Constraints, 6, 141–156, 2001


# Bioinformatics and Constraints

ROLF BACKOFEN backofen@informatik.uni-munchen.dc
*Institut für Informatik, Ludwig-Maximilians-Universität München, Oettingenstraße 67, D-80538 München*

DAVID GILBERT drg@soi.city.ac.uk
*Department of Computing, City University, Northampton Square, London EC1V 0HB, UK*

**Abstract.** This article introduces the topic of bioinformatics to an audience of computer scientists. We discuss the definition of bioinformatics, give a classification of the problem areas which bioinformatics addresses, and illustrate these in detail with examples. We highlight those areas which we believe to be suitable for the application of constraint solving techniques, or where similar techniques are already used. Finally, we give some advice for computer scientists who are considering getting involved in bioinformatics, and provide a resource list and a reading list.

**Keywords:** bioinformatics, computational biology, molecular biology, constraints, databases, algorithms, classification, alignment, comparison, patterns, genome, DNA, RNA, proteins, metabolic pathways, regulatory networks, phylogenetic trees

## 1. What is Bioinformatics?

In this article we aim to introduce the topic of bioinformatics to an audience of computer scientists, and highlight those areas within the subject which we believe to be suitable for the application of constraint solving techniques, or where similar techniques are already used. Bioinformatics is an exciting and rapidly developing field, and we hope that we haven't predicted all the developments in the next few years!

One of the first issues that need to be addressed is what is meant by "bioinformatics"—it is already almost a colloquial word in the scientific community, but its interpretation varies widely. The word bioinformatics has two obvious components—"bio-" and "informatics"; we deal with each of these in turn.

At present the widely accepted interpretation of the "bio" part is *molecular biology*, i.e. the study of the structure and activity of macromolecules essential to life. However, there are other areas within biology which can be considered to be within the remit of bioinformatics, for example the study of evolution, and genetics.

Informatics is a word which has not "officially" entered the English language, but the French, German and Russian traditions broadly agree that its meaning coincides with "computer science." Thus one definition of informatics is "the science of systematic processing of information, using modeling and abstraction of the concrete realisation."

Thus, when considering both parts of the word, we consider the proper meaning to be solving *problems* arising from biology using methodology from computer science. We are strongly of the opinion that bioinformatics is *not* about implementing the mechanisation of existing solutions to problems stemming from biology; in general bioin-

formatics requires either a variant of an existing algorithm from the domain of computer science, or the design of a new algorithm.

An alternative term, more or less coinciding with bioinformatics is *computational biology*, used more in North America than in Europe. Waterman [44] considers that there are three interpretations, all of which are valid:

> One, that it is a subset of biology proper and any required mathematic and computer science can be made up on demand; two, that it is a subset of the mathematical sciences and that biology remains a remote but motivating presence; three: that there are genuine interdisciplinary components, with the original motivation from biology suggesting mathematical problems, which suggest biological experiments.

Of course, there are other fields which broadly apply principles from biology to derive novel approaches in computer science, for example biocomputing, neural computing, genetic algorithms, and evolutionary computing.

Since it is rare to find researchers who are both computer scientists and biologists, it is generally the case that effective research in bioinformatics requires the joint effort from scientists in both fields. An important corollary is that in order to achieve such cooperation all parties must use a common language and be prepared to learn about issues from the other side. In fact many researchers from the biological and physical sciences working in bioinformatics have acquired significant computing skills, and may have greater specialist knowledge in mathematics and statistics than do many computer scientists. An illustration of this is the heavy use of hidden Markov models in bioinformatics, a topic about which most computer scientists know very little. It is the computer scientist's task to apply the approach of problem abstraction together with efficient algorithm design to the problems from the biological domain.

A challenge for computer scientists who are involved in research in bioinformatics is to achieve results that make a contribution to computer science. Of course this is not the main motivation for biologists; moreover there are some exciting projects in bioinformatics which in the short to medium term are unlikely to contribute to computer science.

### *1.1. What Biologists Want from Bioinformatics*

The great aim of research in bioinformatics is to understand the functioning of living organisms in order to "improve the quality of life." This improvement will be achieved by many means including drug design, identification of genetic risk factors, gene therapy, genetic modification of food crops and animals, etc. Some of these, especially the last, are proving to be controversial.

### *1.2. The Central Dogma*

The study of proteins, how they interact with each other, and how genes are regulated is central to the understanding of the basic principles of the functioning of living organisms.

Proteins comprise approximately 60% of the dry mass of a living cell, and are linear heteropolymers that are constructed from a chain or sequence of monomers called amino acids, of which twenty different types are involved in the composition of proteins. It is widely accepted that the function of proteins (and RNA) is determined by their structure, and it is known that structure is determined uniquely by the sequence of amino acids, or nucleotides in the case of RNA.

