Chapter 1
Bioinformatics: a Challenge to Constraint Programming

Pedro Barahona, Ludwig Krippahl, Olivier Perriquet

Abstract Bioinformatics is a rapidly growing field at the intersection of biology and computer science. As such, it poses a wealth of problems, opportunities and challenges for both areas. This paper overviews some of these issues, with an emphasis on those that seem most amenable to Constraint Programming (CP) approaches and where CP has made some progress. Since bioinformatics is tightly focused on real-life applications, this paper does not expand on theoretical principles but, rather, tries to give an idea of the practical issues. At this light, the paper briefly presents the selected problems together with the solutions found so far, that illustrate the versatility of CP techniques that have been used in this area and the need to integrate them with other complementary techniques to handle realistic applications.

1.1 Introduction

Bioinformatics arose from the need to manage the large datasets of sequence information generated by molecular biology research. From the outset, this interdisciplinary field was strongly focused on practical applications, and that is still one of its main features. As molecular biology grew from sequence analysis to structural biochemistry, the scope of bioinformatics broadened to include molecular modeling, molecular dynamics and macromolecular interaction simulations. Nowadays, bioinformatics is still growing, ranging in applications from spectroscopy data processing to biodiversity studies and the modeling of evolutionary processes, and in techniques from simulated annealing to reasoning over ontologies. Of this large field and diverse applications, this article will focus on some problems that seem most amenable to constraint programming (CP) and declarative approaches and on some CP, and related, solutions that have been proposed so far. The goal is to pro-

Pedro Barahona
Centro de Inteligncia Artificial, Dep. de Informtica, Universidade Nova de Lisboa, 2825 Monte de Caparica, Portugal e-mail: pb@di.fct.unl.pt

vide an overview of some interesting applications of CP in this area and to share the authors' perspective on what may be the main challenges in this endeavor.

1.1.1 Data sources

One of the most salient features of current bioinformatics is the abundance of data, which is often overwhelming in both quantity and complexity. Taking bioinformatics in a broad sense, in some areas the information is proprietary and may be costly to obtain. For example, this is the case in drug design, QSAR (Quantitative structure-activity relationship) and other areas that involve large financial investments from private companies. In this paper we will not address those data sources. Rather, we will focus on freely available data on molecular biology, such as gene sequences, protein structure and metabolic pathways, as these are accessible to any academic researchers who may wish to explore CP applications to bioinformatics. Still, one should note that there is also a large amount of data and problems in the private sector, mostly with pharmaceutical companies. Where relevant, we shall mention the most accessible and useful sources of data for researchers interested in those areas where free databases are available.

1.1.2 About this article

In the following five sections we will summarize aspects of different areas of bioinformatics. Neither the topics chosen nor the aspects on which we focus are meant to cover bioinformatics in a comprehensive manner. Rather, the selection aims at bringing out those problems that the authors feel are more interesting for the CP community. Section two covers the analysis of sequence data, such as sequence comparison and pattern matching, but also includes evolutionary models, such as phylogenetic trees, and population genetics problems. Strictly speaking, not all of these involve sequence data. Phylogenetic trees can be calculated from phenotypes and, in the example chosen for population genetics problems, the estimation of genetic diversity from Single Nucleotide Polymorphisms, data can be obtained from restriction fragment lengths instead of direct sequences. The decision to group these problems together is not meant to express some clear partition but rather some basic similarities both at the computational and biochemical level.

Section three is about modeling RNA structures. Since RNA structure is largely dominated by base pairings, a good portion of this problem can be reduced to more abstract and general problems involving graphs and finite domain variables. Though there are open problems in these areas, there are several well-established and efficient solutions using local search or dynamic programming algorithms. Nevertheless, these problems are interesting because the problem domains are close to the classical domains where CP is often applied. In fact, CP approaches have been

used in RNA structure modeling since even before CP became recognized as an autonomous field.

Sections four and five address protein structure and interaction. Proteins are crucial parts in life's machinery, involved in most biochemical reactions, and are the expression of the organism's genes. Proteins are thus a focus of interest (and funding) in current bioinformatics and biochemistry research. The main challenge for CP here is probably the integration of CP algorithms with the software and methods used by this research community. The problems are hard to solve, and though there are successful solutions using molecular dynamics, local search and other approaches, CP seems to have a definite contribution to make in this area, if only it can be meshed with the other applications necessary to make the jump from theoretical studies to solving real-life problems. Section four is an overview of protein structure prediction and determination problems, and section five focuses the problem of modeling interactions between proteins for which the structure is known.

Section six is about systems biology. Admittedly, the authors' decision to include systems biology in bioinformatics would not meet consensual approval among either the bioinformatics or the systems biology community. But from the perspective of the CP researcher, systems biology is part of the same broad problem of applying computer science to solving biological problems. In systems biology, these problems are the complex networks of interactions of which life is made, from gene regulation to ecosystems but, in this paper, we will focus on metabolic pathways and gene regulation.

Finally, section seven concludes the paper by highlighting some common points of interest and challenges with the application of CP technology to these diverse and complex problems.

1.2 Sequence analysis

The study of gene and protein sequences is ideally suited to CP, both because of the power of CP to solve finite-domain combinatorial problems and because, in this field, it is often possible to separate the abstract problem of processing sequences of symbols from the more concrete, and often messier, issues of biological processes and noisy data collection. However, sequence analysis was the original reason for bioinformatics research, and there are well-established algorithms based on dynamic programming (e.g. the Smith-Waterman algorithm [1] for sequence alignments, and the Sankoff recursions in sequence complementarity matching for RNA structure prediction [2]) that, aided by specialized heuristics, can efficiently solve most problems in this field. As a result, it is hard for CP solvers to compete with algorithms such as FASTA [3] or BLAST [4].

Even so, there are some problems in sequence analysis where the versatility of CP can be a determining advantage relative to existing approaches. For example, whenever one wishes to include the parameters that evaluate substitutions and deletions into the problem itself [5] or to consider additional constraints based on prior knowl-

edge of conserved regions [6]. There are also specific related problems where the declarative nature and expressiveness of CP can simplify the implementation. For instance, the determination of the optimal dispensation ordering of nucleotides for pyrosequencing, a technique where DNA is sequenced by coupling a light-emitting reaction to the DNA-polymerase reaction. Each base in the sequence is determined by detecting the emission of light when the right nucleotide is dispensed, and the dispensation order is important to make the process both faster and less expensive [7].

Apart from specific applications involving more restricted data sources, such as when working directly with teams sequencing new genes, the main data source for gene and protein sequence analysis would be GenBank, an open-access gene sequence database supported by the NIH [8]. GenBank contains approximately a hundred million gene sequences with an average of a thousand base pairs each, covering 140,000 different organisms. It also provides basic search features based on BLAST, and several specialized adaptations of this algorithm, to more specific searches (e.g. primers, conserved domains, particular proteins, and so forth).

1.2.1 Phylogenetic trees

Sequence analysis problems in bioinformatics are not restricted to sequence determination or alignment. From that data, much can be inferred about the origin of organisms and species. Phylogenetic trees, which represent the evolutionary relationships between taxonomic groups, are often calculated from gene or protein sequences. Or they can be constructed from phenotype descriptors, sets of states assigned to characters observed in the taxonomic groups being studied, such as the presence or absence of wings or a metabolic pathway. But even in these cases the problem, at least computationally, is similar to generating these trees from sequence data, since in both cases the goal is to distribute all sets of attributes in the way that best represents their relationships according to given criteria.

Phylogenetic trees are usually calculated using algorithms that fall outside the scope of this paper, such as Markov chain bayesian inference or distance matrix methods. However, there are promising results worth noting in the application of CP related techniques to this problem. One example is to use a fundamental property of rooted phylogenetic trees. Since each leaf represents a taxonomic group to classify and each interior node a most recent common ancestor (mrca) of the sub-tree that starts at that node, by labeling each interior node with a measure of how long ago the mrca lived (or how distant it is from its current descendants) we obtain a min-ultrametric tree, a tree for which the path from the root to any leaf goes through nodes labeled in a strictly increasing order. Using this property as a constraint, [9, 10] approached the computation of phylogenetic trees as a constraint satisfaction problem.

