INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Project-Team MODBIO

Computational Models in Molecular Biology

Lorraine

THEME BIO

Table of contents

1. Team

Team Leader
Eric Domenjoud [CR CNRS]

Administrative Assistants
Sophie Drouot [INRIA]
Aurélie Prévost [INRIA, since 6/2005]

Staff member CNRS
Yann Guermeur [CR]

Ph. D. students
Arnaud Courtois [UHP, ATER INPL until 8/2005]
Yannick Darcy [Grant MENRT]
Abdelhalim Larhlimi [UHP, cofinanced by the Région Lorraine]

Post-doctoral Fellows
Sandrine Schermack-Peyrefitte [CNRS, until 02/2005]
Emmanuel Monfrini [UHP, since 10/2005]
Frédéric Sur [CNRS, until 8/2005]

Engineer
Delphine Autard [UHP, since 10/2005]

External Collaborators
Alexander Bockmayr [Professor, Freie Universität Berlin]
François Denis [Professor, Université de Provence, Marseille; research scientist à l'INRIA until 8/2005]

Independent French-German research group, ATIPE CNRS-MPG
Ernst Althaus [Group Leader]
Stefan Canzar [PhD student, cofinanced by the Région Lorraine]

2. Overall Objectives

2.1. Introduction

The aim of the project MODBIO is to develop computational models for molecular and cell biology. We are focusing on two types of problems:

- Determining the structure of biological macromolecules,
- Discovering and understanding the function of biological systems.

We approach these questions by combining techniques from constraint programming, combinatorial optimization, hybrid systems, and statistical learning theory.

2.2. Research themes

- Sequence and structural alignment, phylogeny.
- Determination and analysis of macromolecular envelopes.
- Protein structure prediction and protein docking.
- Modeling alternative splicing regulation.
- Metabolic pathway analysis.

2.3. Scientific and industrial relations

- Participation in the "Génopole Strasbourg Alsace-Lorraine"
- Participation in the Bioinformatics project of the Région Lorraine
- Participation in the ACI project GENOTO3D
- Participation in the ARC INRIA "Process calculi and molecular networks"
- Participation in the "Décrypthon" programme
- Various national and international collaborations

  - Laboratoire «Maturation des ARN et Enzymologie Moléculaire» (MAEM), UMR 7567, Nancy
  - Laboratoire de Cristallographie, LCM3B, Nancy
  - Institut de Biologie et Chimie des Protéines, IBCP, Lyon
  - Institut Supérieur d'Agriculture, ISA, Beauvais, France
  - Center for Bioinformatics, Saarbrücken, Germany
  - DFG Research Center Matheon, Berlin, Germany
  - Institute of Mathematical Problems in Biology, Russian Academy of Sciences
  - University of California, Irvine, USA

3. Scientific Foundations

3.1. Constraint programming

Constraint programming [40] is a declarative programming language paradigm that appeared in the late 80's, and which has become more and more popular since then. A *constraint* is a logical formula that defines a relation to be satisfied by the values of the variables the formula contains. For instance, the formula $x + y \leq 1$ expresses that the sum of the values of the variables x and y must be less than or equal to 1.

In *constraint programming*, the user programs with constraints, i.e., he or she describes a problem by a set of constraints, which are connected by *combinators* such as conjunction, disjunction, or temporal operators (`always`). Each constraint gives some *partial* information about the state of the system to be studied. Constraint programming systems allow one to deduce new constraints from the given ones and to compute *solutions*, i.e., values for the variables that satisfy all constraints simultaneously.

One of the main goals of constraint programming is to develop programming languages that allow one to express constraint problems in a natural way, and to solve them efficiently.

3.1.1. Finite domain constraint programming

In our work, we are first interested in constraint problems over finite domains. In this case, the domain of each variable (the set of values it may take) is a finite set of integer numbers. Theory tells us that most constraint problems over finite domains are NP-hard, which means that there is little hope to solve them by algorithms polynomial in the size of the input. In practice, these problems are handled by tree search methods which try successively different valuations of the variables until a solution is found. Because of the exponential number of possible combinations, it is crucial to reduce the search space as much as possible, i.e., to eliminate *a priori* as many valuations as possible.

There exist two generic methods to solve such problems. The first one is classical *integer linear programming* (see also Sect. 3.3), which has been studied in mathematical programming and operations research for more than 40 years. Here, constraints are linear equations and inequalities over the integer numbers. In order

to reduce the search space, one typically uses the linear relaxation of the constraint set. Equations and inequalities are first solved over the real numbers, which is much easier; then the information obtained is used to prune the search tree.