The central dogma of information flow in biology essentially states that the sequence of amino acids making up a protein and hence its structure (folded state) and thus its function, is determined by transcription from DNA via RNA.

> The central dogma states that once 'information' has passed into a protein it cannot get out again. The transfer of information from nucleic acid to nucleic acid, or from nucleic acid to protein, may be possible, but transfer from protein to protein, or from protein to nucleic acid, is impossible. Information here means the precise determination of sequence, either of bases in the nucleic acid or of amino acid residues in the protein. Francis Crick [11]

Thus bioinformatics is concerned in a major way with the elicitation of DNA sequences from genetic material, the annotation of whole or part of sequences with biologically relevant properties (e.g. function, location, context) the control of gene expression (i.e. under what circumstances proteins are transcribed from DNA), and the relationship between the amino acid sequence of proteins and their structure. At present, the only physical methods to determine protein structure are X-ray crystallography and NMR (nucleo-magnetic resonance), both of which are not only very time-consuming, but cannot be applied to all classes of proteins. One of the holy grails of bioinformatics is to develop computational methods to determine protein structure from amino-acid sequence.

## 2. Bioinformatics Today

### *2.1. A Classification of Problem Areas*

The problem areas in Bioinformatics can be broadly divided into three classes:

**Problems Specifically Related to the Central Dogma:** This includes both those related to a specific level of information, i.e. sequence, structure or function, and those that encompass more than one level. Examples of problems at specific levels are pattern discovery in sequences, structure alignment and analysing patterns of functional relationships within metabolic pathways. Perhaps the most illustrative example of a multilevel problem is determining the relationship between sequence, structure and function for a group of proteins.

**Problems Related to Data in General:** With the exponential growth of knowledge in (molecular) biology, there are rapidly growing problems such as storage, retrieval, and analysis of the data. Hence there are issues of database design for biological resources, representation and visualisation of biological knowledge, and the application of data

analysis methods such as data mining. A key underlying technique is that of **abstraction** of the data; it is of course imperative that the operations over the abstract data preserve the biological meaning of the operations on the original form of the data.

**Simulation of Biological Processes:** This means in general the prediction of dynamic behaviour of a biological system on the basis of its components. Examples include the simulation of protein folding (molecular dynamics) or of metabolic pathways.

In the following we concentrate on the first class of problems, i.e. sequence, structure and function, and select a subset of illustratory examples.

## *2.2. Sequence Related Problems*

### *2.2.1 Physical Map*

In this problem, one has a collection of short, known substrings of the DNA called *probes* with the property that they occur *exactly once* in the DNA, and a set of fragments of the DNA (called *clones*), which (ideally) cover a specific region of interest on the DNA. For both the clones and the probes, the exact location on the DNA and the ordering of the locations are not known. The goal is to find the ordering of the probes and/or clones in the DNA.

The first step is to check for every probe $i$ and every clone $j$, whether clone $j$ contains the substring denoted by probe $i$. This is done by performing hybridisation experiments. This gives a matrix $(a_{ij})$, where $a_{ij}$ is 1 if probe $i$ is on clone $j$, otherwise 0. Now if there were no error in the hybridisation experiment, then the ordering of the probes could be found be reordering the rows and columns of the matrices such that the resulting matrix has the *consecutive ones* property. But since the experiments are faulty, the problem of finding the ordering minimizing the errors is NP-complete (see e.g., [10], [18]).

The ordering of the probes, usually together with a good bound on the distance between to successive probes, constitutes a physical map, which can be used for different purposes. One is to use this map when sequencing the genome. The reason is that sequencing is done by splitting DNA into fragments, which are sequenced in the sequel. The remaining problem is to generate the original DNA-sequence out of sequenced fragments. This is usually done by searching for overlapping fragments. The problem is that DNA contains so-called *repeats*. This are long fragments of DNA which are repeated several times on the DNA. Clearly, such repeats may not be used for the process of generating the original DNA sequence out of overlapping fragments. One way to check this is to use a physical map.

There are several other important computational problems involved in sequencing a genome, but we do not describe them in more detail here.

### *2.2.2 Comparison and Alignment*

The goal of this activity is to compare two sequences or structures, and in addition to return an alignment, i.e. some information regarding those parts which are very similar.

In general, sequence alignment is fast, whereas structure alignment is slow due to its high complexity.

One of the first fields in bioinformatics was DNA sequence alignment. The reason for the interest in sequence alignment stems from the fact that there are many different proteins which have common ancestors, and that these *homologous* (i.e., related by evolution) proteins have a similar structure and function. In addition, homologous proteins often have similar sequences. Using a reverse reasoning, sequence similarity is used to detect the homology of protein structures.