Answer-set programming has also been applied to phylogenetic trees. This form of declarative programming, geared towards processing rules with constraints, was

used to enumerate phylogenetic trees for a set of taxa constrained to a maximum specified number of incompatible characters [11]. Characters are considered incompatible with a phylogenetic tree, in the strict sense, if they appear more than once. The reasoning is that it is more parsimonious to assume that all taxa that share a common character do so because they share a common ancestor with that character.

Another example of applying answer-set programming is in assembling phylogenetic trees by combining phylogenetic quartets. Quartet-based phylogeny reconstruction is a phylogenetic method in two steps. First, unrooted phylogenetic trees are estimated for all combinations of four taxa in the set of taxonomic groups to classify. In the second step, these quartets are combined to generate the complete tree. If all quartets are available and correct, the algorithm is polynomial in time. The problem arises when some quartets are ambiguous and thus cannot be generated with confidence or when the topology of some quartets are inconsistent with each other and cannot all fit in the same tree. In this case the problem becomes one of minimizing the number of quartets that are rejected when assembling the tree, and much harder to solve. This process was implemented using answer-programming by combining the ultrametric tree constraint with the constraint that all used quartets must be compatible [12].

1.2.2 Haplotypes and SNP

The genetic variation of a population is a bioinformatics area where constraint solving and optimisation has been applied. Rather than analysing the external features of the individuals (their phenotypes), these studies focus instead on their genotypes (the ADN of their chromosomes). More specifically, many studies concentrate on variation in specific positions of the genome sequence, known as Single Nucleotide Polymorphisms (SNPs), where mutations are known to have occurred. If not under selection pressure, one of the variant tends to fixate, eventually, in the population, eliminating the polymorphism. Thus, for most SNPs there are only two different nucleotide bases present in a given population.

Diploid individuals inherit one chromosome (or haplotype) from each of its parents. Denoting by A and a the two alleles (the two different nucleotide bases) of an SNP, an individual may inherit the combinations AA, aa, Aa or aA. The latter two cases (biallelic or heterozygous SNP) cannot be easily distinguished by current experimental sequencing techniques that may only distinguish the cases AA, aa and Aa or aA, that we will denote by 0, 1 and 2, respectively.

Given a set of m SNPs sites in some genomic block, all n individuals typically present different genotypes ($n \ll 3^m$). However, when a section of the genotype includes relatively closely spaced SNPs, linkage disequilibrium is higher and recombination is less likely. Although a set of n genotypes is explained by at most 2n haplotypes, it can generally be explained also by a much lower number of haplotypes, and this smaller number of haplotypes is a better measure of population diversity. For example, the six genotypes 21212, 21110, 01112, 11212, 21211 can

be explained by a set of only 4 haplotypes, 01110, 11011, 11110 and 01111, suitably combined. Genotype 21212, for instance, can be explained by the combination of haplotypes 01110 and 11011.

The haplotype inference problem (also known as phasing) can then be stated as follows: given a population of n individuals exhibiting a set of n genotypes find the set of unique haplotypes that exist in the population and find the pair of haplotypes that might exist in each individual, along with the respective probabilities.

The first computational approach to address this problem was Clark's subtraction method [13]. It first creates a set G of all genotypes in the population. The haplotypes that explain the genotypes in G with at most one biallelic SNP can be deterministically inferred and are used to initialise a set H. Then it selects a haplotype from H and checks whether it can explain any of the genotypes in G. For each of these genotypes it creates the complementary haplotype that is added to H and eliminates the genotype from G. The method proceeds until the genotype set G becomes empty.

This greedy algorithm aims at keeping small the cardinality of the resulting haplotype set. However, there are many possible orderings in which the haplotypes are selected from set H only upper bound on the cardinality of the set H are obtained, although certain heuristics (e.g. select the haplotype from H that explains more genotypes in G) have been shown to yield tight upper bounds on the minimum number of haplotypes.

In fact, assuming that nature is parsimonious, the haplotype inference problem can be reformulated into a minimisation problem. As proposed in [14], the Pure Parsimoneous Haplotype Inference (or Haplotype Inference by Pure Parsimony, HIPP) consists of finding the set of minimal cardinality that explains the genotypes of a population. The author then proposed an Integer Programming formulation and a technique, referred to as RTIP, that reduces the problem size without jeopardizing optimality.

The problem was shown to be NP-Hard in [15], who proposed a code (SDPHapler) to find approximate solutions (this problem was shown to be APX-hard [16]). Polynomial solutions were proposed for instances when all genotypes have at most two heterozygous sites [17], and other "islands of tractability" were investigated in [18]. For the general case, some IP based systems such as PolyIP [19] were developed with optimisation techniques (e.g. cutting planes), adapted for the HIPP. Alternatively, Clark's reduction method was adapted to a branch and bound search minimisation algorithm [20].

In addition to other IP and branch and bound formulations exploiting different techniques to improve efficiency, a constraint based system, SHIP, was proposed in [21] that formulated HIPP as a SAT problem. The core algorithm basically encodes a solution of the problem as a set of k haplotypes with $O(n^2m)$ constraints and $O(m^2+nm)$, where n is the population size and m the number of SNPs, and proves whether the problem is satisfiable. The base algorithm obtains an optimal solution by iterating the size k of the haplotype set starting in some lower bound.

The authors present a number of improvements on the model, some of which are common to RTIP to reduce the problem size and others meant to break some sym-

metries. Of course, the computation of a good lower bound is of great importance in this problem and this is done by an approximate solution to a max-clique in a graph encoding incompatibilities of the genotypes. This lower bound estimate was subsequently improved [22] by solving a SAT problem through local search (based on the SKC variant of WalkSAT). An extensive comparison of HIPP solvers is presented in [23] that also compute tight upper bounds for the HIPP problem by making Clark's reduction algorithm less greedy in the inclusion of haplotypes in set H (by a technique named Delayed Selection). Moreover, they present a solver, RPoly, based on pseudo-boolean optimisation and that uses an encoding similar to that of PolyIP, but with some optimisations that allow a significant reduction on the number of variables. The extensive performance comparison of a number of solvers for the HIPP problem on a set of 329 problem instances clearly that the ILP approaches are significantly less efficient than the SAT (SHIP) and PBO (RPoly) approaches, by solving much less instances in 1000 seconds (30% of the problem instances, against 81% of SHIP and 94% by RPoly). Competitive results were obtained with an Answer Set Programming approach that relies on an underlying SAT solver similar to SHIP, which was also applied to HIPAG, a variation of the HIPP problem that only takes into account biallelic genotypes [24]. Finally, the HIPP problem has been addressed by a two level Ant Colony Optimisation approach which reportedly outperforms RPoly in the number of instances solved, but is in general much slower [25]

Although the various approaches seem to show an advantage in solving the HIPP problem with some hybrid techniques, some issues remain regarding the practical use of solutions to the HIPP problem. On the one hand, there may exist a large number of solutions. [23] computed the set of all solutions of a small instance, SU100kb.25, with 34 genotypes and 15 sites, and found 48 different solutions with 17 haplotypes, 14 of which are common to all solutions. Computing the set of all solutions to the HIPP problem is #P-Complete which makes it very difficult to obtain the correct solution. Moreover, [26] have shown with experimental evidence on true haplotype data (not computationally obtained, but experimentally derived) that 3 of 7 sets have solutions that are not parsimonious (in the worst case, the true solution has 32 haplotypes compared to the 28 haplotypes of the HIPP solution). The authors then propose to focus on the computation of backbones (the set of haplotypes that belong to all solutions found), namely those that are implicit (the explicit backbones are easy to compute, and correspond to the initial haplotypes selected in Clark's reduction method).

1.3 RNA Structure

RNA molecules are generally transient messengers carrying genetic information from DNA to the ribosome, where it is translated into proteins. For years, this "Central Dogma" was even thought to hold universally, but it is now known that many non-coding RNA (ncRNA) are directly functional in the cell, some even playing roles similar to those of proteins. The growing number of RNA families being

discovered in the last decades [27] led to an increased interest in modeling RNA structures.

