

Protein Data Condensation for Effective Quaternary Structure Classification

Fabrizio Angiulli¹, Valeria Fionda², and Simona E. Rombo¹

¹ DEIS - Università della Calabria
Via P. Bucci 41C
87036 Rende (CS), Italy
{f.angiulli,simona.rombo}@deis.unical.it

² Dept. of Mathematics
Via P. Bucci 31B
87036 Rende (CS), Italy
fionda@si.deis.unical.it

Abstract. Many proteins are composed of two or more subunits, each associated with different polypeptide chains. The number and the arrangement of subunits forming a protein are referred to as *quaternary structure*. The quaternary structure of a protein is important, since it characterizes the biological function of the protein when it is involved in specific biological processes. Unfortunately, quaternary structures are not trivially deducible from protein amino acid sequences. In this work, we propose a protein quaternary structure classification method exploiting the functional domain composition of proteins. It is based on a nearest neighbor condensation technique in order to reduce both the portion of dataset to be stored and the number of comparisons to carry out. Our approach seems to be promising, in that it guarantees a high classification accuracy, even though it does not require the entire dataset to be analyzed. Indeed, experimental evaluations show that the method here proposed selects a small dataset portion for the classification (of the order of the 6.43%) and that it is very accurate (97.74%).

1 Introduction

Proteins are important components of the cell life, catalyzing most of the living cells reactions and controlling virtually all the cellular processes. Proteins are complex molecules composed by individual units called *amino acids*.

While an increasing number of amino acid sequences is produced and stored in public databases, the geometric conformation of a protein can be determined by slow and expensive methods (such as, for example, crystallography and NMR spectrometry). Thus, a number of computational methods have been developed in the last few years to predict and classify protein secondary, tertiary and quaternary structures [4, 10, 14, 15, 21]. In this paper, we focus on protein quaternary structures. In particular, we deal with the problem of efficiently exploiting available databases of amino acid sequences in order to predict the number of subunits of a given protein.

In the rest of this section, we first recall basic concepts concerning protein structure and then point out our contributions.

Background on protein structure. The sequence of the amino acids of a specific protein is determined by the sequence of the bases in the gene encoding for that protein. Chemical properties of the amino acids composing a protein determine its biological activity. Considering also the order in which amino acid residues lie in the chain, the amino acid sequence is called *primary structure* of the protein. It is possible to univocally identify a protein when the corresponding primary structure is known, but other information about protein spatial conformation are necessary to have a complete description of its biological functions. In particular, the *secondary structure* of a protein is represented by the sequence of the repeated patterns of its backbone folding, without any regard to the conformations of its sidechains; the most common secondary structure configurations are α -helix and β -strand [13]. The *tertiary structure* of a protein refers to the three-dimensional structure of the entire polypeptide, specified by the coordinates of each of its amino acids. Thus, both secondary and tertiary structures contribute to the geometrical description of proteins. Actually, several proteins (e.g., Hemoglobin) are a combination of two or more individual polypeptide chains. The arrangement according to which such subunits assemble is called the protein *quaternary structure*. Quaternary structure refers to the number of subunits involved in forming a protein, to their interconnections and to their organization [12, 17]. Biological processes are often influenced by the quaternary structure of proteins involved therein; e.g., the subunit construction of many enzymes provides the structural basis for the regulation of their activities. Proteins having a quaternary structure are called *oligomers*, and may be further classified as *homo-oligomers*, consisting of identical subunits, and *hetero-oligomers*, made of subunits that are different from one another. Furthermore, based on the number of subunits linked together, an homo-oligomer may be an *homodimer*, an *homotrimer*, an *homotetramer*, etc. Proteins consisting of only one subunit are called *monomers*.

Contributions. In this paper, we propose a classification method to individuate the number of subunits of each protein of a given dataset.

To this aim, we exploit protein functional domain information, as already successfully done in previous literature [21]. We encode each protein by a vector whose elements are associated to PFam domains [3]. The number of subunits included in a given protein is then obtained by assigning that protein to a class (e.g., monomers, homodimers, etc.), on the basis of a previously classified dataset and of a suitable classification method.