Clearly, the quality of this approach depends on the similarity measure used, which is determined by a model of evolution. The usual approaches use a model with substitution, deletion or insertion of a single amino-acid (see e.g. [44] for an overview). In this case, sequence alignment can be performed in polynomial time using a dynamic programming approach. There are also new approaches which deal with more complex models of evolution such as [3], who considers in addition duplication of substrings (tandem repeats). A more complex problem is that of multiple sequence alignment [28].

On the level of structure comparison, there are approaches to compare two different structures by superposing elements using translation and rotation to minimise the atomic coordinate Root Mean Square Deviation (RMSD) [14]. Structures can also be compared at a higher level of abstraction than atomic coordinates by using a topological approach based on secondary structure elements [20]; this can be performed over topology graphs by detecting maximal cliques [29] or by pattern discovery and structural alignment [27].

Protein threading uses a mixture of sequence alignment and structure alignment. In this approach an alignment is made between two sequences, one with an unknown structure and the other with a known structure, taking into account the known structure [32]. This is done by identifying core segments in the structure connected by loop regions of variable length, and then attempting to align, with gaps if necessary, the sequence of unknown structure with those parts of the sequence of known structure which correspond to the core structural elements. This problem has been shown to be NP-hard.

### *2.2.3 Pattern Matching and Pattern Discovery*

In both sequences (DNA and RNA) as well as structures (RNA and protein), there are functionally significant regions that are repeated in different entities; these regions can be often described by patterns. A need has arisen to be able to search through genome or protein databases (which may be very large), and identify entries which match the pattern. Obviously, this has a parallel in formal language theory, see for example Searls' excellent discussion in [38]. In reality, biological data is noisy, and in the case of string languages, stochastic approaches have been developed using for example hidden Markov models [12] and stochastic context-free grammars [31].

Although patterns can be constructed by hand, it is preferable in the general case to use a mechanised (machine learning) approach, i.e., *pattern discovery* [4, 35], with patterns

being validated by an expert. Finding sites regulating gene expression in DNA may require context sensitive patterns.

One active research field is to design appropriate pattern languages and associated discovery mechanisms which are able to express significant properties of structures as opposed to strings [20], [25].

Pattern discovery can also be performed over structures and metabolic pathways, but we do not discuss these very new areas in this paper.

### *2.2.4 Phylogenetic Trees*

If we have any set of species that are related, then the relationship between these species (resp. entities) is called a *phylogeny*. When constructing a phylogenetic tree, the task is to set up a tree to show how the different species have evolved from a common ancestor. In addition, the trees generated are often labelled. The labels indicate the time when the species evolved from a common ancestor, or any other measure of the distance between the different species. Note that the construction of phylogenetic trees is not necessarily applied to species, but to any kind of entities where we can set up some sort of distance information (e.g., phylogenetic trees can be constructed for languages). In this case the tree constructed may not be rooted.

The problem of constructing phylogenetic trees can be formulated in different ways. The first one is to have a finite set of species or entities $S = \{e_1, \ldots, e_n\}$, and a distance matrix $(d_{ij})_{i,\, j \in [1 \ldots n]}$ containing the pairwise distances between the entities. The problem is to construct a tree, where the edges are labeled by distances and the nodes are labeled entities (using new entities for the inner nodes). The tree is correct if for each two entities $e_j, e_k$ from $S$, the distance in the tree (by summing up the edges distances along the path connecting them) out of the ordinal set in the tree is $d_{jk}$. Trees can be constructed from pairwise distances by variety of methods, including UPGMA (unweighted pair group method using arithmetic averages) [39].

Another formulation of the phylogenetic tree construction problem is *parsimony* [17]. Here, one has a set $S$ of sequences (DNA or protein), and a method for calculating costs for relating any two sequences (not restricted to $S$). The task is then to find a tree, where the leafs are labeled by elements of $S$ and the inner nodes are labeled by other sequences. Furthermore, the tree should have minimal costs according to the given method (i.e., the sum of distances between any two sequences that are directly connected in the tree should be minimal).

Since one, or in the case of parsimony several, optimal trees can be generated by tree building algorithms, an approach such as the bootstrap method [16] is commonly used to assess the significance of some phylogenetic feature and thus give some measure of confidence for the tree.

### *2.3. Structure Related Problems*

#### *2.3.1 Structure Prediction*

Here one is concerned about the relation between sequence and structure. The sequence can either be from a protein, in which case the problem is sometimes referred as the *protein folding problem*; a more simple variant is that of RNA folding.