RNA folding is not driven by the same forces that guide protein folding [28, 29], being dominated by the hydrogen bonds formed between complementary bases, so its modeling follows different approaches from those deployed for proteins. The base pairing in RNA can be represented by a graph, describing structural elements such as stems and pseudoknots. RNA secondary structure is defined by the graph of pairings between bases in the same RNA molecule. Secondary structure graphs do not exactly cover the whole set of pairings, but a large portion of them that can be drawn in the plane in a tree-like fashion. More precisely, there is a one-to one correspondence between secondary structures and rooted oriented trees, such a restriction leading to very interesting mathematical properties [30]. Most RNA secondary structure elements involve several bases in a row pairing with another set of contiguous bases running in the opposite direction along the RNA sequence. This arrangement of stacked base pairs is called a stem, or an helix (due to its twisted form in 3D), and one can imagine it as the RNA sequence running to one side, looping and then coming back the other way with the two segments fitting together like a zipper.

Some elements, called pseudoknots, do not fulfill the definition of secondary structure. In a pseudoknot the two sets of contiguous bases zip together in the same direction, a rarer configuration due to the need to twist the RNA chain in order to accommodate this relative placement. As RNA folding is believed to be partially hierarchical [31], this level of representation is not only useful for algorithmic reasons but also corresponds to the biological assumption that RNA starts to fold driven by these base pairings. Thus, the graphs of nucleotide contacts represent an important aspect of the kinetics of RNA folding, even if they do not give the complete picture of its three dimensional structure. While these graphs of secondary structure – structural elements stabilized by these local interactions between base pairs – are generally computed with combinatorial and discrete methods, a more detailed description of the tertiary structure – the overall spacial configuration of the molecule – is best tackled by geometric and continuous approximations methods.

In both cases the related problems (structure prediction or display, structural homology finding, etc.) in the most general form, are NP-hard and require adequate strategies to be solved. However, only a few of these questions were modeled using constraints, probably because in the case of secondary structure efficient polynomial algorithms were found that can solve most instances of this problem in practice, while, in the case of the tertiary structure, the number of known structures only increased recently, and the problem of determining RNA tertiary structure has attracted less interest from the research community in contrast with protein structures.

1.3.1 Questions related to secondary structure

Secondary structure prediction is probably the problem that received most attention and to which a wider range of techniques were applied (for a rapid and recent overview of structure prediction at different levels, see [32] for instance). Disregarding the pseudoknot configurations, the ab initio prediction of RNA secondary structure by combinatorial optimization (maximizing the number of pairings under certain rules or minimizing an energy function) is solvable via a low complexity dynamic programming algorithm for a single sequence [33, 34], or even for a set of aligned sequences using stochastic context-free grammars [35] or other similar approaches. As these solutions are very efficient, a CP formulation of the problem would not be an improvement unless more is required.

In the case of SAPSSARN [36], which was one of the first attempts to introduce constraints for RNA secondary structure prediction, the additional feature is interactivity. The authors propose a dynamic treatment of constraints during structure prediction, where the computation of each predicted structure is interactive with the user, who may add or remove constraints. The interaction allowed by a constraint formulation of the question would not be possible during the computation if using a dynamic programming approach. In a similar perspective, the same authors proposed later, with RNASEARCH [37] a CP approach for RNA secondary structure display in the 2D space that optimizes the layout of the tree-like secondary structure of RNA – including pseudoknots – by trying to minimize stem overlaps.

Another source of interest in constraints is their expressive style. Their proximity to natural language leads to a direct formulation of a problem that may help avoid being trapped in a rigid algorithmic formulation. This is the case with [38, 39], that takes advantage of the declarative nature of constraint network modeling for RNA motif search. A set of conserved RNA features, called a signature, is defined by a series of constraints. A set of template constraint types are defined to handle sequence content, distances, and pair stacking in helices that model the usual structural elements (the approach is not restricted to secondary structure alone). As the problem of finding RNA occurrences that satisfy a given signature is NP-complete for sufficiently general signatures, the previous works traditionally developed two approaches. The grammatical approach models the signature by a context-free grammar, excluding pseudoknots, thus falling into a case where the search can be performed by dynamic programming. Other approaches define the signature as a set of interrelated motifs and perform an exhaustive search using pattern matching techniques. The authors observe that a natural constraint network model formulation emerges from the direct description of the problem: the variables representing the target positions searched in the genomic sequence, the domains being intervals over integers. The specific constraint types and potentially huge domain size call for an adaptation of the usual CP schemes (filtering, backtracking) and for the creation of dedicated reduction operators (a preprocessing of data using specific data structures such as k-factor trees to speed up the search of potential occurrences). Functional RNA are also often interacting with other ligands and the traditional methods are unable to find RNA motifs in interaction with other molecules. The authors simply

define and consider a new type of constraint to model the interactions between different molecules. The introduction of a descriptive language, with in scope the description of new generations of RNA patterns, may have some similarities with an earlier attempt for a programming language dedicated to RNA secondary structure [40] that could not use at that time all the constraint techniques that were developed during the last years. Both these works take full advantage of the handy declarative nature of constraints.

Constraints may also appear as a heuristic reduction for a polynomial algorithm. When several homologous sequences are available, but are not enough to compute a good starting alignment, a possible strategy is to search for a common secondary structure while aligning the sequences at the same time by maximizing a score which reflects both the structure and the alignment. Such an alignment, respectful of a (previously unknown) consensus structure, is called a structural alignment and when dealing with two sequences, the related problem is usually termed pairwise secondary structure prediction, or pairwise structural alignment. An early dynamic programming solution was provided by Sankoff [2] but the high algorithmic complexity of the exact recursion set he proposed makes them inapplicable on natural sequences and calls for heuristic reductions. The method became popular and stimulated a series of work trying to reduce the complexity. Part of the most recent works [41, 42] use alignment constraints, based for instance on nucleotide alignment posterior probabilities. These constraints, defined over the possible structural alignments, drastically reduce the computation requirements but do not really call for dedicated CP techniques.

This reduction on the compute-intensive Sankoff algorithm implicitly suggest that the two approaches – dynamic programming (for which there exists a polynomial algorithm) and CP modeling – could be combined. Aside of a direct use of the constraint technique apparatus, the frequent occurrence of its terminology may lead to unexpected and potentially fruitful connections between remote areas. Secondary structures are, for instance, a special case of outer-planar graphs, which are graphs of low treewidth. Such tree decompositions, are actually often used in CP modeling and it would not be surprising that migrations of these ideas would help bridge CP techniques with different approaches such as dynamic programming.

1.3.2 Ab initio 3D structure prediction

Less surprising is the use of constraints in three-dimensional structure prediction. Beyond the direct descriptivity of the question by constraints, the CP formalism also allows for the integration of information of very different nature, whatever their origin. Concerning RNA ab initio 3D structure prediction, still not much has been done and this should be related to the little number of available known structures, explained by a rather recent increase of interest for RNA. The McSYM research project, started at the University of Montreal in the 90's [43, 45], was the first work addressing that question. MC-Sym builds 3D ribonucleic acid structures from low-

resolution data by combining symbolic and numerical computations. The symbolic step generates all-atom sketches of 3D structures, using constraints derived from different sources, such as NMR spectroscopy data, X-ray crystallography, chemical modifications, secondary structure information, and so forth. The conformational search space is defined by spatial relations among RNA bases, which are encoded by transformation matrices that correspond to the transformation of a base referential into another. The inference engine is implemented as a Boolean constraint solver that accepts or refuses a structure whether or not all the given and inferred constraints are satisfied or not. In a context where the efforts are focused on a tighter connection between the different levels of structural description [46], CP appears to be an easily extensible framework that allows for the exploration and the discovery of more general structural rules. Two decades ago, comparative sequence analysis had started to reveal novel tertiary interactions between more than two bases [47], later confirmed with examples from the accumulated knowledge of 3D structures. The elucidation of the relationships between sequences and RNA motifs – in the broadest sense: recurrent structural elements subjects to constraints [48] – becomes one of the current challenges in RNA structure comprehension.

Although CP terminology is more flexible than alternative approaches, it usually implies an NP-hard formulation of the problem. When a polynomial approach exists, as in RNA secondary structure calculations, the benefits then strongly depend on the new types of information the constraints can handle. When, on the contrary, the question is more directly expressible by constraints, it usually calls for dedicated methods that can enrich the corpus of CP techniques while also providing new domains of application.

1.4 Protein Structure Modeling

Modeling protein structures is a complex problem due to the size and flexibility of these macromolecules, as proteins consist of long polymers of amino acid residues, typically containing thousands of atoms, intricately folded in a structure determined by physical and chemical interactions between these atoms and with the solvent, usually water or a lipid membrane. One can conceive of two different categories of protein modeling problems. One is protein structure prediction, where the structure of the molecule is to be estimated from chemical and physical considerations. The other is the determination of a protein structure given a set of constraints obtained from experimental data, such as Nuclear Magnetic Resonance (NMR) spectroscopy.

