BIOINFORMATICS IN PROTEOMICS: A REVIEW ON METHODS AND ALGORITHMS

I. Popov[1], A. Nenov[2], P. Petrov[3] and D. Vassilev[1]
AgroBioInstitute, Sofia, Bulgaria[1],
Dynomica Ltd, Sofia, Bulgaria[2],
Sofia University, "St. Kliment Ohridsky", Faculty of Mathematics and Informatics, Sofia, Bulgaria[3]
Correspondence to: Dimitar Vassilev
E-mail: jim6329@gmail.com

ABSTRACT

It is often said that bioinformatics is a knowledge based discipline. This means that many of the search and prediction methods that have been used to greatest effect in bioinformatics exploit information that has already been accumulated about the problem of interest, rather than working from first principles. Most of the methods and algorithms discussed in this paper adopt these knowledge-based approaches for protein studies. Typically we have some given examples i.e. data of a given class or function, and we try to identify patterns in that data which characterize these sequences or structures and distinguish them from others that are not in this class. The purpose of this paper is to describe the basic conceptual methods and adjacent algorithms and applications that are used to obtain better and more reliable information of the studied characteristic patterns.

Keywords: bioinformatics, proteomics, algorithms, methods, software

Computational challenges of proteomics

Large scale proteomics experiments have released a vast amount of biological data, collected in variety of repositories worldwide. The information volume of gathered biological data determines the information resource and capacity needed to store, analyze and extract valuable information and knowledge. Proteomics, as an emerging scientific field focused on protein structure and function, accumulates a large amount of biological data through separation of proteins by two dimensional gel electrophoresis, isoelectric focusing, 2D visualization of proteins, mass spectrometry, peptide mass fingerprinting, tandem mass spectrometry, etc. Bioinformatics in the domain of proteomics is a breath taking set of developing methods, as it is an important part of creating knowledge from the experimental data, with its models as protein folding, three dimensional structure of proteins, and prediction for structures and functions of unknown proteins.

On a lower level still, bioinformatics deals with the molecular basics of living organisms - the sequence of nucleic acids, the structure of genes and other functional elements of DNA, the sequence and structure of proteins, membranes and any other compound that comes into the light of scientific research. But no matter how small the physical object is, it can always create an enormous amount of information. The DNA sequences of a whole library of bacterial strains, the coordinates of thousands of atoms in a structure, the data from qPCR runs or the force curves of a hundred AFM-spectroscopy experiments - all these contain that much information, sometimes with such low quality, that it is impossible to gain any understanding of the underlying processes or objects without running some kind of algorithm to determine the quality and to extract the actual information the scientists are after. Thus, it can be said that the main object of bioinformatics is data. The state of the art high-tech equipment used in scientific experiments creates more and more information (like automated pipelines for high-throughput screening of libraries or ELISA assays for example) that can only be handled by computers. Bioinformatics deals with that.

When it comes to **computational biology of proteins**, there is a large set of Internet resources, that integrates the data gathered experimentally for further analysis. One of the milestones in proteomics research is the Protein Data Bank (PDB) (5), that contains over 55 000 three dimensional structures of proteins resulting from crystallography or X-ray studies and created by modelling software. These real and proposed structures, however, do not cover all the proteins found in biological systems, as the resolution of a protein structure is a time consuming and expensive process. Medical related publications in scientific literature leads to quite high redundancy rate in bioinformatics databases, which could be visualized by clustering of the PDB entries with a BLAST algorithm. The clustering similarity evaluation turns up only 20 229 clusters at a 95% cutoff. This redundancy can be observed in most databases.

The PDB is one of the main resources for structural information on proteins, and as such it is widely used in bioinformatics. There are, however, many sources of sequential information or servers that offer classification of proteins by structure and function. Some of those are Structural Classification of Proteins (SCOP), (18), The Universal Protein Resource (UniProt), (28), CATH (Class, Architecture, Topology and Homologous superfamily, a database of classified protein domain structures) (21).

What Bioinformatics aims to achieve

Bioinformatics can both analyze proteomics data and work as a supplement to other proteomics methods in order to increase the quality of results. There are methods like mass spectrometry that depend heavily on Bioinformatics for the analysis of their results. Furthermore, Bioinformatics develops ways to enhance the results of MS-based methods (27) or offer non-bioinformaticians and researchers in specialized fields an easier way to analyze their data (6).

