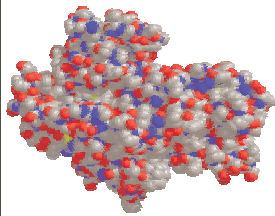




CP(BioNet) : a Constraint Programming framework for the analysis of Biochemical Networks

*Yves Deville, Pierre Dupont,
Grégoire Dooms, Stéphane Zampelli*

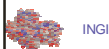


Rennes, IRISA, 6 December 2004



Overview

- Bioinformatics and Biological Networks
- Constraint Programming
- CP(BioNet)
 - The framework
 - Constraints for bio.net path finding
 - Constraints for bio.net matching
- Perspectives



INGI

2



Background

- Sabbatical in 2002-2003
 - Bioinformatics : modeling of biochemical networks
 - *Bioinformatics Research Center*, Glasgow (Prof. David Gilbert)
 - *Molecular Biology and Bioinformatics* Dpt, ULB, Brussels (Prof. Shoshana Wodak)
- Ongoing Research project
 - Analysis of biochemical networks
 - Use of Constraint Programming



INGI

3



What is Bioinformatics ?

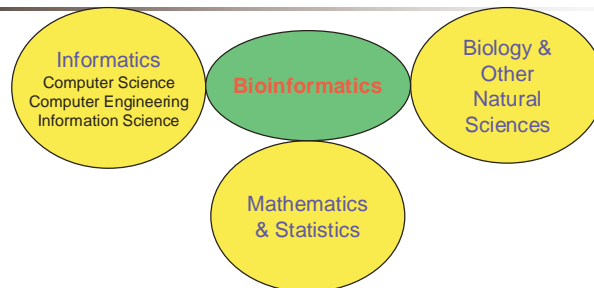
- An intersection of AI and genetics
 - Two very popular (most wanted) sciences
- An opportunity
 - To use some of the most interesting computational techniques to solve some of the most important and rewarding questions
- Where Frankenstein meets the Terminator



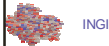
INGI

4

What is Bioinformatics?



- Bioinformatics : the study of the application of
 - molecular biology, computer science, artificial intelligence, statistics and mathematics
 - to model, organize, understand and discover interesting information associated with the large scale molecular biology databases
 - to guide essays for biological experiments



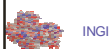
INGI

5

Challenges in Bioinformatics

- Many **NP-hard problems**: multiple alignment, distant homology, motif finding, protein folding, phylogeny, gene relationship in expression data, mining and learning, ...
- Predicting interactions between genes and molecules
- From whole genome to functioning system of a biological organism

Can Constraint Programming be helpful in some of these challenges ?



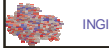
INGI

6

Topics in Bioinformatics

- Sequencing genome
- **Sequence alignment**
- Searching databases
- Machine learning
- Hidden Markov Model
- **Phylogenetic trees**
- Functional genomics
- **Simulation**
- **Structure prediction**
- Microarrays (DNA chips)
- Biochemical databases
- **Biochemical network analysis**
- Ethical, legal & social issues
- ...

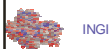
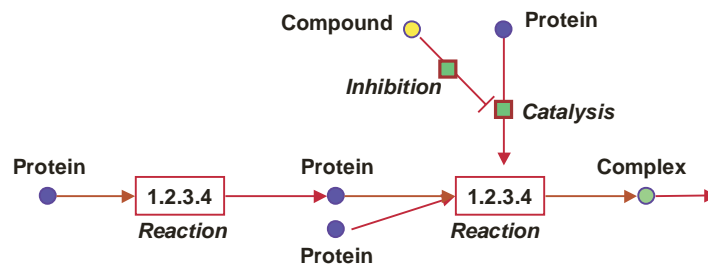
CP already used in some topics



7


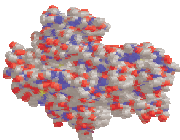
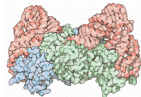
Biological Networks

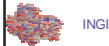
- Networks of **interactions** between **biological entities** within the cell
- Bioentities and Interactions observed in experiments
- Stored in databases



8

Bioentities

- Gene (part of the DNA)
- Polypeptide (Protein)
- Complex (formed by several polypeptides)
- Compound (ATP, ADP, Water, Proline, ...)

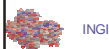


INGI

9

Interactions

- An interaction may be a **transformation** of bioentities into other bioentities
 - **Reaction** : chemical reaction occurring within the cell
 - **Expression** : Gene \rightarrow polypeptide
 - **Assembly** : polypeptide forming a complex
- An interaction may be a **control** of a transformation
 - **Catalysis** of a reaction by some enzyme (protein)
 - **Regulation** of the expression of a gene

