confines model building by homology to the prediction of 3D structure for 10-30% of all protein sequences [Ros98].

Fold Recognition (Threading)

As previously observed, high sequence similarity allows to reliably infer structural similarity, but the majority of proteins with similar three-dimensional structures and functions have low sequence identity. In this case, two proteins are said to be remote homologous (i.e. structurally related). Similarly to homology modeling, threading or fold recognition methods build a model of an unknown fold according to the template of its (remote) homologue, but additional tasks must be solved. The first goal is to detect the existence and identity of a remote homologue or whether there is no known fold to which the input sequence belongs. This is a difficult task, given the large number of possible unrelated sequences with low sequence identity. If the first pass can be solved, the query sequence and the detected remote homologue have to be correctly aligned. These problems are usually solved together. Given a library of folds, which consists of a collection of structural templates derived from experimentally solved protein structures, for each fold in the library the input sequence is *thread* onto the known structure (sequence-structure alignment). It then follows an assessment of how well the query sequence fits each structural template using either amino acid structural propensities [FE96] or mean-force (statistical) potentials [Sip95, JTT94b]. To speed up the fold recognition phase, other methods have been proposed. A prediction-based threading technique is employed in [Ros95], where the alignment is made between the predicted strings of secondary structure and solvent accessibility and those extracted from the fold library. In [Jon99a], profile-based sequence alignments are used to align the query sequence and the sequence of the candidate template. Feed-forward neural networks are then used to score the structural similarity of the two proteins.

The SCOP database [HMBC97] is a database of structural classification of proteins. It contains proteins structures hierarchically grouped into domains, families, superfamilies and folds. Such database can be used to assess performance of remote homology detection algorithms, as proteins of different families but in the same superfamily are likely to be remote homologues. Therefore, a method able to detect a protein of a given family when trained on proteins of other families sharing its same superfamily is actually recognizing a remote homologue. This framework have been cast into a discriminative problem by Jaakkola et al. [JDH00], who paved the way for the use of kernel methods with excellent results. They employed state-of-art HMM methods [KBH98] to generate models of a given protein superfamily, and used them to train a Fisher kernel [JH98a, JH98b], which is a kernel designed to measure the similarity between the generative processes underlying two instances, thus allowing to combine generative and discriminative approaches into a single learning method. The algorithm is trained to discriminate between examples belonging to the given superfamily and examples belonging to all other superfamilies. A wide range of kernels have been developed to the same aim after their work. The spectrum kernel [LEN02] compares two sequences by counting the occurrences of all common substrings of size k. Later variants included allowing mismatches in the common substrings [LEWN03] as well as deletions [LKE04]. Weight matrix [HH91] or regular expression [SSB03] motif databases derived from multiple alignments have been employed in [LMS$^+$01, BHB03]: a sequence is mapped to the feature space of all the comparisons with a motif of the database, and the inner product is computed in such space. Most of these methods employ efficient data structures such as suffix trees [Ukk95] or tries in order to be computationally feasible. A natural way to represent an object belonging to a given set is by its similarity to other elements of the set. This idea is implemented in the kernel framework by the *empirical* feature map [Tsu99], where each example is mapped to the vector of the similarities with all other reference examples. Liao and Noble [LN03] employed pairwise sequence similarity scores obtained by the Smith-Waterman algorithm [SW81]. The feature map for a sequence is thus represented by a vector of pairwise sequence similarities with positive examples (from the superfamily to be modeled) and negative examples (from the other superfamilies), and the size of this vectorization set heavily affects the efficiency of the algorithm. For a detailed treatment of kernel methods for remote protein homology, as well as for other tasks in computational biology, see [Nob04].

Threading methods are very promising but their predictions of 3D structure are not yet reliable. In most of the cases, predictions have to be manually processed by experts to correct false positives. As previously said, they require the remote homologue to exist and to be detectable: assuming all remote homologous could be recognized, threading is in principle applicable for about another 10-20% of newly available sequences.

De novo methods

Overall, it is estimated that knowledge based methods can be applied only for about 20-50% of novel proteins. In the majority of cases, the structure of a novel protein must be assigned *ab initio*, not relying on the protein having a fold similar to a known one. De novo or ab initio approaches [BS01, BTR$^+$01] predict 3D structure using stochastic optimization algorithms. These procedures employ some energy function to search a globally optimal (minimal energy) configuration in the space of allowable structural conformations. Unfortunately, the problem is inherently intractable due to the exponential number of local minima. Exact numerical calculations are beyond the possibilities of present and near future computers and locally-optimal results can be obtained only for small proteins and coarse representations, e.g. using only Cα atoms of the backbone. Moreover, the cost function is often manually designed and its parameters are estimated from statistical considerations. Even if it is possible to find a global optima, this could be different from the native configuration, because the energy function contains some inaccuracies. However, it has been frequently observed that knowledge of topological constraints and other structural features (e.g. secondary structure) can greatly improve the efficiency[4] and performance of de novo approaches.

5.2 Prediction in one and two-dimensional space

As we have seen, the most effective methods for protein fold prediction are comparative modelling and threading techniques. These methods assume that an homologue can be detected, an event estimated to occur for less than half the sequences of newly available genomes. Reliable protein structure prediction from sequence using de novo methods is far from being achieved, mainly because a global search in the conformational space of proteins is inherently intractable. Only recently, this task has been aided by the development of intelligent tools borrowed from machine learning (ML) approaches. These methods extract relevant pieces of information from databases of known structures in order to focus the search on more tractable problems: instead of three-dimensional coordinates, their goal is to predict the values of intermediate and simplified one-dimensional (1D) or two-dimensional (2D) protein structural descriptions, i.e. those aspects of a protein that can be represented as a linear strings of symbols or two-dimensional matrices. Using the rich diversity of information in current biological archives it is possible to predict at some extent of success many structural features, such as:

1D
- Secondary structure;
- Solvent accessibility;
- Topology and topography of transmembrane proteins;
- Residue coordination numbers;
- Cysteines bonding state

2D
- Residues distance and contact maps;
- β-sheet partners;
- Disulphide connectivity patterns

[4] The ratio of number of sampled conformations to the number of random configurations necessary to find a good structure [FL01, RFS98].

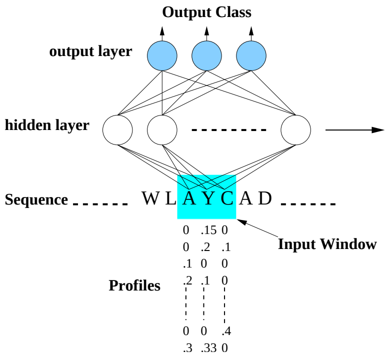


Figure 7: Basic template scheme based on feed-forward neural networks for predicting structural features from amino acid sequences.

There are many advantages in using these features for protein fold prediction, although they represent a partial description of the structure. Firstly, predictions can be obtained in a short amount of time and constitute important steps towards 3D determination. For instance, predicted secondary structure segments can be arranged in space as rigid bodies thus simplifying molecular dynamics simulations or allowing for a combinatorial assessment of a limited number of possibly stable globular folds [CP96]. Predicted features can be used to assist X-ray crystallography during the early stage of structure determination or in molecular biology experiments. Predicted solvent accessibility together with secondary structure is employed in prediction-based threading techniques to produce an alignment of the query sequence with the folds deposited in the library of known structures [Ros95]. Concerning de-novo protein structure prediction, one and two-dimensional structural descriptions represent additional information that can be conveniently exploited to reduce the number of sampled conformations and to guide the conformational search. Knowledge of topological constraints can also assist experimental work in laboratory during the early stage of structure determination and provide guidance for subsequent experiments.

In the context of machine learning, these predictive problems are modelled as learning (i.e. inferring) mappings from input to output spaces. The input space is the domain of the strings of amino acids. In the case of 1D predictions, instances of the output space are also given by sequences represented by strings of symbols. Each symbol represents the predicted structural property of the residue in the same position. 2D predictive systems can be seen as computing representations of relations between the input elements, hence they can be seen as learning a mapping from sequences to graphs. Depending on the problem, an edge in these graphs indicates the presence of some form of relation between the adjacent nodes, e.g. residues, secondary structure segments etc.

Training and prediction of many one-dimensional predictive systems rely on specific variations of a basic scheme, as depicted in Figure 7, where in this case the learner is represented by a feed-forward neural networks. Other algorithms, such as Support Vector Machines are applied in a similar fashion. The figure shows a common approach for solving sequential translation (i.e. predictive) problems in structural genomics: the learner is moved along the input sequence and outputs at each step a vector that encodes the structural class of the residue in current position (Y or Tyrosine in figure). It is usually assumed that structural properties of a residue are affected by its local context (i.e. the neighboring residues). For this reason, the input vector is usually formed by taking a window centered at a given position and replacing each residue in the window with an encoding formed by a fixed size numerical vector. Concerning the connectionist realization of the scheme of Figure 7, the set of adjustable architectural parameters includes the number of output units, the number of hidden layers and units, input encoding and window size. Clearly, the number of output units depends on the number of classes that we decide to model for the specific prediction task. A kernel-based realization of the previous translation task requires to choose among different architectural details, such as the type of machine (e.g. binary or multi-class classifier), the type of kernel (e.g. linear, polynomial or Gaussian) with its corresponding hyper-parameters (e.g. the degree of the polynomial or the Gaussian width) and the regularization parameter. Concerning the input, there are two main types of representation for a position in the sequence: one-hot and profile-based. Suppose sequence position i contains the j-th amino acid, $j = 1, \dots, 20$. Then, in one-hot encoding the input at position i is represented by a binary vector with the j-th component equal to 1 and 0 elsewhere. Instead of a binary vector, profile-based encoding uses the 20 dimensional vector extracted from the PSSM of a multiple alignment (see section 3). The size w of the input window ($w = 3$ in figure) controls the amount of contextual information that is considered for local predictions. Ideally, one could expect that the larger is w, the more is the information given to the predictor, hence performance should necessarily increase. Unfortunately, it is frequently observed that increasing w beyond some limit corresponds to a decrease of the signal to noise ratio. In other words, the amount of (distant) information is negligible with respect to the noise that is introduced. A typical size of the input window goes from 9 to 25 residues.