The development of algorithms for the direct elucidation of biologically significant data is another way for Bioinformatics to provide aid in proteomics research. Nagarai et al. (19) have developed a method for better prediction of transcription factor binding sites using datasets from binding assays, while Cai et al. (8) created their algorithm for the same problem based on the proteins amino acid sequence and physicochemical properties.

Another high value product of Bioinformatics are the results of various structural prediction algorithms and pipelines. There are solutions to predict protein interactions (30) and interaction affinity (29), to find unknown receptor activators (17) or to create models for known protein interactions (26). Evaluation algorithms are also used to classify proteins: Otto et al. (22) used different Bioinformatics tools to identify, annotate and compare analogous and homologous enzymes, that can be used to study the evolution of biochemical pathways and find potential drug targets.

Conceptual methods for protein analysis

Sequence alignment

The simplest and yet powerful method for protein analysis is the comparison of their primary amino acid sequence. The most common information this might provide is the distance of relationship between the compared proteins. By pairwise aligning and scoring a group of proteins, a phylogenetic tree based on the distances can be built. This is usually done by creating global sequence alignments that take into account the whole sequences of the two proteins. When the overall sequence identity of the proteins is low it is possible that a global alignment will miss pockets of high scoring matches in the sequences, especially when they are far apart. In this case local sequence alignment is used. It is characteristic for local alignments to minimize the penalties for introducing gaps into the alignment, thus making possible for far apart regions to be matched and scored higher. This is useful when the proteins in question share only one domain or another smaller structural or functional element.

Multiple sequence alignment is possible in the same way as when two single protein sequences are aligned. The only difference being that the one sequence that is added to the alignment is scored against all of the sequences already in it at the same time and the results being totaled in the scoring matrix.

One algorithm used for global sequence alignment is the Needleman-Wunsch algorithm (20). It uses a scoring matrix that states how good the score between every two amino acids is. When the alignment is built for every position in the two sequences a score is calculated as the sum of the previous score plus the score for the two amino acids in question. If introducing a gap scores better - it is done; if matching the two residues is better - they are matched. Scores are recorded in a matrix that contains the two sequences at its "axes". The final alignment is read backwards from the end of the matrix by following the right scores. A simple change in the algorithm leads to the so called Smith-Waterman algorithm (25). It deals with local alignments by nullifying the gap penalty. This way matching separate domains or other parts of the sequence lead to high scores rather than being offset by the high negative score of a large gap between them (**Table 1**).

TABLE 1

Algorithmic techniques used to solve bioinformatics tasks in proteomics

Bioinformatics Tasks	**Algorithmic Techniques**
Mapping DNA	Brute Force, Exhaustive Search
Sequences Comparison	Dynamic Programming, Divide and Conquer
Gene Prediction	Dynamic Programming
Finding Signals	Brute Force, Exhaustive Search, Greedy Algorithms
DNA Arrays	Clustering, Classification analysis
Genomic Rearrangements	Greedy Algorithms
Molecular Evolution	Clustering, Classification analysis

The most commonly used sequence alignment algorithm is called BLAST (Basic Local Alignment Search Tool) (1). Simply put, BLAST finds high similarity spots in the two sequences and then builds upon them to create the final alignment.

Sequence alignment is an important tool in Bioinformatics and the mentioned algorithms have undergone a lot of changes and improvements to make them faster, able to compare a single sequence to a whole database of proteins or genes, and give more meaningful results. An improvement that aims at that is the Psi-BLAST (Position-Specific Iterated BLAST) (2). It runs a normal BLAST round over the database and then builds a scoring matrix from the results. This matrix is used in the second iteration and gives more distant sequences a chance to emerge in the result. If another iteration is started the matrix is rebuilt from the second results and so on. This way it is possible to find not-so-similar sequences that can still be reasonably related to the search sequence.

Secondary structure recognition

The next step after sequence analysis is the elucidation of the secondary protein structure from said sequence. Secondary structures are divided into several generally used types: alpha helices, beta strands and undefined structure sequences (coils) as a possible classification of secondary structure elements.

Some prediction programs use a beta-loop structure that connects beta strands and the Dictionary of Protein Secondary Structure (DSSP) (16) defines eight different structural elements with a selection of different alpha and beta states, bends, turns, and coils.