A few approaches have been recently introduced to support protein quaternary structure classification [5, 10, 16, 21, 22]. The most successful of them [21, 22] reach at most the 87.5% of overall accuracy, and the maximum dataset size they considered is of about 10,000 proteins. Furthermore, most of the quaternary structure classification methods proposed in the literature store and use the overall dataset, comparing each protein to be classified to each stored protein. This may result hard when large datasets are to be considered.

Our approach gives a contribution in the direction of reducing both the portion of dataset that is necessary to store and, consequently, the number of comparisons to carry out at classification time, allowing sensible space and time savings, while achieving very good accuracy figures.

In particular, we exploit nearest neighbor condensation techniques (as the one recently introduced in [1]) to replace the whole protein dataset with a notable subset that can be

then used for the sake of fast protein quaternary structure prediction. With this aim, we use a training set consistent subset for the nearest neighbor decision rule as reference dataset during classification. Let T be a dataset. Having fixed a meaningful distance metrics, a subset S of T is a training set consistent subset of T for the nearest neighbor rule, if S correctly classifies all the objects of T by means of the nearest neighbor rule. To evaluate our method, we considered two different kind of tests. First, we performed the 10-fold cross-validation on a very large protein dataset including 20,068 proteins taken from the SWISSPROT [2] database. The results confirmed the effectiveness of our approach. In fact, we scored an overall accuracy of 97.74%, by using only the 6.51% of the total dataset. This result is important, since pinpoints that our method can be adopted to correctly classify proteins whose quaternary structures are unknown, notably reducing the portion of dataset to analyze. Such a reduction is particularly attractive in the case of protein quaternary structures classification, where large datasets are often to be considered. The second kind of tests concerns the exploitation of the jackknife cross-validation on a non-redundant dataset already used to test another successful technique proposed in the literature [21]. Also in this case, the results we obtained show that our method is more powerful than the previous ones, being able to obtain comparable accuracy in the classification of quaternary structures, even if using only the 45.39% of the whole dataset.

The rest of the paper is organized as follows. Section 2 briefly surveys related work, addresses differences with our approach, and precisely highlights contribution of this work. Section 3 describes our protein quaternary structure classification method. Section 4 presents experimental results. Finally, Section 5 reports conclusions and future work.

2 Related Work

Recently, some techniques have been proposed for protein quaternary structure classification [5, 10, 16, 21, 22]. Most of them aim at classifying homo-oligomeric proteins, by storing and exploiting the overall considered dataset and comparing each protein to be classified to each stored protein.

In [10] a software, called Quaternary Structure Explorer (QSE), is presented. The method is based on the analysis of protein amino acid sequences to distinguish between homodimers and non-homodimers, by exploiting rule-based classifiers. The $C4.5$ machine-learning algorithm is used, and proteins are represented by 401 amino acid indices. The overall classification accuracy of the approach is 70%.

The method proposed in [16] uses the function of degree of disagreement (FDOD), that is, a measure of information discrepancy, to discriminate between homodimers and other homooligomeric proteins by exploiting information on the primary structure of proteins, enriched with subsequence distributions of such primary sequences to take into account the effect of residue order on protein structure. The technique scores an overall success rate of the 82.5%.

In [22] both the support vector machine (SVM) and the covariant discriminant algorithms are exploited to predict quaternary structure properties from the protein primary sequences. The authors consider the amino acid composition and the autocorrelation functions based on the amino acid index profile of the primary sequence, obtaining at the best an accuracy equal to the 87.5%.

The techniques presented in [10, 16, 22] are able to distinguish just between two classes, that are, homodimers and non-homodimers, whereas our approach is able to discriminate among any number of classes. In this respect, our method is more similar to the two approaches illustrated in the following.

The main contribution of the work [5] is the exploitation of a pseudo amino acid composition to express a protein in terms of a set of discrete numbers, which is more powerful than the standard amino acid composition. In fact, it is able to take into account a considerable amount of sequence-order effects. An overall success rate of 80.1% on a set of 332 test proteins was obtained.

The four approaches described above exploit only protein sequence information, without any regard for protein domain composition. Our method is different, as we consider the protein domain composition that, according also to other studies [19, 11, 6, 4, 20], is able to deliver important information about protein structures and functions, which may be related to protein quaternary structure composition.