Secondary structure

Secondary structure prediction was the first important structural biology problem solved by machine learning techniques. Its significance can be understood by observing the variety of prediction systems that were developed over the last three decades. We can roughly distinguish between first, second and third generation methods. First generation methods were those making predictions using only single residue information. The input for each residue was represented either in the form of its statistical propensity to appear in an α-helix, β-strand and coil region [GOR78], or in the form of explicit biological expert rules [Lim74]. Average accuracy of these methods (known as Q_3 index) was not higher than 55%. Second generation methods were those applying the connectionist architecture described in Figure 7. Local interactions were taken into account by means of a sliding window and one-hot encoding. Values in the output layer discriminate each residue as belonging to one of the three states: α-helix (H), β-strand (E) and coil (C). The first original work [QS88] reported an accuracy of 62.7%. In [RK96], techniques were used to reduce overfitting and to incorporate prior knowledge, improving accuracy to 66.3%. One of the major difficulties of these methods is the correct location of β-strands, because they are predominantly determined by long-ranged interactions[5]. By this, it is generally assumed that only ≈65% of secondary structure depends on local interactions.

Last generation methods started exploiting evolutionary information. The basic observation is that the secondary structure within a family of evolutionary related proteins is more conserved than primary structure. In other words, the evolutionary pressure to conserve function has favored mutations that preserve relevant structural characteristics. This information is first processed doing a multiple alignment between a set of similar sequences and extracting a matrix of profiles (PSSM). Each matrix column represents the input given to the network for the corresponding sequence position. PHD [Ros96] was the first method to incorporate profile-based inputs and achieved an accuracy above 70%. The system is composed of cascading networks: the first one (sequence-structure) is similar to that of Figure 7; a second network (structure-structure) takes as input a window sliding on the previous outputs and refines the output probabilities of the first network. A final stage takes a jury decision by averaging the outputs from independently trained models. Other well-known profile-based methods are PSI-PRED [Jon99b], that uses two neural networks to analyze the profiles generated from a PSI-BLAST search [AMS$^+$97], JNet [CB00] and SecPred. An alternative adaptive model is employed in [BBF$^+$99] to realize bi-directional recurrent neural networks, a non-causal connectionist architecture that exploits contextual knowledge represented by upstream and downstream dependencies (long-range information) stored into hidden state variables. At present, almost all connectionist-based methods achieve performance levels in the range 76-78%.

There are other predictors of secondary structure that are not strictly based on neural network implementations. A nearest-neighbor approach is

[5] A residue in a β-strand has an hydrogen-bonding partner in some other strand that can be distant in sequence.

employed by NNSP [SS95], where the secondary structure state of a test residue is assigned by scoring the information coming from different templates according to their similarities. Template segments are those of proteins with known 3D structure. The web-server JPred [C$^+$98] integrates six different structure prediction methods and returns a consensus based on the majority rule. The program DSC [KS96] combines several explicit parameters in order to get a meaningful prediction. It runs the GOR3 algorithm (an evolution of GOR1 based on information theory applied to local interactions) on each sequence, to provide mean potentials for the three states. A linear combination of the different attributes gives an output which is subsequently filtered. The program PREDATOR [FA97] is based on calculated propensities of the 400 amino acid pairs to interact inside an α-helix or one upon three types of β-bridges. It then incorporates non-local interaction statistics and propensities for α-helix, β-strand and coil derived from a nearest-neighbor approach. In order to use information of homologous proteins, PREDATOR relies on local pairwise alignments. Accuracy is claimed to be 75%. In principle, Hidden Markov Models could be effectively used for prediction of secondary structure, thus allowing for the incorporation of syntactic restraints on the form of the output strings. At present, no HMM based method is able to outperform neural networks. Not surprisingly, the literature reports an improvement on the distribution of the length of predicted segments (SOV, or segment overlap), but not of the Q_3 accuracy measure.

Kernel Machines have entered quite recently [HS01] the arena of secondary structure prediction, and the necessary multiclass extension was either realized with combinations of binary classifiers [HS01, WMBJ03, CFPV03a] or multiclass Support Vector Machines (MSVM) [NR03, GPE$^+$04]. No clear evidence emerged in favour of SVM compared to Neural Networks, taking into account the much higher computational time required. This can be partially explained by the fact the both are *local* classifiers fed with the same inputs, and the great amount of data available make the usual SVM advantages less evident. However, a few recent works [NR03, GPE$^+$04] underlined the effectiveness of multiclass SVM as a filtering stage, to be fed with the output of other predictive methods (which can be MSVM themselves). A simple and effective refinement stage was proposed in [CFPV03a] in order to remove *inconsistencies* of predicted sequences, that is violations of the constraints that can be imposed from the distribution of observed consecutive secondary structure labels. The method builds a finite state automata (FSA) representing all allowed sequences, and turns a predicted labelling sequence into the maximum likelihood sequence given the grammar and the predictions.

All the SVM approaches proposed so far employ standard kernels, such as polynomial or Gaussian ones. The first attempt to develop a kernel especially modeled for secondary structure prediction was presented in [GLV04].

Firstly, it employs a dot product between residues (or profiles) mediated by a substitution matrix, which compares residues according to their biochemical similarity. The substitution matrix was derived from [LRG86] and is especially designed for secondary structure prediction tasks. Secondly, it introduces an adaptive weighting of the window around the target residue, with weights learned by a version of kernel target alignment [CSTEK02] extended to the multiclass case [Ver02]. The proposed kernel was proved superior to a standard MLPs, but given its computational overhead, its usage as a module in big architectures, such as those currently employed for secondary structure prediction [Jon99c, PPRB02, PLN$^+$00], is not straightforward.

Topology and topography of membrane proteins

Membrane proteins represent a biologically important class of biomolecules. Proteins classified into this class are those crossing the outer membrane of cells, mitochondria and chloroplasts (the latter two being the "energetic devices" of eukaryote cells). Membranes play a key role in cell metabolism: they act as regulatory interfaces between inside and outside physico-chemical activities; this role is assured through membrane proteins which form channels for signal and material exchanges.

In spite of their importance, membrane proteins are more difficult to crystallize than globular proteins[6] because of their lipid environment. Moreover, they are hardly tractable by NMR spectroscopy. For this reason, structural predictions are even more needed for this class. In this case, the landscape of prediction can be partitioned according to: 1. type of membrane proteins; 2. prediction task; 3. prediction methodology. At present, only a few structures are known for two types of membrane proteins: α-helical and β-barrel. Belong to the first type those proteins crossing cytoplasmic membranes with α-helices. The second type groups proteins which cross the outer membrane of bacteria with β-strands organized into a sort of cask; there are findings supporting their presence in mitochondria and chloroplasts. Geometrical and physical restraints of membranes allow for the definition of two basic structural problems: prediction of topology and topography. The topology of a membrane protein is the location (inside or outside) of the N- and C-terminus[7] with respect to the membrane bilayer. The topography is the location along the protein sequence of transmembrane segments. Knowledge of topology and topography of membrane proteins is important because their structure and function is closely related to the number and location of membrane spanning segments. For instance, the greater is the number of transmembrane segments, the larger is the channel width for ac-

[6] Hartmut Michel won the Nobel prize in 1987 for the first successful attempt in membrane protein crystallization.

[7] Sequence fragments respectively at the beginning and end of sequence.

tive ion transport. Another problem is of great practical importance: to identify if a protein belongs to the membrane class.

Concerning methodologies, we can still distinguish among first, second and third generation methods. There are systems based on statistical approaches, neural networks and probabilistic graphical models like HMMs. Here we focus on techniques which use sequence profiles. For the prediction of protein topography the outputs discriminate whether a residue is in a transmembrane (TM) or non-transmembrane (NTM) segment. Several neural network-based predictors are available for transmembrane helices: PHDhtm and MEMSAT-2 rank among the most effective ones. PHDhtm follows the basic template scheme of Figure 7 and corrects the length of membrane helices with the introduction of cut-off filters. By using sequence profiles it achieves 95% accuracy in topography prediction [RaPFS95]. It is able to predict protein topology at 86% accuracy [RCF96] using dynamic programming procedures to post-process network outputs. MEMSAT-2 is the profile-based version of MEMSAT [JTT94a] and achieves a success rate of 93%. Significant results can also be obtained with HMMs [KLvHS01, SvHK98]. In this case, topography is predicted with the computation of the optimal Viterbi path. Each state in the probabilistic transition automata is associated either to NTM or TM state of a secondary structure segment.[SvHK98] and HMMTOP.

Prediction of TM β-barrel strands is a more complex task. As previously noted, this depends on the difficulty related to prediction of β-strands and in particular those of membrane proteins. The available methods are based on statistical, neural network and Hidden Markov models. The neural network used in [JFPC01] predicts strands in TM state filtering spurious assignments with dynamic programming constrained optimization and achieves 78% accuracy. Similarly to the methods discussed in [JFPC01, JTT94a], in [MFKC02] an algorithm based on dynamic programming uses HMM outputs to locate the transmembrane β-strands along the protein sequence by model optimization. The HMM is trained with information derived from multiple sequence alignments and can be used to discriminate outer membrane proteins from other protein types. The overall accuracy per residue is as high as 83%.

