

## Chapter 2

# Divide and Conquer Strategies for Protein Structure Prediction

Pietro Di Lena, Piero Fariselli, Luciano Margara, Marco Vassura,  
and Rita Casadio

**Abstract** In this chapter, we discuss some approaches to the problem of protein structure prediction by addressing “simpler” sub-problems. The rationale behind this strategy is to develop methods for predicting some interesting structural characteristics of the protein, which can be useful per se and, at the same time, can be of help in solving the main problem. In particular, we discuss the problem of predicting the protein secondary structure, which is at the moment one of the most successful sub-problems addressed in computational biology. Available secondary structure predictors are very reliable and can be routinely used for annotating new genomes or as input for other more complex prediction tasks, such as remote homology detection and functional assignments. As a second example, we also discuss the problem of predicting residue–residue contacts in proteins. In this case, the task is much more complex than secondary structure prediction, and no satisfactory results have been achieved so far. Differently from the secondary structure sub-problem, the residue–residue contact sub-problem is not intrinsically simpler than the prediction of the protein structure, since a roughly correctly predicted set of residue–residue contacts would directly lead to prediction of a protein backbone very close to the real structure. These two protein structure sub-problems are discussed in the light of the current evaluation of the performance that are based on periodical blind-checks (CASP meetings) and permanent evaluation (EVA servers).

## 2.1 Introduction

Methods developed for the problem of the protein structure prediction in silico aim at finding the three-dimensional (3D) conformation of the protein starting from its amino-acidic residue sequence (primary structure) [7]. Protein function is strictly dependent on the native protein 3D structure and protein structure prediction is one

---

P. Di Lena (✉)

Department of Computer Science, University of Bologna, Italy

e-mail: [dilena@cs.unibo.it](mailto:dilena@cs.unibo.it)

of the most important and mostly studied problems of computational biology [26]. Despite many efforts, an acceptable solution for new sequences, not having homologous sequences for which the 3D structure is known, is still to be found. Given the difficulty to compute directly the protein 3D structure, many intermediate problems have been addressed. One way to simplify the problem is to compute features that are local with respect to the backbone of the protein. These are called secondary structure motifs and are well characterised as alpha-helices, beta-sheets and coil on the basis of specific values of torsion angles. The problem of predicting secondary structures in proteins has been also addressed with machine learning methods and it is presently considered one of the most successful problems of computational biology [43]. In this chapter, we will comment on the most successful implementations of protein secondary structure prediction methods.

However, even when well predicted, secondary structure alone does not carry enough information to understand protein 3D conformation. To this aim, it would suffice to find global distance constraints between each couple of residues. This sub-problem is commonly known as residue–residue contact prediction and it has been again addressed with machine learning methods [3]. Residue–residue contact prediction is today the only method that can grasp in a simplified manner long-range interactions between residues of a protein sequence. Although the problem is still far from being solved, we will review the most efficient algorithms that are presently the state-of-the-art methods in the field.

So far, the most interesting results in secondary structure prediction and residue–residue contact prediction have been achieved by a clever combination of machine-learning methods with evolutionary information available in the ever growing databases of protein structures [1, 11, 18, 20].

In order to make the chapter self-contained as much as possible, in the following sections we briefly review the most basic concepts of machine learning methods (Sect. 2.2) and the most commonly used techniques for extracting evolutionary information from databases of protein sequences (Sect. 2.3). The rest of the chapter is devoted to the detailed description of the most famous secondary structure predictors (Sect. 2.4) and residue–residue contact predictors (Sect. 2.5). For both topics, we also describe in detail the standard evaluation criteria adopted to measure the performance of the predictors and outline what is the state of the art in terms of the respective evaluation criteria according to the experiments performed at CASP meetings<sup>1</sup> and EVA server.<sup>2</sup>

## 2.2 Data Classification with Machine Learning Methods

Machine learning is concerned with the design and development of algorithms for the acquisition and integration of knowledge. Biological data classification is a typical problem usually approached with machine learning methods.

---

<sup>1</sup> <http://predictioncenter.org/>

<sup>2</sup> <http://cubic.bioc.columbia.edu/eva/>

Data classification is the problem of assigning objects to one of the mutually exclusive classes according to statistical properties derived from a training set of examples sharing the same nature of such objects. The problem can be easily formalised in the following way. Assume that the data we want to classify is represented by a set of  $n$ -dimensional vectors  $x \in X = \mathbb{R}^n$  and that each one of such vectors can be assigned to exactly one of  $m$  possible classes  $c \in C = \{1, \dots, m\}$ . Given a set of pre-compiled examples  $E = \{(x_1, c_1), \dots, (x_k, c_k)\}$ , where  $(x_i, c_i) \in X \times C$  and  $|E| < |X|$ , the objective is to learn from  $E$  a mapping  $f : X \rightarrow C$  that assigns every  $x \in X$  to its correct class  $c \in C$ . In the biological context, each entry of the vector  $x \in X$  usually represents a single feature (observation) of the object we want to classify (i.e.,  $x$  is not the object itself), and the number of classes is typically limited to two/three. Moreover, machine learning methods generally do not provide a rigid classification of an object; they instead return the probability that the object belongs to each one of the possible classes (a classification can be obtained by choosing the class with higher probability). In bioinformatics, the most widely used machine learning methods for data classification are neural networks (NN), support vector machines (SVM) and Hidden Markov models (HMM). We do not discuss here the features and the limitations of such methods (for an extensive introduction, see [5]), but we briefly outline the problem of correctly evaluating the performance of predictors of protein structural characteristic.

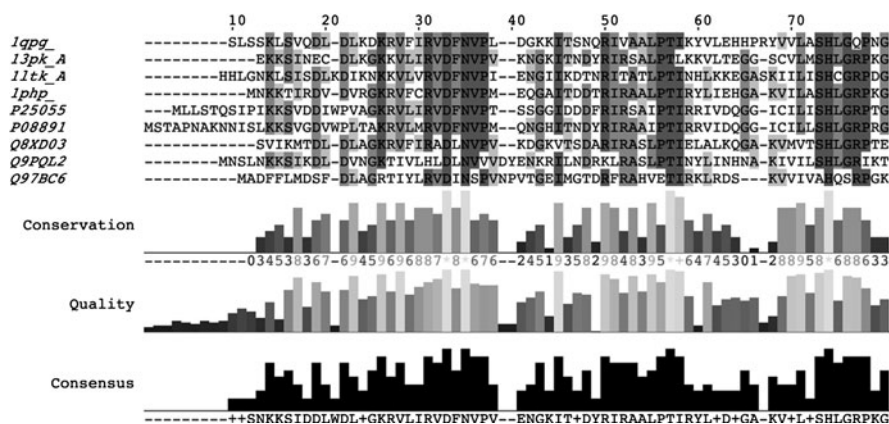
A reliable approach for assessing the performance of data classification is a necessary pre-condition for every machine learning-based method. The cross-validation is the standard technique used to statistically evaluate how accurate a predictive model is. The cross-validation involves the partitioning of the example set into several disjoint sets. In one round of cross-validation, one set is chosen as test set and the others are used as training set. The method is trained on the training set and the statistical evaluation of the performance is computed from the prediction results obtained on the test set. To reduce variability, multiple cross-validation rounds are performed by interchanging training and test sets, and the results obtained are averaged over the number of rounds.

A proper evaluation (or cross-validation) of prediction methods needs to meet one fundamental requirement: the test set must not contain examples too much similar to those contained in the training set. When testing prediction methods for protein features (such as secondary structure or inter-residue contacts), this requirement transduces in having test and training sets compiled from proteins that share no significant pairwise sequence identity (typically  $<25\%$ ). If homologous sequences are included in both training and test set, the average prediction accuracy does not provide a reliable estimation of the performance, and, in particular, it does not reflect the performance of the method for sequences not homologue to those in the training set.