The most important source of macromolecular structure information is the Protein Data Bank (PDB), which contains nearly sixty thousand structures, mostly of proteins but also including nucleic acids and protein/nucleic acid complexes. It is an open access database that can be accessed or downloaded from several organizations (see the Worldwide Protein Data Bank site at www.wwpdb.org). The structure files include the atomic coordinates, the identification of each atom and monomer or ligand in a compound dictionary that specifies additional structural data (such as

chemical bonds), and often specific experimental information such as atom occupancy factors for X-Ray crystallography structures or NMR constraints.

1.4.1 Structure Prediction

Protein structure prediction is generally seen as an optimization problem, the goal being to find the structural configuration that minimizes the free energy of the system. Since the system includes both the protein and all the solvent molecules surrounding it, and since the free energy includes both enthalpy and the contribution of entropy factors that are difficult to compute, this is a computationally intensive problem. Furthermore, the assumption that the correct structure is at the global energy minimum seems not to hold universally, and may in some cases correspond to a local minimum where the structure is retained during folding due to high energy barriers [49]. Thus the traditional approach to protein structure prediction relies on molecular dynamics or simulated annealing, based on models of the physical properties of these macromolecules (ab initio structure prediction). It often resorts to using supercomputers or networks to meet the large computational demands, one of the most famous examples being the Folding@Home project [50]. More recently, protein structure prediction has become dominated by methods that rely on identifying structural features that the target protein has in common with the ever increasing set of known protein structures. Even so, there have been advances in the application of CP to these problems using lattice models of protein structure and interaction.

1.4.1.1 Lattice and HP models

Despite its importance and the interest it generates, a definitive solution to the problem of predicting protein structures still eludes all research efforts and approaches proposed over the last decades, both because of the difficulty in computing the free energy of the system and the complexity of the structure. But, since proteins are composed of chains of amino acids (more accurately amino acid residues) connected by peptide bonds, the problem can be simplified if rather than considering protein models at an atom level, proteins are modeled at the amino acid level. This way the variables represent amino acids, either their centers of mass or the alpha carbon in the protein backbone, with the main difference being the way protein chain is considered, either by following the backbone or the average of the atomic positions at each amino acid residue. In both cases, these models are rather simplified representations, since most physical and chemical properties depend on atoms or small chemical groups and are difficult to assign to amino acids abstractions.

Still, a number of simplified models have been proposed (see [51] for an overview) assuming some simplified characterization of the amino acids and placing them in some lattice structure. Among these models the HP model [52] is worth considering. In this simplified model, amino acids are labeled by their hydrophilic nature,

being classified as either hydrophobic (H) or as hydrophilic or polar (P). Since the solvent is water, H amino acids tend to be packed in the interior of the protein. The HP models this tendency indirectly, by minimizing an energy function modeled by the (negative) number of contacts between H amino acids that are neighbors in the lattice.

The original problem was formulated for a two dimensional square lattice, but it can easily be extended to a three dimensional cubic grid. In both the square and the cubic lattice the problem has been shown to be NP-complete (respectively in [53] and [54]).

Although the problem was not formulated as a constraint problem, and some algorithms were used to solve it not using constraint programming technology (e.g. [55, 56, 57, 58, 59]) such model can be adequately formulated as a finite domain constraint optimization problem: find the position of the amino acids (in the finite set of vertices of the lattice) that satisfy some constraints (successive amino acids in the protein chain must be neighbors, and no two amino acids can occupy the same position) and optimize the objective function (number of contacts between H amino acids).

As such, [60] were the first to attempt to address this problem as a constraint optimization problem with a cubic lattice. An interesting feature of this problem is that it presents many geometric symmetries (namely rotations). To handle this and other types of symmetries the authors proposed what was claimed to be the first declarative method that could be applied to arbitrary symmetries [61]. Among other tests, the authors have shown that they could improve the number of search steps and run times of one to two orders of magnitude to find optimal solutions in cubic lattices for proteins of around 30 amino acids.

This model can be improved in two complementary forms, either by changing the energy function or the lattice that is considered. The HPNX model [62] is an extension of the HP model in the first direction. Now in addition to hydrophobic (H) and polar (P) amino acids are classified as negatively charged (N) and neutral hydrophilic (X), and each amino acid pair in contact has a weighted contribution to the energy function (see Table 1.1, below). These models have been addressed by constraint programming [62] but they do not overcome some important drawbacks of the cubic lattice with respect to structure prediction.

Table 1.1 Comparison of the contact scores for the HP (left) and HPNX (right) models.

	H	P
H	-1	0
P	0	0

	H	P	N	X
H	-4	0	0	0
P	0	1	-1	0
N	0	-1	1	0
X	0	0	0	0

On the one hand, the model prevents, by design, that two amino acids with the same parity in the protein chain establish a contact, which is unreasonable. Moreover, the right angles between amino acids are not very realistic. In fact, [63] has shown that a face-centered cubic lattice, FCC, (a conformation that guarantees optimal packing of spheres [64]) would better approximate the packing of amino acids in a protein, and [65] have shown that the FCC lattice lead to root mean square deviations (RMSD) of 1.78 $\AA$ with respect to the real conformation, rather than the RMSD of 2.84 $\AA$ obtained with a cubic lattice.

Obtaining optimal solutions for the HP model on FCC lattices is very hard. In addition to various heuristic approaches (hydrophobic zipper [66], genetic algorithms [67], chain-growth [59] and approximate algorithms [68]) a number of CP techniques have been applied with significant success. An interesting model has been proposed in [69] that rather than solving the structure determination problem directly, converts it into a threading problem by adjusting the amino acid sequence to pre-computed hydrophobic cores. They showed how to compute such hydrophobic cores for both cubic and FCC lattice models. In particular they were able to find maximally hydrophobic cores for the FCC lattice for up to 100 H amino acids within seconds.

However, threading the sequences to the hydrophobic cores is still a difficult problem. Again a CP approach was proposed [70], combining path constraints and all-different constraints, to obtain a self-avoiding path constraint that was subsequently used in threading the amino acid sequences to the cores. The results obtained for randomly generated proteins are quite satisfactory for small proteins (100% success to proteins with 25 H amino acids with runs of 15 minutes) but the success rate decays for longer proteins (only 50% success for proteins with 100 H amino acids).

A different approach to find minimal energy HP models in FCC lattices was proposed in [71] that applied a tabu search meta-heuristic to local search. However, the authors have shown in [72] how to use CP techniques to improve good solutions previously found with tabu search, in order to exploit such large neighborhoods (LNS – Large Neighborhood Search). For any solution (i.e. a sequence of n amino acids) they randomly select an internal subsequence and perform a systematic search for alternatives, keeping the structure of the prefix and postfix sequences. By adopting a number of relevant modeling decisions (e.g. heuristics and redundant constraints), they obtained quite good results in a set of benchmarks (the Harvard instances). In particular they describe how the LNS search rapidly improves the tabu search solutions, but it must be noticed that the approach uses very substantial computing power (60 Intel base, dual core, dual processor Dell Poweredge 1885 blade server) running for a few seconds after a tabu search taking a few minutes. However, they cannot find optimal solutions that [69] could find in some randomly generated sequences, finding instead solutions within 3% to 10% of the optimum in runs of a few hours.

An improvement of the lattice models is explored in [73] where the authors use more information, namely secondary structures and disulfide bonds as additional constraints and subsequently replace the HP amino acid based model with an all atom model to allow an effective measurement of the root mean square deviation

(RMSD) between some known proteins and the predicted models (PDB code 1YPA, with RMSD of 9.2A within116.9 hours of computation).

The large computational time required led to the proposal of a specialized solver for lattice models [74]. The authors define special purpose encodings for the domains, to improve propagation in a number of specialized constraints useful in these problems (e.g. a constraint, next, to enforce a sequence of amino acids and various constraints that model the spatial distance of amino acids and are useful to handle contacts between spatially near amino acids). Then the authors present COLA, a constraint solver that exploits parallel search in constrained optimization problems and assess its performance in the protein structure determination. The authors also address the specification of rigid groups, useful for the modeling of secondary structures in protein structures, which were addressed in a later implementation [75]. A pure constraint programming was shown to be improved by its hybridization with local search, and discuss how to improve performance in futue work.