Solvent accessibility and coordination number

Residue solvent accessibility (RAcc) is defined as the relative degree to which an amino acid interacts with molecules of the solvent environment. Solvent accessibility can be used to assist prediction of functional sites and subcellular localization. Moreover, it can help threading techniques during the structural alignment of the input string [Ros95]. For this problem, residues are normally classified in two classes: *buried* (RAcc $< 16\%$) or *exposed* (RAcc $\geq 16\%$).

Prediction of relative solvent accessibility has been solved using neural networks with single-sequence input [HMK90] or evolutionary information [RS94], Bayesian techniques [TG96] and simple statistical methods using residue substitution matrices [PPBA98] and residue propensity of exposition [RB99]. The latter and simplest approach achieved accuracy levels (69-71%) comparable to those of other more sophisticated methods that use single-sequence information. Neural networks and Bayesian analysis were able to achieve an accuracy of 75%, showing again the wealth of evolutionary information and the ability of non-linear models to exploit the features provided by this information. Support Vector Machines have been recently applied [YBM02, KP04] with both single sequence and profile based inputs as well as feature weighting schemes, claiming an accuracy as high as 80%.

A related problem involves prediction of residue coordination number, defined as the number of contacts that a residue has in the folded protein. The number of contacts corresponds to the number of spatial neighbors. It is usually assumed that solvent accessibility can be used to predict coordination numbers (less exposed residues tend to belong to the hydrophobic compact core, and have a large number of neighbouring residues). However the two measures show different distributions [FC00], and individual predictors have been often developed. The method in [FC00] is based on the profile-based neural network of Figure 7 and correctly predicts in 69% of the cases whether a residue has a number of contacts lower or higher than its average distribution value, as measured a non-redundant selection of protein structures. State-of-art performances were obtained with ensembles of BRNN [PBFC02] similar to those employed for secondary structure prediction (see relative paragraph).

Contact maps

As previously discussed, learning algorithms can be trained to predict topological features that are both translation and rotation invariant and that constrain the space of structural conformations. Such features are based on intermediate simplified structural representations, such as the distance matrix (DM). The Distance Matrix flattens the set of atom coordinates to a symmetric square matrix where the element in position (i, j) represents the distance among atoms in position i and j. An interesting property of DMs is the independence from the coordinates frame. Unfortunately, predicting DMs is known to be very difficult and no effective method is available. For this reason, a reduced representation is typically considered, known as the (fine-grained) residue contact map (CM). The contact map of a protein with N amino acids is defined as a symmetric $N \times N$ matrix C, with the element C_{ij} defined as:

$$C_{ij} = \begin{cases} 1 & \text{if amino acid i and j are in contact} \\ 0 & \text{otherwise} \end{cases}$$

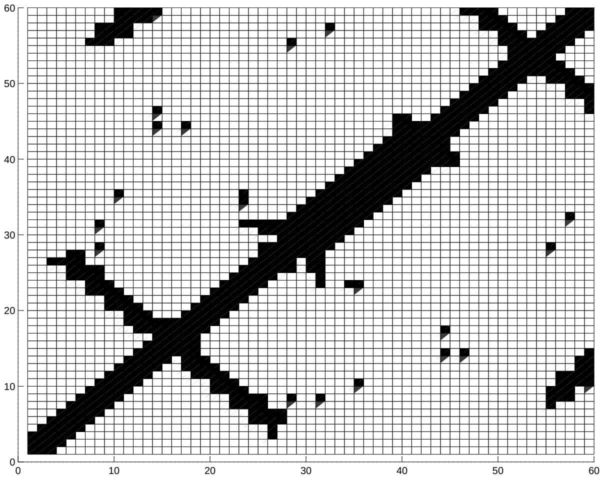


Figure 8: Fine-grained contact map of streptococcal protein g (pdb code: 2igd).

The notion of contact is closely related to a spatial proximity relation and many definitions are possible. Typically, two amino acids are said to be in contact if their distance is below a given threshold (in Å). Commonly used distance definitions are between C_α-C_α (7-9 Å), C_β-C_β (6.5-8 Å) atoms, or the minimal distance between two atoms belonging to the side-chain or backbone of the two amino acids (4.5 Å). Figure 8 is an example of binary contact map defined at residue level. In contact maps, α-helices appear as thick bands of contacts along the diagonal (residues 28-41) and β-sheets as bands parallel or orthogonal to the diagonal (residues 18-25 and 56-60).

The appeal of the contact map representation in the context of folding is essentially due to the possibility of reconstructing 3D atom coordinates from contact maps. Methods for reconstructing 3D coordinates from contact/distance maps have been developed in the NMR literature and elsewhere using distance geometry and stochastic optimization algorithms. Vendruscolo [VKD97] has shown that even partial or noisy knowledge of the CM may be sufficient for recovering the real structure. In the case of NMR spectroscopy, information about residue-residue contacts can be inferred from experimentally derived 2D NOESY peaks. On the other hand, fold prediction methods cannot rely on experimental data and contact maps need to be predicted.

Algorithms have been developed for the prediction of protein distance constraints and contact maps [AGT95, AGT95, GLAB99], in particular using neural networks [FC99, FOVC01, PB02], and Hidden Markov Models [ZJB00, SB03]. An important feature of contact map prediction is the concept of correlated mutations [SKS94, PHCAV97, OV97]. Correlated mutations attempt to model long-ranged residue interactions using constraints on the type of amino acids that can be substituted without the removal of contacts. At the CASP4 competition [LCH01], one of the best method reported a precision of $\approx 21\%$ correct predictions of long-range contacts (i.e. restricting the analysis to pairs whose sequence distance is more than four residues) using neural networks and correlated mutations [FC99, FOVC01]. Correlated mutations are also used in [SVB02] to derive contact likelihood scores for all possible amino acid pairs and then to use these scores to predict contacts. The method in [ZJB00] uses Hidden Markov Models to extract folding initiation sites and to model interactions between those sites. It then applies an A-priori like algorithm to mine association rules between input patterns and contacts. Reported precision is comparable to that of the former method. Since typical predicted contact maps are ambiguous or physically impossible, a variation of this method is applied in [SB03]. In this case, for a given protein sequence an HMM computes the inter-residue contact potentials that are used to align the target to templates. The resulting map is then filtered using a folding pathway method. Another approach is described in [PB02] and is based on a two-dimensional generalization of bidirectional recurrent neural networks [BBF$^+$99]. The method is tested at different distance cutoffs: reported levels of precision are above 50% for the complete map, i.e. not restricting the analysis to pairs that are distant more than a given number of residues.

Contact map representations can be derived not only at the level of amino acids, but also considering contacts between secondary structure segments. Such representations are called coarse-grained contact maps. Contacts in coarse maps are associated with a spatial neighborhood relation on the set of secondary structure segments of a given protein. A coarse map can be strongly informative about the shape of the unknown fold. Hence, the information it provides could be conveniently exploited by de novo 3D reconstruction procedures because it represents additional sets of constraints. Of equal importance is the fact that prediction of residue contact maps is far from being successful. The main advantage of working at the coarse resolution is to obtain a significant dimensionality reduction allowing better but more costly algorithms to be employed[8]. In this perspective, first attempts for predicting protein coarse maps are presented in [VF03, PBVF02] and [BP03]. One of these methods looks at a contact map as an adjacency matrix of an undirected graph whose vertices are secondary structure segments and the edge set is the contact relation defined on pairs of segments. Graphical representations of candidate contact maps are then processed (i.e. scored) by means of a connectionist methods that can deal with structured data. The

[8] An analysis of the Protein Data Bank archive (PDB) [B$^+$77] reveals that the average segment length is about seven residues. Therefore, coarse contact maps are roughly 2% the size of the corresponding residue contact maps.

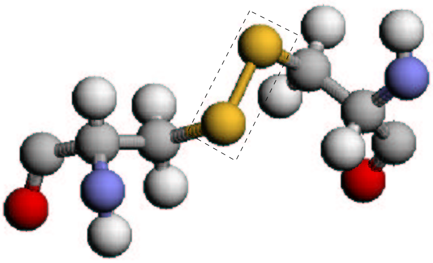


Figure 9: Ball-&-stick representation of a disulphide bridge: the bridge is drawn as a cylinder connecting the sulphur atoms of two oxidized cysteines.

network is trained to score candidate graphs according to a similarity metric with respect to the correct pattern. During prediction, the score computed by the network is used to greedily explore the space of candidate graphs. A second method [BP03] uses the more direct approach introduced in [PB02] to predict inter-residue contact maps and shows comparable performance with respect to the alternative method.

