

Logical and Relational Learning Revisited

Luc De Raedt

luc.deraedt@cs.kuleuven.be

ICML 2008



What is Logical and Relational Learning ?

Inductive Logic Programming

(Statistical) Relational Learning

UNION of

Mining and Learning in Graphs

Multi-Relational Data Mining

They all study the same problem

The Problem

Learning from structured data, involving

- objects, and
- relationships amongst them

and possibly

- using background knowledge

Purpose of this talk

- Relational learning is sometimes viewed as a new problem, but it has a long history
- Emphasize the role of symbolic representations (graphs & logic) and knowledge
- A modern view
 - **logic as a toolbox for machine learning**
- Overview of some of the available tools and techniques
- Illustration of their use in some of our recent work

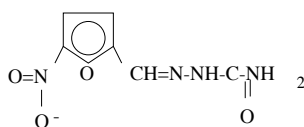
Overview

- A. Motivation for logical and relational learning
- B. Logical and relational learning
- C. A statistical relational learning approach
- D. Challenge

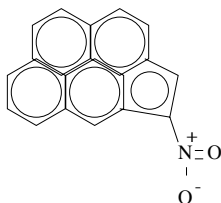
A. Motivation for logical and relational learning

Case I: Structure Activity Relationship Prediction

Active

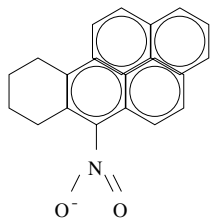


nitrofurazone

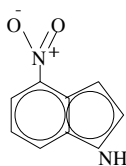


4-nitropenta[cd]pyrene

Inactive



6-nitro-7,8,9,10-tetrahydrobenzo[a]pyrene



4-nitroindole

[Srinivasan et al. AIJ 96]

Structural alert:



General Purpose
Logic Learning System

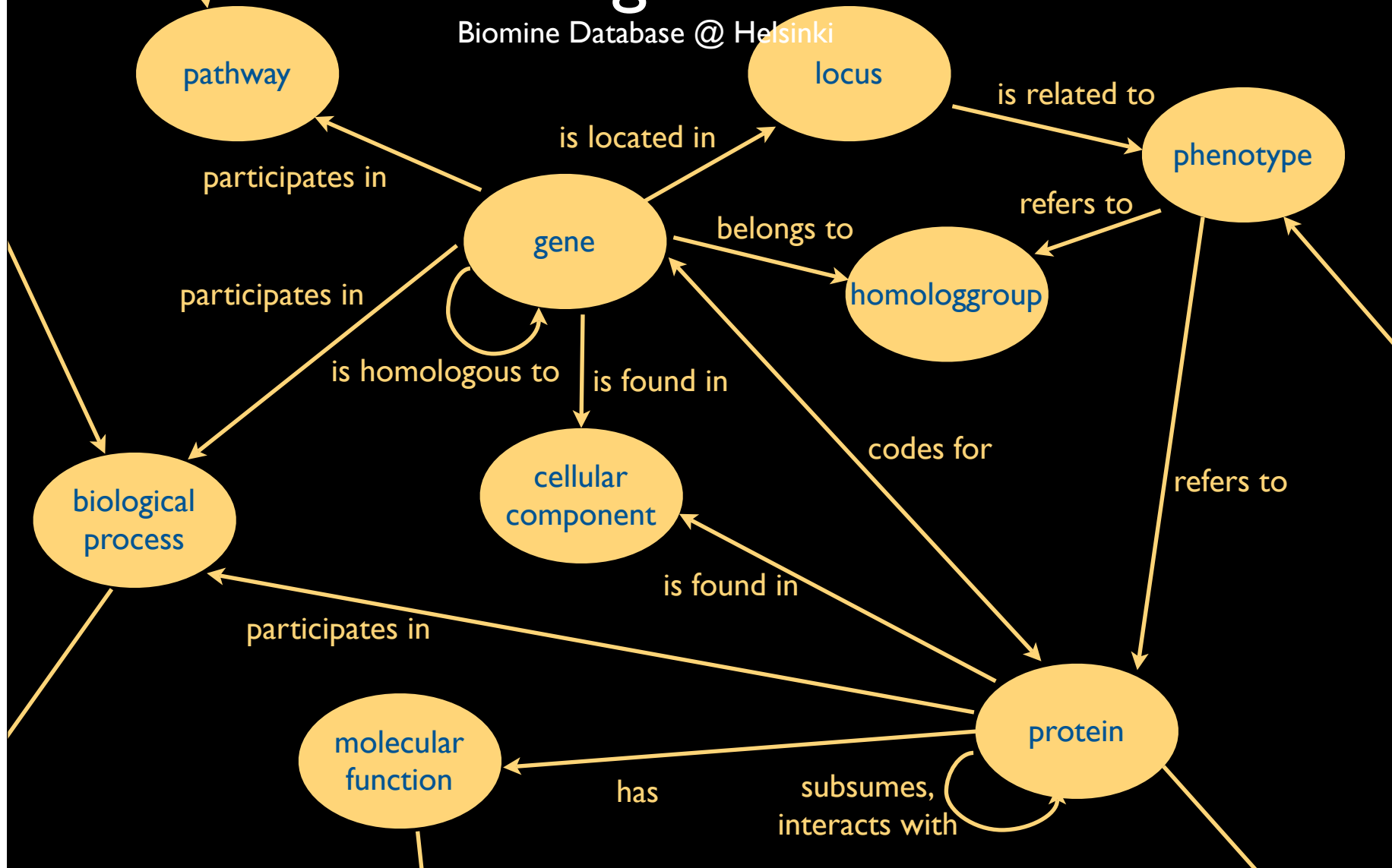
Uses and Produces
Knowledge

Data = Set of Small Graphs

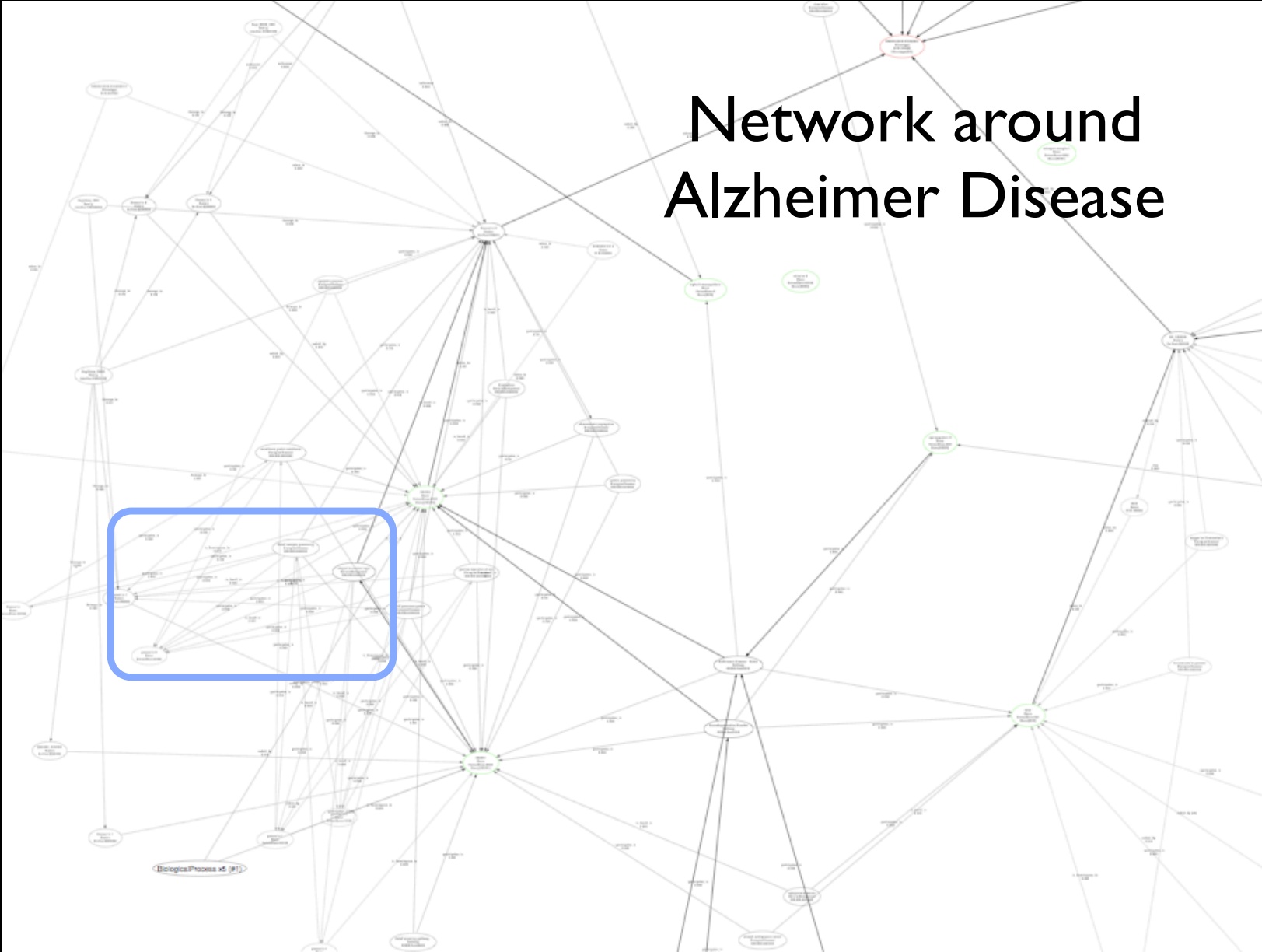
Data = Large (Probabilistic) Network

Case 2: Biological Networks

Biomine Database @ Helsinki

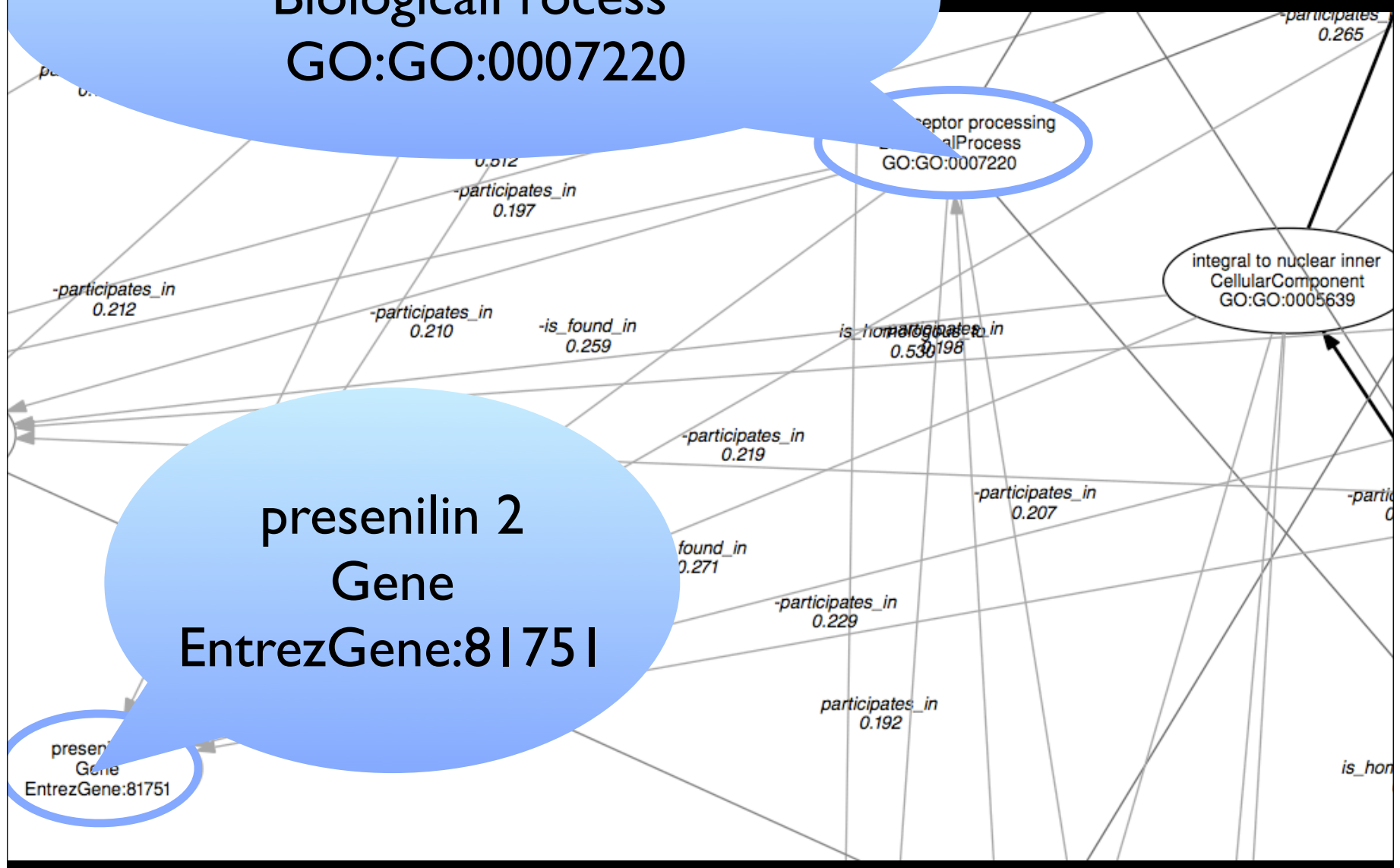


Network around Alzheimer Disease



Notch receptor processing
BiologicalProcess
GO:GO:0007220

presenilin 2
Gene
EntrezGene:81751



Questions to ask

How to support the life scientist in using and discovering new knowledge in the network ?

- What is the probability that gene X is connected to disease Y ?
- Which genes are similar to X w.r.t. disease Y ?
- Which part of the network provides the most information (network extraction) ?
- ...

B. Logical and Relational Learning

B. Logical and Relational Learning

BASIC SETTINGS

Typical Machine Learning Problem

Given

- a set of examples **E**
- a background theory **B**
- a logic language **L_e** to represent examples
- a logic language **L_h** to represent hypotheses
- a **covers** relation on **L_e × L_h**
- a loss function

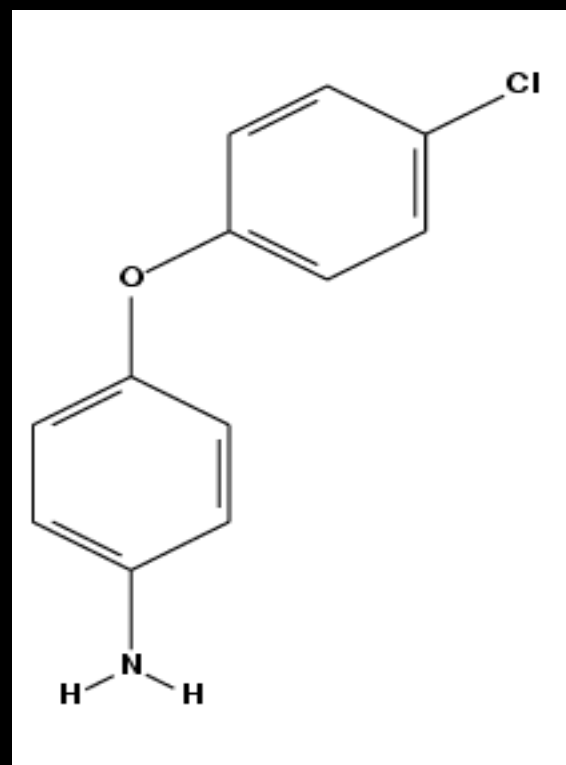
Find

- A hypothesis **h** in **L_h** that minimizes the loss function w.r.t. the examples **E** taking **B** into account

Components

1. Represent the data
and upgrading & downgrading
2. Represent the hypotheses
3. The covers relation
4. The generality relation

Encoding Graphs



Encoding Graphs

atom(1,cl).

atom(2,c).

atom(3,c).

atom(4,c).

atom(5,c).

atom(6,c).

atom(7,c).

atom(8,o).

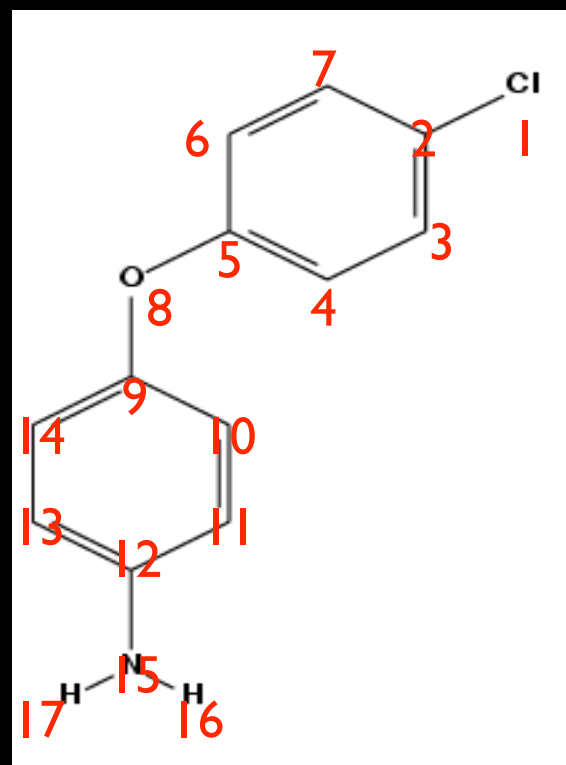
...

bond(3,4,s).

bond(1,2,s).

bond(2,3,d).