Their experiments show significant speed-ups over general purpose constraint solvers such as SICStus and GNU Prolog, but are not directly comparable with the HP model, as they adopt a more comprehensive set of values for any pair of amino acids in contact, without abstracting them into the H and P categories. The comparisons made with protein structures taken from the PDB are also made in terms of the minimal energy solutions obtained, rather than the difference between spatial conformations obtained (e.g. in RMSD) making it hard to assess the results obtained.

HP and lattice models seem a promising way of modeling this problem. One potential advantage of lattice models is the possibility of coupling local search methods in parallel with constraint propagation, improving the enumeration heuristics and the efficiency with which desired solutions can be found. Experience with PSICO (see below), suggests that without the simplicity and geometric elegance of lattice models this coupling of local search with constraint propagation is not feasible, since the domains of the atomic coordinates are too large at the early stages of the computation to allow the application of any energy function. Thus, without the lattice models, it seems that one must separate the process in two stages, constraint processing followed by local search for refining the structures.

However, and despite the promising results obtained with lattice models, it seems that, by themselves, these models are not competitive in real-life folding problems. The best predictors in the Critical Assessment of Techniques for Protein Structure Prediction (CASP) favor very different approaches, such as threading algorithms [76] or meta-servers aggregating predictions from ab initio calculations and ROSETTA fragment insertion [77, 78]. The overview of the first decade of CASP, in 2005, classifies three different types of structural prediction problems [79]. When there is a good sequence similarity between the target protein and a protein, or proteins, with known structure, it is possible to infer the structure of the target by homology modeling followed by suitable refinement with local search. If the target protein is more distantly related to known structures, it is still possible to identify folds, local structures that are stable and common to several proteins, and to use this information to assemble the target structure. Finally, even when no good homology

match is found, the large number of structures known makes it possible to choose structure fragments that are good candidates for assembling the target structure, based on secondary structure propensity and sequence compatibility. This is the approach used in the highly successful ROSETTA algorithm, and the current trend in protein structure prediction. Still, prior to this shift to template-based optimization algorithms, there is some mention of lattice models being used in conjunction with ab initio computations to try to predict protein structure [80], that may possibly be useful in practice when integrated with these more informed methods.

1.4.2 Protein structure determination

From a CP perspective, structure determination from experimental data can be seen as conceptually different from structure prediction, since the latter aims at finding an energy minimum, or simulating folding dynamics, while the former must provide structural models consistent with constraints derived from experimental data. However, the classical approach in the biochemistry community has been to treat structure determination also as an optimization problem, using well established algorithms for simulated annealing and molecular dynamics (e.g. the widely used DYANA/CYANA software [81]). In this approach the experimental constraints are simply treated as additional factors in the function to optimize. This has the advantage of making the method implicitly more resistant to experimental noise, but at the cost of not using the constraints to narrow down the search space.

Although X-Ray crystallography is the main experimental technique for the determination of macromolecular structures, in this case the experimental data contain the positions of all atoms, requiring only the deconvolution of the X-Ray diffraction patterns in order to obtain the structure. Computationally, the more interesting problem is with NMR spectroscopy, which accounts for approximately 15% of known protein structures. This technique provides distance constraints between atom pairs and angular constraints on the relative orientation of inter-atomic bonds, and it is from this set of constraints that the structure must be computed. One example of the application of CP to process these structural constraints is PSICO (Processing Structural Information with Constraint programming and Optimization). This algorithm considers the atomic coordinate triplets as variables with a continuous domain defined by solid shapes defined by sets of cuboid volumes, due to the convenient property of retaining that shape when intersected with other similarly shaped volumes. With this representation of the atomic coordinate domains, it is possible to propagate inter-atomic distance constraints efficiently [82], and even include more generic geometric constraints on the relative coordinates of rigid groups or their orientation with respect to a torsion angle [83].

The propagation of distance constraints is easy to illustrate for the simpler case of an upper bound on the distance between two atoms. This is one kind of information that can be obtained from Nuclear Magnetic Resonance (NMR) experiments. If two atoms can be, at most, separated by a distance of d, then each atom must be within

a neighborhood of distance d of the other atom. These neighborhoods can be computed from the domain of each atom and the constraint propagated by intersecting the domain of each atom with this d neighborhood of the other atom. Lower bounds on the allowed distances can be propagated by adding cuboid volumes contained in the domain of each atom to identify regions from which the atom must be excluded in order to respect these constraints.

This approach led to some promising results. For example, with proteins ranging from 400 to 700 non-Hydrogen atoms and 10.000 to 15.000 constraints, the solutions found had RMSD values from 2$\mathring{A}$ to 3$\mathring{A}$, taking a few minutes to calculate [82]. In addition, there were promising preliminary results with the propagation of higher order constraints defining rigid groups of atoms and their spatial relations. These constraints are propagated by calculating which parts of the domain of each atom are inaccessible to that atom due to the constraint imposed by the group and the domains of the other atoms. These groups range from small parts of amino acid residues joined by bonds that are free to rotate to large secondary structure elements such as alpha helices, and the results on randomly generated groups and groups simulating secondary structure elements showed that this algorithm can improve pruning significantly even for small groups. For larger groups it is even faster than propagating the set of binary distance constraints on those atoms [83].

1.5 Protein interaction

Modeling how two proteins interact (protein docking) is a similar problem to the modeling of protein structures in the sense that the goal is to obtain a macromolecular structure. However, the starting point is the known structures of the interacting partners, so the problem is not so much predicting the folding of the molecules involved but, rather, how two known structures best fit together. Although this fit is governed by intermolecular ”forces”, such as electrostatics and entropy contributions from the solvent, since the interaction is not covalent it is very weak at any single point, requiring a large surface of contact to stabilize the protein complex. Thus, most protein docking algorithms rely on a geometric filtering stage that identifies those configurations with the largest contact surface.

A widely used class of protein docking algorithms is based on the Fast Fourier Transform (FFT) computation of correlation matrices [84]. In this approach, each docking partner is represented as a three-dimensional matrix in which numerical values distinguish between the surface regions (e.g. positive value), the core of the molecule (e.g. negative value) and the empty surrounding space (zero). The correlation matrix indicates the total score for each relative placement of these matrices, thus distinguishing the configurations with a large surface contact from those with smaller surface contacts or forbidden overlaps with the core regions. Thought the FFT algorithm is efficient in time, of O($n^3 log(n)^3$), the need to represent all grids as numerical matrices results in a significant memory footprint. In practice, a CP approach based on a simpler representation of the protein shapes can perform the

calculations in less time and with orders of magnitude lower memory requirements [85].

This algorithm, BiGGER, represents each protein shape as a grid, conceptually, just like in the FFT approach. However, the grids defining core and surface regions for each partner are encoded as sets of line segments that, for each (Z,Y) pair, define the arrays of cells corresponding to surface and core regions along the X axis. Unlike FFT, the search does not involve computing all the correlation matrices but actually searching through the spatial configurations by placing one partner in a position relative to the other. The encoding of the grids makes it easy to restrict the search space by maintaining bounds consistency on the constraints that forbid core-core overlaps, and those requiring a minimum value for the surface-surface overlaps (branch and bound) or, and more significantly, restrict the search space to configurations consistent with geometric constraints obtained from experimental data.

Empirical results suggest that knowing even a few residues in one partner that must be in contact with the other protein partner may be enough, in most cases, to guide the search to the right complex structure [86], and this can be done efficiently in a flexible manner by enforcing a cardinality constraint on a subset of a given set of potential contacts [85]. The reasoning is that experimental data on residue contacts are usually obtained from either the perturbation of the residue during the interaction (as measured by spectroscopy) or the perturbation in the formation of the complex by mutating or otherwise modifying one residue. Such data suggest that the residue is at the interface, but there are other possibilities that give the same results, such as conformational changes in the protein, either during the formation of the complex or due to the changes in that residue. The ability to specify a set of candidate contacts, each of which can be between one residue of one partner and any residue of the other, and to impose a constraint on how many of those must be verified, allows the user to model quite naturally the uncertainty in the experimental data. For example, given ten potential interface residues, one may require that at least five of those, not specified in advance, must be present at the interface of the model complex. As is characteristic of CP, aside from helping to narrow down the right structures, this also reduces the search time (by about an order of magnitude).