Cysteine bonding state and disulphide connectivity

Cysteine (Cys) is one of the twenty amino acids that constitute proteins (see Table 1) and it owns a unique feature. Proteins whose sequence contains cysteine residues are subject to post-translational covalent modifications and cysteines can occur either in *oxidized* or *reduced* (thiol) form. Two oxidized cysteines uniquely pair to form a covalent bond, known as *disulphide bridge* (see fig. 9), whose formation can be described by the following reaction:

$$2\mathrm{CH_2SH} + 1/2\mathrm{O_2} \rightleftharpoons \mathrm{CH_2SSCH_2} + \mathrm{H_2O}$$

Such reactions require an oxidative environment in order to occur. They involve complex pathways in which particular enzymes (oxidoreductases) catalyze oxidation, reduction and isomerization reactions. The required environmental conditions depend on the ratio of the oxidized to reduced form of such enzymes which in turn depends on the sub-cellular localization of the protein. For instance, cytoplasmic and nuclear proteins usually do not have disulphide bridges. The relative rarity of disulphides in cytoplasmic proteins appears to be dependent upon a disulphide-destruction machine [RB01]. Two pivotal cogs in this machine are the thioredoxin and glutathione reductases. The formation of disulphide bridges normally takes place in the periplasm of prokaryotes and in the endoplasmic reticulum (ER) of higher eukaryotes [FG03]. Dedicated enzymatic systems that catalyze the formation of disulphides have been discovered in the periplasm of prokaryotes. Disulphide bond formation in Escherichia coli is catalyzed by at least

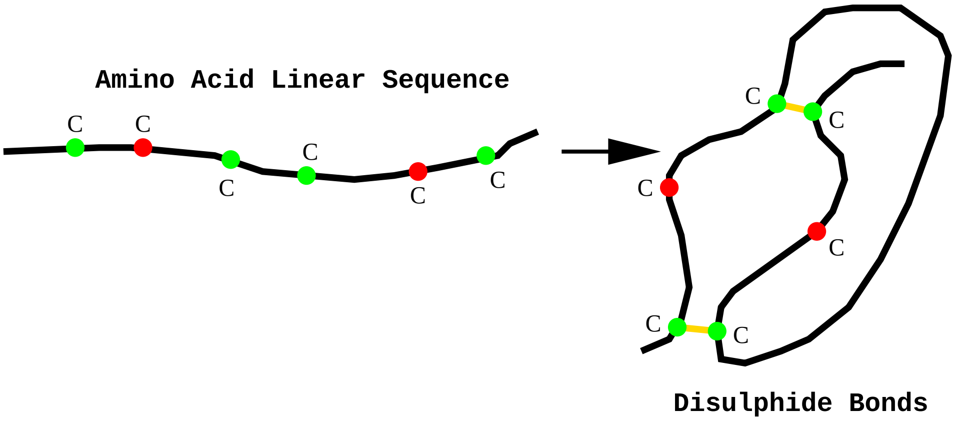


Figure 10: The topological arrangement of the linear chain can be strongly influenced by the location of disulphide bridges. In the example, four cysteines (in green) are oxidized and connect each other to form two intra-chain disulphide bonds.

three 'Dsb' (Disulphide Bond formation facilitator) proteins: Dsb(A-B-C) [Bar94]. The lumen of the ER is a compartment specialized for protein folding; proteins destined for secretion, or for compartments accessed via the secretory pathway, enter the ER lumen unfolded, but only exit when correctly folded and assembled. Protein folding in this context is often associated with the formation of native disulphide bonds, and this is facilitated by the enzyme protein disulphide isomerase (PDI) [BT99, KRDF98].

Disulphides are often vital for the folding and stability of proteins. The incidence that disulphide bridges have on protein structure has been recognized at several levels. Simulations [AS00], experiments in protein engineering [M$^+$89, CF93, KWTR00] and theoretical studies [Bet93, DS95, WWNS00] indicate the importance of a disulphide bond in stabilizing the native state of proteins. Evolutionary models [Dem00] postulate the selective advantage that the introduction of disulphide bonds confers to proteins with unstable folded states.

The stabilizing role of disulphide bonds derives from a reduction of the number of conformational states, thus of the entropic cost of folding a polypeptide chain into its native state [HS94a, WWNS00]. Not only disulphide bridges constitute an important factor in the energetics of folding dynamics. Depending on their number and location, these covalent links can also contribute to catalytic activities of biomolecules [KWTR00]. More importantly, they may connect very distant portions of the sequence, thus representing a set of strong structural constraints in the form of long-range interactions (see Figure 10).

Knowledge of the correct location of native disulphide bridge (S-S) topology is of considerable importance for several important reasons:

- Disulphide-rich protein families, such as EGF-like, defensin-like and plant protease inhibitors, represent a large number of proteins with a biological relevance (growth factors, hormones, toxins etc.). The correct disulphide bridge topology is indispensable for their final structure and function [MAMR$^+$98, Bet93].

- Knowledge of (S-S) topology may assist and speed-up experimental work in laboratory during the early stage of structure determination and provide guidance for subsequent experiments. For instance, structure determination by NMR spectroscopy is an iterative process in which previous structures are used to correct and complete NOE assignments. The information provided by S-S connectivity can improve the quality of initial structures and allows a reliable assignment of ambiguous NOEs [BBSM00].

- Small disulphide-rich (SDFs) protein folds represent a class of proteins having in general mostly irregular secondary structure content [HS96]. In spite of their current success, predictors of secondary structure cannot reliably infer structural regularities for SDFs. For this reason, predicting 3D structure for SDF proteins using fold-recognition (threading) approaches is likely to be difficult, as these algorithms usually exploit the information provided by (predicted) secondary structure.

- It has been observed [CCY$^+$03] that knowledge of disulphide bonds can be used to detect remote (structural) homologues for a given chain. Therefore, knowledge of S-S topology enlarges the range of applicability of threading approaches, because it can be used to discriminate structure similarity.

- Topological constraints provided by S-S connectivity enhance the performance of de novo 3D reconstruction algorithms. This information has been observed to improve the quality of predicted structure and to dramatically decrease the sampling rate of the conformational space [FL01, HSP99, LHBB96].

- Correct Cys-Cys pairing is of importance for determining and exploiting the folds which can be used as scaffolds for medical, pharmaceutical and agronomical applications [DKG$^+$99, VVD$^+$98, OFA$^+$98, FSO$^+$97]. The stability of these folds is largely dependent on the formation of disulphide bonds.

It is therefore clear how knowledge of disulphide bridges could give considerable help to virtually all methods for protein 3D structure prediction (threading or fold recognition, de novo algorithms). Yet, disulphide bonds can generate insights into the structure-function relation of many proteins of interest.

Finally, recent researches enlightened the role of cysteines in determining both prokaryotes and eukaryotes response to oxidative stress (see [LJ03] for a review), which is a major factor of ageing [FH00] as well as of various diseases including cancer [KJ01].

Disulphide bridges prediction can be divided in two steps. Firstly, the bonding state of each cysteine in a given sequence is predicted, as either *reduced* or *oxidized*, the latter meaning that it is involved in a disulphide bridge. Secondly, the connectivity pattern between oxidized cysteines is predicted, pairing each bonded cysteine with its correct partner.

Actually not all bonded cysteines form disulphide bridges, and many other post-translational modifications have been observed or supposed for cysteines [GWG$^+$03], the most important being the binding of ligands, that is the formation of bonds between cysteines and various ligands typically containing metal groups [KH04], such as heme groups and iron-sulfur clusters. While in the following paragraphs we will focus on disulphide bridges prediction, an attempt to discriminate between ligand bound and disulphide bound cysteines was recently described in [PF04].

Cysteine Bondind State Prediction The first step in predicting disulphide bridges in a given protein is that of identifying oxidized cysteines which are involved in disulphide bridge formation. This can be cast to a binary classification task, that is for each cysteine in a given protein, predict whether it is involved in a disulphide bridge or not. This is an active research field, and many different learning algorithms have been developed so far.

The program CYSPRED developed by Fariselli et al. [FRC99] (accessible at `http://gpcr.biocomp.unibo.it/predictors/cyspred/`), uses a neural network with no hidden units, fed by a window of $2k + 1$ residues centered around the target cysteine. Each element of the window is a vector of 20 components (one for each amino acid) obtained from multiple alignment profiles. This method achieved 79% accuracy (correct assignment of the bonding state) measured by 20-fold cross validation and using a non-redundant set of 640 high quality proteins from PDB Select [HS94b] of October 1997. Accuracy was boosted to 81% using a jury of six networks. Still, the bonding state of each cysteine is assigned independently.

Fiser & Simon [FS00] later proposed an improvement based on the observation that cysteines and half cystines[9] rarely co-occur in the same protein. In their algorithm, if a larger fraction of cysteines are classified as belonging to one class, then all the remaining cysteines are predicted in the same state. The accuracy of this method is as high as 82%, measured by a jack-knife pro-

[9] A cystine is the dimer formed by a pair of disulphide-bonded cysteines.

cedure (leave-one-out applied at the level of proteins) on a set of 81 protein alignments. This result suggests that a good method for classifying proteins in two classes is also a good method for predicting the bonding state of each cysteine, even though in this way the accuracy for proteins containing both cysteines and half cystines is sacrificed. The program, called CYSREDOX, is accessible at `http://pipe.rockefeller.edu/cysredox/cysredox.html`.

Later, Mucchielli-Giorgi et al. [MGHT02] have proposed a predictor that exploits both local context (a window centered around the target cysteine) and global protein descriptors. Interestingly, they found that in absence of evolutionary information, prediction of covalent state based on global descriptors was more accurate (77.7%) than prediction based on local descriptors alone (67.3%). Their best predictor, based on a multiple classifier reaches almost 84% accuracy measured by 5-fold cross-validation on a set of 559 proteins from Culled PDB.