...



Encoding Graphs

atom(1,cl,21,0.297)

atom(2,c,21,0.187)

atom(3,c,21,-0.143)

atom(4,c,21,-0.143)

atom(5,c,21,-0.143)

atom(6,c,21,-0.143)

atom(7,c,21,-0.143)

atom(8,o,52,0.98)

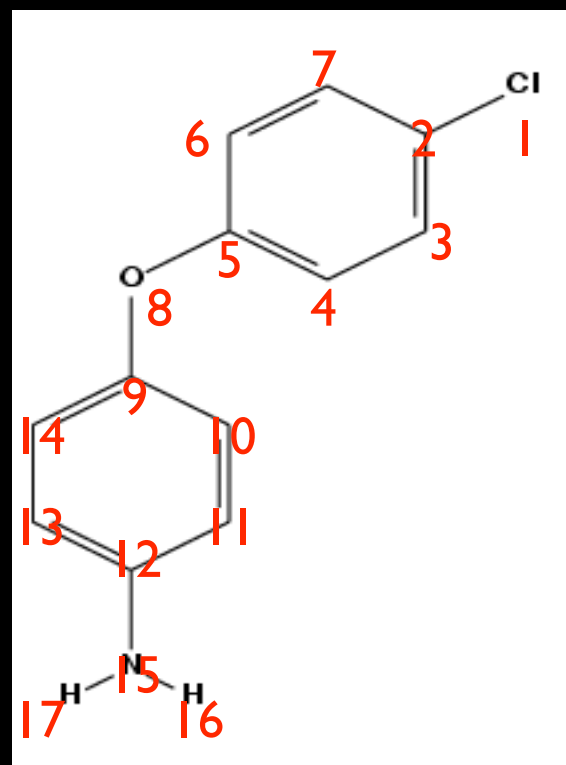
...

bond(3,4,s).

bond(1,2,s).

bond(2,3,d).

...



Encoding Knowledge

Use background knowledge in form of rules

- encode hierarchies

halogen(A):- atom(X,f)

halogen(A):- atom(X,cl)

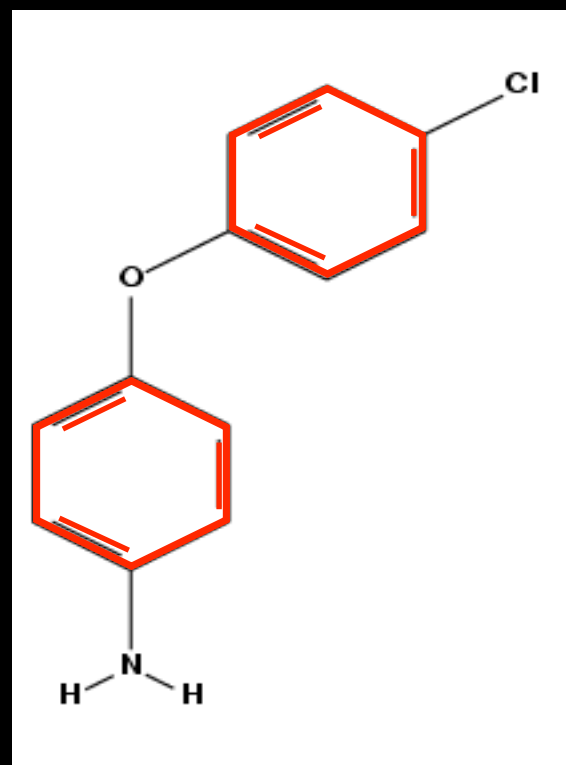
halogen(A):- atom(X,br)

halogen(A):- atom(X,i)

halogen(A):- atom(X,as)

- encode functional group

benzene-ring :- ...



2. The Hypothesis Language

Prolog

OWL

First Order Logic

Graphs

SQL

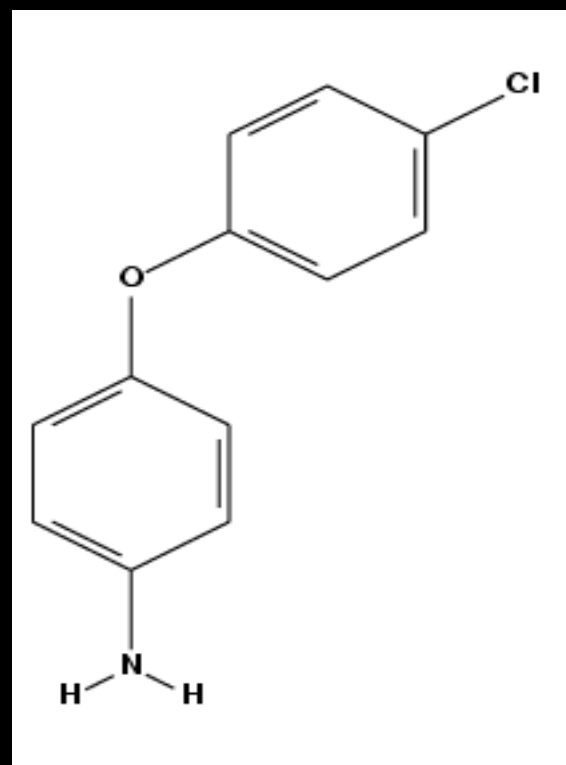
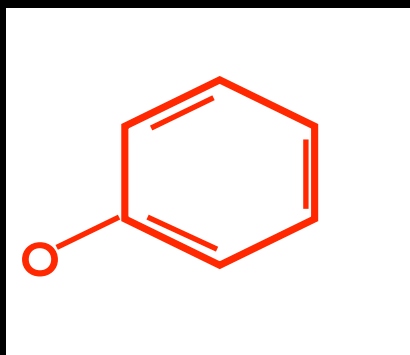
Description Logic

Relational Calculi

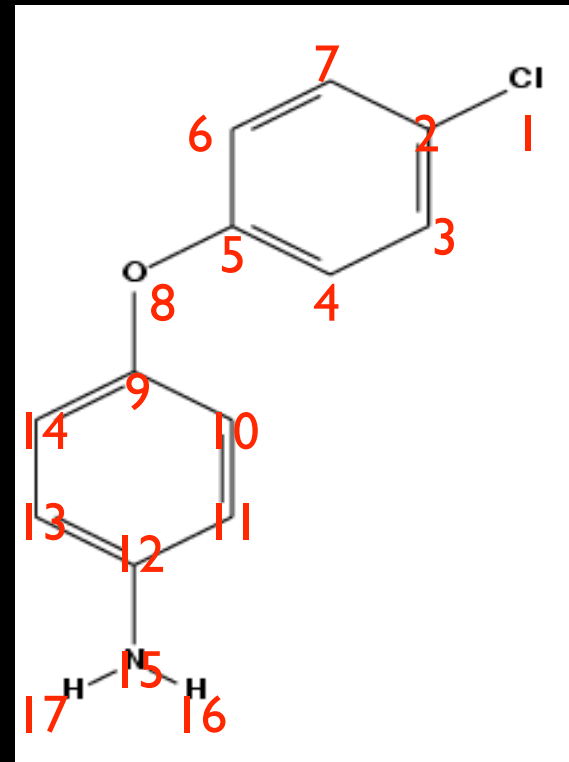
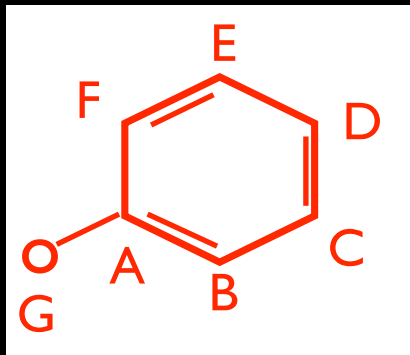
Entity-Relationship Model

Choice probably not that important
though implementation & manipulation

3. Covers Relation



Covers Relation



Subgraph Isomorphism
(bijection)
or
Homomorphism
(injection)

Coverage

positive :- atom(A,c),
 atom(B,c),
 bond(A,B,s),

OI-subsumption
(bijection)
or
theta-subsumption
(injection)

atom(1,c).	
atom(2,c).	
atom(3,c).	
atom(4,c).	
atom(5,c).	
atom(6,c).	bond(3,4,s).
atom(7,c).	bond(1,2,s).
atom(8,o).	bond(2,3,d).
...	...

Coverage

positive :- halogen(A),
 halogen(B),
 bond(A,B,s),

....

halogen(A):- atom(X,f)
halogen(A):- atom(X,cl)
halogen(A):- atom(X,br)
halogen(A):- atom(X,i)
halogen(A):- atom(X,as)

Deduction

atom(1,cl).

atom(2,c).

atom(3,c).

atom(4,c).

atom(5,c).

atom(6,c).

atom(7,c).

atom(8,o).