In general there are three groups of methods that deal with secondary structure prediction (31). The first group comprises statistical methods, based on the propensity of amino acid residues to form a certain structure. The propensity is calculated by analyzing a set of know protein structures and noting the secondary structure elements in which the residues take part. The first such methods used a small amount of resolved protein structures that were available at the time, but the set was gradually expanded for better statistics. More recent methods use whole stretches of amino acid residues in order to take into consideration the local interactions between residues, which are also important for secondary structure formation. Some algorithms use the positions of residues to build scoring matrices (15).

The second group of methods makes use of empirical information about the residues in the polypeptide chain: charge, hydrophobicity, size and shape, H-bond formation, and other physical and chemical properties. Those are called knowledge-based methods. They still use the statistics from the previous group, but they also complement it with more information. Another often used characteristic is the residue conservation, found by multiple sequence alignment. The use of residue conservation generally improves the results of prediction methods.

The last group of methods are the machine learning methods, like neural networks and Hidden Markov Models. They are computer algorithms that are trained on a training dataset and then used for structure prediction on a test dataset to define their accuracy. They are not bound to certain rules and physical models concerning the relationship between the amino acid sequence and the structure of the protein: their internal parameters are just adjusted to best fit the training dataset during the training period, and this makes the proper selection of the training data very important. There are several webservers built that use some kind of neural network implementation for secondary structure prediction such as Jpred (10), PredictProtein (24) and PSIPRED (7, 15). Another server, PREDATOR (14), uses knowledge-based database comparison.

A particular class of proteins deserves a special note when it comes to secondary structure prediction. Transmembrane proteins have regions that span a membrane structure in the cell and those have specific properties due to the highly hydrophobic nature of the membrane. It is possible to predict transmembrane regions by averaging the hydrophobicity over a stretch of amino acid residues, taking into account that the stretch is also limited in length by the membrane. This means looking for groups of 15 to 30 hydrophobic residues, as the thickness of an average membrane (30 angstrom) does not allow a longer or shorter sequence. Peptide chains that cross the membrane more than once may include helices with both hydrophobic and hydrophilic residues, usually separated on opposite sides of the helix. Such helices are called amphypatic.

Structure alignment

Some basic algorithms for structure alignment need a pre-existing sequence alignment of the sequences, and consist of four steps. First, the center of masses for each of the structures at hand should be computed. Second, the two structures need to be overlap, so that their centers of mass are matched. Third, the angle difference between the positions of each pair of corresponding residues is to be computed using the center of mass as a starting point. Fourth, one of the structures is rotated by the median angle difference.

The next step to a more complex structure alignment is to generate a distance scoring matrix from the existing aligned residues and to use it to generate a second sequence alignment. This second alignment is then used in the same way to further rotate the structure and then build the next matrix, and the next alignment, and so on. This is repeated until there is no significant change in the RMSD (root mean square deviation) of the two aligned structures.

Another, more complex algorithm for structure alignment employs the so called double dynamic programming. It uses a two level structure of aligning, that makes it possible to score the pairwise matching of residues and use the scores themselves as information for a scoring matrix. First a series of scoring matrices are created, each of them based on the assumption that a specific pair of residues match and are perfectly aligned in the final structure. From this pair of matching residues an alignment is built with regular dynamic programming in order to give a score for the matching of every other pair of residues. A simple scoring can be created by fixing a reference system in each of the two molecules using the residues before and after the "perfect match" pair, overlapping the two reference systems and then using the distance between the positions of the two residues in their respective reference systems (that can now be observed as a single system). More complex scoring can be achieved by taking into account the direction of the vector between the two residues, the orientation of the residues, their place in the sequence or in space (how far they are from the "perfect match" pair of residues).

When more than two structures are involved, there are few different ways to superimpose them all. A multiple sequence alignment can be used to define the sequence in which the structures will be added to the structure alignment. First the two structures that have the best match in sequence are aligned, and then the next structure that best matches them is added. The process is repeated for all the structures. This presents a logical way of building a multiple alignment, but is vulnerable to biasing the results towards structures, similar to the starting one, that was used in the multiple sequence alignment.

3D structure prediction

The computational biology of proteins is always tightly connected with defining the functional characteristics of

a protein or a set of proteins. One of the important steps in the analysis of functional characteristics is the accurate three dimensional (3D) model of the protein structure. This scientific problem is currently solved by two different approaches:

- ***Ab initio* methods**

  Ab initio methods are used for making predictions about protein features using only a computational model, without extrinsic comparison to existing structures. These methods relate on simple physical and chemical assumptions as free energy and conformation, to build a model of a protein. A key element of these algorithms is the confirmation assessment step, which determines to great extent the method precision.