A different approach was developed in [FPV02, CFPV03b] for exploiting the key fact that cysteines and half cystines rarely co-occur. Prediction in this case is achieved by using two cascaded classifiers. The first classifier predicts the type of protein based on the whole sequence. Classes in this case are “all”, “none”, or “mix”, depending whether all, none, or some of the cysteines in the protein are involved in disulphide bridges. The second binary classifier is then trained to selectively predict the state of cysteines for proteins assigned to class “mix”, using as input a local window with multiple alignment profiles. The method achieves an accuracy of 85% as measured by 5-fold cross validation, on a set of 716 proteins from the September 2001 PDB Select dataset [CFPV03b].

Shortly after, Malaguti et al. [MFMC02] have proposed yet another approach where the disulphide bonding state is predicted as in CYSPRED but predictions are then refined using a Hidden Markov Model [Rab89] trained to recognize the stochastic language that describes the alternate presence of bonding and non-bonding cysteines along the sequence. This improved method achieved the performance level of 88% correct prediction measured by 20-fold cross validation on a non redundant dataset.

Similar results were obtained by adding to the architecture described in [CFPV03b] a global refinement stage [CFPV04] either by bi-directional recurrent neural networks (BRNN) or by the HMM employed in [MFMC02]. In the case of BRNN refinement, a final stage with a finite state automata (FSA) was also employed to force consistent predictions (even number of bonded cysteines in a given chain, interchain bonds not predicted), reaching the best performances to date. The program is accessible at `http://neural.dsi.unifi.it/cysteines` and can be used in combination with the disulphide connectivity predictor.

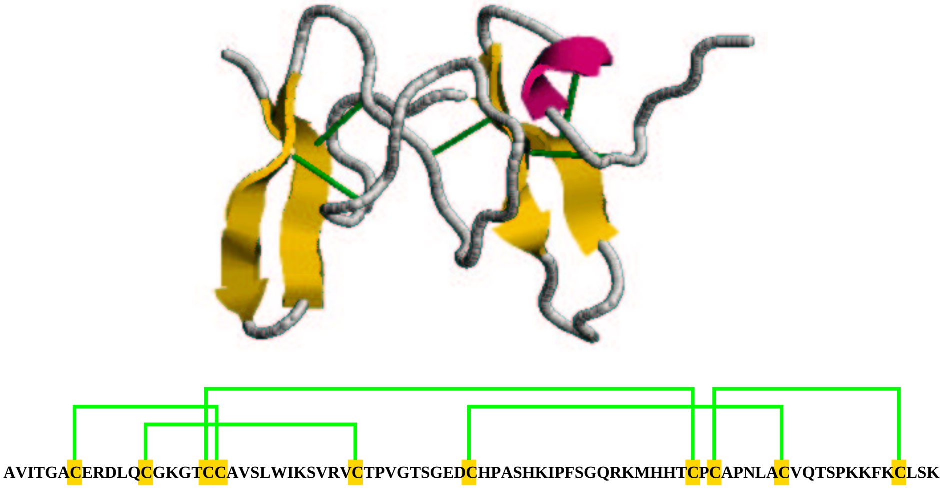

AVITGACERDLQCGKGTCCAVSLWIKSVRVCTPVGTSGEDCHPASHKIPFSGQRKMHHTCPCAPNLACVQTSPKKFKCLSK

Figure 11: 3D structure (top) and schematic picture of disulphide connectivity (bottom) for the 81 AAs chain of Intestinal Toxin 1 (PDB code: 1imt).

Overview of disulphide connectivity prediction Similarly to the more general protein folding problem, determination of disulphide connectivity can be approached with experimental or algorithmic methodologies. Reliable biochemical techniques have been developed, but they are characterized by high degree of complexity and need a significant amount of experimental time [BBSM00, Gra93]

Existing algorithmic (i.e. predictive) approaches provide significant speed gains at the cost of less reliable predictions. Straightforward methods based on the alignment of the protein under study with homologous proteins can be used to detect the location of disulphide bridges. Unfortunately, they require sufficiently high levels of sequence similarity, an event estimated to occur for less than half the sequences from newly available genomes.

Usually, the location of disulphide bridges is predicted starting from knowledge of bonded cysteines. Differently from the problem of predicting the bonding state, this task has received relatively scarce attention in the research community, mainly because of the combinatorial explosion of the problem as the number of bonds increases. Figure 11 shows an example of a protein structure with five disulphide bridges. In this case, a prediction algorithm should be able to discriminate the correct topology among a total of 945 possible alternatives.

Existing predictive approaches make use of stochastic global optimization [FC01, BBSM00], combinatorial optimization [KF03] and hybrid machine learning techniques [FMC02, VF04]. The method in [FC01] represents the set of potential disulphide bridges in a sequence as a complete weighted undirected graph. Vertices are oxidized cysteines and edges are labeled by the strength of interaction (contact potential) in the associated pair of cysteines. First, simulated annealing is used to find an optimal set of weights. After a complete labeled graph is obtained, candidate bridges are located by finding the maximum weight perfect matching[10]. The problem can be solved in polynomial time using linear programming, but the computation of contact potentials is computationally demanding. In a subsequent improvement [FMC02], neural networks were used to learn the contact potentials and then for labeling edges as above, increasing the predictive accuracy and concomitantly reducing training time. This method achieves satisfactory results for the simplest but not trivial case of four oxidized cysteines, assuming perfect knowledge of the set of bonded cysteines.

Another alternative method is described in [KF03]. Finding the location of disulphide bridges is part of a more general protocol aimed at predicting the topology of β-sheets in proteins. The approach assumes hydrophobic rather than hydrogen interactions as the main driving force of β-sheet formation. Residue-to-residue contacts (including Cys-Cys bridges) are predicted solving a series of integer liner programming problems in which customized hydrophobic contact energies must be maximized. Model constraints define allowable sheets and disulphide connectivity configurations. The interesting part of the proposed framework relies on its ability to predict cysteine-cysteine contacts independently from the knowledge of disulphide bonding state of cysteines. Unfortunately, performance of this method is not consistently assessed. The authors report validation results only for two relatively short polypeptides with a small number of bonds (2 and 3). Therefore, it is not possible to compare this method with our and other approaches.

In the approach proposed in [BBSM00], disulphide bridges are predicted with an iterative simulated annealing protocol. The applied reconstruction algorithm uses distance restraints derived from NMR spectroscopy and between any Cys-S$_\gamma$ atom and all other Cys-S$_\gamma$ atoms. This method shares advantages and drawbacks of the approach in [KF03]: knowledge of the disulphide bonding state of cysteines is not required, but assessment of the procedures is reported only for six small polypeptides. Furthermore, the method requires experimental NMR data and the success of the procedure can be strongly affected by the amount and quality of these data.

An alternative and more recent method is proposed in [VF04]. The core of the method is the use of Recursive Neural Networks (RNNs) [FGS98], a class of connectionist models that allows to solve classification and regression tasks on structured data. The model employed in [VF04] is adapted to process arbitrary undirected structured data, like the graphs representing disulphide connectivity patterns. The methodology was first introduced

[10] A perfect matching of a graph (V, E) is a subset $E' \subseteq E$ such that each vertex $v \in V$ is met by only one vertex.

in [VF03] and proved to be effective for the prediction of protein coarse contact maps. It can be seen as a generalization of bi-directional recurrent neural networks for dealing with graphs instead of merely sequential data. The network is trained to score candidate graphs according to a similarity metric with respect to the correct pattern. Vertices of the graphs are labeled by fixed-size vectors of multiple alignment profiles in the local half-cystine environments. During prediction, the score computed by the network is used to exhaustively explore the space of candidate graphs. This approach is tested on the same experimental data as used in [FC01, FMC02] and achieves state-of-the-art results. Similarly to [FC01, FMC02], the complexity of the search procedure prevent the application of the algorithm to chains with more than ten oxidized cysteines. In spite of being limited by the number of predictable disulphides, the method can easily incorporate and effectively exploit evolutionary information and it is shown how it can reliably deal with a broad spectrum of sequences for the disulphide bridge prediction problem.

6 Conclusions

The rapid progress of biological sequencing projects impose the need to assign native 3D conformation to novel proteins both reliably and quickly. X-ray crystallography and NMR spectroscopy are reliable methods, but finding structures in laboratory is extremely difficult, cost prohibitive and can take months in some cases. For these reasons, at the beginning of 2004 the ratio of known 3D conformations to known sequences is approaching 1:50. In the absence of comprehensive theories of folding and high-throughput structure determination projects, prediction tools play an increasingly significant role. Knowledge-based methods (homology modelling and threading) speed up structural assignments and can be very reliable. However, they can only be applied in the presence of detectable sequence or structural similarity, an event estimated to occur for less than half the sequences from newly available genomes. A large fraction of novel proteins can only be processed with de novo methods which are computationally demanding and are still far from being successful. In this scenario, another class of methods mainly based on Machine Learning receives increasing attention. The goal of these methodologies is to predict simplified descriptions of protein 3D conformations. Besides their limited computational requirements, statistical learning methods can take advantage of the huge amount of data deposited in public databases to produce accurate models of important structural characteristics of proteins. Exploiting all these information represents an attractive and promising way for bridging the sequence-structure gap.

References

[AB97] Abagyan and Batalov. Do aligned sequences share the same fold? *J. Mol. Biol.*, 273:355–368, 1997.

[AGT95] A. Aszodi, M.J. Gradwell, and W.R. Taylor. Global fold determination from a small number of distance restraints. *J. Mol. Biol.*, 251:308–326, 1995.

[AMS$^+$97] S.F. Altschul, T.L. Madden, A.A. Schaffer, A.A. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleid Acids Res.*, 25:3389–3402, 1997.

[Anf73] C. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[Aro50] N. Aronszajn. Theory of reproducing kernels. *Trans. Amer. Math. Soc.*, 686:337–404, 1950.