...

bond(3,4,s).

bond(1,2,s).

bond(2,3,d).

...

4. Generality Relation

An essential component of Symbolic Learning systems

G is **more general** than S if all examples covered by S are also covered by G

Using graphs

- subgraph isomorphism or homeomorphism

In logic

- subsumption or $G \models S$

Generality Relation

positive :- atom(X,c) \vDash positive :- atom(X,c), atom(Y,o)

but also

positive :- halogen(X) \vDash positive :- atom(X,c)
halogen(X) :- atom(X,c)

$$G \vDash S$$

S follows *deductively* from G

G follows *inductively* from S

therefore induction is the *inverse* of deduction

this is an operational point of view because there
are many deductive operators \vdash that implement \vDash

take any deductive operator and invert it and one
obtains an inductive operator

Various frameworks for generality

Depending on the form of G and S

single clause

clausal theory

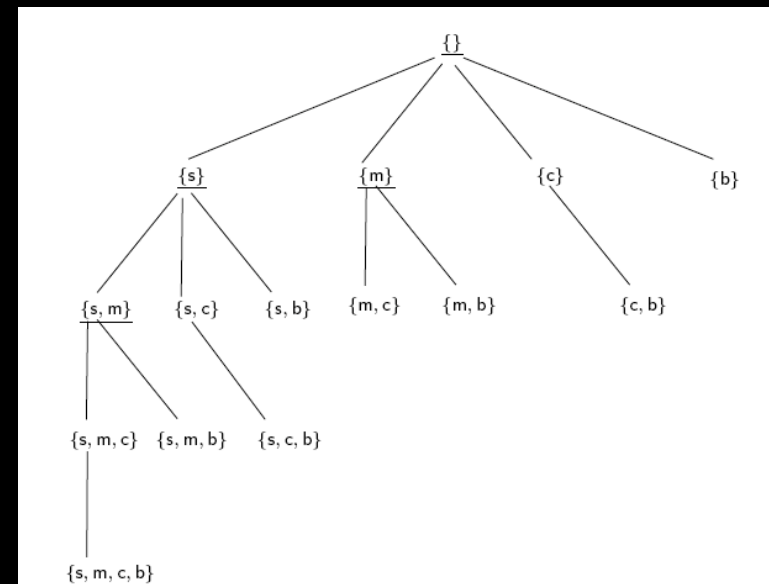
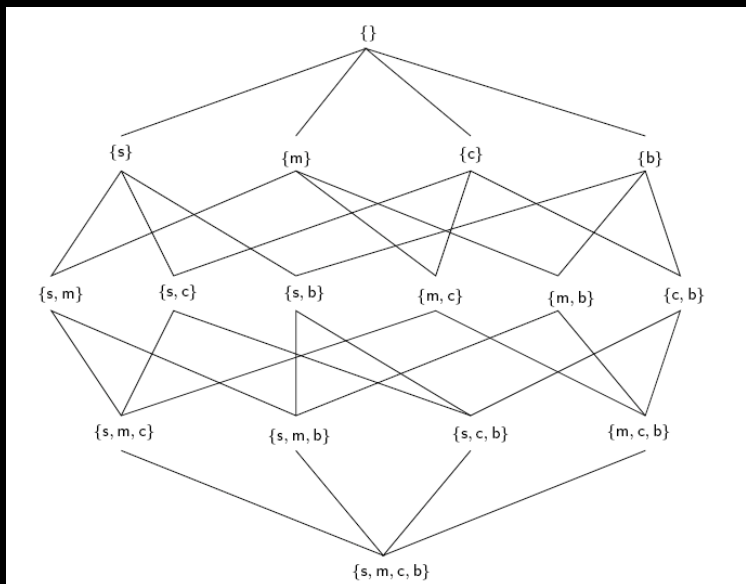
Relative to a background theory $B \cup G \models S$

Depending on the choice of \vdash to invert

subsumption (most popular)

$G \neq S$

Generality relations and refinement operators are well-understood; they apply to simpler structures such as graphs (canonical form -- lexicographic orders)



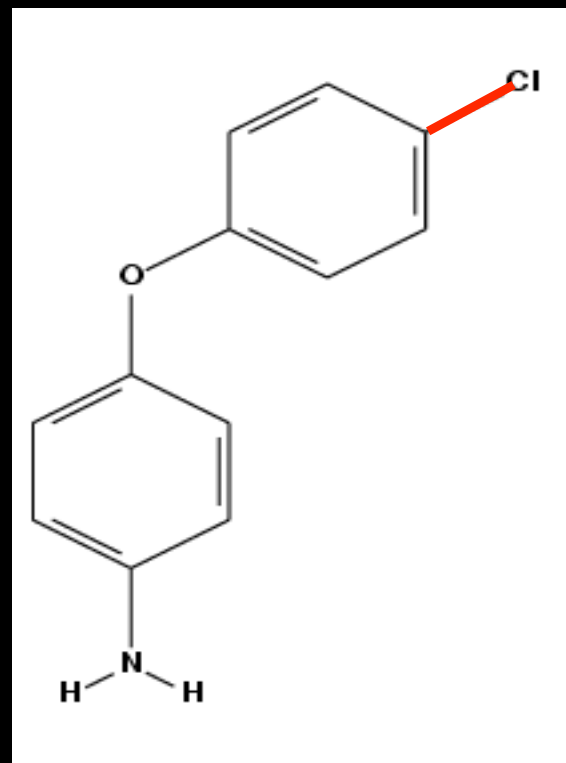
Learning in Graphs

Rule learning

- finds small set of rules (features) that discriminates positive from negative examples (covering algo.)
- interpretable results

Local pattern mining

- find all features that are frequent, or score amongst k best w.r.t. chi-square



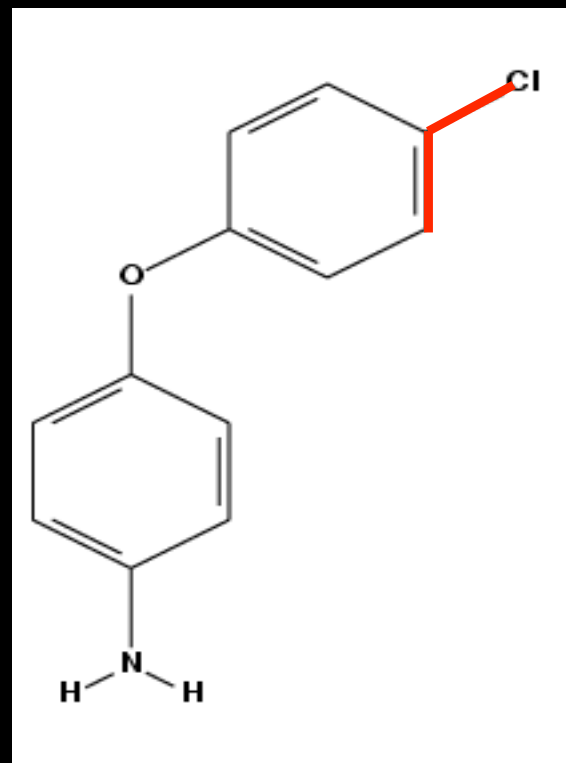
Learning in Graphs

Rule learning

- finds small set of rules (features) that discriminates positive from negative examples (covering algo.)
- interpretable results

Local pattern mining

- find all features that are frequent, or score amongst k best w.r.t. chi-square



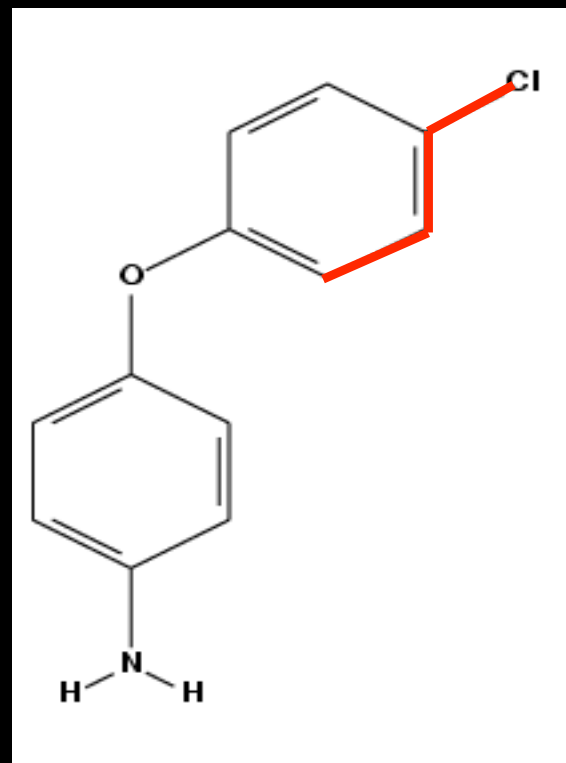
Learning in Graphs

Rule learning

- finds small set of rules (features) that discriminates positive from negative examples (covering algo.)
- interpretable results

Local pattern mining

- find all features that are frequent, or score amongst k best w.r.t. chi-square



B. Logical and Relational Learning

UPGRADING AND DOWNGRADING

Relational versus Graphs

Advantages Relational

- background knowledge in the form of rules, ontologies, features, ...
- relations of arity > 2 (but hypergraphs)
- graphs capture structure but annotations with many features/labels is non-trivial

Advantages Graphs

- efficiency and scalability
- full relational is more complex
- matrix operations

The Upgrading Methodology

Start from existing system for simpler representation

Extend it for use with richer representation
(while trying to keep the original system as a special case)

Illustration in SRL part.