The approach presented in [21] is based on the functional domain composition of proteins and on the nearest neighbor algorithm (NNA). They performed the jackknife cross-validation test on a non-redundant dataset of 717 proteins, and then exploited the non-redundant dataset to classify an independent dataset of 9,951 proteins. They obtained, resp., an overall success rate of 75.17% and 84.11% for the two dataset.

In our experiments, we used the same non-redundant dataset exploited in [21], enriched in the number of considered domains, obtaining some accuracy improvements (see Section 4). Differently from [21], where a *generalized distance* (which is not a metric) has been exploited in the classification method, we used the Jaccard distance as the distance metric. Furthermore, our technique is more efficient than the one presented in [21] and, in general, than the other related techniques, due to its ability of classifying proteins without the necessity of making comparisons with all the elements of the dataset. Indeed, we are able to extract a relatively small subset of the training set to carry out such a classification without any significant loose in precision.

To summarize, our approach is more general than previous methods, such as [10, 16, 22], that are specific for the classification of only two classes of protein quaternary structures. Furthermore, we exploited the protein representation which is shown to be the most complete in terms of protein functional information (i.e., functional domain composition), and we achieve high accuracy values even if exploiting small dataset portions. All these features grant to our method highest overall success rate than the other ones presented in the literature (97.74%), making it attractive especially when large protein datasets are to be handled.

3 Classification method

Next, we describe the classification method to individuate the number of subunits of each protein of a given dataset. In the following we will refer as PQSC-FCNN, for Protein Quaternary Structure Classification through FCNN rule, to the classification method here presented. In order to design an effective and efficient classification method, different issues are to be addressed. Among them: the feature space and distance metrics to adopt, the classification algorithm, and the suitability of the overall method.

As already pointed out, most of the quaternary structure classification methods proposed in the literature, store and use the whole available dataset, comparing each pro-

tein to be classified to each stored protein. This may result hard when large datasets are considered. Hence, we would like to drastically reduce the portion of the dataset that is necessary to store and, consequently, the number of comparisons to carry out, allowing sensible space and time savings.

To this end, we exploit protein functional domain information, and encode each protein by a binary vector whose elements are associated to PFam domains [3]. We adopt the Jaccard metric as our distance measure and exploit the *k nearest neighbor rule* [7], one of the most extensively used nonparametric classification algorithms, which is simple to implement and yet powerful. The rationale underlying this choice is that for this classification rule there exist efficient techniques to reduce both space and time requirements, that we will use in order to reduce the already classified protein dataset.

In the following, the adopted protein representation, distance metrics, classification rule, and data reduction method are detailed.

Protein representation. To characterize proteins, we adopted the functional domain composition, since this kind of representation has been proved to be successful both for the specific problem we analyzed [21], and for the solution of other related problems, such as the prediction of protein-protein interactions [19, 11], of protein structures [6] and of protein functions [4, 20]. Protein functional domains are elements of the protein structure that are self-stabilizing and often fold independently of the rest of the protein chain. According to the functional domain composition, a protein is represented by a binary vector with size equal to the number of exploited domains. In particular, let D be an ordered set of protein domains, which have been considered to characterize the proteins in a dataset P . Then, each protein $p \in P$ is represented by a vector v_p of $|D|$ elements. The element $v_p[i]$ is set to be one if p contains the i -th domain in D , zero otherwise.

Distance metrics. We used the Jaccard metric as our distance metric, which is very suitable for binary data. In particular, the Jaccard distance between two protein vectors v_{p1} and v_{p2} is defined as:

$$d(v_{p1}, v_{p2}) = \frac{n_2 + n_3}{n_1 + n_2 + n_3}$$

where n_1 is the number of indexes for which the correspondent elements are equal to one in both vectors, n_2 is the number of indexes for which the correspondent elements are equal to one in v_{p1} and to zero in v_{p2} , n_3 is the number of indexes for which the correspondent elements are equal to zero in v_{p1} and to one in v_{p2} .

Classification rule. The *nearest neighbor rule* [7] is widely used as a classification algorithm. It is simple to implement and yet powerful, due to its theoretical properties guaranteeing that for all distributions its probability of error is bounded above by twice the Bayes probability of error. The nearest neighbor decision rule can be generalized to the case in which the k nearest neighbors are taken into account. In such a case, a new object is assigned to the class with the most members present among the k nearest neighbors of the object in the training set. This rule has the additional property that it provides a good estimate of the Bayes error and that its probability of error asymptotically approaches the Bayes error [9].