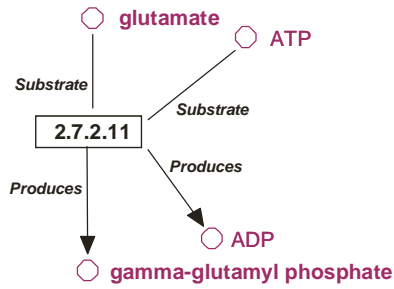


INGI

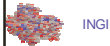
10



Chemical reaction



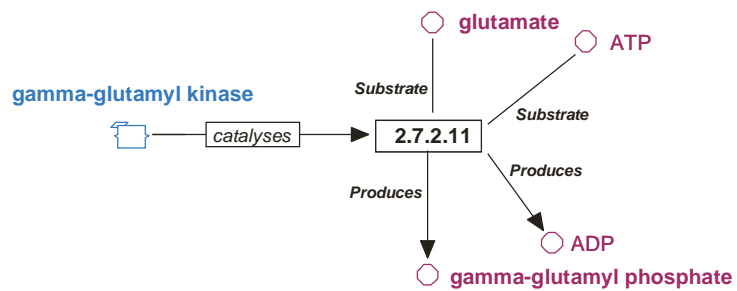
1.5.1.2 EC (reaction) number
○ compound



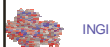
[aMAZE, 2002] 11



Enzymatic catalysis

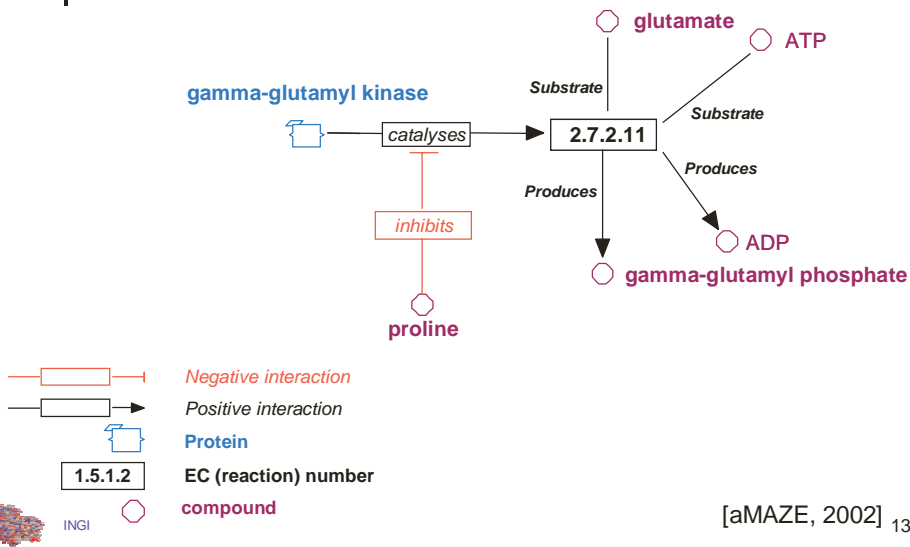


→ Positive interaction
Protein
1.5.1.2 EC (reaction) number
○ compound

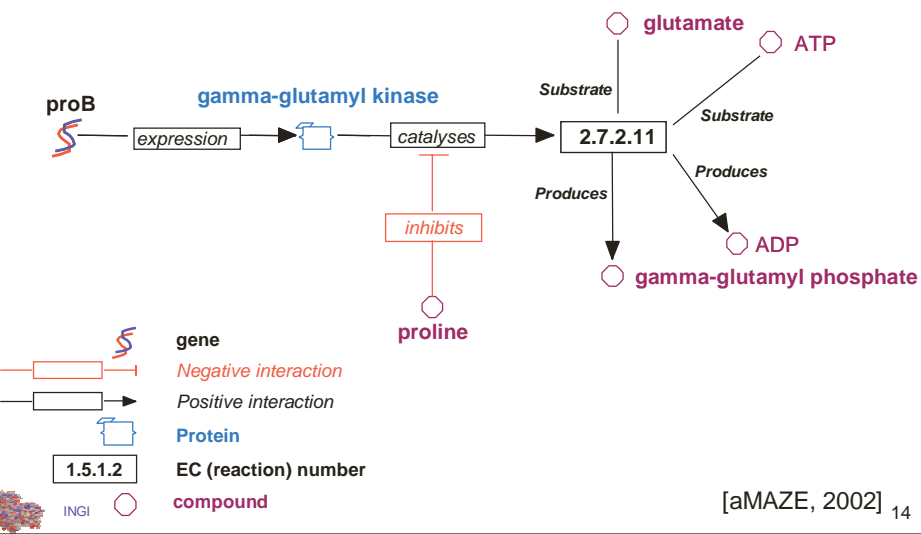


[aMAZE, 2002] 12

Inhibition / Activation



Gene expression





Biochemical networks

One usually distinguish different types of networks :

- **Metabolic** network
 - Series of reactions, possible controlled by enzymes, leading to some specific product
- **Regulatory** network
 - Focus on the regulation of the enzyme activity, or on the stimulation of the enzyme expression
- **Signal transduction** network
 - Transport of information (from membrane to gene)

*These networks are usually represented using
different models, and stored in different databases*



INGI

15



The aMAZE project

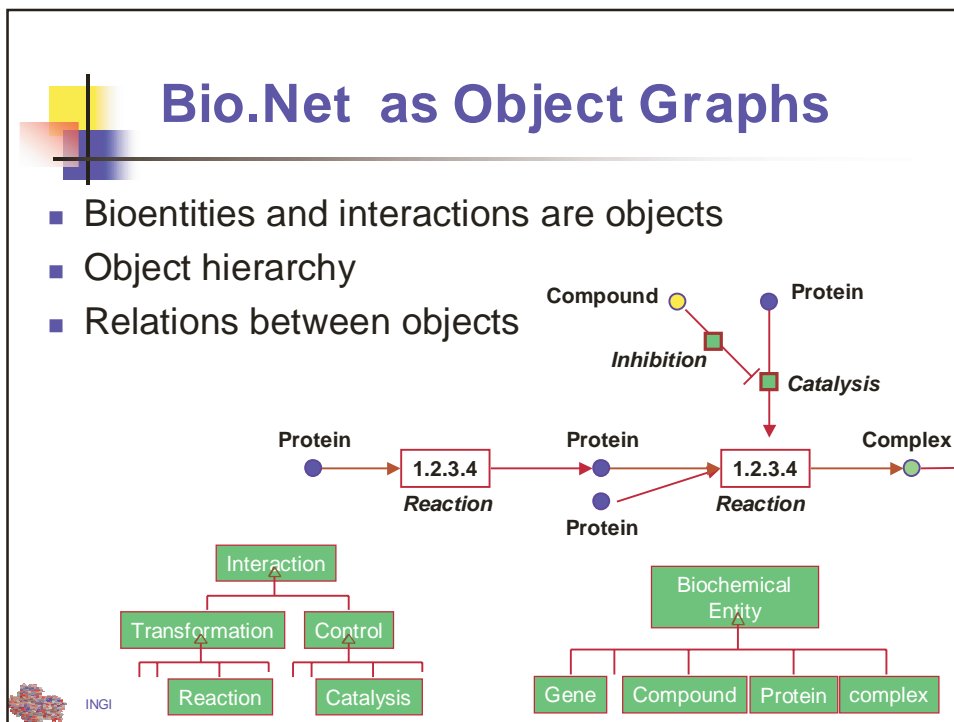
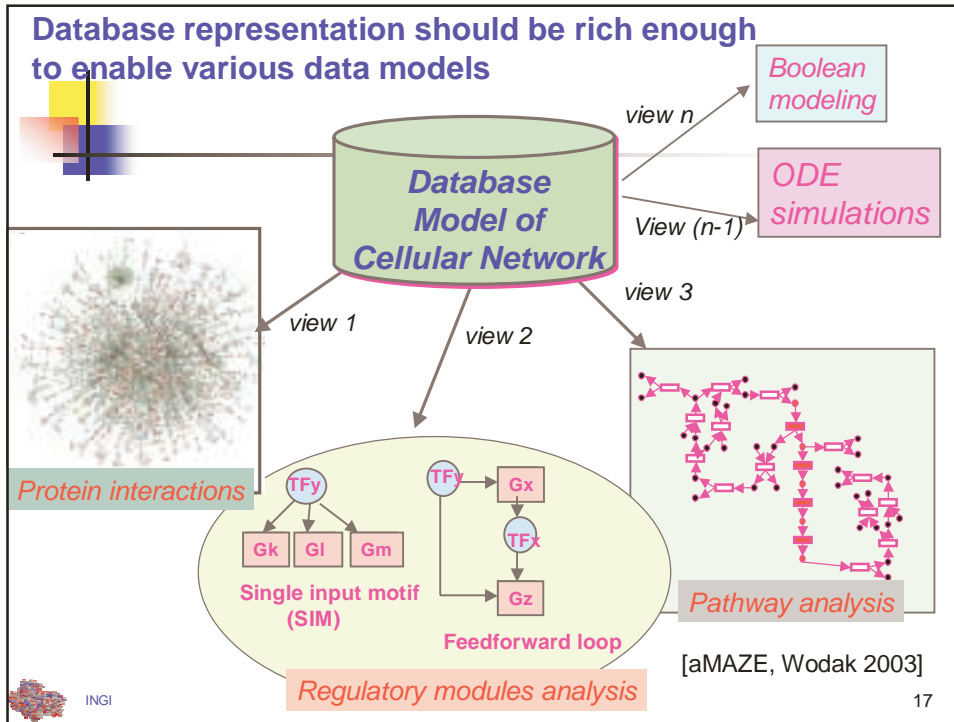
- **Database for biological networks**
- Prof. Shoshana Wodak (Molecular Biology & Bioinformatics – ULB – Belgium, Dpt of Chemistry – Univ of Toronto – Canada)
- Based on a rich **Object Oriented model**:
- **Integration of different types of networks**
 - metabolism
 - regulation
 - signal transduction, etc
- Extendable model