[AS00] V.I. Abkevich and E.I. Shankhnovich. What can disulfide bonds tell us about protein energetics, function and folding: simulations and bioinformatics analysis. *J. Math. Biol.*, 300:975–985, 2000.

[B$^+$77] F. Bernstein et al. The Protein Data Bank: a computer based archival file for macromolecular structures. *Journal of Molecular Biology*, 112:535–542, 1977.

[BA00] A. Bairoch and R. Apweiler. The SWISS-PROT protein sequence database and its supplement TrEMBL. *Nucleic Acids Res.*, 28:45–48, 2000.

[Bar94] J.C. Bardwell. Building bridges: disulphide bond formation in the cell. *Mol Microbiol.*, 14(2):199–205, 1994.

[BB01] P. Baldi and S. Brunak. *Bioinformatics. The Machine Learning Approach.* The MIT Press, 2001.

[BBF$^+$99] P. Baldi, S. Brunak, P. Frasconi, G. Soda, and G. Pollastri. Exploiting the past and the future in protein secondary structure predicton. *Bioinformatics*, 15:937–946, 1999.

[BBSM00] J. Boisbouvier, M. Blackledge, A. Sollier, and D. Marion. Simultaneous determination of disulphide bridge topology and three-dimensional structure using ambiguous intersulphur distance restraints: possibilities and limitations. *Journal of Biomolecular NMR*, 16:197–208, 2000.

[BCR84] C. Berg, J.P.R. Christensen, and P. Ressel. *Harmonic Analysis on Semigroups*. Springer-Verlag, New York, 1984.

[Bet93] S. Betz. Disulfide bonds and the stability of globular proteins. *Proteins, Struct., Function Genet.*, 21:167–195, 1993.

[BHB03] A. Ben-Hur and D. Brutlag. Remote homology detection: a motif based approach. *Bioinformatics*, 19:26–33, 2003.

[BHHSV01] A. Ben-Hur, D. Horn, H.T. Siegelmann, and V.N. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, 2:125–137, 2001.

[BM92] K.P. Bennett and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization Methods and Software*, 1:23–34, 1992.

[BP03] P. Baldi and G. Pollastri. The principled design of large-scale recursive neural network architectures–dag-rnns and the protein structure prediction problem. *Journal of Machine Learning Research, Accepted*, 2003.

[BS01] D. Baker and A. Sali. Protein structure prediction and structural genomics. *Science*, 294:93–96, 2001.

[BT99] C. Branden and J. Tooze. *Introduction to Protein Structure*. Garland Publishing Inc., second edition, 1999.

[BTR$^+$01] R. Bonneau, J. Tsai, I. Ruczinski, D. Chivian, C. Rohl, C.E.M. Strauss, and D. Baker. Rosetta in CASP4: progress in ab initio structure prediction. *Proteins: structure, functions and genetics*, 5:119–126, 2001.

[Bur98] C. Burges. A tutorial on support vector machines for pattern recognition. In *Data Mining and Knowledge Discovery*. Kluwer Academic Publishers, Boston, 1998. (Volume 2).

[C$^+$98] J.A. Cuff et al. Jpred: A consensus secondary structure prediction server. *Bioinformatics*, 14:892–893, 1998.

[CB00] J.A. Cuff and G.J. Barton. Application of multiple sequence alignment profiles to improve protein secondary structure prediction. *Proteins*, 40:502–511, 2000.

[CCY$^+$03] C.C Chuang, C.Y. Chen, J. Yang, P. Lyu, and J. Hwang. Relationship between protein structure and disulfide bonding patterns. *Proteins: structure, function and genetics*, 52:1–5, 2003.

[CD02] M. Collins and N. Duffy. Convolution kernels for natural language. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, 2002.

[CF93] J. Clarke and A.R. Fersht. Engineered disulfide bonds as probes of the folding pathway of barnase - increasing stability of proteins against the rate of denaturation. *Biochemistry*, 32:4322–4329, 1993.

[CFPV03a] A. Ceroni, P. Frasconi, A. Passerini, and A. Vullo. A combination of support vector machines and bidirectional recurrent neural networks for protein secondary structure prediction. In A. Cappelli and F. Turini, editors, *AI*IA 2003*, Advances in Artificial Intelligence, pages 142–153, 2003.

[CFPV03b] A. Ceroni, P. Frasconi, A. Passerini, and A. Vullo. Predicting the disulfide bonding state of cysteines with combinations of kernel machines. *Journal of VLSI Signal Processing*, 35(3):287–295, 2003.

[CFPV04] A. Ceroni, P. Frasconi, A. Passerini, and A. Vullo. Cysteine bonding state: Local prediction and global refinment using a combination of kernel machines and bi-directional recurrent neural networks. In preparation, 2004.

[CL86] C. Chothia and A. Lesk. The relation between the divergence of sequence and structure in proteins. *EMBO J.*, 5:823–826, 1986.

[CP96] F.E. Cohen and S.R. Presnell. *Protein Structure Prediction*, chapter The Combinatorial Approach, pages 207–228. Oxford University Press, 1996.

[CST00] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines*. Cambridge University Press, 2000.

[CSTEK02] N. Cristianini, J. Shawe-Taylor, A. Elisseef, and J. Kandola. On kernel-target alignment. In T.G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*. MIT Press, Cambridge, MA, 2002.

[CV95] C. Cortes and V.N. Vapnik. Support vector networks. *Machine Learning*, 20:1–25, 1995.

[Dem00] L. Demetrius. Thermodynamics and kinetics of protein folding: an evolutionary perpective. *J. Theor. Biol.*, 217:397–411, 2000.

[DKG$^+$99] N.L. Daly, A. Koltay, K.R. Gustafson, M.R. Boyd, J.R. Casas-Finet, and D.J. Craik. Solution structure by NMR of circulin a: a macrocyclic knotted peptide having anti-hiv activity. *J. Mol. Biol.*, 285:333–345, 1999.

[DS95] A. Doig and M. Sternberg. Side chains, conformational entropy in protein folding. *Protein Sci.*, 4:2247–2251, 1995.

[FA97] D. Frishman and P. Argos. Seventy-five percent accuracy in protein secondary structure prediction. *Proteins*, 329-335, 1997.

[FC99] P. Fariselli and R. Casadio. Neural network based predictor of residue contact in proteins. *Prot. Eng.*, 12:15–21, 1999.

[FC00] P. Fariselli and R. Casadio. Prediction of the number of residue contacts in proteins. In *ISMB*, volume 8, pages 146–151, 2000.

[FC01] P. Fariselli and R. Casadio. Prediction of disulfide connectivity in proteins. *Bionformatics*, 17:957–964, 2001.

[FE96] D. Fisher and D. Eisenberg. Fold recognition using sequence derived properties. *Prot. Sci.*, 5:947–955, 1996.

[FG03] D.E. Fomenko and V.M. Gladyshev. Genomics perspective on disulfide bond formation. *Antioxid. Redox Signal.*, 5(4):397–402, 2003.

[FGS98] P. Frasconi, M. Gori, and A. Sperduti. A general framework for adaptive processing of data structures. *IEEE Trans. on Neural Networks*, 9(5):768–786, 1998.

[FH00] T. Finkel and N.J. Holbrook. Oxidants, oxidative stress and the biology of ageing. *Nature*, 408(6809):239–247, 2000.

[FL01] B. Fain and M. Levitt. A novel method for sampling α-helical protein backbones. *J. Mol Biol,*, 305:191–201, 2001.

[FMC02] P. Fariselli, P. L. Martelli, and R. Casadio. A neural network-based method for predicting the disulfide connectivity in proteins. In E. Damiani et al., editors, *Knowledge based intelligent information engineering systems and allied technologies (KES 2002)*, volume 1, pages 464–468. IOS Press, 2002.

[FOVC01] P. Fariselli, O. Olmea, A. Valencia, and R. Casadio. Prediction of contact maps with neural networks and correlated mutations. *Prot. eng.*, 14:835–843, 2001.

[FPV02] P. Frasconi, A. Passerini, and A. Vullo. A two-stage SVM architecture for predicting the disulfide bonding state of cysteines. In *Proc. of the IEEE Workshop on Neural Networks for Signal Processing*, 2002.

[FRC99] P. Fariselli, P. Riccobelli, and R. Casadio. Role of evolutionary information in predicting the disulfide-bonding state of cysteine in proteins. *Proteins*, 36, 340–346 1999.

[FS00] A. Fiser and I. Simon. Predicting the oxidation state of cysteines by multiple sequence alignment. *Bionformatics*, 3:251–256, 2000.

[FSO$^+$97] J.I. Fletcher, R. Smith, S.I. O'Donoghue, M. Nilges, M. Connor, M.E.H. Howden, M.J. Christie, and G.F. King. The structure of a novel insecticidal neurotoxin, omega-atracotoxin-hv1, from the venom of an australian funnel web spider. *Nat. Struct. Biol.*, 4(7):559–566, 1997.

[FSS98] Y. Freund, R.E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. In *Proc. of the* 15^{th} *International Conference on Machine Learning*, San Francisco, 1998. Morgan Kaufmann.

[GBD92] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

[GLAB99] J. Gorodkin, O. Lund, C.A. Andersen, and S. Brunak. Using sequence motifs for enhanced neural network prediction of protein distance costraints. In *Proceedings of the 7th International Conference on Intelligent Systems for Molecular Biology*, pages 95–105. AAAI Press, 1999.