The second method is *finite domain constraint programming* which arose in the last 15 years by combining ideas from declarative programming languages and constraint satisfaction techniques in artificial intelligence. In contrast to integer linear optimization one uses, in addition to simple arithmetic constraints, more complex constraints, which are called *symbolic constraints*. For instance, the symbolic constraint $\texttt{alldifferent}(x_1, ..., x_n)$ expresses that the values of the variables $x_1, ..., x_n$ must be pairwise distinct. Such a constraint is difficult to express in a compact way using only linear equations and inequalities. Symbolic constraints are handled individually by specific filtering algorithms that reduce the domain of the variables. This information is propagated to other constraints which may further reduce the domains.

A state-of-the-art survey of finite domain constraint programming, with special emphasis on its relation to integer linear programming can be found in [5].

3.1.2. Concurrent constraint programming

In *concurrent* constraint programming (cc) [37], different computation processes may run concurrently. Interaction is possible via the *constraint store*. The store contains all the constraints currently known about the system. A process may *tell* the store a new constraint, or *ask* the store whether some constraint is entailed by the information currently available, in which case further action is taken.

Hybrid concurrent constraint programming (`Hybrid cc`) [36] is an extension of concurrent constraint programming which allows one to model and to simulate the temporal evolution of *hybrid systems*, i.e., systems that exhibit both discrete and continuous state changes. Constraints in `Hybrid cc` may be both algebraic and differential equations. State changes can be specified using the combinators of concurrent constraint programming and default logic. `Hybrid cc` is well-suited to model dynamic biological systems, as shown in [4].

3.2. Statistical learning

Statistical learning theory [39] is one of the fields of inferential statistics the bases of which have been established by V.N. Vapnik in the late 1960s. The goal of this theory is to specify the conditions under which it is possible to «learn» from empirical data obtained by random sampling. Learning amounts to solving a problem of function or model selection. Basically, given a task characterized by a joint probability distribution on pairs made up of observations and labels, and a class of functions, of cardinality ordinarily infinite, the goal is to find in the class a function with optimal performance. Training can thus be reformulated as an optimization problem. In many cases, the objective function is related to the capacity of the class of functions [19]. The learning tasks considered belong to one of the three following areas: pattern recognition (discriminant analysis), function approximation (regression) and density estimation.

This theory considers more specifically two inductive principles. The first one, named empirical risk minimization (ERM) principle, consists in minimizing the training error. If the sample is small, one substitutes to this the structural risk minimization (SRM) principle. It consists in minimizing an upper bound on the expected risk (generalization error), a bound sometimes called a guaranteed risk. This latter principle is implemented in the training algorithms of the support vector machines (SVMs), which currently constitute the state-of-the-art for numerous problems of pattern recognition.

SVMs are connectionist models conceived to compute indicator functions, to perform regression or to estimate densities. They have been introduced during the last decade by Vapnik and co-workers [35], as nonlinear extensions of the maximal margin hyperplane [38]. Their main advantage is that they can avoid overfitting in the case where the size of the sample is small [39], [32].

3.3. Combinatorial optimization and integer programming

"Combinatorial optimization is a lively field of applied mathematics, combining techniques from combinatorics, linear programming, and the theory of algorithms, to solve optimization problems over discrete

structures" [34]. A combinatorial optimization problem can be defined as follows: we are given a ground set N and consider a finite collection of subsets, say $\{S_1, S_2, \cdots, S_m\}$. For each subset S_k there is an objective function value, $f(S_k)$, typically a linear function over the elements in S_k. The task is to find the subset S_k that minimizes $f(S_k)$. Typically, the feasible subsets are represented by inclusion or exclusion of members such that they satisfy certain conditions. Well known examples of combinatorial optimization problems are assignment, covering, cutting stock, knapsack, matching, packing, partitioning, routing, sequencing, scheduling (jobs), shortest path, spanning tree, and traveling salesman problems.

This then becomes a special class of integer programs (IP) whose decision variables are binary valued: $x_i = 1$ if the i-th element is in the optimal solution; otherwise, $x_i = 0$. In this case, feasible subsets have to be expressed by linear constraints. IP formulations are not always easy, and often there is more than one formulation, some better than others. Many good formulations have exponential size.

4. Application Domains

4.1. Molecular biology

Participants: Ernst Althaus, Delphine Autard, Alexander Bockmayr, Stefan Canzar, Arnaud Courtois, Yannick Darcy, Eric Domenjoud, Yann Guermeur, Abdelhalim Larhlimi, Emmanuel Monfrini, Sandrine Schermack-Peyrefitte, Frédéric Sur.