www.amaze.ulb.ac.be



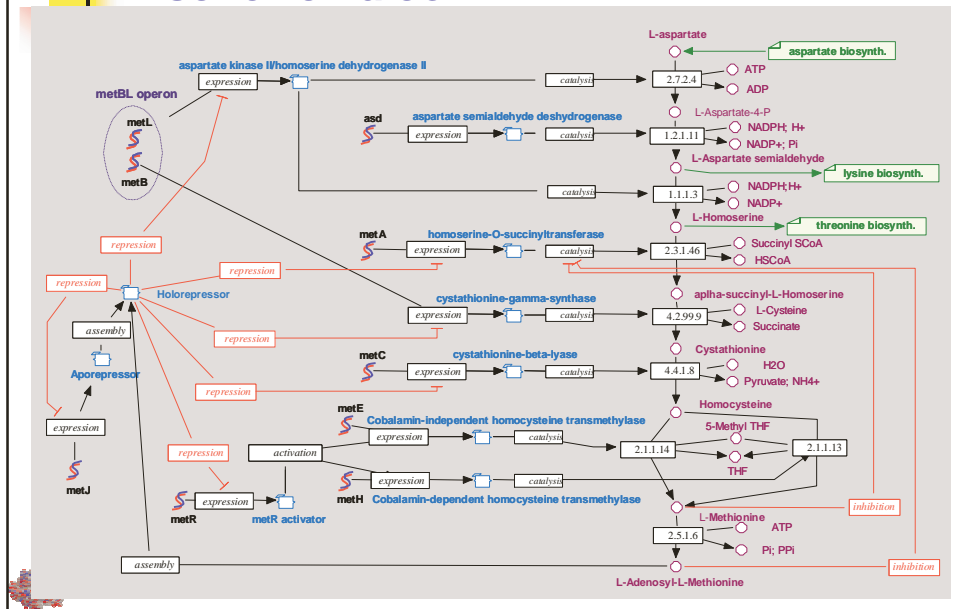
INGI

16



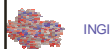
Methionine Biosynthesis in *Escherichia coli*

[aMAZE, van Helden 2003]



The BioMaze Project

- Analysis & Visualisation of biochemical networks
- Closely related to the aMAZE project
- Interdisciplinary and interuniversity project
 - Y. Deville, CS, UCL, Louvain-la-Neuve
 - S. Wodak, Bio, ULB, Brussels
 - J.L. Hainaut, CS, FUNDP, Namur
 - E. Zimany, CS, ULB. Brussels
- Funded by the Walloon Region



INGI

20



Why analyzing networks ?

Examples of biological questions

- Find all **pathways** traversing a set of specified compounds or reactions (e.g. given a set of co-regulated genes, find a pathway that could be formed with the catalyzed reactions).
- Find all **genes** whose expression is directly or indirectly affected by a given compound.
- Show which paths or **pathways** may be **affected** when one or more gene/proteins are turned off or missing.
- **Compare** biochemical **pathways** from different organisms and tissues, or at different stages of annotation; highlight common features and differences; predict missing elements.



INGI

21



Existing Approaches

- Analyses on the database (**SQL queries**)
- Analyses using a **simplistic model** of biochemical networks
 - E.g. *compound graphs* where nodes are bioentities and arcs are the reactions
 - Many useful analysis are meaningless in such models
- Analyses using **specialized algorithms** (often based on graphs)



INGI

22



Problems with existing approaches

- Complex queries
 - Cannot be handled by SQL or simplistic models
 - Specialized algorithms are tough to program
 - Slight changes in queries may be tough to program
 - Difficulties to combine analyses/queries
 - Some queries are really complex (NP-hard)



INGI

23



Objectives of the research

- Design **methods** and associated software **tools** for the analysis of biochemical networks
- Flexibility and genericity :
 - Analysis should be specified/described easily
 - Analysis should be performed incrementally/interactively
 - Possibility to combine analyses
 - Slight changes in the analysis should be supported
 - Support different types of analysis
 - Extendable to new types of analysis
 - Efficiency (but no miracle)
 - Interface with existing DB and visualisation tools



INGI

24



Overview

- Bioinformatics and Biological Networks
- **Constraint Programming**
- CP(BioNet)
 - The framework
 - Constraints for bio.net path finding
 - Constraints for bio.net matching
- Perspectives



INGI

25



Motivation

- Hard Problems
 - Combinatorial Search / Optimization Problems
 - Scheduling Problems
 - Resource Allocation Problems
 - Highly Nonlinear Problems
 - Nonlinear Optimization Problems
- Computational Complexity
 - NP-Hard or worse (at least **exponential**)
 - No reasonable approximation
- **Yet, it is important to solve them**
- The best specific algorithms
 - tough to design and implement
 - experimental