## 2.3 Evolutionary Information and Multiple Sequence Alignments

One of the most successful tools in bioinformatics is the introduction of evolutionary information as a key ingredient for protein structure and function predictions. The evolutionary information contained in a set of (related) protein sequences can be extracted from a multiple alignment of all the sequences in the set. The multiple sequence alignment (MSA) refers to the problem of aligning three or more sequences in order to identify their regions of similarity. An MSA of a set of protein sequences is represented as a matrix, where each row corresponds to a single sequence and each column corresponds to a set of aligned residues, one for each protein in the set (Fig. 2.1).<sup>3,4</sup> When properly computed, each column of the MSA encodes the possible evolutionary mutations that can occur at the corresponding positions in the sequences included in the MSA. Those columns of the MSA that exhibit low variability correspond to regions that are highly conserved with respect to the evolutionary mutations of protein sequences.

In a pioneering work, Benner and Gerloff [4] introduced the idea that multiple sequence alignments can improve protein structure prediction. Their basic concept relies on the fact that the most conserved regions of a protein sequence (in terms of multiple alignments) are those regions which are either functionally important, and/or buried in the protein core. By this, Benner and Gerloff demonstrated that the degree of solvent accessibility of an amino acid residue could be predicted with



**Fig. 2.1** Multiple sequence alignment taken from the BALiBASE3 database (example BB50004 from RV50 reference set) and visualised with the Jalview software. Only the first 80 positions of the alignment are visualised. The symbol “-” denotes a gap. Darker columns of the MSA correspond to higher conserved regions. The only perfectly conserved positions are 33, 35, 57 and 74

<sup>3</sup> BALiBASE3 database: <http://www-bio3d-igbmc.u-strasbg.fr/balibase/>

<sup>4</sup> Jalview software: <http://www.jalview.org/>

reasonable accuracy by clustering the sequences in an aligned family, and assessing the degree of sequence variability observed between very similar pairs. Lately, this idea was exploited by Rost and Sander, who showed that it was possible to improve the accuracy of the prediction of secondary structures and solvent accessibility introducing evolutionary information in the form of sequence profiles as input to neural networks [42].

Differently from an MSA, whose dimension increases linearly with the number of aligned sequences, a sequence profile of a protein is a matrix  $P$  whose columns represent the sequence positions and whose rows are the 20 possible residue symbols. The profile matrix  $P$  is computed from a MSA and it is relative to a specific sequence of interest  $p$ . Each element  $P_{ai}$  of the sequence profile represents the normalised frequency of the residue type  $a$  in the aligned position  $i$ . In practice, given an MSA that contains the sequence of interest  $p$ , we derive the column  $i$  of the corresponding profile by computing the frequencies of occurrence of each residue in the column of the MSA corresponding to the  $i$ th residue of  $p$ . In this way, the information contained in a profile  $P$  is not dependent on the number of aligned sequences so that it becomes easy to use fragments of the matrix  $P$  as input for machine learning methods.

The computation of an MSA for a query sequence is a complex process both in terms of time and care required. It consists of two steps. First, a search of the query sequence against a non-redundant dataset of protein sequences is needed in order to select a set of chains that are similar to the query one. There are several optimal and near-optimal pairwise-alignment algorithms to perform such searches. Currently, the heuristic basic local alignment search tool (BLAST) [2] is considered the standard-de-facto software for pairwise sequence comparison. Despite the fact that exact algorithms are available for pairwise sequence comparison, the heuristic BLAST is the most widely used due to its speed (non-redundant datasets can contain millions of different protein sequences) and good performance compared to exact algorithms. The selection of similar sequences must be performed carefully in order to avoid the introduction of meaningless sequences in the MSA, such as sequences with low complexity regions. Low complexity regions represent sequences of very non-random composition ("simple sequences," "compositionally-biased regions"). They are abundant in natural sequences and may determine high scoring matching segments in unrelated protein sequences. To avoid this problem, BLAST implements a filter procedure based on the SEG [49] software. SEG provides a measure of compositional complexity of a sequence segment and divides sequences into contrasting segments of low complexity and high complexity. Typically, globular domains have higher sequence complexity than fibrillar or conformationally disordered protein segments. When used in BLAST, SEG replaces the low complexity regions within the input sequence with  $X$ 's to prevent spurious matching with unrelated sequences.

When the set of similar sequences has been selected, the second step consists of building an MSA. Differently from the pairwise sequence alignment problem, building an optimal multiple alignment is a difficult task and it is not computable in reasonable time. Several software implementations of heuristic algorithms for MSA

are available (MAXHOM [44], CLUSTALW [47], T-Coffee [29] and MUSCLE [10] are currently the most widely used) and none of them is globally accepted as a standard.

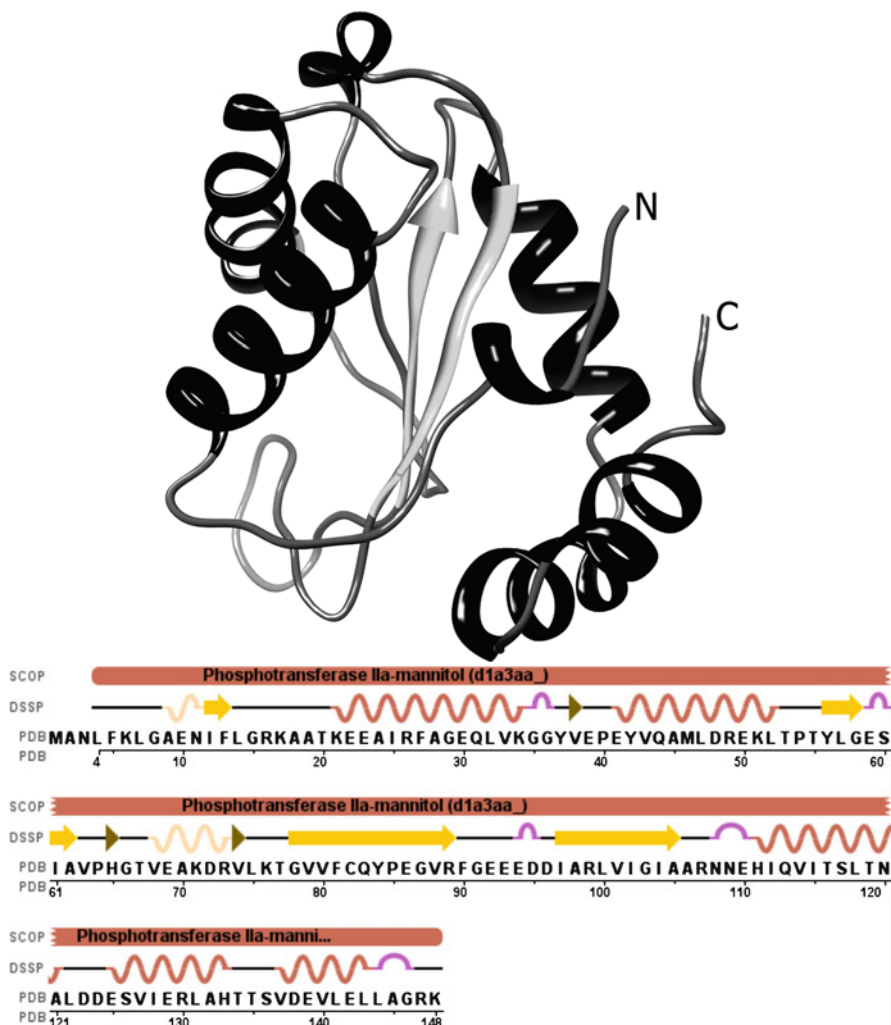
Few years ago, the procedure described above for building MSA was almost standard and time-consuming; thus, during the construction and tuning of new prediction methods most of the researchers used the homology-derived secondary structure of proteins (HSSP) precompiled multiple sequence alignments generated with the MAXHOM software. Currently, a faster and more accurate method for the construction of reliable sequence profiles is the adoption of the position specific iterative (PSI) feature in BLAST [2]. In PSI-BLAST a sequence profile and a position-specific scoring matrix (PSSM) are automatically constructed from a pseudo-multiple alignment of the highest scoring hits in an initial BLAST search. The PSSM is generated by calculating position-specific scores for each position in the alignment. Highly conserved positions receive high scores and weakly conserved positions receive scores near zero. The profile is used to perform a further BLAST search and the current profile is refined according to the outcomes of the new search. This iterative procedure is performed until the retrieved sequences remain constant or a fixed number of iterations are achieved. In [21], the prediction accuracy of secondary structure was improved by using directly the PSI-BLAST PSSM to feed a neural network system.