Molecular biology is concerned with the study of three types of biological macromolecules: DNA, RNA, and proteins. Each of these molecules can initially be viewed as a string on a finite alphabet: DNA and RNA are nucleic acids made up of nucleotides A,C,G,T and A,C,G,U, respectively. Proteins are sequences of amino acids, which may be represented by an alphabet of 20 letters.

Molecular biology studies the information flow from DNA to RNA, and from RNA to proteins. In a first step, called *transcription*, a DNA string ("gene") is transcribed into messenger RNA (mRNA). In the second step, called *translation*, the mRNA is translated into a protein, where each triplet of nucleotides encodes one amino acid ("genetic code"). During transcription, an intermediate maturation step can occur, which happens mainly in eukaryotic cells. In the so-called *splicing* process, introns are removed from the premessenger RNA. The remaining exons are concatenated yielding the mature RNA molecule.

Biological macromolecules are not just sequences of nucleotides or amino acids. Actually, they are complex three-dimensional objects. DNA shows the famous double-helix structure. RNA and proteins fold into complex three-dimensional structures, which depend on the underlying sequence. RNA is a single-stranded chain of nucleotides. However, a nucleotide in one part of the molecule can base-pair with a nucleotide in another part, following the Watson-Crick complementarity rules. This results in a folding of the molecule. The *secondary structure* of RNA indicates the set of base pairings in the three dimensional structure of the molecule. This information can be represented by a graph.

Proteins have several levels of structure. Above the primary sequence is the *secondary structure*, which involves three basic types: α*-helices*, β*-sheets*, and structure elements that are neither helices nor sheets, called *loops*. A *domain* of a protein is a combination of secondary structure elements with some specific function. It contains an *active site* where an interaction with an external molecule may happen. A protein may have one or several domains.

The ultimate goal of molecular biology is to understand the *function* of biological macromolecules in the life of the cell. Function results from the *interaction* between different macromolecules, and depends on their structure. The overall challenge is to make the leap from sequence, through structure, to understand about the function.

4.2. Crystallography

Participants: Alexander Bockmayr, Eric Domenjoud.

X-ray structure analysis is the main tool to establish the three-dimensional atomic structure of biological macromolecules and their complexes. The determination of a structure in X-ray crystallography passes through several stages:

- purification and crystallization of the object under study (a protein, DNA, RNA, virus, or a huge macromolecular complex, such as ribosome or lipoprotein particles);
- X-ray experiment (usually at synchrotron accelerators); data collection (up to a million of independent observations) and their primary processing;
- the solution of the inverse problem of the theory of diffraction to find the electron density distribution in the studied object and to interpret it in terms of atoms.

A key problem of X-ray structure analysis is the so-called *phase problem*. In an X-ray experiment, one can measure only the magnitudes of the complex Fourier coefficients of the electron density distribution under study, but not their phases. Half of the necessary information is therefore lost, and must be restored by other means.

4.3. Operations research

Participants: Ernst Althaus, Alexander Bockmayr, Eric Domenjoud.

While molecular biology has become the main application area of our work, we continue to study selected problems from other domains, in particular operations research. During this year, we have been working on graph and network design problems, and also on problems from computational geometry and linguistics. The corresponding results are presented in Sect. 6.10.

5. Software

5.1. M-SVM: Multi-class Support Vector Machine

Participant: Yann Guermeur [correspondent].

We have extended the functionalities and optimized the code of the application devoted to the standard M-SVM (M-SVM1 in [9]). The corresponding pieces of software have been registred at the APP under the IDDN number IDDN.FR.001.170014.000.R.P.2005.000.10000.

6. New Results

6.1. Structural risk minimization inductive principle for multi-class discriminant analysis

Keywords: *Statistical learning theory*, *model selection*, *support vector machine*.

Participants: Yannick Darcy, Yann Guermeur, Frédéric Sur.

We have continued our study of the generalization error of large margin multi-class discriminant models, laying emphasis on the use of bounds for model selection. A first algorithm of model selection, dedicated to M-SVMs, was based on a bound on the entropy numbers of the evaluation operator [27]. The computation of tighter bounds on those entropy numbers is still a work in progress. It takes the form of the derivation of a generalized formulation of the Maurey-Carl theorem [33]. Those bounds will then be compared with those involving extended Sauer's lemmas and generalized VC dimensions [18]. In parallel, the work on the computation of estimates of the risk based on the leave-one-out procedure has given birth to a first theorem [31], extending Chapelle's "radius-margin bound". All the aforementioned bounds are progressively incorporated in our M-SVM software, where they can be used to select the soft margin parameter C.