Learning Tasks

- rule-learning & decision trees [Quinlan 90], [Blockeel 96]
- frequent and local pattern mining [Dehaspe 98]
- distance-based learning (clustering & instance-based learning) [Horvath, 01], [Ramon 00]
- probabilistic modeling (cf. statistical relational learning)
- reinforcement learning [Dzeroski et al. 01]
- kernel and support vector methods

Logical and relational representations can (and have been) used for all learning tasks and techniques

From Upgrading to Downgrading

Work at the right level of representation

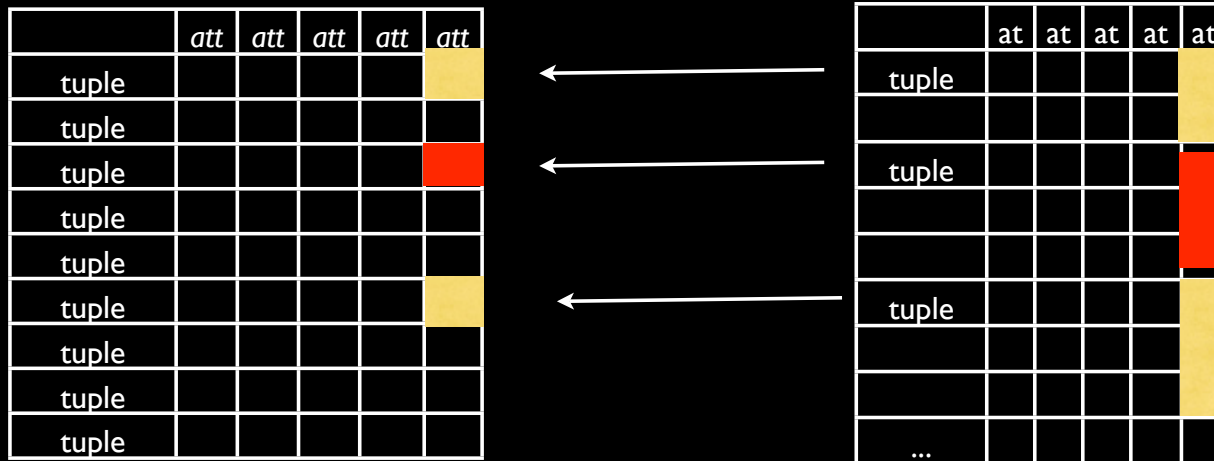
- trade-off between **expressivity & efficiency**

The **old** challenge: **upgrade** learning techniques for simpler representations to richer ones.

The **new** challenge: **downgrade** more expressive ones to simpler ones for efficiency and scalability; e.g. graph miners.

Note: systems using rich representations form a **baseline**, and can be used to test out ideas.

Aggregation



from multi-tuple relations to single-tuple

Propositionalization and Aggregation

Often useful to reduce more expressive representation to simpler one but almost always results in information loss or combinatorial explosion

Shifts the problem

- how to find the right features / attributes

One example

- features = paths in a graph (for instance)
- which ones to select ?

C. Statistical Relational and Logical Learning

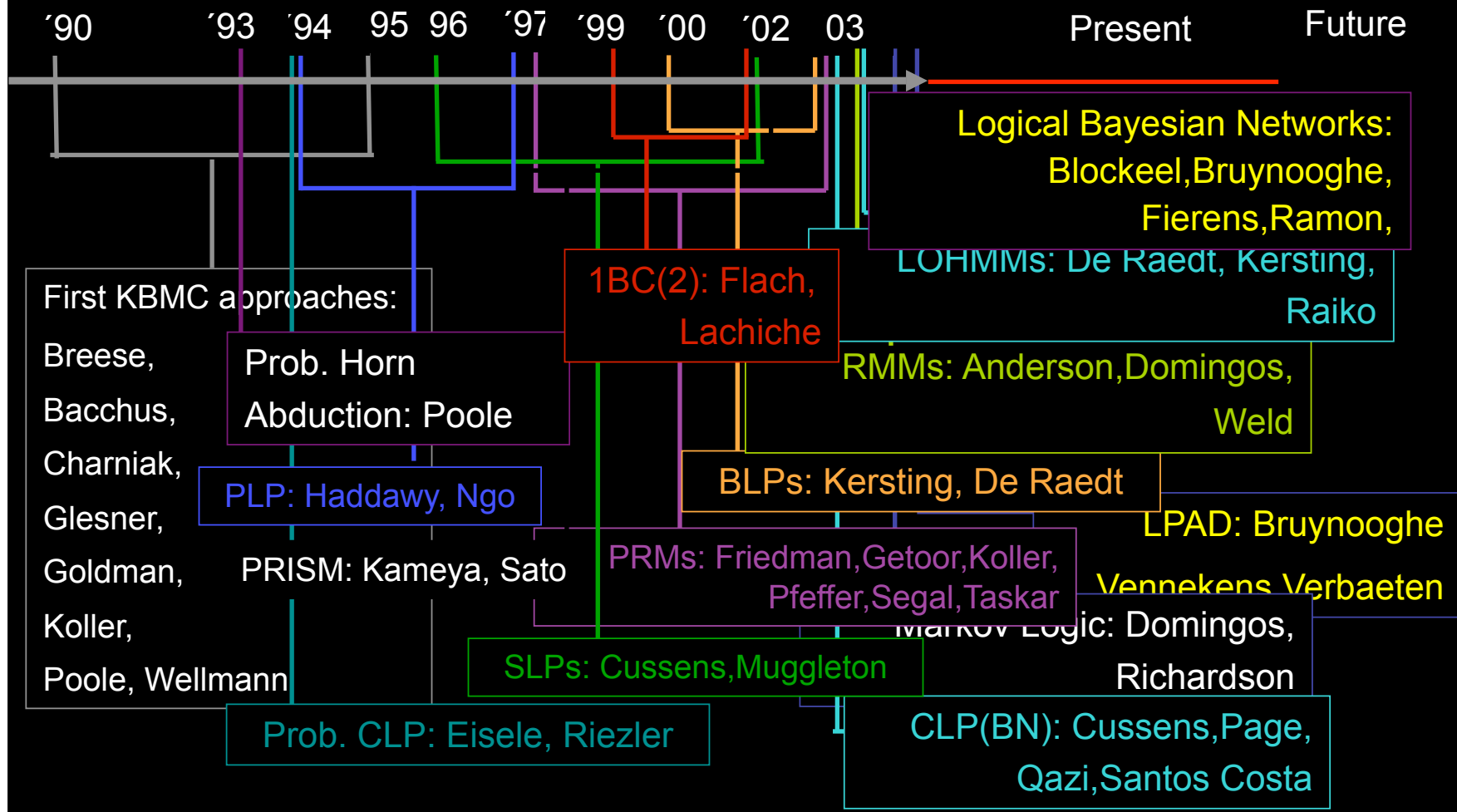
Joint work with Bernd Gutmann, **Angelika Kimmig**,
Kristian Kersting, Niels Landwehr, Vitor Santos Costa,
Ingo Thon, **Hannu Toivonen**, ...