## 2.4 Secondary Structure Prediction

In biochemistry and structural biology, the protein secondary structure refers to the three-dimensional shape of consecutive residue segments. The most common secondary structure elements are alpha-helices and beta-sheets. The formation of secondary structure elements is mostly guided by local inter-residue interactions mediated by hydrogen bonds. For example, an alpha-helix is formed when hydrogen bonds occur regularly between positions  $i$  and  $i + 4$  in a protein segment. When hydrogen bonds occur between positions  $i$  and  $i + 3$ , then a  $3_{10}$  helix is formed. A beta-sheet is formed when two strands are joined by hydrogen bonds involving alternating residues on each participating strand. In the 1950s, Pauling correctly guessed the formation of helices and strands [31, 32], before any protein structure had been determined experimentally.

There are several methods for defining protein secondary structure elements. The dictionary of protein secondary structure (DSSP) method [23] is actually considered the de facto standard for secondary structure definition. The DSSP defines eight types of secondary structure elements, based on hydrogen-bonding patterns as those initially proposed by Pauling (Fig. 2.2):

- G = 3-turn helix ( $3_{10}$  helix). Min length three residues.
- H = 4-turn helix (alpha helix). Min length four residues.
- I = 5-turn helix (pi helix). Min length five residues.
- T = hydrogen bonded turn (3, 4 or 5 turn).



**Fig. 2.2** Graphical representation of the *Escherichia coli* phosphotransferase IIamannitol (1a3a chain A, 148 residues). The figure above shows the three-dimensional structure, highlighting helices, strands (arrows) and coils (irregular loops). The figure below shows the amino-acidic sequence and the respective DSSP secondary structure elements

- E = extended strand in parallel and/or anti-parallel beta sheet conformation. Min length two residues.
- B = residue in isolated beta bridge (single pair beta-sheet hydrogen bond formation).
- S = bend (the only non-hydrogen bond-based assignment).
- C = every residue that cannot be assigned to any of the above conformations.



It is worth noting that the eight-state DSSP vocabulary is just a simplification of the possible variations of hydrogen-bonding patterns present in proteins. For example, the class C stands for loops or irregular elements, which are often called coils or random coils. In order to simplify the DSSP classification, most of the secondary structure prediction methods reduce further the DSSP vocabulary into three most characteristic states, helix (*H*), strand (*E*) and other (*L*), according to the scheme proposed in the secondary structure section of EVA server (EVAsec<sup>5</sup>): *H* includes (H,G,I), *E* includes (E,B) and *L* includes all the others.

Predicting the protein tertiary structure from only its amino acid sequence is actually one of the most challenging problems in structural bioinformatics. In contrast, the secondary structure prediction is more tractable and has been successfully addressed in the last decades. In particular, the successful results in this field have been achieved by combining machine learning methods with evolutionary information available in the ever-growing databases of protein structures. Early secondary structure prediction methods were based on statistics derived from protein segments [6, 16]. The statistics were used to predict how likely the central residue in the segment is in some particular secondary structure element. Several different methods (machine learning based and not) were exploited to derive statistics from protein segments. The accuracy of all these methods was limited to slightly more than 60%. A first significant step-forward in prediction accuracy was made by exploiting evolutionary information encoded in MSA [43]. The PHD predictor by Rost and Sander [42] is the first method that used MSA successfully for secondary structure prediction and that was able to achieve a prediction accuracy >70%. The next step-forward was made using more accurate evolutionary information resulting from improved searches and larger databases. The PSIPred method by Jones [21] is historically the first method for secondary structure prediction that cleverly used position-specific alignments from PSI-BLAST and that achieved a further improvement of slightly more than 5% in accuracy. The accuracy of modern secondary structure prediction methods is currently about 77%. While this is not the best possible we can do, due to the approximations made by the DSSP in assigning secondary structure classes, the theoretical limit of prediction accuracy has been estimated approximately 88% [41].

The performances of secondary structure prediction methods were evaluated in CASP1 experiments from CASP1 (1994) to CASP5 (2002). Starting from CASP6, the secondary structure prediction category was not included in the experiments since the progress in this area was too little to be detected with the few amounts of data available in CASP sessions. Currently, larger scale benchmarking is continuously assessed by the EVAsec experiments. In the following section (Sect. 2.4.1), we review the most important measures of secondary structure prediction accuracy (as defined in EVAsec) and we also provide a comparison of some secondary structure prediction methods in terms of these measures. We conclude (Sect. 2.4.2) with the detailed description of two secondary structure predictors, PHD (Sect. 2.4.2.1) and PSIPred (Sect. 2.4.2.2).

---

<sup>5</sup> [http://cubic.bioc.columbia.edu/eva/doc/intro\\_sec.html](http://cubic.bioc.columbia.edu/eva/doc/intro_sec.html)



### 2.4.1 EVAsec: Evaluation of Secondary Structure Prediction Servers

The objectives of EVAsec<sup>3</sup> are to provide a continuous, fully automated and statistically significant analysis of protein secondary structure prediction servers. EVAsec continuously evaluates *secondary structure prediction servers* in real time, whenever new data are available. Secondary structure prediction servers are fully automated websites that accept prediction tasks on request and provide answers in electronic format. At the moment (April 2009), EVAsec is running since 303 weeks and monitors 13 servers.

The most simple and widely used measure of secondary structure prediction accuracy used in EVAsec is the *per-residue prediction accuracy*:

$$Q_3 = 100 \cdot \frac{1}{N} \sum_{i=1}^3 M_{ii}, \quad (2.1)$$

where  $N$  is the length of the protein and  $M \in \mathbb{N}^{3 \times 3}$  is the confusion matrix, i.e.,  $M_{ij}$  is equal to the number of residues observed in state  $i$  and predicted in state  $j$  with  $i, j \in \{H, E, L\}$ . Since a typical protein contains about 32%  $H$ , 21%  $E$ , 47%  $L$ , the correct prediction of class  $L$  tends to dominate the overall accuracy. There are several other measures defined in EVAsec (such as per-state/per-segment accuracy) that can be used to limit this effect. The per-state measures are based on the *Matthews correlation coefficient*:

$$C_i = \frac{p_i \cdot n_i - u_i \cdot o_i}{\sqrt{(p_i + u_i) \cdot (p_i + o_i) \cdot (n_i + u_i) \cdot (n_i + o_i)}}, \quad (2.2)$$

where  $i \in \{H, E, L\}$ ,  $p_i = M_{ii}$  (true positives),  $n_i = \sum_{j \neq i}^3 \sum_{k \neq i}^3 M_{jk}$  (true negatives),  $o_i = \sum_{j \neq i}^3 M_{ji}$  (false positives) and  $u_i = \sum_{j \neq i}^3 M_{ij}$  (false negatives). The most important per-segment accuracy is the *Segment Overlap* (SOV) measure, based on the average segment overlap between the observed and predicted segment instead of the average per-residue accuracy:

$$\text{SOV} = \frac{100}{N} \sum_{(s_1, s_2) \in S} \frac{\min \text{OV}(s_1, s_2) + \delta(s_1, s_2)}{\max \text{OV}(s_1, s_2)} \cdot \text{len}(s_1), \quad (2.3)$$

where

- $s_1, s_2$  are, respectively, the observed and predicted secondary structure segments in state  $i \in \{H, E, L\}$ , i.e., all residues of  $s_1, s_2$  are in state  $i$ ,
- $S$  is the set of segment pairs  $(s_1, s_2)$  that are both in the same state  $i$  and that overlap at least by one residue. Conversely,  $S'$  is the set of observed segments  $s_1$  for which there is no predicted overlapping segment  $s_2$ .
- $\text{len}(s_1)$  is the number of residues in segment  $s_1$ ,
- $N = \sum_{(s_1, s_2) \in S} \text{len}(s_1) + \sum_{s_1 \in S'} \text{len}(s_1)$  is the normalization value,

- $\text{minOV}(s_1, s_2)$  is the length of actual overlap of  $s_1$  and  $s_2$ , i.e., the extent for which both segments have residues in state  $i$ ,
- $\text{maxOV}(s_1, s_2)$  is the length of the total extent for which either of the segments  $s_1$  or  $s_2$  has a residue in state  $i$ ,
- $\delta(s_1, s_2)$  is equal to

$$\min \left\{ \text{maxOV}(s_1, s_2) - \text{minOV}(s_1, s_2), \text{minOV}(s_1, s_2), \begin{Bmatrix} \text{int}(\text{len}(s_1)/2), \text{int}(\text{len}(s_2)/2) \end{Bmatrix} \right\}$$

The accuracy of prediction of the 13 servers currently monitored by EVAsec is given in Table 2.1. The second column of the table gives the number of proteins predicted by each method and the third column gives the average accuracy ( $Q_3$ ) over all proteins for the respective method. The results in Table 2.1 cannot be used for comparison, since different sets of proteins are used for each method. In Table 2.2, six methods are compared on their largest common subset of 80 proteins. In Table 2.2, also SOV and per-state accuracy measures  $C_H$ ,  $C_E$ ,  $C_L$  are included.