6.2. Probabilistic automata inference

Keywords: *Statistical learning theory*, *grammatical inference*, *probabilistic automata*, *rational languages*.

Participant: François Denis.

In Probabilistic Grammatical Inference, it is supposed that learning data consist in a sequence of words over a finite alphabet Σ drawn according to a fixed but unknown probability distribution P called a *stochastic language*. Then, the goal is to find a model, which can be a probabilistic automata (PA) or a Hidden Markov Model (HMM) for instance, consistent with the data. Hidden Markov Models and Probabilistic Automata have the same expressivity and their relationship have been precisely studied in [17]. With Yann Esposito, from the "Laboratoire d'informatique fondamentale de Marseille" (LIF), we have proved in [25] that stochastic languages p generated by probabilistic automata A depend continuously on the parameters of A, for the $||\cdot||_\infty$ norm. As a corollary, we prove that probabilistic automata can be identified in the limit and that the identification is exact when the parameters of the target are rational numbers. However, this result is theoretical and does not lead to a practical learning algorithm. The main difficulty is to infer an appropriate structure from the data: this is possible when natural components of the model correspond to intrinsic components of the target language. We defined the notions of *residual languages* of a stochastic language and *Probabilistic Residual Automata*. A PRA is a PA whose states directly correspond to the residual of the language it generates. When the target stochastic language can be generated by a PRA, an efficient learning algorithm can be defined (see [25]). Stochastic languages defined from probabilistic automata are rational languages and we feel necessary to study Rational Stochastic Languages from a Language Theoretical point of view. Main results have been described in [26]. A main publication is in preparation.

6.3. Semi-supervised learning; application to the disulfide bridges prediction

Keywords: *Statistical learning theory*, *bio-informatics*, *semi-supervised learning*.

Participant: François Denis.

Semi-supervised learning algorithms aimed to exploit simultaneously labeled and unlabeled data for classification. We have been working for several years on a specific semi-supervised learning problem: binary classification from positive and unlabeled data. Theoretical results, strengthened by experimental results, have proved that many learning algorithm can be adapted to this context (see [16]). With Christophe Magnan, who is doing a PhD on this subject at the LIF, we are currently studying applications of this paradigm to a biological problem: disulfide bridges prediction [28]. We are also working, with Liva Ralaivola (MdC, Université de Provence), on a more sophisticated model in order to deal with contact maps in proteins.

6.4. Boosting blast

Keywords: *Statistical learning theory*, *bio-informatics*, *boosting*.

Participant: François Denis.

The function of a single protein is mainly carried out by a *domain* which is a subsequence of amino-acids within the whole sequence of the protein. During evolution, the sequence of such a domain can be significantly modified while the function is still conserved. Our work deals with functional families whose domains are not well conserved during evolution. Let F be a functional family, let $P = \{p_1, ..., p_n\}$ a set of annotated proteins which are known to belong or not to F, our problem is to decide whether any new protein p belongs to F.

In many cases, comparing a new sequence of protein p with some sequences of the family F is enough for predicting whether $p \in F$. Such a similarity search may be achieved by using either an alignment program such as BLAST or any model of the family's sequences, for example stochastic and probabilistic models such as Hidden Markov Models. Unfortunately, none of these methods is satisfactory whenever the sequences of the domains of the family are not conserved. Our proposal is to use a *boosting* algorithm associated with BLAST to deal with this problem. First results have published in [24], [23]. Cécile Capponi, at the LIF, is leader on this thema.

6.5. Shape recognition in digital images

Keywords: *Shape recognition*, *a contrario models*, *cluster analysis*.

Participant: Frédéric Sur.

Shape recognition is the field of computer vision which addresses the problem of finding out whether a query shape lies or not in a shape database, up to a certain invariance. Most shape recognition methods simply sort shapes from the database along some (dis-)similarity measure to the query shape. Their Achilles' heel is the decision stage, which should aim at giving a clear-cut answer to the question: "do these two shapes look alike?" In [13], [21], the proposed solution consists in bounding the number of false correspondences of the query shape among the database shapes, ensuring that the obtained matches are not likely to occur "by chance". As an application, one can decide with a parameterless method whether any two digital images share some shapes or not. In a paper submitted to VISAPP'06, we propose to apply the above *a contrario* methodology to shapes which are described by size functions, in order to design a perceptual matching algorithm.