Now for natural protein sequences, the protein folds into one stable structure (which is believed to be a structure where the free energy has a global minima), which is completely determined by its amino acids sequence. This native structure determines the function of a protein. Since it is very easy to determine the sequence of a protein, the structure prediction problem consists of determining the structure from a given sequence. This is one of the holy grails of bioinformatics, since protein structure prediction is a very important but notoriously hard problem. It is subject of many ongoing attempts to solved this problem by a variety of methods (see for example the CASP competitions [7], [37]). Note that for artificial sequences, the sequence usually does not determine the structure (i.e., the artificially designed protein will not fold to a stable structure in general).

Proteins have a high level of local organisation (called secondary structure), which consist of $\alpha$-helices, $\beta$-strands and turns. For this reason, there are approaches for predicting secondary structure, where the overall tertiary structure, as well as approaches with try to predict tertiary structure directly. It is presently believed that protein structure prediction cannot be done purely on the level of secondary structure alone.

A problem related to the protein folding problem is the *inverse protein folding*, which consists of the following. Given a three-dimensional structure, generate a sequence that will fold uniquely into the given structure. Naively, this can be solved using structure prediction (generate a sequence, then predict the structure, and compare the result with the given structure). Clearly, this problem is of interest for drug design, although inverse protein folding is not used in drug design yet. The reason simply that the problem is unsolved (see e.g. [22], where this problem is treatment for lattice proteins).

For RNA, secondary structure is usually related to base pair bonding, and structure prediction is possible on this level (under some restrictions) taken into account thermodynamical energies [46]. However, for another approach see [15] in this volume.

#### *2.3.2 Protein Docking and Ligand Binding*

Protein docking attempts to find the most stable mode of association between two protein molecules, starting from the atomic coordinate of the two isolated components. It can be likened to a 'lock and key' mechanism, where both lock and key are plastic, and distort according to mutual interactions. The protein-protein interfaces are closely packed, similar to protein cores. The aim of any docking algorithm is to optimise the surface area and attractive forces and to minimise the loss of energy due to interaction with the solvent. This is a difficult area of research, but there are general rules. Optimisation must be performed on many degrees of freedom, since this is an example of 6-D problem of

rigid body movement—3 translations and 3 rotations, all of which must be searched. The approaches to rigid surfaces are broadly

1. Given the information of a pair of proteins crystallised together, to *reconstruct* the docking
2. Given the individual proteins separately crystallised, to *predict* their docking requires trying all combinations of degrees of freedom. Note that *ligand binding*—small ligands tend to bind in big pockets; ligands are more flexible than proteins

### *2.4. Function Related Problems*

#### *2.4.1 Metabolic Pathways*

Living organisms function by a complex set of interactions at the molecular level which occur in a highly organised manner. They involve metabolic reactions which transform some compounds (*substrates*) into others (*products*). In general a reaction $S \to P$ can be described by a transition $S \to S' \to P$, transforming the set of substrates $S$ into the set of products $P$ via a transition state $S'$ in which the substrate molecules are distorted into some electronic conformation which more readily converts to the products. In order to occur, $S \to P$ has a negative free energy, i.e. the free energy of $S$ is greater than that of $P$; however $S \to S'$ has a *positive* free energy change, termed the energy of activation. This energy is a barrier preventing $S \to P$ occurring spontaneously, without which all reactions would occur in an uncontrolled way. Most reactions are catalysed by special proteins called *enzymes* which control the reaction by lowering the energy barrier (i.e. increasing the rate of flow). They do this by binding substrates at combining sites within active sites, positioning substrate molecules in the most favourable orientations for reactions to occur, as well as distorting them in order to favour transition state formation. During this process the enzyme may change shape in order to induce a fit with the substrate, rather than just rely on a rigid 'lock and key' mechanism. In general, reactions can be chained together into paths so that the products of one reaction become the substrates of another [13].

#### *2.4.2 Regulatory Networks*

Metabolic reactions can be regulated in two ways. The first is by the direct activation or inhibition of activity of enzymes by small molecules. This method is relatively fast in action, since it directly affects the chain of reactions. Another method of regulation is that of transcriptional regulation, in which the production of the enzyme itself is controlled by a transcription factor (a protein which activates the capacity of a gene to produce another protein). This method is relatively slow, since it indirectly affects the reaction path.

Reactions can be self-regulated using either the direct or transcriptional method, since it is common that products of an immediate or eventual reaction act have a direct

or transcriptional effect on enzymes involved earlier in the chain of reactions. These regulatory relationships can be quite complex in that products from one path can regulate enzymes involved in another path.

## 3. Bioinformatics and Constraints

We now describe problem areas in bioinformatics, where constraints have been used successfully, or where we think that constraints can be useful. The reader should note that computing in bioinformatics is often data driven. Since the data is empirical it can often be "dirty" and moreover the biologists who interpret the data in general will not be able to classify the data cleanly. The effect of this is that the constraints used to describe the data will have to reflect this by being stochastic or alternatively "soft" rather than being "hard"; this will also have an effect on the design of the constraint solvers to be used. However, some problems in bioinformatics can be formalised as pure optimisation problems and do not require stochastic approaches.