**Table 2.1** Average prediction accuracy (third column) for each secondary structure server monitored in EVAsec (data updated at April 2009). Different sets of proteins are used for each method (the number of proteins used is given in the second column)

Method	Num. proteins	$Q_3$
APSSP2 [39]	122	75.5
PHDpsi [37]	229	75.0
Porter [33]	73	80.0
PROF_king [30]	230	72.1
PROFsec [40]	232	76.6
PSIpred [21]	224	77.9
SABLE [36]	232	76.1
SABLE2 [36]	159	76.8
SAM-T99sec [24]	204	77.3
SCRATCH (SSpro3) [34]	207	76.2
SSpro4 [34]	144	77.9
Yaspin [27]	157	73.6

**Table 2.2** Performance comparison of six secondary structure prediction methods on their largest common subset of 80 proteins as evaluated in EVAsec (data updated at April 2009). The average of three different accuracy measures  $\pm$  standard deviation are given:  $Q_3$  (see (2.1)), SOV (see (2.3)) and  $C_H$ ,  $C_E$ ,  $C_L$  (see (2.2)). The first column of the table gives the rank of the corresponding predictor

Rank	Method	$Q_3$	SOV	$C_H$	$C_E$	$C_L$
1	PROFsec	$75.5 \pm 1.4$	$74.9 \pm 1.9$	$0.65 \pm 0.03$	$0.70 \pm 0.04$	$0.56 \pm 0.02$
	PSIpred	$76.8 \pm 1.4$	$75.4 \pm 2.0$	$0.67 \pm 0.03$	$0.73 \pm 0.04$	$0.55 \pm 0.02$
	SAM-T99sec	$77.2 \pm 1.2$	$74.6 \pm 1.5$	$0.67 \pm 0.03$	$0.71 \pm 0.03$	$0.59 \pm 0.02$
2	PHDpsi	$73.4 \pm 1.4$	$69.5 \pm 1.9$	$0.64 \pm 0.03$	$0.68 \pm 0.04$	$0.52 \pm 0.02$
3	PROF_king	$71.6 \pm 1.5$	$67.7 \pm 2.0$	$0.62 \pm 0.03$	$0.68 \pm 0.04$	$0.51 \pm 0.02$

Most of the 13 methods are based on NN. The exceptions are PORTER, SCRATCH, SSPro4 (based on bidirectional recurrent NN), SAM-T99sec (based on HMM) and Yaspin (based both on NN and HMM).

### 2.4.2 Secondary Structure Prediction Methods

In this section, we describe in detail two of the most famous secondary structure prediction methods: PHD<sup>6</sup> and PSIPred.<sup>7</sup> Both methods are based on NN and share similar network topology. The main difference between the two methods is the way evolutionary information is extracted from MSA and encoded into the NN input. Early version of PHD used HSSP pre-computed multiple alignments generated by MAXHOM. PSIPred uses the position-specific scoring matrix (PSSM) internally computed by PSI-BLAST. As discussed in [41], the improvement of PSIPred with respect to PHD is mostly due to the better alignments used to feed the NN. The better quality of the alignments is in part due to the growth of the databases and the filtering strategy used by Jones to avoid pollution of the profile through unrelated proteins. A more recent version of PHD uses PSSM input and it is called PHDpsi to distinguish it from the older implementation. The only difference between PHD and PHDpsi is the use of PSSM input instead of frequency profile input.

Also for all the other secondary structure predictors, the main source of information is the sequence profile or the PSSM. The main difference between the different approaches relies on the technique used to extract knowledge from these two sources of information. The particular technique is specific to the machine learning method used. Here we decided to describe only PHD and PSIPred because, historically, they represent the two most important step-forward in secondary structure prediction.

#### 2.4.2.1 PHD

PHD has been described in [42]. The PHD method processes the input information in two different levels, corresponding to two different neural networks: (1) *sequence-to-structure NN* and (2) *structure-to-structure NN*. The final prediction is obtained by filtering the solution obtained from consensus between differently trained neural networks (3).

1. At the first level, the input units of the NN encode local information taken from sequence profiles (from PSSM in PHDpsi). For each residue position  $i$ , the local information is extracted from a window of 13 adjacent residues centered in  $i$ . For each residue position in the window, 22 input units are used: 20 units encode the corresponding column in the sequence profile, 1 unit is used to detect

---

<sup>6</sup> <http://www.predictprotein.org/>

<sup>7</sup> <http://bioinf.cs.ucl.ac.uk/psipred/>

when the position is outside the N/C-terminal region (1 if outside and 0 if not) and 1 unit accounts for the conservation weight at that position (see below for definition). The output of the first level NN consists of three nodes, one for each possible secondary structure element helix/strand/coil, corresponding to the state of the central residue in the window. The first level NN classifies (13-residues long) protein segments according to the secondary structure class of their central residue. This classification does not reflect the fact that different segments can be correlated, being, for example, consecutive and overlapping in the protein sequence. Particularly, at this level, the NN has no knowledge of the correlation between secondary structure elements. For example, it has no way to know that a helix consists of at least three consecutive elements.

2. The second level is introduced to take into account the correlation between consecutive secondary structure elements. The input of the second level NN is compiled from the output of the first level NN. For every residue position, the input unit encodes a window of 17 consecutive elements taken from the secondary structure prediction of the first NN. Every position in the window is encoded with 5 units: three for the predicted secondary structure, one to detect whether the position is outside the boundaries of the protein and one for the conservation weight. The output is set as in the first NN and, also in this case, corresponds to the state of the central residue in the window.
3. The consensus is a simple arithmetic average over (typically four) differently trained networks. The highest value of the three output units is taken as the final prediction. To every such prediction, a reliability index can be associated with the following formula

$$RI = \lceil 10 \cdot (o_1 - o_2) \rceil, \quad (2.4)$$

where  $o_1$  and  $o_2$  are the highest and the second highest values in the output vector, respectively. The prediction obtained is finally filtered (with the help of the reliability index) in order to fix some eventually unrealistic local predictions that neither the second level NN nor the consensus were able to detect (particularly, too short alpha-helix segments).

The conservation weight provides a score for positions in the MSA with respect to their level of conservation: the more conserved is a position the higher is the conservation weight score. Such a weight is contained in the HSSP database and it is defined by

$$CW_i = \frac{\sum_{r,s=1}^N w_{rs} \cdot \text{sim}_{rs}^i}{\sum_{r,s=1}^N w_{rs}} \quad (2.5)$$

with

$$w_{rs} = 1 - \frac{1}{100} \cdot \text{ident}_{rs},$$

where  $N$  is the number of sequences in the multiple alignment,  $\text{ident}_{rs}$  is the percentage of sequence identity (over the entire length) of sequences  $r, s$  and  $\text{sim}_{rs}^i$  is the value of the similarity between sequences  $r, s$  at position  $i$  according to the Dayhoff similarity matrix [8].

### 2.4.2.2 PSIPred

PSIPred has been described in [21]. The original implementation is based on neural networks. An almost equivalent implementation with SVM has been described in [48] and compared with the original version.