INGI

26



Constraint Programming

The Hope

- To reduce the development time
- To preserve efficiency

The CP framework

- Solving Constraint Satisfaction Problems
- Constraints as basic computational elements
- Declarative semantics, modeling tool
- **Search** : explore the search space
- **Propagation** : consistency technique to reduce the search space



INGI

27



Constraint Satisfaction Problem

- What is a CSP ?
 - A set of **variables**, defined over **domains**
 - A set of **constraints** over the variables
- What is a solution?
 - A **solution** is a consistent **assignment** of values to the variables
 - **Consistent** assignment : does not violate any constraint
- Possibility to have an **objective** function
 - Find a solution maximizing the objective function



INGI

28

N-queens

The problem

Place n queens on a $n \times n$ board such that no queen attacks each other

Variables

- X_i : position (column) of the queen on row i ($1 \leq i \leq 8$)

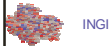
Domains

- Domain of X_i : $\{1,2,3,4,5,6,7,8\}$

Solution

$$\begin{aligned} X_1 &= 1 & X_5 &= 3 \\ X_2 &= 5 & X_6 &= 7 \\ X_3 &= 8 & X_7 &= 2 \\ X_4 &= 6 & X_8 &= 4 \end{aligned}$$

	1	2	3	4	5	6	7	8
X1	♛							
X2					♛			
X3								♛
X4						♛		
X5			♛					
X6							♛	
X7		♛						
X8				♛				



INGI

29

N-queens

Constraints

- Two queens cannot be on the same column

$$X_i \neq X_j$$

- Two queens cannot be on the same diagonal

$$X_i - X_j \neq i - j$$

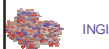
$$X_i - X_j \neq j - i$$

8-queens

- 8 variables
- 84 constraints

$X_1 \neq X_2, X_1 \neq X_3, X_1 \neq X_4, X_1 \neq X_5, X_2 \neq X_3, X_2 \neq X_4, X_2 \neq X_5, X_3 \neq X_4, X_3 \neq X_5, X_4 \neq X_5, \dots$

$X_1 - X_2 \neq -1, X_1 - X_2 \neq 1, X_1 - X_3 \neq -2, X_1 - X_3 \neq 2, X_1 - X_4 \neq -3, X_1 - X_4 \neq 3, X_1 - X_5 \neq -4, X_1 - X_5 \neq 4, X_2 - X_3 \neq -1, X_2 - X_3 \neq 1, X_2 - X_4 \neq -2, X_2 - X_4 \neq 2, X_2 - X_5 \neq -3, X_2 - X_5 \neq 3, X_3 - X_4 \neq -1, X_3 - X_4 \neq 1, X_2 - X_5 \neq -2, X_3 - X_5 \neq 2, X_4 - X_5 \neq -1, X_4 - X_5 \neq 1, \dots$



INGI

30

Program

```
var int queens[1..8] in 1..8
solve {
  forall( ordered i, j in 1..8 ) { queens[i] <> queens[j];
    queens[i] + i <> queens[j] + j;
    queens[i] - i <> queens[j] - j;
  }
};
search {
  forall( i in 1..8 ) { generate(queens[i]); }
};
```

variables

domains

constraints

search

INGI

31

Propagation

- Objective
 - Remove values that cannot be part of a solution
 - Reduce the domains of the variables
 - Efficient algorithms
- Principle
 - Solve a relaxation of the problem
 - Consider the constraints **locally**
- Consistency techniques
 - Forward checking
 - Arc-consistency
 - Global constraints

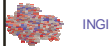
INGI

32

Forward checking

- When a variable X has a value, consider the constraints involving this variable
- Remove the values inconsistent with the value of X

	1	2	3	4	5	6	7	8
X1	👑							
X2			👑					
X3						👑		
X4								
X5								
X6								
X7								
X8								



33

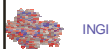
Arc consistency

$$X \geq Y + 2$$

$$X \in \{1, 2, 3, 4, 5\}$$

$$Y \in \{2, 3, 4, 5, 6\}$$

- remove any values from the domain of X for which there is no value in the domain of Y that satisfies the constraint (and vice versa)



34

Arc consistency

$$X \geq Y + 2$$



- Remove any values from the domain of X for which there is no value in the domain of Y that satisfies the constraint (and vice versa)
- Apply arc consistency to each constraint until more values can be removed



INGI

35

Arc consistency

	1	2	3	4	5	6	7	8
X1	👑							
X2			👑					
X3					👑			
X4							✗	
X5								✗
X6				👑				
X7		✗				✗		
X8								

There is no value in the domain of X4 supporting the value 8 of X5



INGI

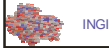
36



Domains of computation

CP frameworks and systems have been designed for different computation domains

- CP(B) : Boolean $X \in \{0, 1\}$
- CP(FD) : Finite domain $X \in \{a_1, a_2, \dots, a_n\}$
- CP(FS) : Finite set $X \subseteq \{a_1, a_2, \dots, a_n\}$
- CP(R) : Intervals (floats) $X = [r, l]$



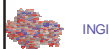
INGI

37



Overview

- Bioinformatics and Biological networks
- Constraint Programming
- CP(BioNet)
 - **The framework**
 - Constraints for bio.net path finding
 - Constraints for bio.net matching
- Perspectives



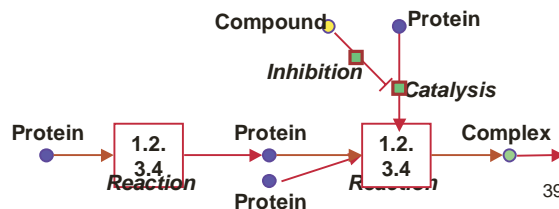
INGI

38

Framework

- Storage of biochemical networks
 - Integrated Object-Oriented model
 - Relational model
- Analysis
 - Non-directed graph model: typed nodes (entities, transforms, controls) + arcs
 - Types will be exploited in the analyses

A graph representation of Biochemical network is used for the analysis



Framework of CP(BioNet)

- Introduction of **graph domain** variable
 - Values are *biochemical networks*
 - Domains are sets of biochemical networks
- Introduction of **node domain** variable
- Introduction of **arc domain** variable
- Definition of **constraints**
 - Dealing with graph, node and arc domain variables
 - Useful for the analysis of *biochemical networks*
 - Extendable framework
 - Implementation of **propagator** for the constraints