The naive implementation of the NN rule has no learning phase, since it requires to store all the previously classified data, and then to compare each sample point to be

classified to each stored point. In order to reduce both space and time requirements, several techniques to reduce the size of the stored data for the NN rule have been proposed (see [18] for a survey). In particular, among those techniques, the *training set consistent* ones, aim at selecting a subset of the training set that correctly classifies the remaining data through the NN rule.

Data reduction. In order to reduce the reference protein quaternary structure dataset used during classification, we exploited the Fast Condensed Nearest Neighbor rule [1], FCNN for short, an algorithm computing a training set consistent subset for the NN rule. Informally, having fixed a meaningful distance metrics and a dataset T , a subset S of T is a training set consistent subset of T for the nearest neighbor rule, if S correctly classifies all the objects of T by means of the nearest neighbor rule. Thus, loosely speaking, the objects of the subset S can be regarded as representing the objects of T which are not in S , and training set consistent subset methods for the nearest neighbor rule can be regarded as methods to filter out dataset instances which can be considered unessential to correctly classify new incoming objects.

The method is recalled next. We provide some definitions first. We define T as a labelled training set from a metric space with distance metrics d . Let x be an element of T . Then we denote by $nn_k(x, T)$ the k th nearest neighbor of x in T , and by $nns_k(x, T)$ the set $\{nn_i(x, T) \mid 1 \leq i \leq k\}$. $l(x)$ will be the label associated to x . Given a point y , the k -NN rule $NN_k(y, T)$ assigns to y the label of the class with the most members present in $nns_k(y, T)$. A subset S of T is said to be a *k -training set consistent subset of T* if, for each $y \in (T - S)$, $l(y) = NN_k(y, S)$. Let S be a subset of T , and let y be an element of S . By $Vor(y, S, T)$ we denote the set $\{x \in T \mid \forall y' \in S, d(y, x) \leq d(y', x)\}$, that is the set of the elements of T that are closer to y than to any other element y' of S , called the *Voronoi cell* of y in T w.r.t. S . Furthermore, by $Voren(y, S, T)$ we denote the set $\{x \in (Vor(y, S, T) - \{y\}) \mid l(x) \neq NN_k(x, S)\}$, whose elements are called *Voronoi enemies* of y in T w.r.t. S . $Centroids(T)$ is the set containing the centroids of each class label in T . The FCNN rule relies on the following property: a set S is a training set consistent subset of T for the nearest neighbor rule if for each element y of S , $Voren(y, S, T)$ is empty.

The FCNN algorithm initializes the consistent subset S with a seed element from each class label of the training set T . In particular, the seeds employed are the centroids of the classes in T . The algorithm is incremental. During each iteration the set S is augmented until the stop condition, given by the property above, is reached. For each element of S , a *representative* element of $Voren(y, S, T)$ w.r.t. y is selected and inserted into S . Such a representative element it is the nearest neighbor of y in $Voren(y, S, T)$, that is, the element $nn(y, Voren(y, S, T))$ of T .

As for the time complexity of the method, let N denote the size of the training set T and let n denote the size of the computed consistent subset S . Then the FCNN rule requires Nn distance computations to compare the elements of T with the elements of S . However, if the distance employed is a metric, a technique exploiting the triangle inequality further reduces this worst case computational cost [1].

4 Experiments

In this section, we illustrate the experimental evaluation of the method proposed in this paper. To build our datasets, we downloaded proteins from the SWISSPROT database

Large dataset						
Classes	PQSC-FCNN, $k = 2$		PQSC-FCNN, $k = 3$		PQSC-FCNN, $k = 4$	
	Corr/Tot	% Accuracy	Corr/Tot	% Accuracy	Corr/Tot	% Accuracy
Monomer	6,114/6,184	99.45%	6,130/6,184	99.13%	6,135/6,184	99.21%
Homodimer	8,408/8,690	96.75%	8,427/8,690	96.97%	8,402/8,690	96.68%
Homotrimer	1,154/1,190	96.97%	1,150/1,190	96.64%	1,136/1,190	95.46%
Homotetramer	2,422/2,513	96.38%	2,452/2,513	97.57%	2,380/2,513	94.71%
Homopentamer	232/237	97.89%	232/237	97.89%	232/237	97.89%
Homohexamer	759/784	96.81%	761/784	97.07%	742/784	94.64%
Homoheptamer	4/5	80.00%	4/5	80.00%	4/5	80.00%
Homooctamer	457/465	98.28%	458/465	98.49%	458/465	98.49%
Overall	97.60%		97.74%		97.11%	
% Dataset Exploitation	6.43%		6.51%		6.70%	

Table 1. Overall success rate obtained by the *PQSC-FCNN* method for $k \in \{2, 3, 4\}$ on a dataset of 20,068 proteins.