[GLV04] Y. Guermeur, A. Lifchitz, and R. Vert. A kernel for protein secondary structure prediction. In B. Schölkopf, K. Tsuda, and J. P. Vert, editors, *Kernel Methods in Computational Biology.* The MIT Press, Cambridge, MA, 2004. In press.

[GOR78] J. Garnier, D.J. Osguthorpe, and B. Robson. Analysis of the accuracy and implications of simple methods for predictiong the secondary structure of globular proteins. *J. Mol. Biol.*, 120:97–120, 1978.

[GPE$^+$04] Y. Guermeur, G. Pollastri, A. Elisseeff, D. Zelus, H. Paugam-Moisy, and P. Baldi. Combining protein secondary structure prediction models with ensemble methods of optimal complexity. *Neurocomputing*, 56C:305–327, 2004.

[Gra93] W.R. Gray. Echistatin disulfide bridges: selective reduction and linkage assignment. *Protein Sci.*, 2(10):1749–1755, 1993.

[GVB$^+$92] I. Guyon, V.N. Vapnik, B. Boser, L. Bottou, and S. Solla. Structural risk minimization for character recognition. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems IV*, San Mateo, CA, 1992. Morgan Kaufmann Publishers.

[GWG$^+$03] N.M. Giles, A.B. Watts, G.I. Giles, F.H. Fry, J.A. Littlechild, and C. Jacob. Metal and redox modulation of cysteine protein function. *Chemistry & Biology*, 10:677–693, 2003.

[HH91] S. Henikoff and J.G. Henikoff. Automated assembly of proteins blocks for database searching. *Nucleic Acids Res.*, 19(23):6565–6572, 1991.

[HMBC97] T. Hubbard, A. Murzin, S. Brenner, and C. Chothia. Scop: a structural classification of proteins database. *Nucleic Acids Res.*, 25(1):236–9, January 1997.

[HMK90] S.R. Holbrook, S.M. Muskal, and S.H. Kim. Predicting surface exposure of amino acids from protein sequence. *Prot. Eng.*, 3:659–665, 1990.

[HS94a] P. M. Harrison and M.J.E. Sternberg. Analysis and classification of disulfide connectivity in proteins. *J. Mol. Biol.*, 244:448–463, 1994.

[HS94b] U. Hobohm and C. Sander. Enlarged representative set of protein structures. *Protein Science*, 3:522–524, 1994.

[HS96] P.M. Harrison and M.J.E. Sternberg. The disulfide β-cross: from cysteine geometry and clustering to classification of small disulfide-rich protein folds. *J. Mol. Biol.*, 264:603–623, 1996.

[HS01] S. Hua and Z. Sun. A novel method of protein secondary structure prediction with high segment overlap measure: Support vector machine approach. *J Mol. Biol.*, 308(2):397–407, 2001.

[HSP99] E.S. Huang, R. Samundrala, and J.W. Ponder. Ab initio fold prediction of small helical proteins using distance geometry and knowledge-based scoring functions. *J. Mol. Biol.*, 290:267–281, 1999.

[JDH00] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. *Journal of Computational Biology*, 7(1–2):95–114, 2000.

[JFPC01] I. Jacoboni, P. Fariselli, V. De Pinto, and R. Casadio. Prediction of transmembrane regions of β-barrel membrane proteins with a neural network-based model. *Prot.Sci.*, 10:779–787, 2001.

[JH98a] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *Proc. of Neural Information Processing Conference*, 1998.

[JH98b] T. Jaakkola and D. Haussler. Probabilistic kernel regression models. In *Proc. of Neural Information Processing Conference*, 1998.

[Joa98] T. Joachims. Text categorization with support vector machines: Learning with many relevant features. In *Proceedings of the European Conference on Machine Learning*, 1998.

[Jon99a] D.T. Jones. GenThreader: an efficient and reliable protein fold recognition method for genomic sequences. *J.Mol.Biol.*, 287:797–815, 1999.

[Jon99b] D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J.Mol.Biol.*, 292:195–202, 1999.

[Jon99c] D.T. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J Mol. Biol.*, 292:195–202, 1999.

[JTT94a] D.T. Jones, W.R. Taylor, and J.M. Thornton. A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry*, 33:3038–3049, 1994.

[JTT94b] D.T. Jones, W.R. Taylor, and J.M. Thornton. A new approach to protein fold recognition. *Nature*, 358:86–89, 1994.

[KBH98] K. Karplus, C. Barrett, and R. Hughey. Hidden markov models for detecting remote protein homologies. *Bioinformatics*, 14(10):846–856, 1998.

[KF03] J.L. Klepeis and C.A. Floudas. Prediction of β-sheet topology and disulfide bridges in polypeptides. *J. Comput. Chem.*, 24:191–208, 2003.

[KH04] J. Kuchar and R.P. Hausinger. Biosynthesis of metal sites. *Chem. Rev.*, 104:509–525, 2004.

[KJ01] P. Kovacic and J.D. Jacintho. Mechanisms of carcinogenesis: focus on oxidative stress and electron transfer. *Curr Med Chem 2001; 8: 773*, 8:773–796, 2001.

[KLvHS01] A. Krogh, B. Larsson, G. von Heijne, and E.I. Sonnhammer. Predicting transmembrane protein topology with a hidden markov model. *J. Mol. Biol.*, 305:567–580, 2001.

[KP04] H. Kim and H. Park. Prediction of protein relative solvent accessibility with support vector machines and long-range interaction 3d local descriptor. *Proteins*, 54(3):557–562, 2004.

[KRDF98] P. Klappa, L.W. Ruddock, N.J. Darby, and R.B. Freedman. The b' domain provides the principal peptide-binding site of protein disulfide isomerase but all domains contribute to binding of misfolded proteins. *EMBO.J.*, 17:927–935, 1998.

[KS96] R.D. King and M.J.E. Sternberg. Identification and application of the concepts important for accurate and reliable protein secondary structure prediction. *Prot. Sci.*, 5:2298–2310, 1996.

[KW71] G. Kimeldorf and G. Wahba. Some results on tchebycheffian spline functions. *J. Math. Anal. Applic.*, 33:82–95, 1971.

[KWTR00] T.A. Klink, K.J. Woycechosky, K.M. Taylor, and R.T. Raines. Contribution of disulfide bonds to the conformational stability and catalytic activity of ribonuclease A. *Eur. J. Biochem.*, 267:566–572, 2000.

[LCH01] A. M. Lesk, L. Lo Conte, and T. J. P. Hubbard. Assessment of novel fold targets in casp4: prediction of three-dimensional structures, secondary structures and interresidue contacts. *Proteins*, 45(S5):98–118, 2001.

[LEN02] C. Leslie, E. Eskin, and W.S. Noble. The spectrum kernel: a string kernel for svm protein classification. In *Proc. of the Pacific Symposium on Biocomputing*, pages 564–575, 2002.

[LEWN03] C. Leslie, E. Eskin, J. Weston, and W.S. Noble. Mismatch string kernels for svm protein classification. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 1417–1424. MIT Press, Cambridge, MA, 2003.

[LHBB96] O. Lund, J. Hansen, S. Brunak, and J. Bohr. Relationship between protein structure and geometrical constraints. *Protein Science*, 5:2217–2225, 1996.

[Lim74] V.I. Lim. Algorithms for prediction of α-helical and β-structural regions in globular proteins. *J. Mol. Biol.*, 88:873–894, 1974.

[LJ03] K. Linke and U. Jakob. Not every disulfide lasts forever: Disulfide bond formation as a redox switch. *Antioxid. Redox Signal.*, 5(4):425–434, 2003.

[LKE04] C. Leslie, R. Kuang, and E. Eskin. Inexact matching string kernels for protein classificatio. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology.* MIT Press, 2004. In press.

[LMS$^+$01] B. Logan, P. Moreno, B. Suzek, Z. Weng, and S. Kasif. A study of remote homology detection. Technical report, Cambridge Research Laboratory, June 2001.

[LN03] L. Liao and W.S. Noble. Combining pairwise sequence similarity and support vector machines for detecting remote protein evolutionary and structural relationships. *Journal of Computational Biology*, 10(6):857–868, 2003.

[LRG86] J.M. Levin, B. Robson, and J. Garnier. An algorithm for secondary structure determination in proteins based on sequence similarity. *FEBS*, 205(2):303–308, 1986.

[M$^+$89] M. Matsumura et al. Substantial increase of protein stability by multiple disulfide bonds. *Nature*, 342:291–293, 1989.

[MAMR$^+$98] J.M Mass, P. Aloy, M.A. Marti-Renom, B. Oliva, C. Blanco-Aparicio, M.A Molina, R. de Llorens, E. Querol, and F.X. Avils. Protein similarities beyond disulphide bridge topology. *Journal of Molecular Biology*, 284:541–548, 1998.

[MFKC02] P.L. Martelli, P. Fariselli, A. Krogh, and R. Casadio. A sequence-profile-based hmm for predicting and discriminating beta barrel membrane proteins. *Bioinformatics*, 18, 2002.

[MFMC02] P.L. Martelli, P. Fariselli, L. Malaguti, and R. Casadio. Prediction of the disulfide-bonding state of cysteines in proteins at 88% accuracy. *Protein Sci.*, 11(2735–2739), 2002.

[MGHT02] M.H. Mucchielli-Giorgi, S. Hazout, and P. Tuffèry. Predicting the disulfide bonding state of cysteines using protein descriptors. *Proteins*, 46:243–249, 2002.

[Mit97] T.M. Mitchell. *Machine Learning.* McGraw-Hill, 1997.