### *3.1. NMR Structure Determination*

When protein structure is determined by the NMR technique, then the result is not a unique structure, but a set of data that can easily be interpreted as a set of distance constraints. The problem of determining the structure from this data set is equivalent to finding atom coordinates which are consistent with the given distance constraints. Of course, there may be errors, which lead to inconsistent constraints. In this case, the task is to find a (maximal) consistent subset as well. This is the problem handled by the technique called "Distance Geometry" [9]. Distance geometry is also used for the problem of homology protein modeling, where the distance constraints are obtained by an alignment of the new sequence with templates of the known structure (see e.g. [24], [23], [42], [40], [19]; for an overview on the subject see [43]). An alternative approach for structure determination using threading techniques is given in [6]. A recent approach using constraint solving techniques is given in [27].

### *3.2. Alignment and Threading*

Sequence alignment and protein threading are both used in the case of homologous proteins, where one of the sequences has a known structure, and one wants to know the structure of the other sequence. In sequence alignment, only the sequences are investigated in order to find conserved regions, whereas in protein threading the known structure is also considered.

When considering multiple sequences, then the problem of aligning these sequences is known to be NP-complete [28]. In [36], an integer linear programming (ILP) approach for solving the problem has been described. This is of interest for the constraint community,

since (1) an ILP-formulation is a special kind of constraint formulation, (2) it is also interesting to integrate e.g. user defined constraint, where an ILP-formulation does not work anymore. The latter approach has been described in [33].

In the case of protein threading, the problem is also known to be NP-complete [32], where a branch-and-bound approach for solving this problem was introduced. It is even hard to approximate the problem, as shown in [1]. Constraint optimization could be used for a more efficient algorithm, or for integrating user-defined constraints as it is done in multiple sequence alignment.

### *3.3. Protein Structure Prediction, Protein Docking*

The problem of ab initio structure prediction is described as follows: Given a sequence of amino-acids, what is the native structure this sequence will have? Since the problem is very important, but very hard (for NP-completeness results see e.g. [34], [41]), there are a lot of different formulations of this problem. For the constraint community, simplified models are of special interest. There is a huge variation of simplified models.

The first class are the off-lattice models, where one describes a backbone conformation (i.e., the positions of the $C_\alpha$ atoms, the $N$ atoms of the amino groups, and the $C$ atoms of the carboxy groups) as a combination of $\phi$-angles (i.e., the rotation angles between $N$ and $C_\alpha$ atoms) and $\psi$-angles (i.e., the rotation angles between $C_\alpha$ and $N$ atoms). Since there are preferred combinations of $\phi, \psi$-angles, a discrete set of combinations are used in the off-lattice models. For finding the minimal energy model, one can span the search tree over all possible combinations. Constraints can be used to cut of the search tree, as it is done in the Geocore system of Yue and Dill [45].

The other class consist of the lattice models, where the positions of the amino acids are positions in a regular lattice. Here, constraints have been already been successfully used for the cubic lattice model. Even for the simplest lattice model (Dill's HP-model [30]), the problem of finding the conformation has been shown to be NP-complete in two dimensions [8] and in three dimensions [5].

Protein docking uses algorithms to optimise the surface area and attractive forces and to minimise the loss of energy due to interaction with the solvent. To the best of our knowledge there has been no work which has attempted to apply constraint optimisation techniques to protein docking, and thus this difficult area is one which should be seen as a challenge to the constraint programming community.

### *3.4. Metabolic Pathway Analysis*

Although data about reactions has been available for some time, the recent flood of information about the genome, transcription to proteins and protein structure has made this area one of the 'hot topics' in bioinformatics. Overall, metabolic pathways and their associated regulatory networks are complex, since they can describe the functioning of entire organisms, and are often represented as graphs; the amount of data involved can be

very large. The types of analysis performed can range from *simulation* of the reactions and calculation of the quantities of substrates and products, to a topological *analysis* of the patterns of connectivity (reactions and regulation) within and between pathways.

Some questions which could be addressed by such an analysis might include the number of pathways, and how many steps within each pathway, which lead from one compound to another, those pathways that contain or lack specified compounds, and identifying those paths which may be affected when one or more genes or proteins are turned off or are missing. At a more general level, there is a need to compare biochemical pathways from different organisms and tissues in terms of both common features and differences. Pathway reconstruction aims to predict elements which are missing from experimental data; indeed data in pathway databases can be annotated with different levels of certainty, ranging from empirical to that computed from other data (e.g. protein structure as predicted from sequence data).