Statistical Relational Learning

Logic and relations alone are often insufficient

- but can be combined with probabilistic reasoning and models
- use logic as a toolbox

Some SRL formalisms

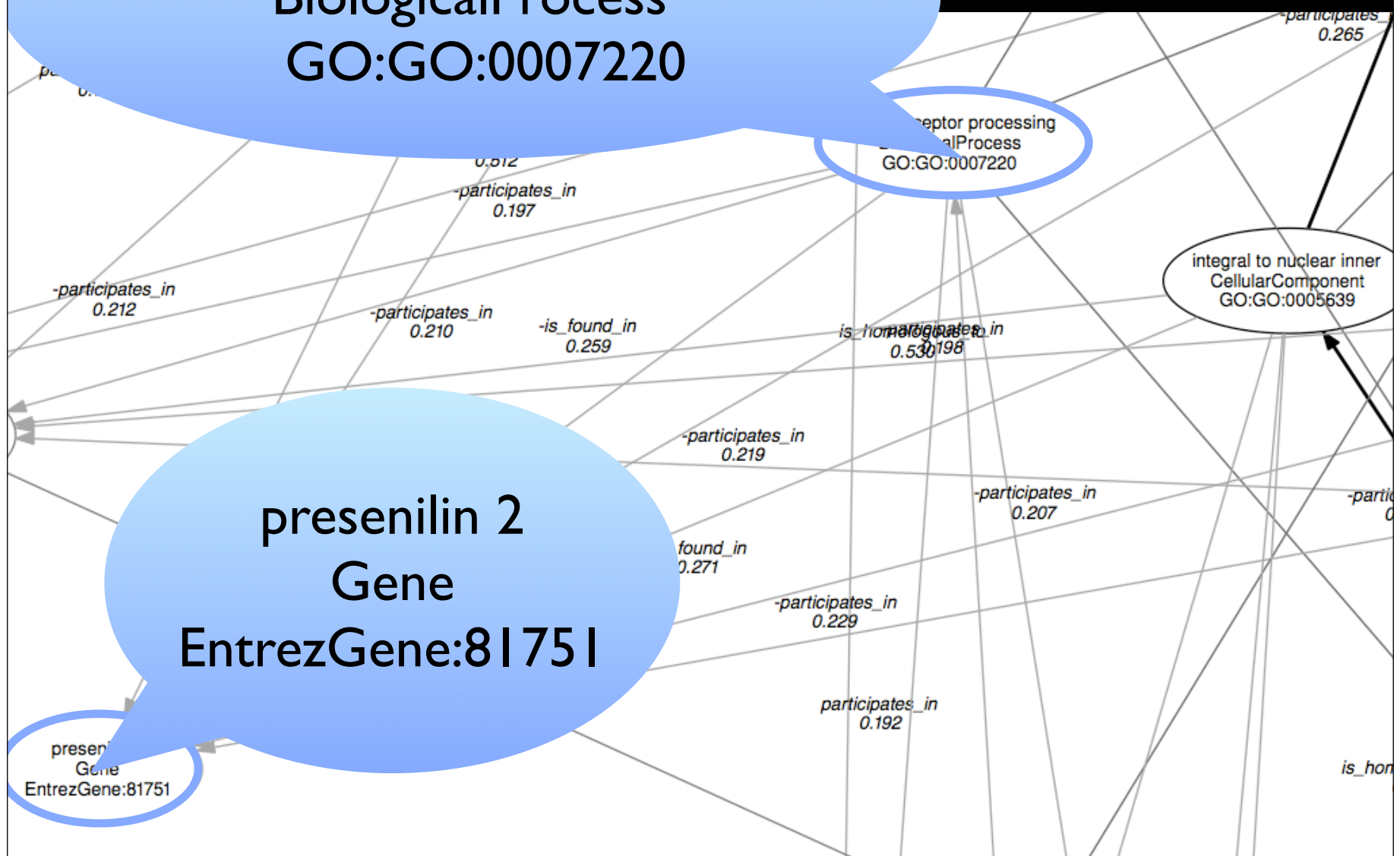


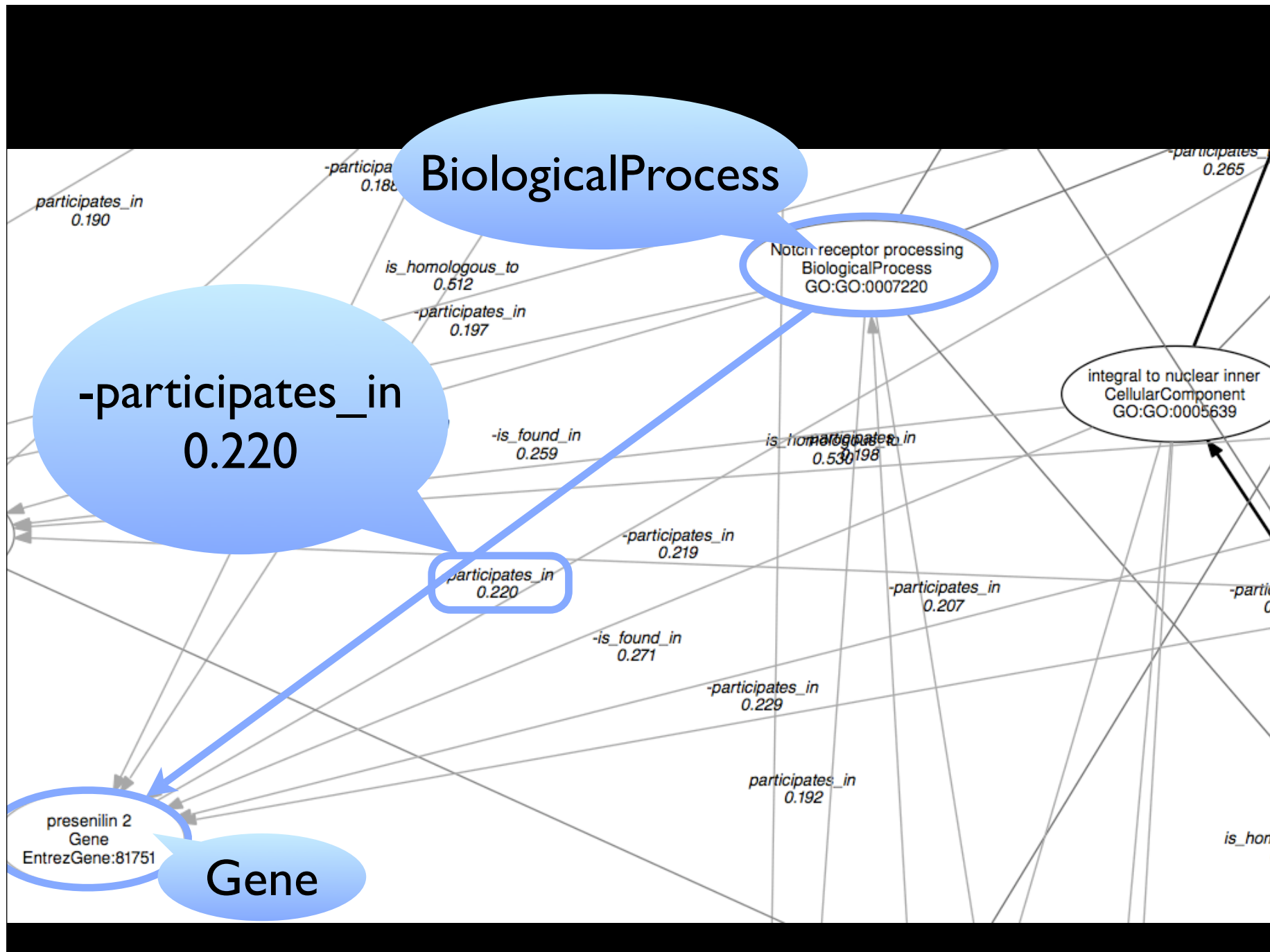
Two key approaches

- Logical Probability Models [MLNs, PRMs, BLPs, ...]
 - Knowledge Based Model Construction, use (clausal) logic as a template
 - generate graphical model on which to perform probabilistic inference and learning
- Probabilistic Logical Models [ICL, PRISM, ProbLog, SLPs, ...]
 - Annotate logic with probabilities
 - perform inference and learning in logic
 - illustrate the idea of **upgrading**