A further step consists in grouping matching shapes that share the same respective positions in two corresponding images. In [30], we intend to form spatially coherent groups of shapes. Each pair of matching shape elements indeed leads to a unique transformation (similarity or affine map.) A unified *a contrario* detection method is proposed to solve three classical problems in clustering analysis. The first one is to evaluate the validity of a cluster candidate. The second problem is that meaningful clusters can contain or be contained in other meaningful clusters. A rule is needed to define locally optimal clusters by inclusion. The third problem is the definition of a correct merging rule between meaningful clusters, permitting to decide whether they should stay separate or unit. As an application, the present theory on the choice of the right clusters is used to group shapes by detecting clusters in the transformation space.

6.6. Protein structure prediction

Keywords: *Statistical learning*, *disulphide bridges*, *kernel engineering*, *protein secondary and tertiary structure*.

Participants: Yannick Darcy, Yann Guermeur, Frédéric Sur.

Knowing the three-dimensional structure of a protein can greatly help to infer its function. Predicting this *tertiary structure* from the sequence of amino acids (or *primary structure*), remains one of the central open problems in structural biology. This is the subject of the «GENOTO3D» project that we coordinate. This year, our main efforts have been concentrated on the development of a new kernel for our M-SVM dedicated to protein secondary structure prediction, a kernel based on a pair-HMM.

Our collaboration with Nicolas Sapay and Gilbert Deléage, at IBCP, in Lyon, on the prediction of amphipathic in-plane membrane anchors in motopic proteins, has given birth to a new prediction method, «AmphipaSeek» [29], which is available from the website of the PBIL, at the following address : http://npsa-pbil.ibcp.fr/cgi-bin/npsa_automat.pl?page=/NPSA/npsa_amphipaseek.html.

6.7. Modeling the FAK protein (ATIPE)

Keywords: *clustering*, *homology modeling*, *protein structures*.

Participants: Ernst Althaus, Stefan Canzar.

Imposing a classification on an otherwise unordered protein fold space aids our understanding of protein evolution and the relationship between three-dimensional structure and function. We describe a similarity model that provides the objective basis for clustering proteins of similar structure. More specifically, we consider the following variant of the protein-protein similarity problem: We want to find proteins in a large database *pdbase* that are very similar to a given query protein in terms of geometric shape. We give experimental evidence, that the shape similarity model of Osada, Funkhouser, Chazelle and Dobkin can be transferred to the context of protein structure comparison. This model is very simple and leads to algorithms that have attractive space requirements and running times. For example, it took 0.39 second to retrieve the

eight members of the seryl family out of 26,600 domains. Furthermore, a very high agreement with one of the most popular classification schemes proved the significance of our simplified representation of complex proteins structure by a distribution of C_α-C_α distances.

6.8. Multiple sequence alignment (ATIPE)

Keywords: *cutting planes*, *integer programming*, *multiple sequence alignment*.

Participants: Ernst Althaus, Stefan Canzar.

In [14], we propose a cutting plane approach for the alignment of multiple sequences, which is a central problem in computational biology, considering the general case in which (arbitrary) gap costs, besides the customary alignment costs, are specified. An interesting and unusual aspect of our approach is that the three (exponentially large) classes of natural valid inequalities that we considered since the beginning of our study turn out to be both facet defining for the convex hull of the integer solutions and separable in polynomial time. Both the facet defining proofs and the separation algorithms are far from trivial. Experimental results on instances from the BAliBase library of reference alignments show that our method outperforms the best tools developed so far, in that it produces alignments which are better from a biological point of view.

At the moment, we are working in improving the computational efficiency of your approach. The bottleneck of our implementation is the solution of the (exponentially large) linear relaxation of our integer program. We try to develop methods to approximate the value of the linear program efficiently.

6.9. Computing Steiner minimum trees in Hamming metric (ATIPE)

Keywords: *Steiner minimum trees*.

Participant: Ernst Althaus.

In [22] we consider the problem of computing a Steiner minimum tree in Hamming metric. Given a set $T \subseteq U$ of required points (terminals) in an universe U and a cost function $c : U \times U \mapsto \mathbb{R}$, a Steiner tree is a tree connecting $T \cup S$ for a subset $S \subseteq U$. A Steiner minimum tree SMT(T), is a Steiner tree of minimal cost.

The Steiner tree problem is one of the most studied NP-hard optimization problems (probably second after the Traveling Salesman problem). Here we are interested in the variant where U is the set of strings of a certain length d and c is the Hamming distance between two strings.

The main application of this variant of the Steiner tree problem is to compute evolutionary trees in bioinformatics and computational linguistics.

Among all methods for finding such trees, algorithms using variations of a branch and bound method developed by Penny and Hendy have been the fastest for more than 20 years. We describe a new pruning approach that is far superior to previous methods and outline its implementation.

6.10. Approximating k-hop minimum spanning trees (ATIPE)

Keywords: *minimum spanning tree*.