Constraints have not yet been used in any large scale for in the area of metabolic pathways, unlike in chemistry [26], but could be used to describe patterns in the graphs (for example constraints over the number of nodes that a metanode in a pattern may represent), or as part of the equations describing the chemical reactions.

### *3.5. Patterns and Databases*

The amount of data now available to researchers working in the field of bioinformatics is not only large, but also increasing rapidly at a great rate. This situation has arisen due to advances in the technology associated with data collection; the need is now to analyse this data.

One approach is to induce or learn patterns or rules (i.e. constraints) from bio-data. Because such data is often "dirty" and in any case describes populations whose individuals vary in some or all of their attributes, stochastic constraints will often be appropriate; alternatively pattern languages which describe disjunction can be used to describe clustered observations. If the objective is to discover classifying patterns[1] then learning must be over negative as well as postitive examples, implying that some form of negation needed—in general this is hard. The challenge for the constraint community is to rival the approaches that have been well established in this field, including Hidden Markov Models and stochastic grammar (see [2], [12]).

Furthermore, one important source of data are scientific texts, and there is now a great deal of interest in extracting high-level information from large sets of such texts. The combination of constraint–based induction and textual analysis is likely to be a very fruitful area in the future.

## 4. Conclusions

The speed at which data collection is being made is indeed impressive if not somewhat daunting—for example the first pass at sequencing of the human genome has been com-

pleted in the summer of 2000. Thereafter, of course, effort will be put into sequencing different populations and also individuals.

In general we expect that the emphasis on research in sequencing will decrease whilst research on sequence analysis (e.g. gene finding and annotation) will increase. Also we predict that there will be an increase in the research emphasis on structure and function-related topics, and that research in the area of metabolic pathways will become very active.

Overall there will be an increasing need for computer scientists to be involved in the way in which the large volumes of data generated in the bioinformatics field will be manipulated, presented and reasoned about; this in itself will create new research problems.

Some tips to bear in mind for computer scientists who are considering getting involved in bioinformatics are:

- Work alongside with molecular biologists otherwise as a computer scientist the danger is that you design the solution to a problem, neither of which have any relevance to the community.
- There are a lot of problems out there in bioinformatics which look fun to tackle, but do check out what has been done before by people from the biological sciences.
- You will need to carefully and rigorously test any systems that you develop and this will have to be done with scientists who can interpret the results—i.e. people from the biological sciences. Since the Web is heavily used to deliver applications in the bioinformatics community, be prepared to interface your system to the Web.
- The bottom line is that biologists are interested the speed of solutions, and are used to systems delivering results which are not 'black-and white' (i.e. trading off coverage against accuracy) from dirty data. If your system is slower than the one they already have then you will have to convince them of its extra functionality, but don't expect that they will appreciate its declarative nature, at least at first.

Our experience has convinced us that constraint programming can be used for real on these large and challenging problems, and if you find the area interesting, then do go for it!

## Resources

European Bioinformatics Institute: www.ebi.ac.uk

National Center for Biotechnology Information: www.ncbi.nlm.nih.gov

Protein Data Bank: www.rcsb.org/pdb

Swiss-Prot Database: www.expasy.ch/sprot/sprot-top.html

CATH Database of Folds: www.biochem.ucl.ac.uk/bsm/cath

SCOP Database: scop.mrc-lmb.cam.ac.uk/scop

DALI: www2.ebi.ac.uk/dali

Structural Genomics: www.structuralgenomics.org

3D Search: gene.stanford.edu/3dsearch

Bioinformatics course: cmgm.stanford.edu/biochem201

The Bioinformatics Resource: www.hgmp.mrc.ac.uk/CCP11

Pattern discovery: industry.ebi.ac.uk/brazma/patterns.html

Metabolic: www.ebi.ac.uk/research/pfmp

## Reading List

Attwood, T. & Parry-Smith, D. J. (1999). *An Introduction to Bioinformatics*. Longman.

Baldi, P. & Brunak, S. (1998). *Bioinformatics: The Machine Learning Approach*. MIT Press.

Branden, C. & Tooze, J. (1998). *Introduction to Protein Structure*. Garland Publishing.

Clote, P. & Backofen, R. (2000). *Computational Molecular Biology*. John Wiley & Sons, Ltd. Chichester.

Creighton, T. E. (1992). *Protein Folding*. New York: Freeman.

Creighton, T. E. (1993). *Proteins: Structures and Molecular Properties*, 2nd ed. New York: Freeman.

Dary, N. J. & Creighton, T. E. (1993). *Protein Structure*. Oxford: IRL Press.

Durbin, R., Eddy, S., Krough, A. & Mitchison, G. (1998). *Biological Sequence and Analysis*. CUP.