This approach contrasts with the classical approach in the biochemistry community which is, much like with the determination of protein structures, to include the experimental constraints as additional factors in an energy function that is being minimized. HADDOCK [87, 88] (High Ambiguity Driven biomolecular DOCKing), for example, is a docking application based on local-search NMR assignment and structure calculation software that predicts protein complexes by minimizing the violation of geometric constraints included in the global function.

As with protein structure modeling, it seems that the potential of CP techniques has not yet been fully appreciated by the biochemistry community. The problem, we feel, lies mostly with the difficulty of crossing the gap between theoretical studies and proof-of-concept of the algorithms, and the actual application to real-life problems, as the latter requires a tighter collaboration with the researchers involved in those problems and the integration of the CP solvers in the often large and complex processing pipeline that goes from the data to the final refined model.

1.6 Systems Biology

Systems Biology is a new area in biology which aims at understanding biological systems at the higher level of the interactions between all components [89]. In particular, biological networks of gene expression and regulation, protein interaction, metabolic pathways and such processes required for life. The development of computational models plays a key role in systems biology [90], and a number of techniques have been used to model different levels of abstractions of such systems. In Boolean networks for gene regulation, only the presence or absence of substances is represented and the systems dynamics is modeled by state transitions: variables denote whether a gene is expressed and Boolean functions relate variables in different states [91]. A richer expressiveness is obtained in qualitative networks [92] where multi-valued variables are used to represent various degrees of, for example, gene expression. Other computational formalisms developed for systems verification have also been applied to biological systems such as Petri-nets [93], p-calculus [94] and other types of spatial/temporal calculi [95]. Still, the continuous behavior of biological systems is better captured by means of sets of ordinary equations. Various systems exist that exploit one or more of these formalisms, such as BIOCHAM [96] that allows the representation and reasoning about these systems at different levels of detail ranging from Boolean networks, to temporal logics and differential equations.

Despite existing proposals for incorporating differential equations as first order constraints in the constraint programming paradigm [97] its use in modeling biological processes has been limited [98] given the problems of scalability of the approach.

Nevertheless, constraint programming has been applied to specific problems in systems biology. For example, [99] analyses and proposes extensions to the concurrent CP paradigm (CC) to model systems biology problems. They analyze Timed CC, Timed Default CC and Hybrid CC extensions (Hcc), and show how the latter can express differential equations and initial value problems in addition to algebraic constraints. This allows the modeling of system behavior by means of a sequence of alternating point and interval phases. The authors also show how to model a number of key aspect of biological systems in Hcc constructs, such as Reaching thresholds, time and concentrations, kinetics, gene interaction and stochastic behavior, and illustrate these concepts in a comprehensive example of cell differentiation for a population of X. Laevis cells.

An alternative to modeling changes is to model steady state behaviour of molecular networks. These can now be represented by means of graphs, and reason about the behaviour of the networks by finding some patterns of reaction as paths in these graphs. This was a major motivation for the development of CP(Graphs) [100], where a new domain (graphs) was introduced in the CP and a set of (global) constraints where specified (such as path and reachable constraints). Some filtering algorithms where proposed and tested in the analysis of metabolic networks, namely for finding metabolic pathways that are in use in the cell, given a list of reactions detected with DNA chips.

Answer-set programming has also been applied to metabolic pathway and gene expression data, as a declarative approach to processing the rules and constraints that characterize these problems. One example is the check for consistency between metabolic pathway databases, which store the theoretical or presumed knowledge about the reactions occurring inside an organism, and experimental data, such as gene expression data derived from DNA microarrays or direct measures of metabolite levels in the organism [101]. This approach processes the rules implied by the presumed influence of regulator genes and the assumed reaction pathways with the experimental data about metabolite levels and gene expression. If data and theroetical assumptions are inconsistent, this method also identifies minimal subsets of regulatory influences that can account for the inconsistency.

Another example is an action language designed to describe metabolic pathways or other biological networks, their changes and queries about such systems [102]. This language describes the properties that change and the actions that cause such changes, and has the important property that any description in this language be automatically translated into an answer-set program and take advantage of the efficiency of answer-set solvers.

In essence, systems biology poses several challenges to CP and declarative approaches in general, by addressing system dynamics problems. Some problems fall somewhat outside the main stream of current CP research but open the possibility of addressing new domains and exploit constraint technology (e.g. filtering) to these domains.

As for data sources, a good starting point would be the Kyoto Encyclopedia of Genes and Genomes (KEGG), [103] a knowledge base combining information on metabolic pathways, functional hierarchies that include genes, proteins, drugs, diseases and organisms, and several annotated databases on genes and ligand compounds. From a CP perspective, perhaps the most interesting part of KEGG would be the PATHWAY database, describing metabolic pathways, which are chains of biochemical reactions catalyzed by enzymes and regulated by the activity of the genes that code for each enzyme. The study of metabolic pathways combines graph problems with dynamic systems modeled by differential equations, both fields of interest to CP researchers.

1.7 Conclusion

This article is an overview of the application of CP technology to the broad area of bioinformatics, an application domain with increasing importance due to the advances in biological and biochemical experiments, and the huge amount of data that must be subsequently processed. In this paper we have shown that the relationship between CP and bioinformatics holds in both directions. On the one hand, CP programming has shown its potential in many bioinformatics applications, and for some specific models and problems we have provided examples where it is the most appropriate computational approach to deal with them. Nevertheless, we acknowledge

that realistic applications in bioinformatics (as well as in other application areas), given the wealth of questions raised by the ever increasing amounts of experimental data being collected, usually demand a variety of computational techniques. When attempting to solve realistic bioinformatics problems, a comprehensive analysis of these problems should thus be made, both by computer scientists and biologists and biochemists, to determine whether CP can effectively be used, in which subproblems and how these relate to the whole application.

Once such cautions are taken, one must recognize that the computational problems that bioinformatics poses, some of which were addressed in the previous sections, have been a source of inspiration for CP. They have provided new domains where CP can be applied, of which we can refer lattice structures, spatial cuboids, graphs and even temporal domains (differential equations and timed events). The complexity of such problems demand the exploitation of advanced CP techniques to improve search and a number of them have been already applied, namely various types of global constraints, specialized heuristics, the interaction of CP and local search, and the exploitation of some forms of parallel execution. We can only foresee that, given the increasingly importance of bioinformatics and the rich problems it poses, this trend can only continue in the years to come.

References

1. Smith TF, Waterman MS (1981). "Identification of Common Molecular Subsequences". Journal of Molecular Biology 147: 195197
2. Sankoff D (1985) Simultaneous solution of the RNA folding, alignment and protosequence problems. SIAM Journal on Applied Mathematics. 45:810-825.
3. Lipman DJ, Pearson WR, (1985), Rapid and sensitive protein similarity searches. Science. 1985 Mar 22;227(4693):1435-41.
4. Altschul SF, Gish W, Miller W, Myers EW, Lipman DJ (1990). "Basic local alignment search tool". J Mol Biol 215 (3): 403410
5. Roland H. C. Yap, (2001) Parametric Sequence Alignment with Constraints. Constraints 6(2/3):157-172
6. Sebastian Will, Anke Busch, Rolf Backofen (2008) Efficient Sequence Alignment with Side-Constraints by Cluster Tree Elimination. Constraints 13(1-2): 110-129
7. Carlsson, Mats and Beldiceanu, Nicolas (2004) Multiplex dispensation order generation for pyrosequencing. In: CP'2004 Workshop on CSP Techniques with Immediate Application, 27 Sep 2004, Toronto, Canada.
8. Benson DA, Karsch-Mizrachi I, Lipman DJ, Ostell J, Wheeler DL. (2004), GenBank: update. Nucleic Acids Res. 2004 Jan 1;32(Database issue):D23-6.
9. Gent IP, Prosser P, Smith BM, Wei W. (2003), Supertree Construction Using Constraint Programming. LNCScience 2833, Proc. CP2003, Springer.
10. Moore NC, Prosser P (2008). The ultrametric constraint and its application to phylogenetics. Journal of Artificial Intelligence Research 32 (Aug 2008) 901938
11. Daniel R. Brooks, Esra Erdem, Selim T. Erdogan, James W. Minett, Donald Ringe: Inferring Phylogenetic Trees Using Answer Set Programming. J. Autom. Reasoning 39(4): 471-511 (2007)
12. Wu G, You JH, Lin G (2007). Quartet-Based Phylogeny Reconstruction with Answer Set Programming. IEE/ACM Trans. CBB. January-March 2007 (vol. 4 no. 1) pp. 139-152.