Related Work in CP

- **Path constraints** [Sellmann 1998] [Cambazard, Bourreau, 2004]
 - Sellmann : arc variables only
 - Cambazard : node variables only
- **Graph matching**
 - [Rudolf 1998], [Larrosa & Valiente 2000] : CP approach to graph matching
 - [Sorlin & Solnon 2004] : global constraint for graph isomorphism
- **Monadic second-order logic of graphs** [Courcelle]
 - CP with set variables = MSOL + counting
 - MSOL on graph with quantif. node/arc > MSOL on graph with quantif. Node
- **Set constraints** (e.g. [Gervet, Dovier])
 - Implementation techniques
 - Propagation techniques
- **Global constraints** (e.g. [Beldiceanu, Caseau, Müller])
 - Graph algorithms
 - Implementation techniques



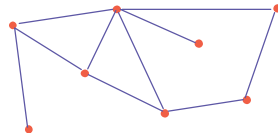
INGI

41



Graph

- A biochemical network is represented by a graph
- A graph $g=(N,A)$ is defined by
 - N : set of **nodes**
 - A : set of **arcs** ($A \subseteq N \times N$)



For simplicity of presentation, no types for the nodes

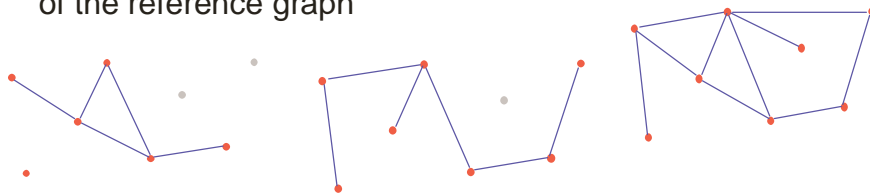


INGI

42

Graph domain variable

- A **graph domain variable** is declared with an initial domain, called the **reference graph** of G
- The (initial) domain a gd-variable G is the set of all subsets of the reference graph



- We assume here that the gd-variables have the same reference graph
 - Analysis of a single biological network



INGI

43

Overview

- Bioinformatics and Biological networks
- Constraint Programming
- CP(BioNet)
 - The framework
 - **Constraints for bio.net path finding**
 - Constraints for bio.net matching
- Perspectives



INGI

44



Constraints on gd-variables

- $NodeInGraph(n,G)$
- $ArcInGraph(a,G)$
- $SubGraph(S,G)$
- $Path(P,ns,ne,max)$
- $EveryArc(G)$
- $ExistPath(ns,ne,max,G)$
- $Connex(G)$
- $NoCycle(G)$
- $Tree(G)$
- ...



INGI

45



Path finding in CP(BioNet)

- State a constrained path finding problem (CPFP) using:
 - $Path(G,...)$ constraint
 - Additional constraints
- **Example**
 - $Path(G,a,b,\infty) \wedge NodeInGraph(G,c) \wedge NodeInGraph(G,d)$

Note CPFP is an NP-Complete problem



INGI

46



Path finding in CP(BioNet)

More advanced queries : Find TIM-PER pathway

- 3 variables : $G, P1, P2$

$Path(P1, tim, TIM-PER, \infty)$

$\forall r \in RegulationNodes : NotInGraph(P1,r)$

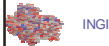
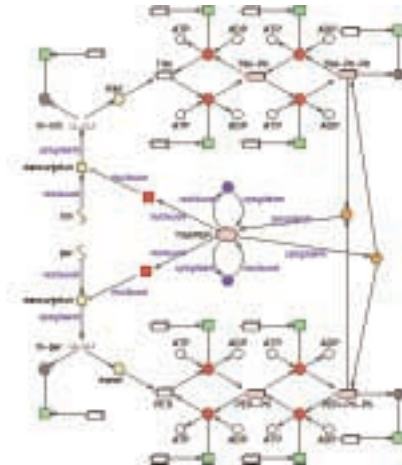
$Path(P2, per, TIM-PER, \infty)$

$\forall r \in RegulationNodes : NotInGraph(P2,r)$

$SubGraph(P1,G)$

$SubGraph(P2,G)$

$AllSubsProdsControls(G)$

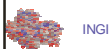


47



Representation of a gd-variable

- A graph domain variable G is represented by
 - its **reference graph** g
 - a **finite set domain variable** N
 - a **finite set domain variable** A
 - The **constraint** $A \subseteq N \times N$
- We denote
 - $N = node(G)$
 - $A = arc(G)$

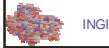


48

Implementation

Choice of a constraint programming environment

- Choice of **Oz**
- Facility to develop new propagators
- Local expertise at UCL
 - Peter Van Roy and his research team

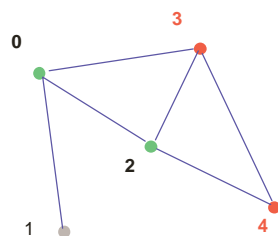


INGI

49

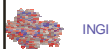
Implementation of gd-variable

- **node(G)** : fs-domain variable
 - Vector of n Boolean domain variables
 - State the presence/absence of the node in G



node(G)

0	1	2	3	4
0-1	0	0-1	1	1

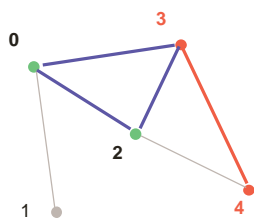


INGI

50

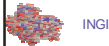
Implementation of gd-variable

- **arc(G)** : fs-domain variable
 - Adjacency matrix of n^2 Boolean domain variables
 - State the presence/absence of the arcs in G



arc(G)

	0	1	2	3	4
0	0	0	0-1	0-1	0
1		0	0	0	0
2			0	0-1	0-1
3				0	1
4					0

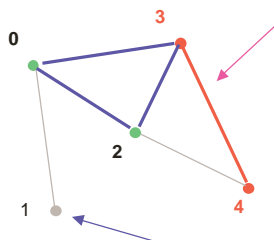


51

Implementation of gd-variable

Internal constraint : $arc(G) \subseteq node(G) \times node(G)$

- Represented by n^2 propagators
 $arc(G)_{ij} \Rightarrow node(G)_i \wedge node(G)_j$

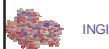


node(G)

	0	1	2	3	4
0	0-1	0	0-1	1	1

arc(G)