Participant: Ernst Althaus.

In [15], we consider the problem of computing minimum-cost spanning trees with depth restrictions. Specifically, we are given an n-node complete graph G, a metric cost-function c on its edges, and an integer $k \geq 1$. The goal in the *minimum-cost k-hop spanning tree* (kHMST) is to compute a spanning tree T in G of minimum total cost such that the longest root-leaf-path in the tree has at most k edges.

Our main result is an algorithm that computes a tree of depth at most k and total expected cost $O(\log n)$ times that of a minimum-cost k-hop spanning-tree. The result is based upon earlier work on metric space approximation due to Fakcharoenphol et al, and Bartal. In particular, we show that the kHMST problem can be solved exactly in polynomial time when the cost metric c is induced by a so called *hierarchically well-separated tree*.

7. Other Grants and Activities

7.1. Regional projects

We participate in the «Génopole Strasbourg Alsace-Lorraine» together with the laboratory MAEM («Maturation des ARN et Enzymologie Moléculaire»), UMR 7567, in Nancy and the IGBMC in Strasbourg.

In the framework of the CPER Lorraine 2000-2006, we participate in the project «Bioinformatics and Applications to Genomics» of the PRST «Intelligence Logicielle». Our partners here are the Laboratory of Crystallography LCM3B (UMR 7036), the «équipe de Dynamique des Assemblages Membranaires» (eDAM, UMR 7565) and the MAEM (UMR 7567) at the University Henri Poincaré, Nancy 1.

7.2. National projects

Since February 2002, we have been participating in the cooperative research action ARC CPBIO «Process calculi and Biology of Molecular Networks». Our partners are the project team CONTRAINTES from INRIA Rocquencourt (F. Fages), the Genoscope (V. Schächter) and the laboratory PPS (V. Danos) in Paris.

We have regular contacts with the INRIA project teams HELIX (Rhône-Alpes), SYMBIOSE (Rennes) and COMORE (Sophia-Antipolis). In particular, we have been collaborating with Hidde de Jong (HELIX) in modeling the regulation of alternative splicing.

Since September 2003, we are coordinating a project called GENOTO3D, which is funded by the «Action Concertée Incitative» (ACI) «Masses de Données». The aim of this project is to apply machine learning approaches to the prediction of the tertiary structure of globular proteins. Our partners are the IBCP in Lyon, the LIF in Marseille, the project team SYMBIOSE from IRISA, the LIRMM in Montpellier, and the MIG laboratory of INRA in Jouy-en-Josas.

7.3. International relations

Within the French-Russian Institute Liapunov, we have a joint project with the Institute for Mathematical Problems in Biology (IMPB) of the Russian Academy of Sciences in Pushchino (V. Y. Lunin).

We have been collaborating with researchers from Carnegie-Mellon University (E. Balas, John N. Hooker), the Center of Operations Research CORE in Louvain-la-Neuve (L. Wolsey), the Max Planck Institute for Computer Science in Saarbrücken (working groups of K. Mehlhorn and F. Eisenbrand), SAP AG (T. Kasper), the University of California at Irvine (P. Baldi), IBM at Zurich (A. Elisseeff), and the Wiener laboratories in Rosario (D. Zelus).

8. Dissemination

8.1. Serving the scientific community

François Denis has been the program commitee chair of CAP05, the french national conference on Machine Learning, which was held in Nice on June 1-3, 2005.

Yann Guermeur has been a member of the program committee of CAP'05.

8.2. Teaching

Ernst Althaus has taught the following lectures (unless otherwise indicated, all teaching has been done at the Universität der Saarlandes, Saarbrücken, Germany):

- *Datastructures and Algorithms*, October 2004 - February 2005 (together with Dr. Ulrich Meyer)
- Part of the course *Pépites Algorithmiques*, March 2005 at the École des Mines, Nancy, France
- Lecture *Optimization*, April 2005 - July 2005 (together with Dr. Benjamin Dörr)

- Lecture *Datenstrukturen und effiziente Algorithmen*, October 2005 - March 2006 at the Johannes-Gutenberg Universität, Mainz, Germany
- Seminar *Online-Algorithmen*, October 2005 - March 2006 at the Johannes-Gutenberg Universität, Mainz, Germany (together with Elmar Schömer and Marcel Marquardt)
- Lecture *Bioinformatik*, October 2005 - March 2006 at the Johannes-Gutenberg Universität, Mainz, Germany (together with several lectures from the department of biology)

Yann Guermeur has been teaching bioinformatics at a Master of the INPL and the M2P speciality "Génomique et Informatique" of the Master "Sciences de la Vie et de la Santé" (SVS), at the UHP.