Gusfield, D. (1997). *Algorithms on Strings, Trees and Sequences*. CUP.

Searls, D. B. (1995). String variable grammar: a logic grammar formalism for the biological language of DNA. *Journal of Logic Programming*, 24(1–2): 73–102.

Schulz, G. E. & Schirmer, R. H. (1985). *Principles of Protein Structure*. New York: Springer-Verlag.

Stryer, L. *Biochemistry*, 4th ed. (1995). Freeman.

Waterman, M. (1995). *Introduction to Computational Biology*. London: Chapman & Hall.

## Notes

1. Given a set of data believed to belong to a certain family and another set believed to be outside this family, a *classifier function* returns TRUE for members of the family, and FALSE for non-members. When only family members are given, then a *conservation function* encapsulates features characterising the family [4].

## References


1. Akutsu, T., & Miyano, S. (1997). On the approximation of protein threading. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, NM, pages 3–8.
2. Baldi, P., & Brunak, S. (1998). *Bioinformatics: The Machine Learning Approach*. Cambridge, MA: MIT Press.
3. Benson, G. (1997). Sequence alignment with tandem repeats. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, pages 27–36.
4. Brazma, A., Jonassen, I., Eidhammer, I., & Gilbert, D. R. (1998). Approaches to the automatic discovery of patterns in biosequences. *Journal of Computational Biology*, 5(2): 277–303.
5. Berger, B., & Leighton, T. (1998). Protein folding in the hydrophobic-hydrophilic (HP) modell is NP-complete. In *Proceedings of the Second Annual International Conferences on Compututational Molecular Biology (RECOMB98)*, New York, pages 30–39.
6. Baxter, K., Steeg, E., Lathrop, R., Glasgow, J., & Fortier, S. (1996). From electron density and sequence to structure: Integrating protein image analysis and threading for structure determination. In *Proceedings of the 4th International Conference on Intelligent Systems for Molecular Biology (ISMB'96)*, St. Louis, MO, pages 25–33.
7. CASP3. (1998). Third community wide experiment on the critical assessment of techniques for protein structure prediction, http://predictioncenter.llnl.gov/casp3/casp3.html.
8. Crescenzi, P., Goldman, D., Papadimitriou, C., Piccolboni, A., & Yannakakis, M. (1998). On the complexity of protein folding. In *Proceedings of STOC*. Short version in *Proceedings of RECOMB'98*, pages 61–62.
9. Crippen, G. M., & Havel, T. F. (1988). *Distance Geometry and Molecular Conformation*. Taunton, Somerset, UK: Research Studies Press.
10. Christof, T., Juenger, M., Kececioglu, J., Mutzel, P., & Reinelt, G. (1997). A branch-and-cut approach to physical map with end-probes. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, NM, pages 84–92.
11. Crick, F. H. C. (1958). On protein synthesis. *Symposium of the Society of Experimental Biology*, 12: 138–167.
12. Durbin, R., Eddy, S., Krough, A., & Mitchison, G. (1998). *Biological Sequence Analysis*. CUP.
13. Elliott, W. H., & Elliott, D. C. (1997). *Biochemistry and Molecular Biology*. OUP.
14. Eidhammer, I., Jonassen, I., & Taylor, W. R. (1999). Structure comparison and structure patterns. Technical Report 174, Department of Informatics, University of Bergen, Bergen, Norway.
15. Eidhammer, I., Jonassen, I., Grindhang, S. H., Gilbert, D., Ratnayke, M., (2000). A constraint based structure description language for biosequences, *Constraints,* 6: 173–200.
16. Felsenstein, J. (1985). Confidence limits on phylogenies: an approach using the bootstrap. *Evolution*, 39: 783–791.
17. Fitch, W. M. (1971). Toward defining the course of evolution: minimum change for a specified tree topology. *Systematic Zoology*, 20: 406–416.
18. Greenberg, D. S., & Istrail, S. (1995). Physical mapping by sts-hybradisation: algorithmic strategies and the challenge of software evaluation. *Journal of Computational Biology*, 2(2): 219–273.
19. Gunn, J. R. (1998). Hierarchical minimization with distance and angle constraints. In *Proceedings of the 6th International Conference on Intelligent Systems for Molecular Biology (ISMB'98)*, Montréal, Québec, Canada.
20. Gilbert, D. R., Westhead, D. R., Nagano, N., & Thornton, J. M. (1999). Motif-based searching in tops protein topology databases. *Bioinformatics*, 15(4): 317–326.

21. Gilbert, D., Westhead, D., Thornton, J., & Viksna, J. (1999). Tops cartoons: formalisation, searching and comparison. *RECOMB99 (poster)*.

22. Hart, W. E. (1997). On the computational complexity of sequence design problems. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, NM, pages 128–136.