	0	1	2	3	4
0	0	0	0-1	0-1	0
1		0	0	0	0
2			0	0-1	0-1
3				0	1
4					0



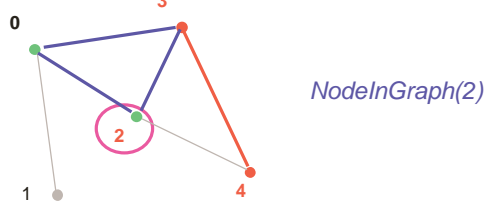
52

NodeInGraph(n,G)

- G : gd-variable
- n : node variable
- Constraint : $n \in \text{node}(G)$

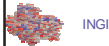
Implementation

- $\text{node}(G)_n$



node(G)				
0	1	2	3	4
0-1	0	1	1	1

Basic constraints that can be negated



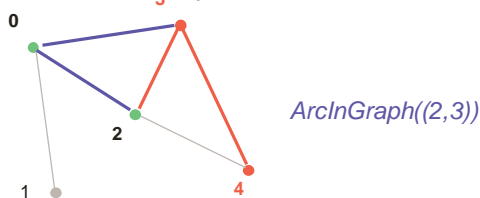
53

ArcInGraph(a,G)

- G : gd-variable
- a : arc variable
- Constraint : $a \in \text{arc}(G)$

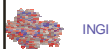
Implementation

- $\text{arc}(G)_{ij}$ with $a = (i,j)$



arc(G)					
	0	1	2	3	4
0	0	0	0-1	0-1	0
1		0	0	0	0
2			0	1	0-1
3				0	1
4					0

Basic constraints that can be negated



54

SubGraph(S,G)

- S, G : gd-variables
- **Constraint** : S is a subgraph of G
 $node(S) \subseteq node(G)$ and $arc(S) \subseteq arc(G)$

Implementation

- $n^2 + n$ propagators
 $arc(S)_{ij} \Rightarrow arc(G)_{ij}$
 $node(G)_i \Rightarrow node(G)_j$



INGI

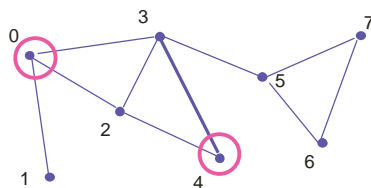
55

Path(P,ns,ne,max)

- P : gd-variable
- ns, ne : node variable
- max : integer
- **Constraint** : P is a path from ns to ne , length $\leq max$

$$ns=n_0 \wedge ne=n_k \wedge node(P) = \{n_0, \dots, n_k\} \wedge k \leq max$$

$$\wedge arc(P) = \{ (n_i, n_{i+1}) \mid 0 \leq i < k \}$$



Path(P,0,4,3)



INGI

56

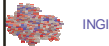
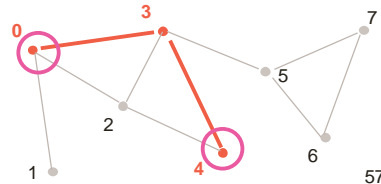
Path propagator

- ns and ne must be in the path P
 $NodeInGraph(ns,P) \wedge NodeInGraph(ne,P)$
- $\#nodes \leq max + 1$
- Degree of the nodes in P (i.e. number of neighbors)
 - $degree(ne)=degree(ns)=1$
 - Other nodes : $degree(n)=2$
- **Implemented** by n propagators

$$node(P)_i \Leftrightarrow \sum arc(P)_{ij} = 2 \quad (ne \neq i \neq ns)$$

$$\sum arc(P)_{nsj} = 1$$

$$\sum arc(P)_{nej} = 1$$

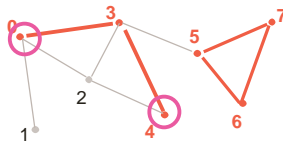


INGI

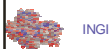
57

Path propagator

- nodes in P must be constrained to be a single connected component



- **Implemented** by a stateful propagator
 - Prune the nodes in the other connected components
 - Search of connected components
 - Standard breadth-first, depth-limited (max) search, starting from ns

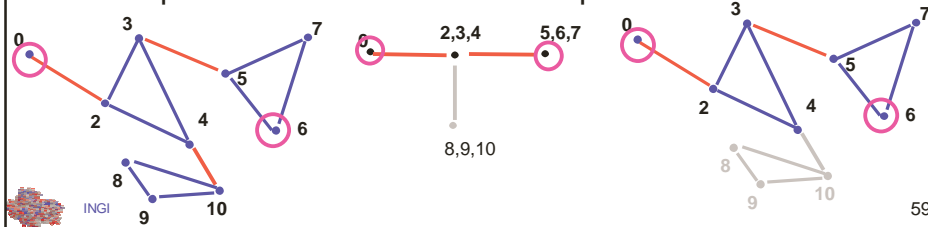


INGI

58

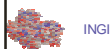
Path propagator

- Search of connected components
 - Standard breadth-first, depth-limited (*max*) search, starting from *ns*
- Propagate information about bridges and articulation nodes
 - Bridge: arc which removal breaks the connected component into two unconnected parts.



Complexity of Path propagator

- Sum constraints
 - At most $|Nref|$ constraints
 - Amortized complexity of one constraints $O(|Aref|)$
 - Hence complexity $O(|Nref| \cdot |Aref|)$
- Connected components
 - Construction of *ConGraph* : $O(|Aref|)$
 - Search of the connected components : $O(|Aref|)$
 - Constraint executed when an arc is removed from *P* : $O(|Aref|)$
 - Hence complexity $O(|Aref|^2)$
- Hence a global complexity of $O(|Aref|^2)$ for the constraint
- Could be reduced to $O(|Aref| \cdot \log(|Nref|)^2)$
 - Dynamic graph connectivity algorithms [Holn & al., 1998]





Experimental results

Objectives

- To show the feasibility of the approach CP(BioNet)
- Testing different implementations
 - Directed vs undirected graphs
 - gd-variable implemented by
 - matrix of boolean variables
 - List/record of boolean variables
 - set variables

Overview


- The experimental data
- Path finding
- Combined constraints

61



The experimental data

- Graphs (50, 100, 200, and 500 nodes) extracted from metabolic network (KEGG)
 - 4.492 chemical entities and 5.281 reactions
 - one connected component
 - average degree of nodes = 4
 - maximum degree = 18 % number of nodes
- Running time measures of constrained path finding , results plotted according to 2 criteria :
 - size of graphs : **Sub-exponential growth**
 - number of internal mandatory nodes. **Independent on average** (not for standard dev.)