[MP92] D.C. Montgomery and E.A. Peck. *Introduction to Linear Regression Analysis.* John Wiley and Sons, Inc., 2nd edition edition, 1992.

[MR03] S. Mika and B. Rost. Uniqueprot: creating representative protein sequence sets. *Nucleic Acids Research*, 31(13):3789–3791, 2003.

[Nob04] W.S. Noble. Support vector machine applications in computational biology. In B. Schölkopf, K. Tsuda, and J.-P. Vert, editors, *Kernel Methods in Computational Biology.* MIT Press, 2004. In press.

[NR03] M.N. Nguyen and J.C. Rajapakse. Multi-class support vector machines for protein secondary structure prediction. *Genome Informatics*, 14:218–227, 2003.

[OFA$^+$98] D.A. Oren, O. Froy, E. Amit, N. Kleinberger-Doron, M. Gurevitz, and B. Shaanan. An excitatory scorpion toxin with a distinctive feature: an additional alpha helix at the c terminus and its implications for interaction with insect sodium channels. *Structure*, 6(9):1095–1103, 1998.

[OV97] O. Olmea and A. Valencia. Improving contact predictions by the combination of correlated mutations and other sources of sequence information. *Fold. Des.*, 2:25–32, 1997.

[PB02] G. Pollastri and P. Baldi. Prediction of contact maps by giohmms and recurrent neural networks using lateral propagation from all four cardinal corners. *Bioinformatics*, 2002. In press.

[PBFC02] G. Pollastri, P. Baldi, P. Fariselli, and R. Casadio. Prediction of coordination number and relative solvent accessibility in proteins. *Proteins*, 47(2):142–153, 2002.

[PBVF02] G. Pollastri, P. Baldi, A. Vullo, and P. Frasconi. Prediction of protein topologies using giohmms and gbrnns. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems*, volume 15. MIT-Press, 2002.

[PF04] A. Passerini and P. Frasconi. Learning to discriminate between ligand bound and disulfide bound cysteines. *Protein Eng.*, 2004. (submitted).

[PHCAV97] F. Pazos, M. Helmer-Citterich, G. Ausiello, and A. Valencia. Correlated mutations contain information about protein-protein interations. *J. Mol. Biol.*, 271:511–523, 1997.

[PLN$^+$00] T.N. Petersen, C. Lundegaard, M. Nielsen, H. Bohr, J. Bohr, S. Brunak, G.P. Gippert, and O. Lund. Prediction of protein secondary structure at 80% accuracy. *Proteins*, 41(1):17–20, October 2000.

[PPBA98] S. Pascarella, R. De Persio, R. Bossa, and P. Argos. Easy method to predict solvent accessibility from multiple protein sequence alignments. *Proteins*, 32:190–199, 1998.

[PPRB02] G. Pollastri, D. Przybylski, B. Rost, and P. Baldi. Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins*, 47(2):228–235, 2002.

[QS88] N. Qian and T.J. Sejnowski. Predicting the secondary structure of globular proteins using neural networks models. *J. Mol. Biol.*, 202:865–884, 1988.

[Rab89] L.R. Rabiner. A tutorial on hidden markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–286, 1989.

[RaPFS95] B. Rost, R. Casadio adn P. Fariselli, and C. Sander. Prediction of helical transmembrane segments at 95% accuracy. *Protein Science*, 4:521–533, 1995.

[RB99] C.J. Richardson and D.J. Barlow. The bottom line for prediction of residue solvent accessibility. *Prot.Eng.*, 12:1051–1054, 1999.

[RB01] D. Ritz and J. Beckwith. Roles of thiol-redox pathways in bacteria. *Annu. Rev. Microbiol.*, 55:21–48, 2001.

[RCF96] B. Rost, R. Casadio, and P. Fariselli. Topology prediction for helical transmembrane proteins at 86% accuracy. *Prot.Sci.*, 5:1704–1718, 1996.

[RFS98] B. Reva, A.V. Filkenstein, and J. Skolnik. What is the probability of a chance prediction of a protein structure with rmsd of 6 å? *Fold. Des.*, 3:141–147, 1998.

[RK96] S.K. Riis and A. Krogh. Improving prediction of protein secondary structure using structured neural networks and multiple sequence alignments. *J. Comp. Biol.*, 3:163–183, 1996.

[Ros95] B. Rost. TOPITS: threading one-dimensional predictions into three dimensional structures. In C. Rawlings et al., editors, *Third International Conference on Intelligent Systems for Molecular Biology*, pages 314–321. AAAI press, 1995.

[Ros96] B. Rost. PHD: predicting one-dimensional protein structure by profile based neural networks. *Meth. Enzymol.*, 266:525–539, 1996.

[Ros98] B. Rost. Protein structure prediciton in 1d, 2d and 3d, 1998.

[RS94] B. Rost and C. Sander. Conservation and prediction of solvent accessibility in protein families. *Proteins*, 20:55–72, 1994.

[SB03] Y. Shao and C. Bystroff. Predicting interresidue contacts using templates and pathways. *Proteins: Structure, Function, and Genetics*, 53:497–502, 2003.

[Sip95] M.J. Sippl. Knowledge-based potential for proteins. *Curr. Opin. Struct. Biol.*, 5(2):229–235, 1995.

[SKS94] I.N. Shindyalov, N.A. Kolchanov, and C. Sander. Can three-dimensional contacts of proteins be predicted by analysis of correlated mutations? *Prot. Eng.*, 349-358, 1994.

[SS95] A. Salamov and V. Solovyev. Prediction of protein secondary structure by combining nearest-neighbor algorithms and multiple sequence alignment. *J. Mol. Biol.*, 247:11–15, 1995.

[SS97] R. Sanchez and A. Sali. Evaluation of comparative protein structures modeling by MODELLER-3. *Proteins Suppl.*, 1:50–58, 1997.

[SS02] B. Schölkopf and A.J. Smola. *Learning with Kernels.* The MIT Press, Cambridge, MA, 2002.

[SSB03] Q. Su, S. Saxonov, and D. Brutlag. eblocks: an automated database of protein conserved regions maximizing sensitivity and specificity, 2003. http://fold.stanford.edu/eblocks/.

[SVB02] M.S. Singer, G. Vriend, and R.P. Bywater. Prediction of protein residue contacts with a pdb-derived likelihood matrix. *Protein Engineering*, 15(9):721–725, 2002.

[SvHK98] E. Sonnhammer, G. von Heijne, and A. Krogh. A hidden markov model for predicting transmembrane helices in protein sequences. In *Proceedings 6th international conference on intelligent systems for molecular bioology*, volume 6, pages 175–182. AAAI Press, 1998.

[SW81] T.F. Smith and M.S. Waterman. Identification of common molecular subsequences. *J. Mol. Biol.*, 147:195–197, 1981.

[TG96] M.J. Thompson and R.A. Goldstein. Predicting solvent accessibility: Higher accuracy using bayesian statistics and optimized residue substitution classes. *Proteins*, 25:38–47, 1996.

[Tik63] A.N. Tikhonov. On solving ill-posed problem and method of regularization. *Dokl. Akad. Nauk USSR*, 153:501–504, 1963.

[Tsu99] K. Tsuda. Support vector classification with asymmetric kernel function. In M. Verleysen, editor, *Proc. of ESANN*, pages 183–188, 1999.

[Ukk95] E. Ukkonen. On-line construction of suffix trees. *Algorithmica*, 14(3):249–260, 1995.

[Vap79] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data [in Russian].* Springer-Verlag, Nauka, Moscow, 1979. (English translation: Springer-Verlag, New York, 1982).

[Vap95] V.N. Vapnik. *The Nature of Statistical Learning Theory.* Springer, New York, 1995.

[Vap98] V. Vapnik. *Statistical Learning Theory.* John Wiley, New York, 1998.

[Ver02] R. Vert. Designing a m-svm kernel for protein secondary structure prediction. Master's thesis, DEA informatique de Lorraine, 2002.

[VF03] A. Vullo and P. Frasconi. Prediction of protein coarse contact maps. *Journal of Bioinformatics and Computational Biology*, 1(2):411–431, 2003.

[VF04] A. Vullo and P. Frasconi. Disulfide connectivity prediction using recursive neural networks and evolutionary information. *Bioinformatics*, 20:653–659, 2004.

[VKD97] M. Vendruscolo, E. Kussel, and E. Domany. Recovery of protein structure from contact maps. *Fold. Des.*, 2:295–306, 1997.

[VVD$^+$98] C. Vita, J. Vizzavona, E. Drakopoulou, S. Zinn-Justin, B. Gilquin, and A. Mnez. Novel miniproteins engineered by the transfer of active sites to small natural scaffolds. *Biopolymers*, 47(1):93–100, 1998.

[WMBJ03] J.J. Ward, L.J McGuffin, B.F. Buxton, and D.T. Jones. Secondary structure prediction with support vector machines. *Bioinformatics*, 19(13):1650–1655, 2003.

[WWNS00] W.J. Wedemeyer, E. Welkler, M. Narayan, and H.A. Scheraga. Disulfide bonds and protein-folding. *Biochemistry*, 39:4207–4216, 2000.

[YBM02] Z. Yuan, K. Burrage, and J.S. Mattick. Prediction of protein solvent accessibility using support vector machines. *Proteins*, 48(3):566–570, 2002.

[ZJB00] M.J. Zaki, S. Jin, and C. Bystroff. Mining residue contacts in proteins using local structure predictions. In *IEEE international symposium on bioinformatics and biomedical engineering*, pages 168–175, 2000.