13. Clark,A.G. (1990) Inference of haplotypes from PCR-amplified samples of diploid populations. Mol. Biol. Evol., 77, 111122.
14. D. Gusfield. Haplotype inference by pure parsimony. In 14th Annual Symposium on Combinatorial Pattern Matching (CPM03), pages 144155, 2003.
15. Huang,Y.-T. et al. (2005) An approximation algorithm for haplotype inference by maximum parsimony. J. Comput. Biol., 12, 12611274.
16. Lancia,G. et al. (2004) Haplotyping populations by pure parsimony: Complexity of exact and approximation algorithms. INFORMS J. Comput., 16, 348359.
17. Cilibrasi,R. et al. (2005) On the complexity of several haplotyping problems. In 5th Workshop on Algorithms in Bioinformatics (WABI 2005). Mallorca, Spain, pp. 128139.
18. Sharan,R. et al. (2006) Islands of tractability for parsimony haplotyping. IEEE/ACM Trans. Comput. Biol. Bioinform., 3, 303311.
19. D. Brown and I. Harrower. Integer programming approaches to haplotype inference by pure parsimony. IEEE/ACM Transactions on Computational Biology and Bioinformatics, 3(2):141154, 006.
20. Wang,L. and Xu,Y. (2003) Haplotype inference by maximum parsimony.Bioinformatics, 19, 17731780.
21. I. Lynce and J.Marques-Silva. Efficient haplotype inference with Boolean satisfiability. In AAAI Conference on Artificial Intelligence, pages 104109, July 2006.
22. I. Lynce, J. Marques-Silva, and S. Prestwich. Boosting haplotype inference with local search. Constraints, 13(1):155179, 2008.
23. I. Lynce, A. Graa, J. Marques-Silva and A. L. Oliveira. Haplotype inference with boolean constraint solving: An overview. In Proc. of 20th IEEE Intl Conf. on Tools with Artificial Intelligence (ICTAI 08), Dayton, OH, 2008.
24. Esra Erdem, Ozan Erdem and Ferhan Tre: HAplo-ASP: Haplotype Inference using Answer Set Programming, LPNMR09, LNCS 5753, Springer, 573-578.
25. Stefano Benedettini, Andrea Roli, Luca Di Gaspero: Two-Level ACO for Haplotype Inference Under Pure Parsimony, ANTS Conference 2008: 179-190.
26. Sharlee Climer, Gerold Jager, Alan R. Templeton, and Weixiong Zhang. How frugal is mother nature with haplotypes? Bioinformatics, 25(1):6874, 2009.
27. S. R. Eddy. Non-coding RNA genes and the modern RNA world. Nat Rev Genet , 2(12):919 929, December 2001.
28. I. Tinoco and C. Bustamante. How RNA folds. JMB , 293:271281, 1999.
29. Peter B. Moore. The RNA world, Second Edition, chapter 15 : The RNA folding problem, pages 381401. Cold Spring Harbor Laboratory Press, Cold Spring Harbor, 1999.
30. M.S. Waterman. Introduction to Computational Biology , chapter 13 : RNA secondary structure, pages 327343. Chapman & Hall, 1995.
31. M. Wu and I. Tinoco. RNA folding causes secondary structure rearrangement. PNAS, 95:1155511560, 1998.
32. Emidio Capriotti and Marc A. Marti-Renom. Computational RNA structure prediction. Current Bioinformatics , 3(1):3245, January 2008.
33. R. Nussinov and A.B. Jacobson. Fast algorithm for predicting the secondary structure of single stranded RNA. PNAS, 77:63096313, 1980.
34. J.A. Jaeger, D.H. Turner, and M. Zuker. Improved predictions of secondary structures for RNA. PNAS, 86:77067710, october 1989.
35. B. Knudsen and J. Hein. RNA secondary structure prediction using stochastic context-free grammars and evolutionary history. Bioinformatics , 15(6):446454, 1999.
36. Christine Gaspin and Eric Westhof. An interactive framework for rna secondary structure prediction with a dynamical treatment of constraints. Journal of Molecular Biology, 254(2):163174, November 1995.
37. Christine Gaspin. Rna secondary structure determination and representation based on constraints satisfaction. Constraints, 6(2/3):201221, 2001.
38. P Thebault, S de Givry, T Schiex, and C Gaspin. Searching rna motifs and their intermolecular contacts with constraint networks. Bioinformatics, July 2006.

39. Matthias Zytnicki, Christine Gaspin, and Thomas Schiex. Darn! a weighted constraint solver for rna motif localization. Constraints, 13(1-2):91109, 2008.
40. B. Billoud, M. Kontic, and A. Viari. Palingol : a declarative programming language to describe nucleic acids secondary structures and to scan sequence databases. NAR , 24(8):13951404, april 1996.
41. Arif O. Harmanci, Gaurav Sharma, and David H. Mathews. Efficient pairwise rna structure prediction using probabilistic alignment constraints in dynalign. BMC Bioinformatics, 8, April 2007.
42. Robin D. Dowell and Sean R. Eddy. Efficient pairwise rna structure prediction and alignment using sequence alignment constraints. BMC Bioinformatics, 7:400+, September 2006.
43. F. Major, M. Turcotte, D. Gautheret, G. Lapalme, E. Fillion, and R. Cedergren. The combination of symbolic and numerical computation for three-dimensional modeling of RNA. Science, 253:12551260, september 1991.
44. B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald. Bridging the gap in RNA structure prediction. Curr Opin Struct Biol , 17(2):157165, April 2007.
45. D. Gautheret, F. Major, and R. Cedergren. Modeling the threedimensional structure of RNA using discrete nucleotide conformational sets. JMB, 229:10491064, 1993.
46. B. A. Shapiro, Y. G. Yingling, W. Kasprzak, and E. Bindewald. Bridging the gap in rna structure prediction. Curr Opin Struct Biol, 17(2):157165, April 2007.
47. R. R. Gutell, A. Power, G. Z. Hertz, E. J. Putz, and G. D. Stormo. Identifying constraints on the higher-order structure of RNA: continued development and application of comparative sequence analysis methods. Nucleic. Acids. Res., 20:57855795, 1992.
48. N. B. Leontis, A. Lescoute, and E. Westhof. The building blocks and motifs of RNA architecture. Curr Opin Struct Biol , 16(3):279287, June 2006.
49. Lazaridis T, Karplus M. (2000), Effective energy functions for protein structure prediction. Curr Opin Struct Biol. 2000 Apr;10(2):139-45.
50. M. R. Shirts and V. S. Pande. (2000). "Screen Savers of the World, Unite!". Science 290: 19031904.
51. K.A. Dill, S. Bromberg, K. Yue, K.M. Fiebig, D.P. Yee, P.D. Thomas and H.S. Chan, Principles of protein folding a perspective of simple exact models, Protein Science, 4, 561-602, 1995.
52. Kit Fun Lau, Ken A. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. Macromolecules 1989, 22, 3986-3997.
53. P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis, On the complexity of Protein Folding, Journal of Computational Biology, 5 (3), 423-466, 1998.
54. B. Berger, T. Leighton, Protein Folding in the hydrophobic-hydrophilic (HP) model is NP-complete, Journal of Computational Biology, 5 (3), 27-40, 1998.
55. K. Yue, K.A. Dill, Folding proteins with a simple energy function and extensive conformational serach, Protein Science, 5(2), 254-261, 1996
56. V.I. Abkevitch, A.M. Gutin, and E.I. Shakhnovich, Impact of local and non-local interactions oin thermodynamics and kinetics of protein folding, Journal of Molecular Biology, 252, 460-471, 1995.
57. R. Unger and J. Moult, Local interactions dominate folding in a simple protein model, Journal of Molecular Biology, 259, 988-994, 1996.
58. D.A. Hinds and M. Levitt, From structure to sequence and back again, Journal of Molecular Biology, 258, 201-209, 1996.
59. E. Bornberg-Bauer, Chain growth algorithms for HP-type lattice proteins, Procs. of RECOMB97, First Int. Conf. on Research in Computational Molecular Biology 47-55, 1997.
60. R. Backofen Constraint techniques for solving the protein structure prediction problem, Procs of CP98, LNCS 1520, 72-86, 1998.
61. Backofen, R., and Will, S. 2002. Excluding symmetries in constraint-based search. Constraints 7(3):333349.
62. R. Backofen, S. Will and E. Bornberg-Bauer, Application of Constraint Programming techniques for structure prediction of lattice proteins with extended alphabets, Bioinformatics 15(3), 234-242, 1999.