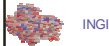
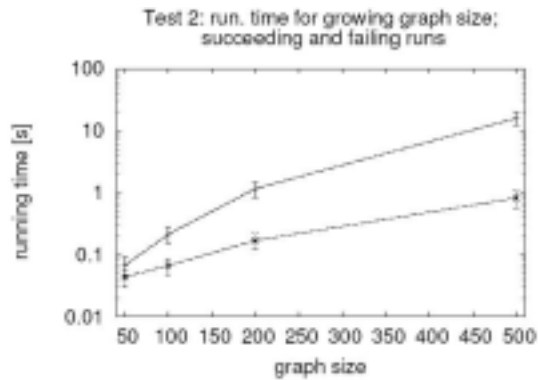


INGI

62

Size of the graph

- Sub-Exponential Growth of running time with respect to the size of the graph

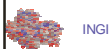
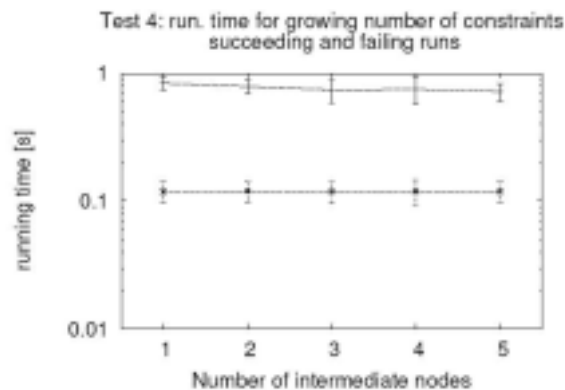


INGI

63

Additional constraints

- $Path(G,a,b,\infty) \wedge NodeInGraph(G,c1) \wedge \dots \wedge NodeInGraph(G,cn)$
- Constant average running time with respect to the number of mandatory internal nodes



INGI

64



Exploiting bio.net properties

- The benchmarks did not exploit the biological properties of the networks
 - Types of the nodes
 - Structure of the relations
- These properties could reduce the search space drastically



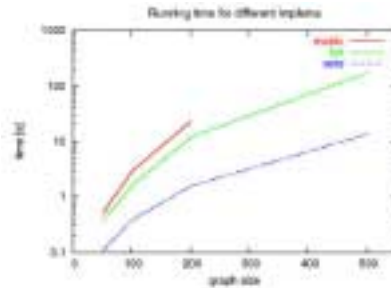
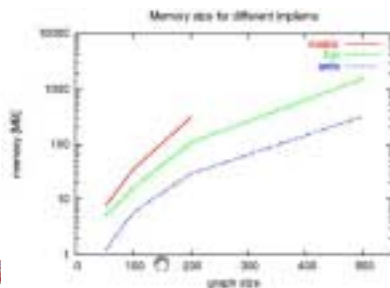
INGI

65



Comparing implementations

- gd-variable implemented by
 - matrix of boolean variables
 - List/record of boolean variables
 - set variables
- **increasing graph size** : random start/end, 2 random internal nodes

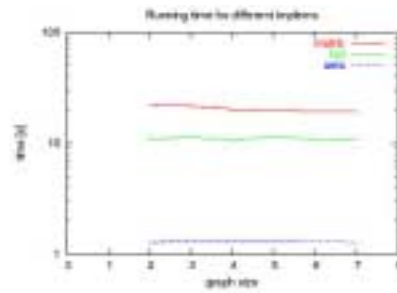
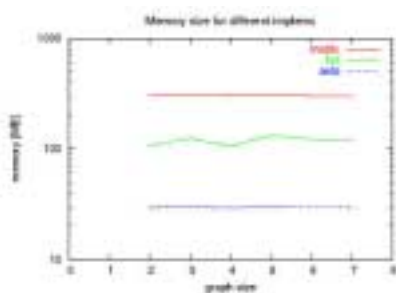


66



Comparing implementations

- gd-variable implemented by
 - matrix of boolean variables
 - List/record of boolean variables
 - set variables
- **Increasing number of internal nodes** : random path

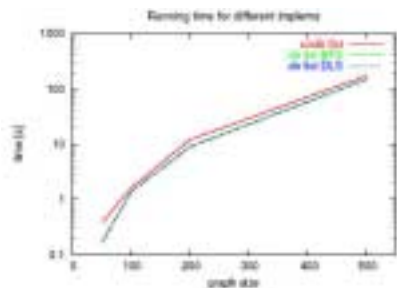
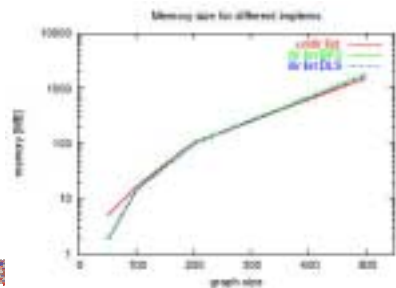


67



Comparing implementations

- Undirected vs graph
 - List representation of gd-variable
 - BFS & DLS search in propagator
 - Adapted propagator for directed graph
 - **Increasing graph size** : random start/end

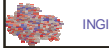


68



Ongoing work

- Integration of the tools within the AMAZE environment
 - Eclipse Plugin
 - Integration with visualization tools
- Optimization
 - Weights on nodes and arcs, best path (B&B)
- Biological applications
- New constraints for biological applications



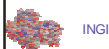
INGI

69



Overview

- Bioinformatics and Biological networks
- Constraint Programming
- CP(BioNet)
 - The framework
 - Constraints for bio.net path finding
 - **Constraints for bio.net matching**
- Perspectives

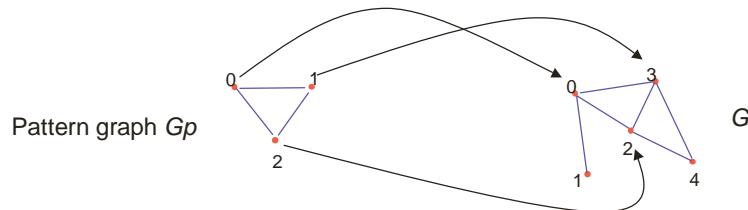


INGI

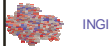
70

Subgraph isomorphism

- $G_p = (N_p, A_p)$: **pattern graph**
- $G = (N, A)$ with $|N_p| \leq |N|$
- Find a **function** $f: N_p \rightarrow N$ such that
 - f is injective
 - $\forall n_1, n_2 : (n_1, n_2) \in A_p \Rightarrow (f(n_1), f(n_2)) \in A$



- Subgraph isomorphism is NP-complete

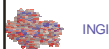


INGI

73

Family of problems

- Graph vs subgraph isomorphism
- Exact vs inexact matching
 - Distance from the initial pattern and the chosen subpattern
 - Nodes with attributes, matching between two node is a *distance*
 - Some nodes/arcs from the pattern graph are optional.