Notch receptor processing
BiologicalProcess
GO:GO:0007220

presenilin 2
Gene
EntrezGene:81751





BiologicalProcess

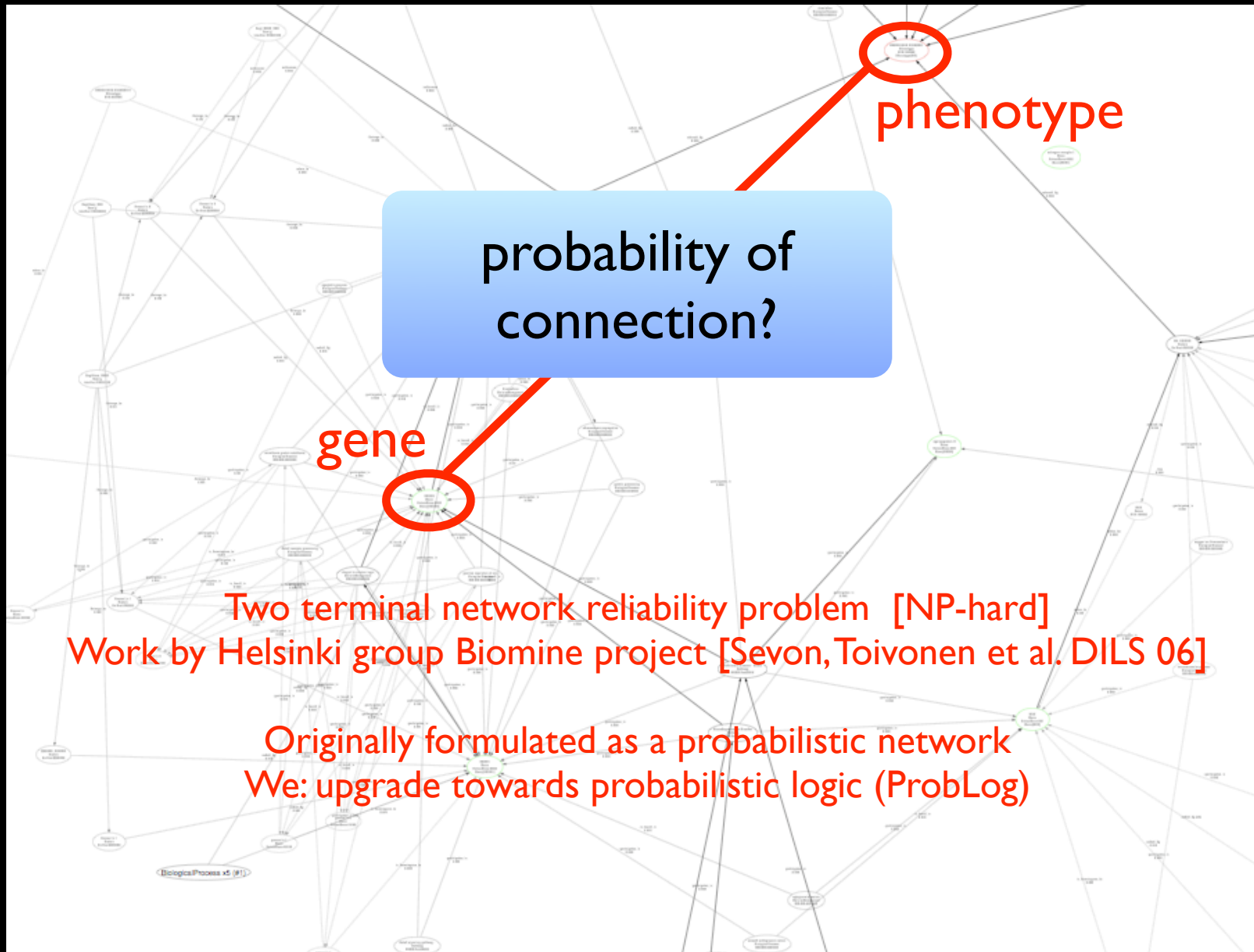
-participates_in
0.220

Gene

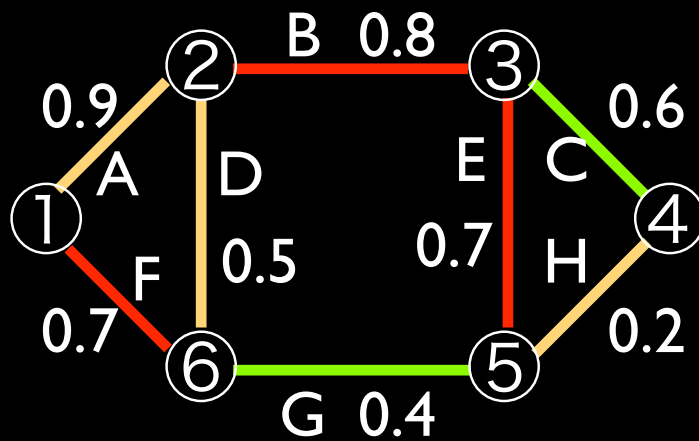
Notch receptor processing
BiologicalProcess
GO:GO:0007220

integral to nuclear inner
CellularComponent
GO:GO:0005639

presenilin 2
Gene
EntrezGene:81751



Example in ProbLog



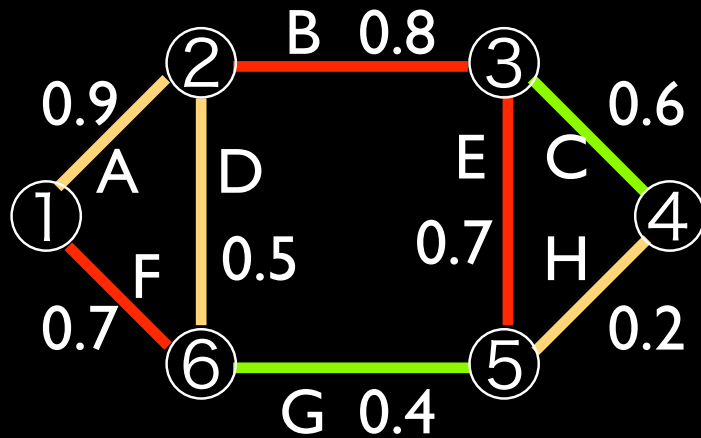
facts mutually independent

logical part L

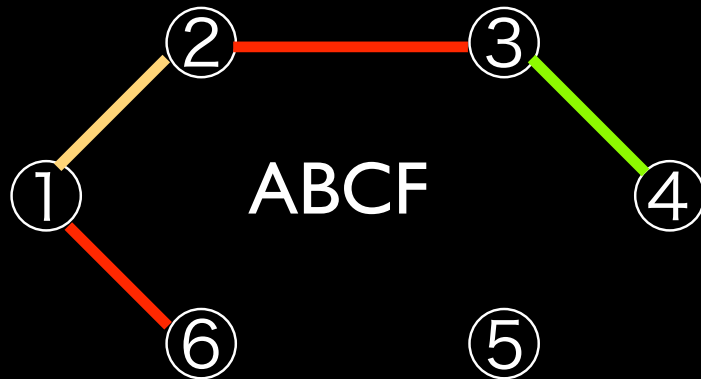
ProbLog theory T

```
0.9 : y_edge(1,2).  
0.8 : r_edge(2,3).  
0.6 : g_edge(3,4).  
...
```

Sampling Subprograms



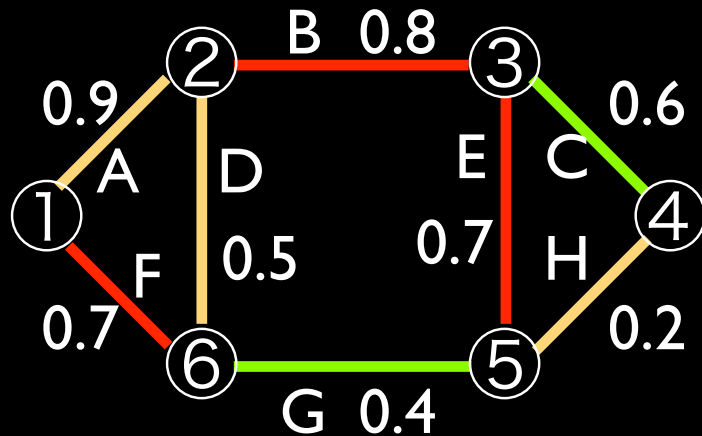
- Biased coins
- Independent



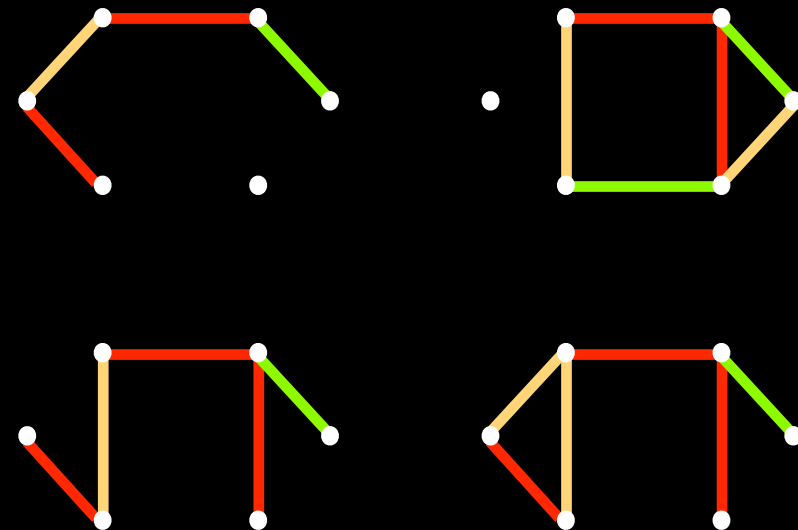
A	B	C	D	E	F	G	H
+	+	+	-	-	+	-	-

$$P = 0.9 \cdot 0.8 \cdot 0.6 \cdot (1 - 0.5) \cdot (1 - 0.7) \cdot 0.7 \cdot (1 - 0.4) \cdot (1 - 0.2)$$