The neural network topology of PSIPred is very similar to the one used in PHD: in both methods the input is processed in two different levels, and the final result is obtained as the consensus between differently trained networks. The main differences are the lengths of the windows used in the first and second levels: in both networks PSIPred uses 15-residue long windows, while PHD uses lengths 13 and 17, respectively. Moreover, the conservation weight is not included in the input of PSIPred (it showed poor improvement also in PHD [42]). The most important difference between early PHD version and PSIPred is the way evolutionary information is treated. In particular, the position-specific scoring matrix (PSSM) is used to feed the NN instead of the classical frequency profile computed from MSA.

Here we review in detail the procedure used by Jones to produce meaningful position-specific profiles with PSI-BLAST, as described in [21]. Although PSI-BLAST is much more sensitive than BLAST in picking up distant evolutionary relationships, it must be used carefully in order to avoid false-positive matches. In particular, PSI-BLAST is very prone to incorporate repetitive sequences into the intermediate profiles. When this happens, the searching process tends to find highly scored matches with completely random sequences. In order to maximise the performances of PSI-BLAST, Jones builds a custom sequence data bank by first compiling a large set of non-redundant protein sequences and then by filtering the databank in order to remove low complexity regions [49], transmembrane segments [22] and regions which are likely to form coiled-coil regions (these filtering are now automatically performed by PSI-BLAST).

Finally, the input of the NN is computed from the PSSM of PSI-BLAST after three iterations, scaled to values between 0 and 1 with the logistic function  $1/(1 + e^x)$ , where  $x$  is the raw profile value.

## 2.5 Residue–Residue Contact Prediction

Residue–residue contact prediction refers to the prediction of the probability that two residues in a protein structure are spatially close to each other. Inter-residue contacts provide much information about the protein structure. A contact between two residues that are distant in the protein sequence can be seen as a strong constraint on the protein fold. If we could predict with high precision even a small set of (non-trivial) residue pairs in contact, we could use this information as extra constraints to guide the protein structure prediction. The prediction of inter-residue contact is a difficult problem, and no satisfactory improvements have been achieved in the last 10 years of investigation. On the other end, even if residue contact predictors are highly inaccurate, they still have higher accuracy compared to contact predictions derived from the best 3D structure prediction methods [45].

In the following sections, we describe the standards adopted for contact definition and contact prediction evaluation (Sect. 2.5.1). We next describe the most important statistics used to extract contact information from MSA (Sect. 2.5.2) and the best performing contact predictors, as evaluated in the last five CASP editions (Sect. 2.5.3).

### 2.5.1 *EVAcon: Evaluation of Inter-Residue Contact Prediction Servers*

Equivalently to EVAsec, the objectives of EVAcon<sup>8</sup> are to provide a continuous, fully automated and statistically significant analysis of inter-residue contact prediction servers. Differently from EVAsec, the statistics of EVAcon are not so frequently updated and only very few servers are monitored at the moment. Anyway, EVAcon provides the standards for contact definition and evaluation criteria for contact prediction. These measures are also those adopted at CASP meetings.

There are several ways to define inter-residue contacts; all definitions are more or less equivalent. In EVAsec, two residues are defined to be in contact if the Euclidean distance between the coordinates of their *beta carbon atoms* ( $C_\beta$ ) is  $\leq 8$  Angstroms ( $\text{\AA}$ ) (Fig. 2.3). For Glycines, the coordinate of the *alpha carbon atom* ( $C_\alpha$ ) is considered instead of the  $C_\beta$  coordinate, which is missing (i.e., Glycines have only a unique carbon atom).

The most important measure for the evaluation of contact predictors is the accuracy of prediction. The accuracy of prediction is defined as

$$\frac{\text{Number of correctly predicted contacts}}{\text{Number of predicted contacts}}$$

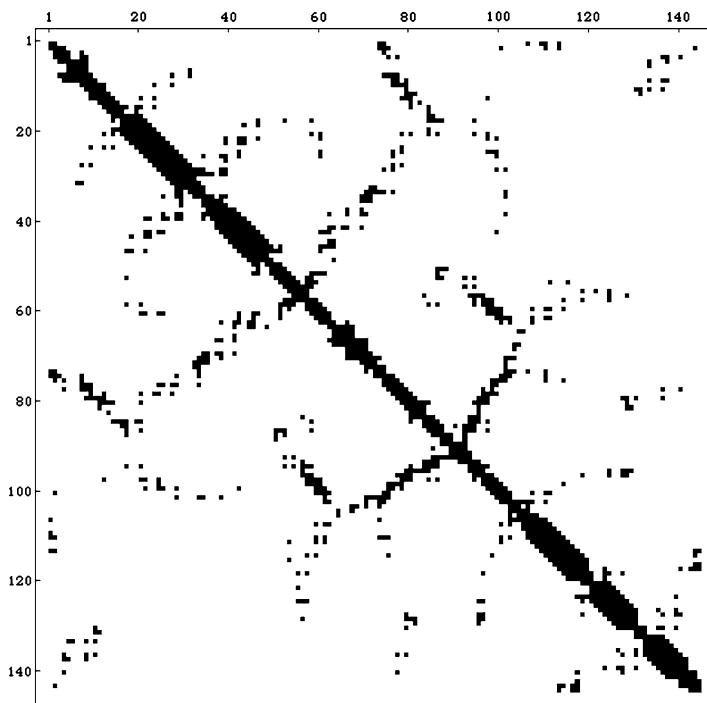
Since contact predictors usually return the probability that two residues are in contact, the above formula is computed in slightly different way: the list of residue pairs is sorted in decreasing order according to the predicted contact probability, and the accuracy is computed by taking the first  $2L$ ,  $L$ ,  $L/2$ ,  $L/5$  and  $L/10$  (most probable) pairs, where  $L$  here denotes the length of the protein. More formally, the accuracy of prediction with respect to length  $l \in \{2L, L, L/2, L/5, L/10\}$  is defined as

$$Acc_l = \frac{nc_l}{l}, \quad (2.6)$$

where  $nc_l$  is the number of correctly predicted contacts among the first  $l$  high-scored pairs. It makes sense to distinguish between short-range contacts (i.e., contacts between residues that are close in the protein sequence) and long-range contacts (i.e., contacts between residues that are distant in the sequence). Long-range contacts are much more sparse than short-range contacts, but they provide much more

---

<sup>8</sup> [http://cubic.bioc.columbia.edu/eva/doc/intro\\_con.html](http://cubic.bioc.columbia.edu/eva/doc/intro_con.html)



**Fig. 2.3** Map of the  $C_\beta$ – $C_\beta$  contacts at threshold 8 Å of the protein 1a3aA of Fig. 2.2. Black dots represent contacts

information about the protein structure and for this they are much more difficult to predict correctly. For this reason, for the calculation of the accuracy with (2.6), the predicted pairs of residues are split in three sets according to the separation of the two residues in the pair (i.e., the number of residues between them): short-range (from 6 to 11), medium-range (from 12 to 23) and long-range ( $\geq 24$ ) sequence separation. Residue contacts whose sequence separation is below 6 do not provide useful information about the protein folding and are not evaluated. Among all these measures, the most important evaluation parameter is the accuracy for sequence separation  $\geq 24$  and L/5 number of pairs.

Since the performances of contact predictors are not monitored as extensively as secondary prediction servers, the statistics about the state-of-the-art accuracy of inter-residue contact prediction is not very significant. According to the results obtained in the last two CASP events [11, 20], we can evaluate the state of the art in prediction accuracy as few percentage points above the 20% for sequence separation  $\geq 24$  and L/5 number of contacts.

### 2.5.2 Contact Prediction with Correlated Mutations

The most simple approach to predict residue–residue contacts in a protein is based on the evaluation of statistical properties derived from paired-columns of the MSA.



This approach relies on a very simple but significant hypothesis: since evolutionary mutations tend to preserve more protein structures than its sequence, residue substitutions must be compensated by other mutations in the spatially close neighbours in order to not destabilise the protein fold. That is, during the evolutionary course of protein sequences, a pair of residues in contact are more likely to co-mutate than residues not in contact. This basic idea has been exploited in the reverse direction: the probability of two residues to be in contact can be inferred by measuring how much changes in one column of the MSA (corresponding to one of the two residues) affect changes in the other column. There are various measures which can be used to extract correlated mutation statistics from the MSA. Despite the intensive investigation of correlated mutation-based methods, this approach alone resulted in a limited success in predicting residue–residue contacts. A possible explanation of these performances can be that the statistical measures exploited so far are too weak to discriminate between true correlation and background noise. Nevertheless, these methods are still interesting on their own, and they have been often used proficiently in conjunction with machine learning approaches for the contact prediction problem.