INGI

74

Existing algorithms

Many existing algorithms based on various techniques

- Cliques
- Fuzzy set theory
- Elastic graph matching
- Multiple graph matching
- Error correction
- Genetic algorithms
- Decision tree
- Neural networks
- Clustering
- Connected components
- **Constraint programming** [Rudolf, 1998] [Valiente, 2000]
- ...

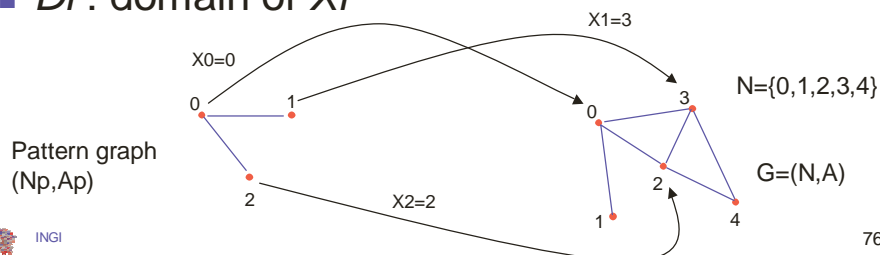


INGI

75

SI as a CSP

- $G_p = (N_p, A_p)$: **pattern graph**
- $G = (N, A)$ with $n = |N_p| \leq |N| = d$
- We use $N_p = (X_1, \dots, X_n)$
- **Domain variables** $X_i \in N$
- D_i : domain of X_i



INGI

76



The problem

- Design a constraint for subgraph isomorphism (pattern matching)
- Design and implement a propagator for this constraint
- Existing approach
 - [Rudolf, 1998], [Larrosa & Valiente, 2000]
- Extensions
 - Exploiting the structure of biochemical network (type and attribute information)
 - Approximate matching problem



INGI

77



Results

Exploiting type information

- Extension/enhancement of propagator
- Outperforms a classical benchmark [Larrosa & Valiente, 2000] by 27%

Approximate matching problem

- Some nodes/arcs of the pattern graph may not be assigned to nodes/arc in the target graph
- Extension of the propagator to optional nodes/arcs
- Ongoing work



INGI

78



Overview

- Bioinformatics and Biological networks
- Constraint Programming
- CP(BioNet)
 - The framework
 - Constraints for bio.net path finding
 - Constraints for bio.net matching
- **Perspectives**



INGI

79



CP(BioNet)

- Design of a new computation domain : **graph**
- A framework for the analysis of biochemical networks
- The combination of constraints allows for the expression of sophisticated and varied analysis
- Simplicity and versatility of the analysis : no code to develop for new analysis
- A prototype shows the feasibility of the approach



INGI

80



Perspectives

- What next ?
 - Design and implementation of a full CP(BioNet)
 - Integration of constraints and pattern matching
 - Constraints on the pattern
 - Generic arc in pattern representing a path
 - Pattern as a constraint pattern variable
 - Experimentations in collaboration with biologists
- The preliminary results show the potential of this CP approach



INGI

81



References

<http://www.info.ucl.ac.be/people/YDE/yde.html>

- *An Overview of Data Models for the Analysis of Biochemical Pathways*
Yves Deville, Jacques van Helden, Soshana Wodak David Gilbert
Briefings in Bioinformatics, August 2003
- *An Object-Oriented Data Model for Signal Transduction*
Yves Deville, David Gilbert, Christian Lemer, Jacques van Helden, Shoshana J. Wodak. *ECCB2003*.
- *The aMAZE LightBench: a Web interface to a relational database of cellular processes*
Christian Lemer, Erick Antezana, Fabian Couche, Frédéric Fays, Xavier Santolaria, Rekins's Janky, Yves Deville, Jean Richelle, Shoshana J. Wodak. *Nucl. Acid Research*, 2004
- *A Mozart implementation of CP(BioNet)*. G. Dooms, Y. Deville, P. Dupont (2004), "In Proceedings of the second International Mozart/Oz Conference, October 2004.
- *Recherche de chemins contraints dans les réseaux biochimiques* G. Dooms, Y. Deville, P. Dupont
Treizèmes Journées Francophones de Programmation en Logique et de programmation par Contraintes, JFPLC 2004, June 2004.
- *Finding Patterns in Biochemical Networks : a Constraint Programming Approach*
S. Zampelli, Y. Deville, P. Dupont. In *Proceedings of 5èmes Journées Ouvertes Biologie Informatique Mathématiques, JOBIM 2004, June 2004.*
- *Constrained path finding in biochemical networks* G. Dooms, Y. Deville, P. Dupont. In *Proceedings of 5èmes Journées Ouvertes Biologie Informatique Mathématiques, JOBIM 2004, June 2004.*
- *Towards a CLP framework for the analysis of Biochemical Networks* Y. Deville. *Invited talk SweConsNet 2004, Swedish Constraint Network workshop, Linköping 15 January 2004.*



INGI

82

Acknowledgments

aMAZE team

ULB - Belgium

- Erick Antezana
- Fabian Couche
- Fred Fays
- **Olivier Sand**
- **Christian Lemer**
- **Jean Richelle**
- **Jacques van Helden**
- **Soshana Wodak**

(*)*Yves Deville (on sabbatical)*

BioMaze Project

UCL - Belgium

- **Yves Deville**
- **Pierre Dupont**
- **Stéphane Zampelli**
- **Grégoire Doods**
- **Sébastien Vast**
- **Jonathan Fallon**

ULB - Belgium

- Esteban Zimani
- Sabri

FUNDP - Namur

- Jean-Luc Hainaut
- Jean-Marc Hick

Collaborators

Institut Pasteur - France

- Georges Cohen

Birkbeck College - UK

- Lorenz Wernisch

University of Glasgow - UK

- **David Gilbert**

Univ. Köln - Germany

- Dietmar Schomburg

EBI-EMBL

- Sandra Orchard

External data sources

- Swissprot
- Genbank
- KEGG/LIGAND
- BRENDA
- PUBMED

Sponsors

- Astra-Zeneca
- Aventis, Organon
- Roche, (Monsanto)
- EC
- **Brussels Gov.**
- **Walloon Region**