63. Z. Bagci, R.L. Jernigan and I. Bahar, Residue coordination in proteins conforms to the closest packing of spheres, Polymer, 43, 451-459, 2002.
64. B. Cipra, Packing challenge mastered at last, Science, 281, 1267, 1998
65. B.H. Park and M. Levitt, The complexity and accuracy of discrete state models of protein structure, Journal of Molecular Biology, 249, 493-507, 1995.
66. Cooperativity in protein-folding kinetics, Procs. Natl. Acad. Science USA, 90, 1942-1946, 1993
67. R. Unger and J. Moult, Genetic algorithms for protein folding simulations, Journal of Molecular Biology, 231, 75-81, 1993.
68. R. Agarwala, S.Batzoglou, V. Dancik, S.E. Decatur, M. Farach, S. Hannenhalli, S. Muthukrishnan and S. Skiena, Local rules for protein folding on a triangular lattice and generalized hydrophobicity in the HP-model, Journal of Computational Biology, 4 (2), 275-296, 1997.
69. Backofen, R., and Will, S. 2006. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. Constraints 11(1):530.
70. R. Backofen and S. Will, Fast, Constraint-Based Threading of HP-Sequences to Hydrophobic Cores, Procs. of CP01, LNCS 2239, 494-508, 2001.
71. M. Cebrian, I. Dotu, P. Van Hentenryck, P. Clote, Protein Structure Prediction on the Face Centered Cubic Lattice by Local Search, in Procs. AAAI08, 241-245, 2008.
72. Ivn Dot, Manuel Cebrin, Pascal Van Hentenryck, Peter Clote, Protein Structure Prediction with Large Neighborhood Constraint Programming Search. Procs. CP08, LNCS 5202, 82-96, 2008
73. A. Dal Pal, A. Dovier and F. Fogolari, Constraint Logic Programming approach to protein structure prediction, BMC, Bioinformatics 5, 186, 2004.
74. A. Dal Pal, A. Dovier and E. Pontelli, A constraint solver for discrete lattices, its paralelization and application to protein structure prediction, Softw. Pract. Exper. 37(13), 1405-1449, 2007
75. Raffele Cipriano, Alessandro Dal Pal and Agostino Dovier, A Hybrid approach mixing local search and constraint programming applied to the protein structure prediction problem, in Procs WCB08, Paris, May 2008.
76. Yang Zhang. (2008) I-TASSER server for protein 3D structure prediction. BMC Bioinformatics, vol 9, 40.
77. Fischer, D. (2006) Servers for Protein Structure Prediction. Curr. Opin. Struc. Biol; 16:178-182
78. Bonneau, R., Jerry Tsai, Ingo Ruczinski, Dylan Chivian, Carol Rohl, Charlie EM Strauss, David Baker. (2001) Rosetta in CASP4: Progress in ab initio protein structure prediction. Proteins 45(S5)119-126.
79. Moult, J. (2005). A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. Curr Opin Struct Biol. 15:285-9
80. Jeffrey Skolnick, Andrzej Kolinski, Daisuke Kihara, Marcos Betancourt, Piotr Rotkiewicz, and Michal Boniecki, (2001), Ab Initio Protein Structure Prediction via a Combination of Threading, Lattice Folding, Clustering, and Structure Refinement, PROTEINS: Structure, Function, and Genetics Suppl 5:149156
81. Gntert, P., Mumenthaler, C. & Wthrich, (1997), K. Torsion angle dynamics for NMR structure calculation with the new program DYANA. J. Mol. Biol. 273, 283298
82. Krippahl, L., Barahona, P., PSICO: Solving Protein Structures with Constraint Programming and Optimisation, Constraints 2002, 7, 317-331
83. Krippahl, L, Barahona P, Propagating N-ary Rigid-Body Constraints, (2003). Principles and Practice of Constraint Programming, CP'2003 (Procs.), Francesca Rossi (Ed.), Lecture Notes in Computer Science, vol. 2833, Springer, pp. 452-465, October, 2003.
84. Katchalski-Katzir E, Shariv I, Eisenstein M, Friesem AA, Aflalo C, Vakser IA. (1992) Molecular surface recognition: determination of geometric fit between proteins and their ligands by correlation techniques. Proc Natl Acad Sci U S A. 1992 Mar 15;89(6):2195-9.
85. Krippahl, L, Barahona P. Applying Constraint Programming to Rigid Body Protein Docking (2005), Principles and Practice of Constraint Programming, CP'2005 (Procs.), Peter van Beek (Ed.), Lecture Notes in Computer Science, vol. 3709, Springer, pp. 373-387

86. Krippahl, L., Moura, J.J., Palma, P.N., Modeling Protein Complexes with BiGGER (2003). Proteins. 2003 Jul 1;52(1):19-23.
87. Cyril Dominguez, Rolf Boelens and Alexandre M.J.J. Bonvin (2003). HADDOCK: a protein-protein docking approach based on biochemical and/or biophysical information. J. Am. Chem. Soc. 125, 1731-1737.
88. S.J. de Vries, A.D.J. van Dijk, M. Krzeminski, M. van Dijk, A. Thureau, V. Hsu, T. Wassenaar and A.M.J.J. Bonvin "HADDOCK versus HADDOCK: New features and performance of HADDOCK2.0 on the CAPRI targets." Proteins: Struc. Funct. & Bioinformatic 69, 726-733 (2007).
89. H. Kitano (ed), Foundations of System Biology, MIT Press, 2001.
90. J.M. Bower and H. Bolouri (eds.), Computational modeling of genetic and biochemical networks, MIT Press, 2001.
91. S.A. Kauffman, The origins of order, Oxford University Press, 1993.
92. D. Thieffry and R. Thomas, Qualitative analysis of gene networks, Pacific Symposium on Biocomputing, 3, 77-88, 1998.
93. V.N. Reddy, M.L. Mavrovouniotis and M.L. Liebman, Petri net representation in metabolic pathways, Intelligent Systems for Molecular Biology, ISMB93, AAAI Press, 328-336, 1993.
94. A. Regev, W.Silverman and E. Shapiro, representation and simulation of bio-chemical processes using the pcalculus process algebra, Pacific Symposium on Biocomputing, 6, 459-470, 2001.
95. Luca Cardelli, Abstract machines of systems biology, Transactions on Computational Systems Biology, 3737, 145-168, 2005
96. L. Calzonne, F. Fages and S. Soliman, BIOCHAM. An environment for modeling biological systems and formalizing experimental knowledge, Bioinformatics 22(14), 1805-1807, 2006.
97. J. Cruz and P. Barahona, Constraint Satisfaction Differential Problems, Procs of CP03, LNCS 2833, 259-273, 2003.
98. J. Cruz and P. Barahona, Constraint Reasoning in Deep Biomedical Models, Artificial Intelligence in Medicine (34), 77-88, 2005.
99. A. Bockmayr and A. Courtois, Using hybrid concurrent constraint programming to model dynamic biological systems, ICLP02, LNCS 2401, 85-99, 2002.
100. G. Dooms , Y. Deville and P. Dupont, CP(Graph): Introducing a graph computation domain in Constraint Programming, Procs. of CP05, LNCS 3709, 211-225, 2005.
101. Gebser M, Schaub T, Thiele S, Usadel B, Veber P (2008). Detecting inconsistencies in large influence networks with answer set programming. International Conference on Logic Programming, 2008.
102. Dworschak S, Grell S, Nikiforova VJ, Schaub T, Selbig J (2008). Modeling Biological Networks by Action Languages via Answer Set Programming. Constraints 13(1-2): 21-65
103. Kanehisa, M. and Goto, S.(2000), KEGG: Kyoto Encyclopedia of Genes and Genomes. Nucleic Acids Res. 28, 27-30.