[2], available at www.ebi.ac.uk/swissprot/, and domains from the PFam database [3], available at <http://www.sanger.ac.uk/Software/Pfam/>.

We considered two different experiments. The first experiment consisted in running the 10-fold cross-validation on a very large protein dataset consisting of 20,068 proteins. The number of considered domains is 1,816. The results of this experiment are shown in Table 1. The first column of the table contains the homo-oligomeric class names, the second, third and fourth ones report both the number of correctly predicted proteins w.r.t. their total number and the percentage of accuracy scored by *PQSC-FCNN* for $k = 2$, $k = 3$ and $k = 4$, respectively, for each class. In the last two rows of the table, the overall accuracy and the percentage of exploited dataset are reported. The results confirmed the effectiveness of our approach. In fact, the maximum overall success rate obtained on the entire dataset is of the 97.74%, and the minimum dataset exploitation is drastically reduced to the 6.43% of the original dataset. In general, as for the classification accuracy the three values of k were comparable, being equivalent on the homopentamers and on the homoheptamers, while only for $k = 3$ and for $k = 4$ the method returns the same results for homo-octamers.

Table 2 shows detailed information about the condensed set generated by the method on the overall dataset of 20,068 proteins. In particular, for each class, both the number of elements of the condensed set belonging to that class, and the reduction percentage w.r.t. the total number of elements in that class, are reported. The number of elements and the reduction percentage of the overall condensed set are shown on the last row of the table. By using these three values of k , the method extracted condensed sets with the same size per class for homopentamers, homoheptamers and homo-octamers. For the homoheptamer class, the reduction percentage is notably higher than for the other classes, due to the few elements belonging to that class (only 5 ones). The reduction percentage on the overall dataset is 6.01% for $k = 2$, 6.10% for $k = 3$ and 6.32% for $k = 4$. This shows the power of the method, as it is sufficient to explore only a bit more than the 6% of the overall dataset to (most probably) classify a new protein.

Both the exploited protein datasets and the correspondent domains can be downloaded at the web-links reported in Section 5.

Condensed set						
Classes	PQSC-FCNN, $k = 2$		PQSC-FCNN, $k = 3$		PQSC-FCNN, $k = 4$	
	Number of elements	Percentage	Number of elements	Percentage	Number of elements	Percentage
Monomer	98/6,184	1.58%	153/6,184	2.47%	157/6,184	2.54%
Homodimer	643/8,690	7.40%	649/8,690	7.47%	718/8,690	8.26%
Homotrimer	145/1,190	12.18%	101/1,190	8.49%	108/1,190	9.08%
Homotetramer	197/2,513	7.84%	199/2,513	7.92%	157/2,513	6.25%
Homopentamer	17/237	7.17%	17/237	7.17%	17/237	7.17%
Homoexamers	74/784	9.44%	74/784	9.44%	79/784	10.08%
Homoheptamer	3/5	60.00%	3/5	60.00%	3/5	60.00%
Homooctamer	29/465	6.24%	29/465	6.24%	29/465	6.24%
Overall	1,206	6.01%	1,225	6.10%	1,268	6.32%

Table 2. Condensed sets detailed information for the dataset of 20,068 proteins.

In order to compare our method with a related one, in the second kind of experiments we considered the non-redundant protein dataset discussed by Yu et al. in [21]. The main goal of this comparison is to show that our method may have accuracy comparable to those of related methods, while sensibly reducing the amount of labeled data to exploit during classification. In particular, we point out that the method presented in [21] utilizes a non redundant version of the overall protein dataset in order to cope with problems associated with management of large data sets. As we will show in the following, our method will be able to halve even this non redundant dataset, while maintaining the same accuracy as the competitor method.