23. Havel, T. (1993). Predicting the structure of the flavodoxin from *Escherichia coli* by homology modeling, distance geometry and molecular dynamics. *Molecular Simulation*, 10: 175–210.

24. Havel, T., & Snow, M. (1991). A new method for building protein conformations from sequence alignments with homologoues of known structure. *Journal of Molecular Biology*, 217: 1–7.

25. Jonassen, I., Eidhammer, I., & Taylor, W. R. (1999). Discovery of local packing motifs in protein structures. *Proteins*, 34(2): 206–219.

26. Jourdan, J., & Valdés-Pérez, R. E. (1989). Constraint logic programming applied to hypothetical reasoning in chemistry. In E. L. Lusk and R. A. Overbeek, eds., *Logic Programming, Proceedings of the North American Conference*, Cleveland, pages 154–172. Cambridge, MA: The MIT Press.

27. Krippahl, L., & Barahona, P. (1999). Applying constraint programming to protein structure determination. In J. Jaffar, ed., *Proceedings Principles and Practice of Constraint Programming—CP99*, pages 289–302. New York: Springer-Verlag.

28. Kececioglu, J. D. (1993). The maximum weight trace problem in multiple sequence alignment. In *Proceedings 4th Symposium Combinatorical Pattern Matching*, volume 684 of *Lecture Notes in Computer Science*, pages 106–119. New York: Springer-Verlag.

29. Koch, I., Lengauer, T., & Wanke, E. (1996). An algorithm for finding maximal common subtopologies in a set of protein structures. *Journal of Computational Biology*, 3(2): 289–306.

30. Lau, K. F., & Dill, K. A. (1989). A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22: 3986–3997.

31. Lefebvre, F. (1996). A grammar-based unification of several alignment and folding algorithms. In D. J. States, P. Agarwal, T. Gaasterland, L. Hunter, & R. Smith, eds., *Proceedings of the Fourth International Conference on Intelligent Systems for Molecular Biology*, pages 143–154. Menlo Park, CA: AAAI Press.

32. Lathrop, R. H., & Smith, T. F. (1996). Global optimum protein threading with gapped alignment and empirical pair score functions. *Journal of Molecular Biology*, 255: 641–665.

33. Myers, G., Selznick, S., Zhang, Z., & Miller, W. (1997). Progressive multiple alignment with constraints. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, NM, pages 220–225.

34. Ngo, J. T., & Marks, J. (1992). Computational complexity of a problem in molecular structure prediction. *Protein Engineering*, 5: 313–321.

35. Rigoutsos, I., & Floratos, A. (1998). Combinatorial pattern discovery in biological sequences. *Bioinformatics*, 14(1): 55–67.

36. Reinert, K., Lenhof, H.-P., Mutzel, P., Melhorn, K., & Kececioglu, J. P. (1997). A branch-and-cut algorithm for multiple sequence alignment. In *Proceedings of the First Annual International Conferences on Compututational Molecular Biology (RECOMB97)*, Santa Fe, NM, pages 241–249.

37. Sternberg, M. J., Bates, P. A., Kelley, L. A., & MacCallum, R. M. (1999). Progress in protein structure prediction: assessment of CASP3. *Current Opinions in Structural Biology*, 9(3): 368–373.

38. Searls, D. (1993). The computational linguistics of biological sequences. In L. Hunter, ed., *Artificial Intelligence and Molecular Biology*, chapter 2, pages 47–120. Menlo Park, CA: AAAI/MIT Press.

39. Sokal, R. R., & Michener, C. D. (1958). A statistical method for evaluating systematic relationships. *University of Kansas Scientific Bulletin*, 28: 1409–1438.

40. Sánchez, R., & Šali, A. (1997). Comparative protein modeling as an optimization problem. *Journal of Molecular Structure (Theochem)*, 398: 489–496.

41. Unger, R., & Moult, J. (1993). Finding the lowest free energy conformation of a protein is an NP-hard problem: proof and implications. *Bull. Math. Biol.*, 55(6): 1183–1198.

42. Šali, A., & Blundell, T. (1993). Comparative protein modelling by satisfaction of spatial restraints. *Journal of Molecular Biology*, 234: 779–815.

43. Šali, A. (1996). Comparative protein modeling by satisfaction of spatial restraints. http://guitar.rockefeller.edu/publications/papers/molmed-95/html/rev.html.

44. Waterman, M. (1995). *Introduction to Computational Biology*. London: Chapman & Hall.

45. Yue, K., & Dill, K. A. (1996). Folding proteins with a simple energy function and extensive conformational searching. *Protein Science*, 5(2): 254–261.

46. Zuker, M. (1989). On finding all foldings of an RNA molecule. *Science*, 244: 48–52.