8.3. Miscellaneous

Frédéric Sur has been a member of the board of the "Banque PT" entrance examination in mathematics ("Grandes Écoles" entrance examination).

9. Bibliography

Major publications by the team in recent years

[1] E. ALTHAUS, A. CAPRARA, H.-P. LENHOF, K. REINERT. *Multiple sequence alignment with arbitrary gap costs: Computing an optimal solution using polyhedral combinatorics.*, in "Proc. European Conference on Computational Biology", Bioinformatics, vol. 18, n° Supplement 2, October 2002, p. S4–S16.

[2] E.T ALTHAUS, K. MEHLHORN. *Traveling Salesman-Based Curve Reconstruction in Polynomial Time*, in "SIAM Journal on Computing", vol. 31, n° 1, 2001, p. 27–66.

[3] E. BALAS, A. BOCKMAYR, N. PISARUK, L. WOLSEY. *On unions and dominants of polytopes*, in "Mathematical Programming, Ser. A", vol. 299, 2004, p. 223-239.

[4] A. BOCKMAYR, A. COURTOIS. *Using hybrid concurrent constraint programming to model dynamic biological systems*, in "18th International Conference on Logic Programming, ICLP'02, Copenhagen", Springer, LNCS 2401, 2002, p. 85-99.

[5] A. BOCKMAYR, J. N. HOOKER. *Constraint Programming*, in "12: Discrete Optimization", K. AARDAL, G. NEMHAUSER, R. WEISMANTEL (editors). , Handbooks in Operations Research and Management Science, chap. 10, Elsevier, 2005, p. 559–600.

[6] A. BOCKMAYR, V. WEISPFENNING. *Solving numerical constraints*, in "Handbook of Automated Reasoning", A. ROBINSON, A. VORONKOV (editors). , vol. 1, chap. 12, Elsevier, 2001, p. 751-842.

[7] Y. GUERMEUR, A. ELISSEEFF, D. ZELUS. *Bound on the risk for M-SVMs*, in "Statistical Learning, Theory and Applications", 2002, p. 48–52.

[8] Y. GUERMEUR, C. GEOURJON, P. GALLINARI, G. DELÉAGE. *Improved performance in protein secondary structure prediction by inhomogeneous score combination*, in "Bioinformatics", vol. 15, n° 5, 1999, p. 413–421.

[9] Y. GUERMEUR. *Combining discriminant models with new multi-class SVMs*, in "Pattern Analysis and Applications", vol. 5, n° 2, 2002, p. 168–179.

[10] Y. GUERMEUR, A. LIFCHITZ, R. VERT. *A kernel for protein secondary structure prediction*, in "Kernel Methods in Computational Biology", B. SCHÖLKOPF, K. TSUDA, J.-P. VERT (editors). , The MIT Press, 2004, p. 193–206.

[11] Y. GUERMEUR, H. PAUGAM-MOISY. *Théorie de l'apprentissage de Vapnik et SVM, Support Vector Machines*, in "Apprentissage Automatique", M. SEBBAN, G. VENTURINI (editors). , Hermès, 1999, p. 109–138.

[12] VLADIMIR Y. LUNIN, ALEXANDRE URZHUMTSEV, ALEXANDER BOCKMAYR. *Direct phasing by binary integer programming*, in "Acta Crystallographica Section A", vol. 58, 2002, p. 283-291.

[13] P. MUSÉ, F. SUR, F. CAO, Y. GOUSSEAU, J.-M. MOREL. *An a contrario decision method for shape element recognition*, Technical report, n° 2004-16, CMLA, ENS Cachan, 2004.

Articles in referred journals and book chapters

[14] E. ALTHAUS, A. CAPRARA, H.-P. LENHOF, K. REINERT. *Aligning Multiple sequences by Cutting Planes*, in "Mathematical Programming", to appear.

[15] E. ALTHAUS, S. FUNKE, S. HAR-PELED, J. KÖNEMANN, E. A. RAMOS, M. SKUTELLA. *Approximating k-hop minimum-spanning trees*, in "Operations Research Letters", vol. 33, n° 2, March 2005, p. 115–120.

[16] F. DENIS, R. GILLERON, F. LETOUZEY. *Learning From Positive and Unlabeled Examples*, in "Theoretical Computer Science", vol. 348, 2005, p. 70-83.

[17] P. DUPONT, F. DENIS, Y. ESPOSITO. *Links between Probabilistic Automata and Hidden Markov Models: probability distributions, learning models and induction algorithms*, in "Pattern Recognition: Special Issue on Grammatical Inference Techniques & Applications", vol. 38/9, 2005, p. 1349-1371.