Yu et al. adopted an approach based on the functional domain composition and employed the nearest neighbor algorithm (NNA) to classify protein quaternary structures. They represented the 717 considered proteins by 540 domains. Here, we enlarge the number of considered domains to 1,253 in order to obtain a more accurate representation of data. Thus, we compare *PQSC-FCNN* with *NNA* by running the jackknife cross-validation on the non-redundant dataset, by considering the same 1,253 domains representation for all methods.

We run *PQSC-FCNN* exploiting the Jaccard metric, whereas *NNA* has been run with the generalized distance exploited in [21]. The results are illustrated in Table 3. The first column of the table contains the homo-oligomeric classes, the second, third, fourth and fifth ones illustrate both the number of correctly predicted objects w.r.t. the total number of them and the percentage of accuracy scored by *PQSC-FCNN* for $k = 2$, for $k = 3$ and for $k = 4$, and *NNA*, respectively, for each considered class. In the last two rows of the table, the overall accuracy and the percentage of exploited dataset for each method are reported. We can observe that all the considered techniques return the same results for the two classes of homopentamers and homooctamers. The only class for which *PQSC-FCNN* does not obtain higher accuracy than the *NNA* is the homodimer class. However, for $k = 4$ it obtains a success rate of 78.51% for that class, w.r.t. the 80.00% scored by the *NNA*, and thus, also in this case, the two methods achieve comparable results.

For the homotrimer and the homotetramer classes, the *PQSC-FCNN* for $k = 2$ and the *NNA* have the same accuracy scores, whereas in the remaining cases (i.e., monomers and homoexamers), *PQSC-FCNN* always scores better accuracy than *NNA*, also with

Non-redundant dataset								
Classes	PQSC-FCNN, $k = 2$		PQSC-FCNN, $k = 3$		PQSC-FCNN, $k = 4$		NNA	
	Corr/Tot	% Accuracy	Corr/Tot	% Accuracy	Corr/Tot	% Accuracy	Corr/Tot	% Accuracy
Monomer	177/208	85.10%	174/208	83.65%	178/208	85.58%	168/208	80.77%
Homodimer	250/335	74.63%	250/335	74.63%	263/335	78.51%	268/335	80.00%
Homotrimer	28/40	70.00%	27/40	67.50%	27/40	67.50%	28/40	70.00%
Homotetramer	53/95	55.79%	52/95	54.74%	42/95	44.21%	53/95	55.79%
Homopentamer	11/11	100.00%	11/11	100.00%	11/11	100.00%	11/11	100.00%
Homohexamer	9/23	39.13%	9/23	39.13%	10/23	43.48%	7/23	30.43%
Homooctamer	2/5	40.00%	2/5	40.00%	2/5	40.00%	2/5	40.00%
Overall accuracy	73.92%		73.22%		74.34%		74.90%	
% Dataset Exploitation	46.79%		47.35%		45.39%		100%	

Table 3. Comparison of overall success rate obtained by PQSC-FCNN and NNA.

some sensible improvements. In particular, for $k = 4$, *PQSC-FCNN* scores the accuracy value of 85.58% for monomers, which is about 4.81% higher than the success rate obtained by the *NNA*, whereas it scores the accuracy value of 43.48% for homohexamers, which is about 13.05% higher than the success rate obtained by the *NNA* for the same class and represents the best accuracy improvement w.r.t. *NNA* we obtained.

These results are significant since, for monomers and homohexamers, our method has been able to obtain more accurate results than *NNA* even exploiting only the 45.39% of the overall dataset, whereas [21] did not make any dataset reduction.

Summarizing, the *PQSC-FCNN* method is able to reach an overall success rate that is greater than the *NNA*, even if it exploits only the 45.39 – 47.35% of the original dataset. This means that *PQSC-FCNN* is more efficient than the method [21], allowing both time and space savings without any significant loss in accuracy but, rather, often allowing success rate improvements.

5 Concluding Remarks

We proposed a classification method for protein quaternary structures, by exploiting protein functional domain information and FCNN rule. Experimental evaluations showed that our approach is able to reduce both the portion of protein dataset that is necessary to store and the number of comparisons to carry out, allowing sensible space and time savings even if guaranteeing high values of accuracy. Thus, the method is competitive w.r.t. other methods presented in the literature. We illustrated some tests we carried out on homo-oligomeric proteins, confirming the validity of our approach. Future efforts will be devoted to investigate the problem of quaternary structure prediction for hetero-oligomers.