In the following sections, we shortly describe just few of the statistical measures used to evaluate correlated mutations; more detailed information can be found in [14, 19].

### 2.5.2.1 Pearson Correlation

The best known implementation of the correlated mutation approach [17] uses the Pearson correlation coefficients to quantify the amount of co-evolution between pair of sites.

The Pearson product-moment correlation coefficient is a measure of the linear dependence between two variables  $X, Y$ , and it is defined as

$$C(X, Y) = \frac{1}{N} \sum_{k=1}^N \frac{(X_k - \bar{X})(Y_k - \bar{Y})}{\sigma_X \sigma_Y}, \quad (2.7)$$

where  $N$  is the number of elements contained in  $X$  and  $Y$ ,  $\bar{X}$  is the average of  $X$  and  $\sigma_X$  is its standard deviation. The coefficient  $-1 \leq C(X, Y) \leq 1$  quantifies the degree of linear dependence between  $X$  and  $Y$ . If  $C(X, Y) = 1$ , then  $X$  is equal to  $Y$  up to a linear transformation. In general, if  $C(X, Y) \sim 1$ ,  $X$  and  $Y$  are considered positively correlated, not correlated if  $C(X, Y) \sim 0$  and anti-correlated if  $C(X, Y) \sim -1$ .

To evaluate the Pearson correlation of a pair of sites (columns)  $i, j$  in an MSA, we have to define two substitution score vectors. Assuming that the MSA matrix  $M$  contains  $t$  aligned sequences, the substitution vector corresponding to position  $i$  is defined as

$$X = S_i = (\delta(M_{1i}, M_{2i}), \delta(M_{1i}, M_{3i}), \dots, \delta(M_{1i}, M_{ti}), \delta(M_{2i}, M_{3i}), \dots, \delta(M_{t-1i}, M_{ti})),$$

where  $\delta(M_{ki}, M_{li})$  is the score assigned to the substitution (mutation)  $M_{ki} \rightarrow M_{li}$ . The substitution vector  $Y = S_j$  corresponding to position  $j$  is computed in the same way. The substitutions with gaps are not considered; hence, if  $M_{ki} \rightarrow M_{li}$  is a gap-substitution then it is excluded from  $S_i$  and  $M_{kj} \rightarrow M_{lj}$  is also excluded from  $S_j$  (the conversely holds for position  $j$ ). The coefficient  $C(S_i, S_j)$  quantifies the degree of linear correlation for the evolutionary mutations as observed at the  $i$ th column of the MSA with respect to the mutations occurring at the  $j$ th column. Perfectly conserved columns and columns with more than 10% of gaps are usually excluded from the analysis since they are uninformative.

This approach requires a similarity matrix to weight residue substitutions  $M_{ki} \rightarrow M_{li}$ : that is, the substitution vector is defined in terms of a scoring scheme  $\delta(M_{li}, M_{ki})$ . The substitution scores are generally provided by the McLachlan similarity matrix [28], which defines residue similarity in terms of their physico-chemical properties. The choice to use the McLachlan is not critical since there are several different similarity matrices that perform equally well [9]. Other related implementations of this method have been proposed (a comprehensive review can be found in [35]). These approaches differ from the original method [17] essentially in the measures adopted to weight the co-evolving substitutions.

### 2.5.2.2 Mutual Information

The mutual information measures the mutual dependence of two variables  $X, Y$ . It is defined as

$$I(X, Y) = \sum_{x \in X} \sum_{y \in Y} p(x, y) \log \frac{p(x, y)}{p_1(x)p_2(y)}, \quad (2.8)$$

where  $p_1(x)$  is the marginal probability distribution of  $x$  in  $X$ ,  $p_2(y)$  is the marginal probability distribution of  $y$  in  $Y$  and  $p(x, y)$  is the joint probability of  $x, y$ , i.e., the probability that  $x$  and  $y$  occur in conjunction. The mutual information is  $I(X, Y) = 0$  if and only if  $X$  and  $Y$  are independent.

To evaluate the mutual information of two columns  $i, j$  of the MSA, we have to compute the marginal probabilities of residues occurring in each respective column and their joint probability. The variable  $X$  contains the different residues occurring in the  $i$ th column of the MSA, and  $p_1(x), x \in X$  is the probability of residue  $x$  of being in the  $i$ th column, i.e.,  $p_1(x)$  is the frequency of residue  $x$  in the  $i$ th column of the MSA. The marginal probabilities of column  $j$  are computed in the same way. The joint probability  $p(x, y), x \in X, y \in Y$  is the frequency of the pair  $x, y$  in the columns  $i, j$  of the MSA. In order to compute the mutual information of two positions  $i, j$ , the MSA is filtered from sequences containing a gap in position  $i$  or  $j$ . Note that, if either position  $i$  or  $j$  are perfectly conserved in the MSA, their mutual information reduces to 0.

A comparison in terms of prediction accuracy between Pearson correlation and mutual information has been analysed in [14]. According to this analysis, mutual information shows poor performances in contact prediction. Nevertheless, a more deep analysis described in [45] shows that the significance of the observed mutual information results in a much more strong measure for correlated mutations.

### 2.5.2.3 Joint Entropy

The joint entropy is a measure of how much entropy (variation) is contained in two variables  $X, Y$ . It is defined as

$$J(X, Y) = - \sum_{x \in X, y \in Y} p(x, y) \log p(x, y), \quad (2.9)$$

where  $p(x, y)$  is the joint probability of  $x, y$ .

To compute the joint entropy of a pair of columns in the MSA,  $X, Y$  and  $p(x, y)$  are defined as in Sect. 2.5.2.2. Note that for perfectly conserved positions, the joint entropy reduces to 0. Highly conserved positions are more likely to correspond to residues buried in the core of the protein, which is the most stable portion of a protein structure and thus less subjected to evolutionary mutations. Most of the residue–residue contacts are, in fact, localised in the core of a protein structure. Therefore, residue pairs with lower joint entropy are more likely to be in contact than pairs with higher entropy. In this sense, joint entropy is complementary to Pearson correlation and mutual information, which cannot extract information from highly conserved columns of the MSA (recall that perfectly conserved position are excluded from the Pearson analysis and have mutual information equal to 0).

### 2.5.3 Contact Prediction with Neural Networks

We describe the best performing NN contact predictors CORNET,<sup>9</sup> PROFcon,<sup>10</sup> and SAM-T06con,<sup>11</sup> as evaluated in the last five editions of CASP experiments (from CASP4 in 2000 to CASP8 in 2008).

All best known implementations of NN contact predictors have some common similarities. First of all, due to the high variability of protein lengths, the NN input cannot be set in order to directly encode the overall protein sequence. For this reason, the NN input encodes specific information related to pair of residues and only coarse-grained global features of the protein are taken into account. This information is usually derived from the MSA and from structural/statistical properties of the protein. We can identify three different kinds of information used in NN input units:

- *Local information*, derived from the respective local environments of the two residues;
- *Global information*, derived from the overall protein structure and/or sequence;
- *Paired-residue statistics*, which include statistical properties derived from paired columns of the MSA.

<sup>9</sup> [http://gpcr.biocomp.unibo.it/cgi/predictors/cor-net/pred\\_cmap.cgi](http://gpcr.biocomp.unibo.it/cgi/predictors/cor-net/pred_cmap.cgi)

<sup>10</sup> <http://cubic.bioc.columbia.edu/services/profcon/>

<sup>11</sup> [http://compbio.soe.ucsc.edu/SAM\\_T06/T06-query.html](http://compbio.soe.ucsc.edu/SAM_T06/T06-query.html)

All NN predictors described here differ essentially only by the features chosen to capture these three different kinds of information.

The output layer of the NN contains a unique node, which during the training phase is set to 1 if the two residues are in contact and to 0 if they are not. Accordingly, in the prediction phase the output of the NN (a value between 0 and 1) is interpreted as the probability that the two input residues are in contact.