- **Comparative modelling methods**

  These methods are based on previously determined features and their connections with certain 2D or 3D structural patterns. These methods, including threading (fold recognition) and homology modeling (comparative modeling), rely on detectable similarities between the modeled sequence and a set of known structures.

Modeller is a computer program that models three-dimensional structures of proteins and their assemblies by satisfaction of spatial restraints. A three dimensional model is generated by optimization of a molecular probability density function, which is optimized with the variable target function procedure in Cartesian space that employs methods of conjugate gradients and molecular dynamics with simulated annealing. In practice, Modeller could be generally divided into four main bioinformatics protocols, such as: model a sequence with high identity to a template; model a sequence based on multiple templates and bound to a ligand; increase the accuracy of the modeling exercise by iterating the 4 step process, and model a sequence based on a low identity to a template. Modeller can also perform multiple comparisons of protein sequences and/or structures, clustering of proteins, and a search of sequence databases.

Nest is a program for modelling protein structure based on a given sequence-template alignment. Nest could execute: homology model building; composite model building; model building based on multiple templates; structure refinement, and sequence-alignment tuning. The final step of the Nest's calculation is based on the energy function, consisting of the following parameters: van der Waals energy, hydrophobic, electrostatics, torsion angle energy, hydrogen-bond network energy of the template, and statistical energy of a residue's solvent accessibility.

When compared, the Nest algorithm is producing the most accurate loop conformations, although the difference between Modeller and Nest in loop building is not statistically significant (11).

Mutation analysis

The computer simulation of a mutation and its effect on protein structural and functional properties is closely related to homology modeling. However, there are other possible ways to look at the change induced by a mutation and properties that homology modeling cannot describe. One such method is proposed by Piana et al. (23), who investigated the effect of point mutations on the stability of protein folds using an atomistic approach and molecular dynamics simulations.

Fold recognition

Fold recognition uses several sources of lower level information on the protein structure like predicted or known secondary structure, contact energy functions (12), sequence profiling, evolutionary analysis and Hidden Markov Models (described later) to reach the folded state of the protein. There are also databases and methods for protein classification based on folds like SCOP or PFRES (9).

Domain classification

As a functional unit, a protein domain is more conservative as a structure in 3D space. The relations between different domains and their functional representation are of interest both to proteomics and bioinformatics. There are several main resources, dealing with domain classification.

The CATH database sets protein domains in a four-level hierarchy according to their CLASS (secondary structure composition), ARCHITECTURE (shape formed by the secondary structures), TOPOLOGY (connectivity order of the secondary structures) and HOMOLOGOUS SUPERFAMILY (structural and functional similarity). The classification of individual protein domains is performed by several algorithms: CATHEDRAL, SSAP, DETECTIVE, PUU, and DOMAK. CATH is partially manual, since classification of Architecture is done by visual inspection.

Some algorithmic concepts in Bioinformatics

A common task in bioinformatics is to classify sequences into a number of different categories. In the simplest case there are only two categories: 1) the sequences belong to a group of sequences of interest, and 2) when the sequences do not belong to it. A more general case is when there are several categories and it is necessary to decide into which category the new sequence best fits. Probabilistic models can be developed to address questions of this type (3) (**Table 2**).

Hidden Markov Models (HMMs) are abstract mathematical machines which are used to model certain types of probabilistic processes which operate, typically but not necessarily, over time. Time is, in most cases, viewed as a discrete concept, i.e. it is viewed as a sequence of time steps - step 0, step 1, step 2, ..., step t, etc. The HMMs are probabilistic machines because at each moment of time the symbol that is to be output and the next state the machine will go into are defined by two probabilistic functions – these are the so-called transition and emission functions. The transition function specifies given a current state what the probability is that the next state will be as it is expected. And it has been done so, for each couple of states. This is the so called Markov property – the next state depends only on the current state and not on any of the states

further ahead in the future. Generalizations of HMMs exist where the next state depends on the last ***k*** states only (in the classical HMM case ***k=1***). What is viewable from the outside (of the HMM) is just the sequence of the emitted /generated symbols - hence the term 'hidden' (4).