Additional material. The protein datasets we exploited are available on-line. In particular, the non-redundant dataset of 717 proteins is listed at:

http://siloe.deis.unical.it/PQSC_FCNN/non_redundant_dataset.pdf, together with the corresponding 1,253 domains we exploited for the functional domain composition, while the 20,068 proteins of the second dataset and their 1,816 are available at:

http://siloe.deis.unical.it/PQSC_FCNN/20068_dataset.pdf.

References

1. F. Angiulli. Fast condensend nearest neighbor rule. In *Proc. of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.
2. A. Bairoch and R. Apweiler. The swiss-prot protein sequence data bank and its new supplement trembl. *Nucleic Acids Research*, 24(1):21–25, 1996.
3. A. Bateman, E. Birney, L. Cerruti, R. Durbin, L. Etwiller, S. R. Eddy, S. Griffiths-Jones, K. L. Howe, M. Marshall, and E. L. L. Sonnhammer. The pfam protein families database. *Nucleic Acids Reserch*, 30(1):276–280, 2002.
4. Y. D. Cai and A. J. Doig. Prediction of saccharomyces cerevisiae protein functional class from functional domain composition. *Bioinformatics*, 20(8):1292–1300, 2004.
5. K. C. Chou and Y. D. Cai. Predicting protein quaternary structure by pseudo amino acid composition. *Proteins: Structure, Function, and Genetics*, 53(2):282–289, 2003.
6. K. C. Chou and Y. D. Cai. Predicting protein structural class by functional domain composition. *Biochemical and biophysical research communications*, 321(4):1007–1009, 2004.
7. T.M. Cover and P.E. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Inform. Th.*, 13(1):21–27, 1967.
8. L. Devroye, L. Györfy, and G. Lugosi. *A Probabilistic Theory of Pattern Recognition*. Springer-Verlag, New York, 1996.
9. K. Fukunaga and L.D. Hostetler. k -nearest-neighbor bayes-risk estimation. *IEEE Transactions on Information Theory*, 21:285–293, 1975.
10. R. Garian. Prediction of quaternary structure from primary structure. *Bioinformatics*, 17(6):551–556, 2000.
11. W. K. Kim, J. Park, and J. K. Suh. Large scale statistical prediction of protein-protein interaction by potentially interacting domain (pid) pair. *Genome informatics. International Conference on Genome Informatics*, 13:42–50, 2002.
12. I. M. Klotz, N. R. Langerman, and D. W. Darnall. Quaternary structure of proteins. *Annual review of biochemistry*, 39:25–62, 1970.
13. A. M. Lesk. *Introduction to Protein Architecture*. Oxford University Press, 2001.
14. J. Meiler and D. Baker. Coupled prediction of protein secondary and tertiary structure. *Proceedings of the National Academy of Sciences of the United States of America*, 100(21):12105–12110, 2003.
15. 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*, 47:228–235, 2002.
16. J. Song and H. Tang. Accurate classification of homodimeric vs other homoooligomeric proteins using a new measure of information discrepancy. *Journal of chemical information and computer sciences*, 44(4):1324–1327, 2004.
17. H. Sund and K. Weber. The quaternary structure of proteins. *Angewandte Chemie (International ed. in English)*, 5(2):231–245, 1966.
18. D.R. Wilson and T.R. Martinez. Reduction techniques for instance-based learning algorithms. *Machine Learning*, 38(3):257–286, 2000.
19. J. Wojcik and V. Schachter. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, 17(1):296–305, 2001.
20. X. Yu, J. Lin, T. Shi, and Y. Li. A novel domain-based method for predicting the functional classes of proteins. *Chinese Science Bulletin - English Edition-*, 49(22):2379–2384, 2004.
21. X. Yu, C. Wang, and Y. Li. Classification of protein quaternary structure by functional domain composition. *BMC Bioinformatics*, 7(187), 2006.
22. S. W. Zhang, Q. Pan, H. C. Zhang, Y. L. Zhang, and H. Y. Wang. Classification of protein quaternary structure with support vector machine. *Bioinformatics*, 19(18):2390–2396, 2003.