[18] Y. GUERMEUR, A. ELISSEEFF, D. ZELUS. *A comparative study of multi-class support vector machines in the unifying framework of large margin classifiers*, in "Applied Stochastic Models in Business and Industry", vol. 21, n° 2, 2005, p. 199–214.

[19] Y. GUERMEUR, O. TEYTAUD. *Estimation et contrôle des performances en généralisation des réseaux de neurones*, in "Apprentissage Connexionniste", Y. BENNANI (editor). , to appear, Hermès, 2005.

[20] P. MUSÉ, F. SUR, F. CAO, Y. GOUSSEAU, J.-M. MOREL. *An a contrario decision method for shape element recognition*, in "International Journal of Computer Vision", to appear, 2006.

[21] P. MUSÉ, F. SUR, F. CAO, Y. GOUSSEAU, J.-M. MOREL. *Shape recognition based on an a contrario methodology*, in "Statistics and analysis of shapes", H. KRIM, A. YEZZI (editors). , Birkhauser, 2006.

Publications in Conferences and Workshops

[22] E. ALTHAUS, R. NAUJOKS. *Computing Steiner Minimum Trees in Hamming Metric*, in "Proceedings of the 17th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA-06), Miami, USA", to appear, 2006.

[23] C. CAPPONI, G. FICHANT, Y. QUENTIN, F. DENIS. *Boosting Blast*, in "Applied Stochastic Models and Data Analysis ASMDA 2005, Brest, France", 2005.

[24] C. CAPPONI, G. FICHANT, Y. QUENTIN, F. DENIS. *Boosting BLAST for classifying protein domains*, in "Actes de JOBIM 2005, Lyon, France", 2005.

[25] F. DENIS, Y. ESPOSITO. *Learning classes of Probabilistic Automata*, in "COLT 2004", LNAI, n° 3120, 2004, p. 124-139.

[26] F. DENIS, Y. ESPOSITO. *Rational stochastic languages*, in "TAGI 2005", 2005.

[27] Y. GUERMEUR, M. MAUMY, F. SUR. *Model selection for multi-class SVMs*, in "ASMDA'05, Brest, France", 2005, p. 507–516.

[28] C. MAGNAN. *Apprentissage semi-supervisé asymétrique et estimations d'affinités locales dans les protéines*, in "Actes de CAP 05", F. DENIS (editor). , PUG, 2005, p. 297-312.

[29] N. SAPAY, Y. GUERMEUR, G. DELÉAGE. *Prediction of in-plane amphipathic membrane segments based on an SVM method*, in "JOBIM'05, Lyon, France", 2005, p. 299–311.

Internal Reports

[30] F. CAO, J. DELON, A. DESOLNEUX, P. MUSÉ, F. SUR. *A unified framework for detecting groups and application to shape recognition*, (submitted to a journal), Technical report, n° RR-5695, INRIA, 2005.

[31] Y. DARCY, Y. GUERMEUR. *Radius-margin bound on the leave-one-out error of multi-class SVMs*, Technical report, n° RR-5780, INRIA, 2005.

Bibliography in notes

[32] C.J.C. BURGES. *A tutorial on support vector machines for pattern recognition*, in "Data Mining and Knowledge Discovery", vol. 2, n° 2, June 1998, p. 121–167.

[33] B. CARL, I. STEPHANI. *Entropy, compactness, and the approximation of operators*, Cambridge University Press, Cambridge, UK, 1990.

[34] W. J. COOK, W. H. CUNNINGHAM, W. R. PULLEYBLANK, A. SCHRIJVER. *Combinatorial Optimization*, Wiley, 1998.

[35] C. CORTES, V.N. VAPNIK. *Support-Vector Networks*, in "Machine Learning", vol. 20, 1995, p. 273–297.

[36] V. GUPTA, R. JAGADEESAN, V. SARASWAT. *Computing with Continuous Change*, in "Science of computer programming", vol. 30, n° 1-2, 1998, p. 3-49.

[37] V. A. SARASWAT. *Concurrent constraint programming*, ACM Doctoral Dissertation Awards, MIT Press, 1993.

[38] V.N. VAPNIK. *Estimation of Dependences Based on Empirical Data.*, Springer-Verlag, N.Y., 1982.

[39] V.N. VAPNIK. *Statistical learning theory*, John Wiley & Sons, Inc., N.Y., 1998.

[40] P. VAN HENTENRYCK, V. SARASWAT. *Strategic directions in constraint programming*, in "ACM Computing Surveys", vol. 28, n° 4, 1996, p. 701–726.