In a Markov model of protein sequence the probability that any amino acids will occur at a given position depends on which amino acid lies immediately before it. In an Hidden Markov Model (HMM) the probability of an amino acid depends on the values of hidden variables determining which state of sequence it is in at that point (e.g., helix or loop region of a membrane protein).

HMMs can be trained to represent families of sequences. When a given sequence is compared to the model, the optimal path of the sequence through the states of the model can be calculated. This allows to be predicted which parts of a sequence are corresponding to which states in the model. A particular case of this is profile HMMs where the model describes a sequence alignment, and calculating the optimal path through the model corresponds to aligning the new sequence with the profile.

Pattern recognition methods based on protein sequence alignment are attempted either to encode the conservative regions or to apply certain probabilistic approach to work with some statistical inferences and properties of the sequences. In bioinformatics applications the data usually are short segments of protein or DNA sequence - say 10 - 20 residues or nucleotides. A sequence can be encoded into inputs in several ways. The input signals determine the signals produced by the hidden layer, which in turn determine the signals of the output layer. Neural Networks (NNs) are machine learning algorithms particularly suited for classification and pattern recognition problems (4). Some of the most successful methods of protein secondary and tertiary structure prediction use NNs. Artificial Neural Network (ANN) is a computing paradigm inspired by the way biological Neural Networks work. The building block of each biological neural network is the neuron. In biological terms this is a cell which accepts input from many other neurons via its dendrites and generates output via its single axon. The axon though is used to pass the output to many other neurons at once - this is done at junction places called synapses. Artificial neural networks can be viewed as a more extended variant of a directed graph where each node is an artificial neuron. One of the functions related to neurons is the output function. The decision whether to have a transition or not is typically left to another function (related to the node *n*) which is usually the so-called threshold function.

The principles of SVMs (Support Vector Machines) are close to these of NNs as a machine learning algorithm. For any given input the algorithm gives a Yes or No output. The value of the output depends on the internal variables of the programme. As with NNs, SVMs are trained by optimizing the internal variables of the algorithm such that it gives a correct answer for as many as possible of the examples in the training set. When the SVM will be used further it will predict that proteins with similar profiles to the positive examples are also members of the class. SVMs are another way of solving problems that are not linearly separable. SVMs are worked mostly with gene-expression data and classification analysis in DNA or protein microchips (3).

TABLE 2

Models used for representing some bioinformatics tasks in proteomics

Bioinfromatics Tasks	**Theoretical Models**
Sequencing DNA	Graph Models
Sequences Comparison	Combinatorial Models
Finding Signals	Hidden Markov Models, Probabilistic Models
Identifying Proteins	Graph Models
Repeat Analysis	Combinatorial Models
DNA Arrays	Graph Models, Tree Models
Molecular Evolution	Tree Models

In the context of this paper on pattern recognition a general point about prediction methods is that the most successful are the knowledge based methods - i.e., they look for similarities to sequences of known structure, or they use training sets of known examples, like the machine learning approaches. *Ab initio* methods, beginning only with a single sequence and fundamentals such as interatomic forces tend to be less successful. Thus pattern recognition techniques are providing an answer to the practical question of structure prediction and gradual improvements are being made.

Conclusions

In the post genomic era and in the beginning of the era of synthetic biology, the methods of bioinformatics in proteomics are crucial as a tool for achieving not only high performance of knowledge generation, but also an increased quality of the resulting information. It has been proven that not the simplified accumulation of data, but its integration will be the key to a new knowledge discovery. A trend in the past several years shows that many repositories were redesigned into highly integrated databases; many proteomics databases have implemented new algorithms for data analysis, prediction and error checking; many databases were inter-connected and integrated into larger databanks, utilizing state-of-the-art information technology methods like artificial intelligence. This trend shows the extreme need for data integrability and interdisciplinary view over the vast amount of generated experimental data in proteomics.