In order to filter most of the (uninformative) data related to local contacts, the set of training examples is computed only from residue pairs whose sequence separation is larger than some threshold, typically  $>6$ . Moreover, to avoid the over-estimation of non-contacts (which are much more abundant than contacts), the training examples are usually balanced. The balancing is generally obtained by randomly selecting only a fraction of the negative examples (typically 5%) from each epoch of the training phase. This technique has the effect from speeding up the learning process and assures that most of the negative examples are seen by the NN.

The performances and the limits of NN predictors are strictly related to their input encodings. Different from the secondary structure prediction problem, the contact probability is not a property that can be inferred locally, since it is a consequence of repulsive and attractive inter-atomic forces over all the protein sequence. Due to the limit imposed by the different protein lengths, the NN predictors are forced to infer global information about the protein structure mostly from local information only. This is probably the main reason why residue–residue contact prediction is not as successful as secondary structure prediction. Nevertheless, the NN-based approaches are actually the state of the art in residue–residue contact prediction and they provide much better performances than the contact prediction derived from tertiary structure modeling (for free-modeling domains).

In the following sections, for each NN predictor, we focus on the specific encoding of the input information. More detailed description of the implementations together with the analysis of their performances can be found elsewhere [13, 38, 45].

### 2.5.3.1 CORNET

The implementation of CORNET and its performances have been described in [12, 13].

In total, the input encodings requires 1,071 units. Most of the NN input encodes paired-residue statistics. For each residue pair  $i, j$  ( $j > i + 6$ ) in the protein sequence, the NN input encodes local information (a) in terms of sequence conservation of positions  $i, j$  and in terms of predicted secondary structure of their immediate neighbours, i.e., the two windows  $[i - 1, i + 1]$  and  $[j - 1, j + 1]$  are considered. Two distinct paired-residue statistics are used (b): Pearson correlated mutations and paired evolutionary information as observed in the two neighbouring windows. No global information is taken into account.

- a. For each position in the two neighbouring windows three input units encode the secondary structure information (alpha/beta/coil). If the secondary structure in

one position is predicted as alpha, then the corresponding entry in the input unit is 1 and the remaining two entries are set to 0. The same holds for the other secondary structure elements. When the neighbouring window is outside the boundaries of protein, all entries of the secondary structure input units are set to 0. The sequence variability, as computed in [15], is included only for positions  $i$  and  $j$  (2 units).

- b. The evolutionary information for the pair  $i, j$  is encoded as an input vector containing  $210 = 20 \cdot (20 + 1)/2$  elements, one entry for each distinct pair of aminoacids (symmetric pairs are considered equivalent). Every entry of the vector contains the frequency of the occurrence of the related pair of amino-acids in the multiple alignment with respect to positions  $i, j$ . The evolutionary information of the neighbours of  $i, j$  is also taken into account. The positions considered to introduce the evolutionary information are  $(i - 1, j - 1)$ ,  $(i + 1, j + 1)$  (parallel pairings) and  $(i - 1, j + 1)$ ,  $(i + 1, j - 1)$  (anti-parallel pairings). The correlated mutation information (1 unit) is defined as described in (2.7). For perfectly conserved positions, the correlation between  $i$  and  $j$  is set by default to 0 and for positions with more than 10% of gaps to  $-1$ .

### 2.5.3.2 PROFcon

The implementation of PROFcon and its performances have been described in [38].

In total, the input encodings require 738 units. For every pair of residues  $i, j$ , the neural network input incorporates local information from the neighbours of  $i, j$  and from their connecting segment (a). Several global properties of the protein are taken into account (b) but not the paired-residue statistics.

- (a) The local information of the two residues is derived from two windows of width nine centered in  $i, j$  and from the segment connecting  $i, j$ . The connecting segment information is captured by taking a window of five consecutive residues from  $k - 2$  to  $k + 2$  where  $k = \lceil i - j \rceil$ . Each residue position in the three windows is described by the frequency of occurrence of the 20 amino acid types in that position (20 input units plus 1 more unit to detect when the position is outside the boundaries of the protein), predicted secondary structure (4 units, helix/strand/coil and reliability of the prediction at that position as defined in (2.4)), predicted solvent accessibility (3 units, buried/exposed and prediction reliability) and conservation weight (1 unit) as defined in (2.5). Some more features are introduced to better characterise the biophysical properties of the pair  $i, j$  (7 input units: hydrophobic-hydrophobic, polar-polar, charged-polar, opposite charges, same charges, aromatic-aromatic, other) and if they are in low-complexity regions, as computed by the SEG software (2 input units). Global features of the entire connecting segment are also considered: amino acid composition (20 units), secondary structure composition (4 units) and the fraction of SEG-low-complexity residues in the whole connecting segment

(1 node). Finally, the length of the segment connecting  $i$  and  $j$  is encoded in 11 input units corresponding to sequence separations 6, 7, 8, 9, 10–14, 15–19, 20–24, 25–29, 30–39, 40–49, >49.

- (b) This global information includes amino acid composition of the entire protein (20 units), secondary structure composition (3 units) and protein length (4 units, lengths 1–60, 61–120, 121–240 and >240).

### 2.5.3.3 SAM-T06con

This NN contact predictor is included in the protein structure prediction architecture SAM-T06. The implementation of the contact predictor and its performances have been described in [45].

In total, the input encoding of the NN requires 449 units. The local information (a) is accounted by taking a windows of length five centered in each one of the two residues. Four distinct paired-residue statistics are used (b) and just the length of the protein is taken into account as global information (c).

- (a) For each position in the two windows, the NN input encodes the amino acids distribution according to a Dirichlet mixture regularizer [46] (20 units), the predicted secondary structure and predicted burial [25] (13 and 11 units, respectively). Moreover, the entropy of the amino acids distribution (1 unit for each window) and the logarithm of the sequence separation between the two residues (1 unit) are included.
- (b) The NN input encodes four paired-residue statistics (1 input unit for three of them and 2 for the last one). The most simple statistics counts the number of different pairs observed in the MSA columns corresponding to the two residues. Other statistics considered are the joint entropy (2.9), the propensity of contact, and a mutual information-based statistics (2.8). For these three last measures, the logarithm of the rank of the statistic's value is taken into the input, except for the mutual information for which both the logarithm of the rank and the exact value are added. The rank of a statistic value is computed as the rank of the value in the list of values for all pairs of columns.

The propensity for two residue to be in contact is the log odds of a contact between the residues vs. the probability of the residues occurring independently. This measure has been slightly modified in order to give more weight to high-separation with respect to low-separation contacts. Here the mutual information statistics is introduced by computing its p-value (i.e., the probability of seeing the observed mutual information by chance). The significance of the mutual information shows better performances in contact prediction than the statistics itself, as computed in (2.8). More detailed information about the propensity of contact and the mutual information-based statistics can be found in [45].

- (c) The only global information added is the logarithm of the length of the protein (1 unit).

## 2.6 Conclusions

In this chapter, we presented two different aspects of the protein structure prediction problem: the prediction of protein secondary structure, which is simpler in its formulation than the protein folding problem and from which sequential annotations can be derived, and the most demanding problem of residue contact prediction in proteins. The first relevant message from our analysis of the current state-of-the-art methods is that a key-role is played by evolutionary information. This knowledge, which can be exploited by using different multiple sequence alignment methods, is one of the major resources to identify relevant domains of the protein that are related to secondary structure elements or packing regions. A second relevant message is that the most successful predictors are based on machine-learning tools, indicating that for the described tasks (at least up-to-now) bottom-up approaches compete favorably with the methods that directly predict the 3D structure of the proteins.