REFERENCES

1. **Altschul S.F., Gish W., Miller W., Myers E.W. and Lipman D.J.** (1990) J. Mol. Biol., **215**, 403-410.
2. **Altschul S.F., Madden T.L., Schäffer A.A., Zhang J., Zhang Z., Miller W. and Lipman D.J.** (1997) Nucleic Acids Res., **25**, 3389-3402.
3. **Attwood T.K. and Higgs P.G.** (2005) Bioinformatics and Molecular Evolution. Blackwell Science, p. 365.
4. **Baldi P. and Brunak S.** (2001) Bioinformatics and machine Learning Approach. MIT Press, Cambridge, Mass. USA.
5. **Berman H.M., Henrick K., Nakamura H.** (2003) Nature Structural Biology, **10**(12), p. 980.
6. **Brusniak M.Y., Bodenmiller B., Campbell D., Cooke K., Eddes J., Garbutt A., Lau H., Letarte S., Mueller L.N., Sharma V., Vitek O., Zhang N., Aebersold R., Watts J.D.** (2008) BMC Bioinformatics, **9**(1), p. 542.
7. **Bryson K., McGuffin L.J., Marsden R.L., Ward J.J., Sodhi J.S. and Jones D.T.** (2005) Nucleic Acids Res., **33**(Web Server issue), W36-W38.
8. **Cai Y., He J., Li X., Lu L., Yang X., Feng K., Lu W., Kong X.** (2008) J. Proteome Res., Dec. 19, [Epub ahead of print].
9. **Chen K., Kurgan L.** (2007) Bioinformatics, **23**(21), 2843-2850.
10. **Cole C., Barber J.D. and Barton G.J.** (2008) Nucleic Acids Res., **36(**suppl. 2), W197-W201.
11. **Dalton J.A.R. and Jackson R.M.** (2007) Bioinformatics, **23**(15), 1901-1908.
12. **Duan M.J., Zhou Y.H.** (2005) Genomics Proteomics Bioinformatics, **3**(4), 218-224.
13. **Eswar N., Marti-Renom M.A., Webb B., Madhusudhan M.S., Eramian D., Shen M., Pieper U., Sali A.** (2006) Comparative Protein Structure Modeling With MODELLER. Current Protocols in Bioinformatics, John Wiley & Sons, Inc., **Supplement 15**, 5.6.1-5.6.30, p. 200.
14. **Frishman D. and Argos P.** (1996) Protein Engineering, **9**, 133-142.
15. **Jones D.T.** (1999) J. Mol. Biol., **292**, 195-202.
16. **Kabsch W., Sander C.** (1983) Biopolymers, **22**(12), 2577-2637.
17. **Kortagere S., Chekmarev D., Welsh W.J., Ekins S.** (2008) Pharm. Res., Dec. 30, [Epub ahead of print].
18. **Murzin A.G., Brenner S.E., Hubbard T., Chothia C.** (1995) J. Mol. Biol., **247**, 536-540.
19. **Nagaraj V.H., O'Flanagan R.A., Sengupta A.M.** (2008) BMC Biotechnol., **8**(1), p. 94.
20. **Needleman S.B., Wunsch C.D.** (1970) J. Mol. Biol., **48**(3), 443-453.
21. **Orengo C.A., Michie A.D., Jones D.T., Swindells M.B., Thornton J.M.** (1997) Structure, **5**, 1093-1108.
22. **Otto T.D., Guimaraes A.C., Degrave W.M., de Miranda A.B** (2008) BMC Bioinformatics, **9**(1), p. 544.
23. **Piana S., Laio A., Marinelli F., Van Troys M., Bourry D., Ampe C., Martins J.C.** (2008) J. Mol. Biol., **375**(2), 460-470.
24. **Rost B., Yachdav G. and Liu J.** (2004) Nucleic Acids Res., **32**(Web Server issue), W321-W326.
25. **Smith T.F., Waterman M.S.** (1981) J. Mol. Biol., **147**, 195-197.
26. **Subramaniam S., Mohmmed A., Gupta D.** (2009). J. Biomol. Struct. Dyn., **26**(4), 473-80.
27. **Sundstrom J.M., Sundstrom C.J., Sundstrom S.A., Fort P.E., Rauscher R.L., Gardner T.W., Antonetti D.A.** (2009) J. Proteome Res., Jan 6, [Epub ahead of print].
28. **The UniProt Consortium** (2008) Nucleic Acids Res., **36**, D190-D195.
29. **Tsukamoto K., Yoshikawa T., Hourai Y., Fukui K., Akiyama Y.** (2008) J. Bioinform. Comput. Biol., **6**(6), 1133-1156.
30. **Xue B., Faraggi E., Zhou Y.** (2008) Proteins, Nov. 18, [Epub ahead of print].
31. **Zvelebil M., Baum J.** (2008) Understanding Bioinformatics, Garland Science.