## References

1. Aloy, P., Stark, A., Hadley, C., Russell, R.B.: Predictions without templates: new folds, secondary structure, and contacts in CASP5. *Proteins* **53**, 436–456 (2003)
2. Altschul, S.F., Madden, T.L., Schaffer, A.A., Zhang, J., Zhang, Z., Miller, W., Lipman, D.J.: Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res.* **25**, 3389–3402 (1997)
3. Bartoli, L., Capriotti, E., Fariselli, P., Martelli, P.L., Casadio, R.: The pros and cons of predicting protein contact maps. *Methods Mol Biol.* **413**, 199–217 (2008)
4. Benner, S.A., Gerloff, D.: Patterns of divergence in homologous proteins as indicators of secondary and tertiary structure: a prediction of the structure of the catalytic domain of protein kinases. *Adv. Enzyme Regul.* **31**, 121–181 (1991)
5. Bishop, C.M.: *Pattern Recognition and Machine Learning*. Springer, Heidelberg (2007)
6. Chou, P.Y., Fasman, G.D.: Conformational parameters for amino acids in helical, beta-sheet, and random coil regions calculated from proteins. *Biochemistry* **13**, 211–222 (1974)
7. Cozzetto, D., Tramontano, A.: Advances and pitfalls in protein structure prediction. *Curr Protein Pept Sci.* **9**, 567–577 (2008)
8. Dayhoff, M.O.: *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Washington DC (1978)
9. Di Lena, P., Fariselli, P., Margara, L., Vassura, M., Casadio, R.: On the Upper Bound of the Prediction Accuracy of Residue Contacts in Proteins with Correlated Mutations: The Case Study of the Similarity Matrices. *Lecture Notes in Computer Science* 5488, 210–221 (2009)
10. Edgar, R.C.: MUSCLE: multiple sequence alignment with high accuracy and high throughput. *Nucleic Acids Res.* **32**, 1792–1797 (2004)
11. Ezkurdia, I., Graña, O., Izarzugaza, J.M., Tress, M.L.: Assessment of domain boundary predictions and the prediction of intramolecular contacts in CASP8. *Proteins* **77**, 196–209 (2009)
12. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Prediction of contact maps with neural networks and correlated mutations. *Protein Eng.* **14**, 835–843 (2001)
13. Fariselli, P., Olmea, O., Valencia, A., Casadio, R.: Progress in predicting inter-residue contacts of proteins with neural networks and correlated mutations. *Proteins* **5**, 157–162 (2001)
14. Fodor, A.A., Aldrich, R.W.: Influence of conservation on calculations of amino acid covariance in multiple sequence alignments. *Proteins* **56**, 211–221 (2004)



15. Garcia-Boronat, M., Diez-Rivero, C.M., Reinherz, E.L., Reche, P.A.: PVS: a web server for protein sequence variability analysis tuned to facilitate conserved epitope discovery. *Nucleic Acids Res.* **36**, 35–41 (2008)
16. Garnier, J., Osguthorpe, D.J., Robson, B.: Analysis of the accuracy and implications of simple methods for predicting the secondary structure of globular proteins. *J. Mol. Biol.* **120**, 97–120 (1978)
17. Göbel, U., Sander, C., Schneider, R., Valencia, A.: Correlated mutations and residue contacts in proteins. *Proteins* **18**, 309–317 (1994)
18. Graña, O., Baker, D., MacCallum, R.M., Meiler, J., Punta, M., Rost, B., Tress, M.L., Valencia, A.: CASP6 assessment of contact prediction. *Proteins* **61**, 214–224 (2005)
19. Horner, D.S., Pirovano, W., Pesole, G.: Correlated substitution analysis and the prediction of amino acid structural contacts. *Brief. Bioinform.* **9**, 46–56 (2008)
20. Izarzugaza, J.M., Graña, O., Tress, M.L., Valencia, A., Clarke, N.D.: Assessment of intramolecular contact predictions for CASP7. *Proteins* **69**, 152–158 (2007)
21. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.* **292**, 195–202 (1999)
22. Jones, D.T., Taylor, W.R., Thornton, J.M.: A model recognition approach to the prediction of all-helical membrane protein structure and topology. *Biochemistry* **33**, 3038–3049 (1994)
23. Kabsch, W., Sander, C.: Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers* **22**, 2577–2637 (1983)
24. Karplus, K., Barrett, C., Hughey, R.: Hidden Markov models for detecting remote protein homologies. *Bioinformatics* **14**, 846–856 (1998)
25. Karplus, K., Katzman, S., Shackleford, G., Koeva, M., Draper, J., Barnes, B., Soriano, M., Hughey, R.: SAM-T04: what is new in protein-structure prediction for CASP6. *Proteins* **61**, 135–142 (2005)
26. Lesk, A.: *Introduction to Bioinformatics*. Oxford University Press, London (2006)
27. Lin, K., Simossis, V.A., Taylor, W.R., Heringa, J.: A simple and fast secondary structure prediction method using hidden neural networks. *Bioinformatics* **21**, 152–159 (2005)
28. McLachlan, A.D.: Tests for comparing related amino-acid sequences. Cytochrome c and cytochrome c 551. *J. Mol. Biol.* **61**, 409–424 (1971)
29. Notredame, C., Higgins, D.G., Heringa, J.: T-Coffee: A novel method for fast and accurate multiple sequence alignment. *J Mol Biol.* **302**, 205–217 (2000)
30. Ouali, M., King, R.D.: Cascaded multiple classifiers for secondary structure prediction. *Protein Sci.* **9**, 1162–1176 (2000)
31. Pauling, L., Corey, R.B.: Configurations of polypeptide chains with favored orientations around single bonds: two new pleated sheets. *Proc. Natl. Acad. Sci. USA* **37**, 729–740 (1951)
32. Pauling, L., Corey, R.B., Branson, H.R.: The structure of proteins: two hydrogen-bonded helical configurations of the polypeptide chain. *Proc. Natl. Acad. Sci. USA* **37**, 205–211 (1951)
33. Pollastri, G., McLysaght, A.: Porter: a new, accurate server for protein secondary structure prediction. *Bioinformatics* **21**, 1719–1720 (2005)
34. Pollastri, G., Przybylski, D., Rost, B., Baldi, P.: Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins* **47**, 228–235 (2002)
35. Pollock, D.D., Taylor, W.R.: Effectiveness of correlation analysis in identifying protein residues undergoing correlated evolution. *Protein* **10**, 647–657 (1997)
36. Porollo, A., Adamczak, R., Wagner, M., Meller, J.: Maximum Feasibility Approach for Consensus Classifiers: Applications to Protein Structure Prediction. In *proceedings of CIRAS 2003*
37. Przybylski, D., Rost, B.: Alignments grow, secondary structure prediction improves. *Proteins* **46**, 197–205 (2002)
38. Punta, M., Rost, B.: PROFcon: novel prediction of long-range contacts. *Bioinformatics* **21**, 2960–2968 (2005)
39. Raghava, G.P.S.: APSSP2: A combination method for protein secondary structure prediction based on neural network and example based learning. *CASP5 A-132* (2002)
40. Rost, B.: <http://cubic.bioc.columbia.edu/predictprotein>

41. Rost, B.: Rising accuracy of protein secondary structure prediction. In: Chasman D (ed.) Protein structure determination, analysis, and modeling for drug discovery, pp. 207–249. Dekker, New York (2003)
42. Rost, B., Sander, C.: Prediction of protein secondary structure at better than 70% accuracy. *J. Mol. Biol.* **232**, 584–599 (1993)
43. Rost, B., Sander, C.: Third generation prediction of secondary structures. *Methods Mol. Biol.* **143**, 71–95 (2000)
44. Sander, C., Schneider, R.: Database of homology-derived protein structures and the structural meaning of sequence alignment. *Proteins* **9**, 56–68 (1991)
45. Shackelford, G., Karplus, K.: Contact prediction using mutual information and neural nets. *Proteins* **69**, 159–164 (2007)
46. Sjölander, K., Karplus, K., Brown, M., Hughey, R., Krogh, A., Mian, I.S., Haussler, D.: Dirichlet mixtures: a method for improved detection of weak but significant protein sequence homology. *Comput. Appl. Biosci.* **12**, 327–345 (1996)
47. Thompson, J.D., Higgins, D.G., Gibson, T.J.: CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res.* **22**, 4673–4680 (1994)
48. Ward, J.J., McGuffin, L.J., Buxton, B.F., Jones, D.T.: Secondary structure prediction with support vector machines. *Bioinformatics* **19**, 1650–1655 (2003)
49. Wootton, J.C., Federhen, S.: Statistics of local complexity in amino acid sequences and sequence databases. *Comput. Chem.* **17**, 149–163 (1996)

Mathematical Approaches to Polymer Sequence  
Analysis and Related Problems

Bruni, R. (Ed.)

2011, X, 248 p., Hardcover

ISBN: 978-1-4419-6799-2