Queries



① → ④

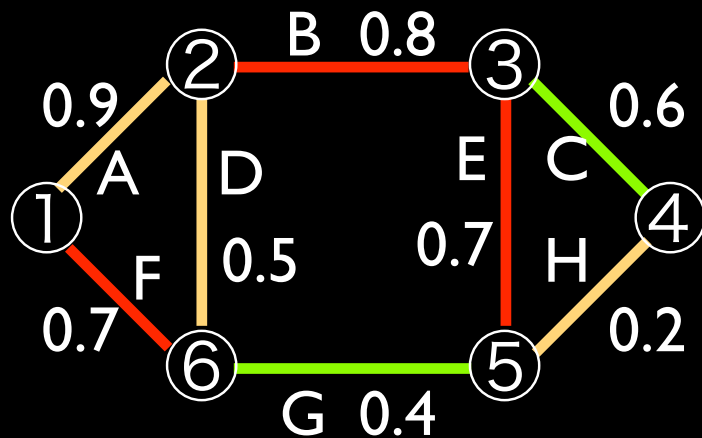


path(x,y) :- edge(x,y)
 path(x,y) :- edge(x,z), path(y,z)

$$P(q|T) = \sum_{S \subseteq L, S \models q} P(S|T)$$

...

Queries



① → ④

path(x,y) :- edge(x,y)
path(x,y) :- edge(x,z), path(y,z)

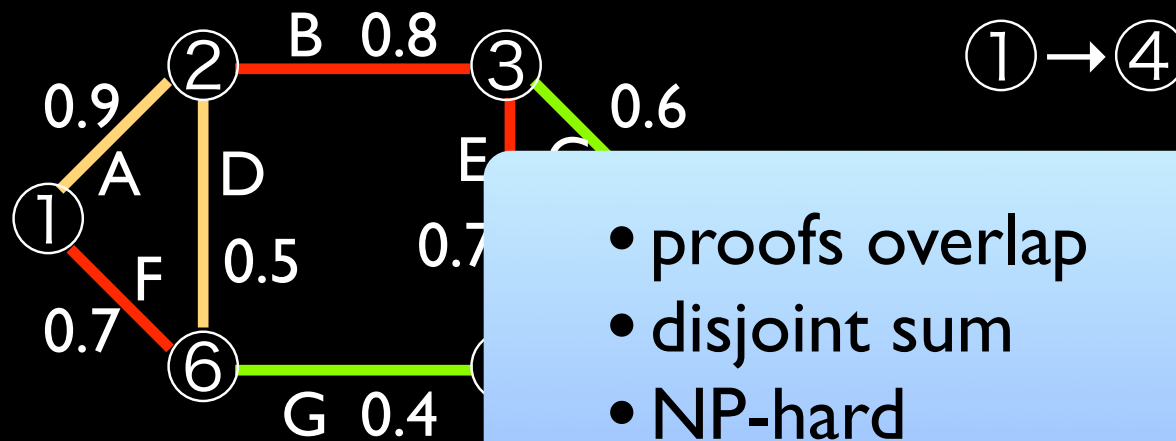
$$P(q|T) = \sum_{S \subseteq L, S \models q} P(S|T)$$

Key Point
of ProbLog and Logic

any relation can be defined

Query Probability

using proofs

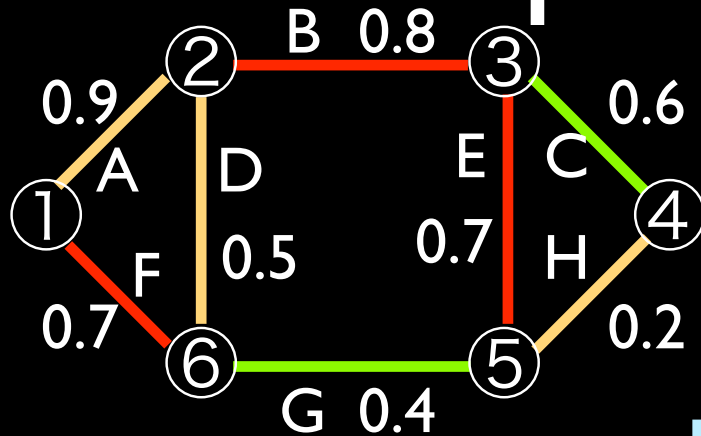


- proofs overlap
- disjoint sum
- NP-hard
- approximation algorithm
[De Raedt et al, IJCAI 07]

$$P(\text{path}(1, 4) | T)$$

$$= P(\underbrace{ABC}_{\text{proof 1}} + \underbrace{ABEH}_{\text{proof 2}} + \dots + \underbrace{FDBEH}_{\text{proof 3}})$$

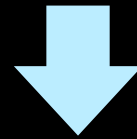
Most likely proof / explanation



example

① → ④

Abduction



ABC

Semantics ProLog

Not really new, rediscovered many times

Intuitively, a probabilistic database

Formally, a distribution semantics [Sato 95]

Other systems, such as Sato's Prism and Poole's ICL avoid the disjoint sum problem

- assume that explanations / proofs are mutually exclusive, that is,
- $P(A \vee B \vee C) = P(A) + P(B) + P(C)$

Long term vision: develop an optimized probabilistic Prolog implementation in which other SRL formalisms can be compiled. (work together with Vitor Santos Costa and Bart Demoen, integration in YAP Prolog planned)

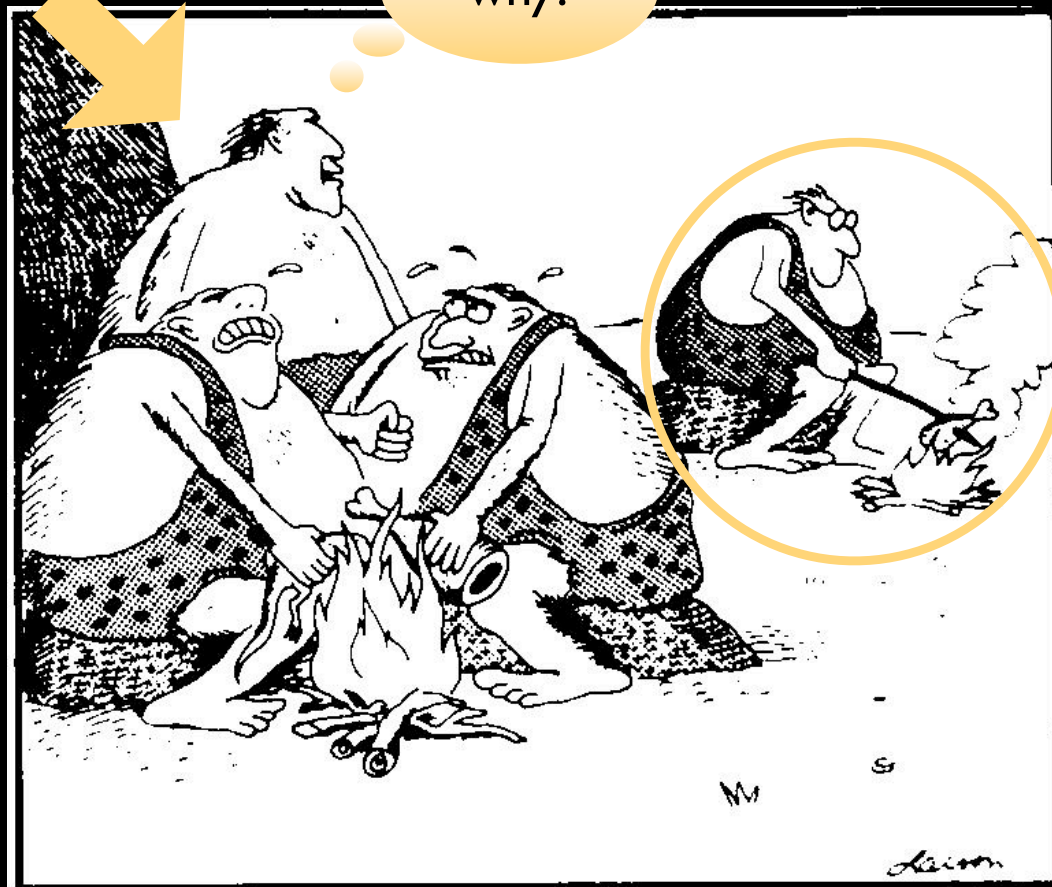
Some learning tasks

Following the upgrading idea

1. explanation based learning
2. local pattern mining
3. theory compression
4. parameter learning

I. Explanation Based Learning as presented by Gerald DeJong

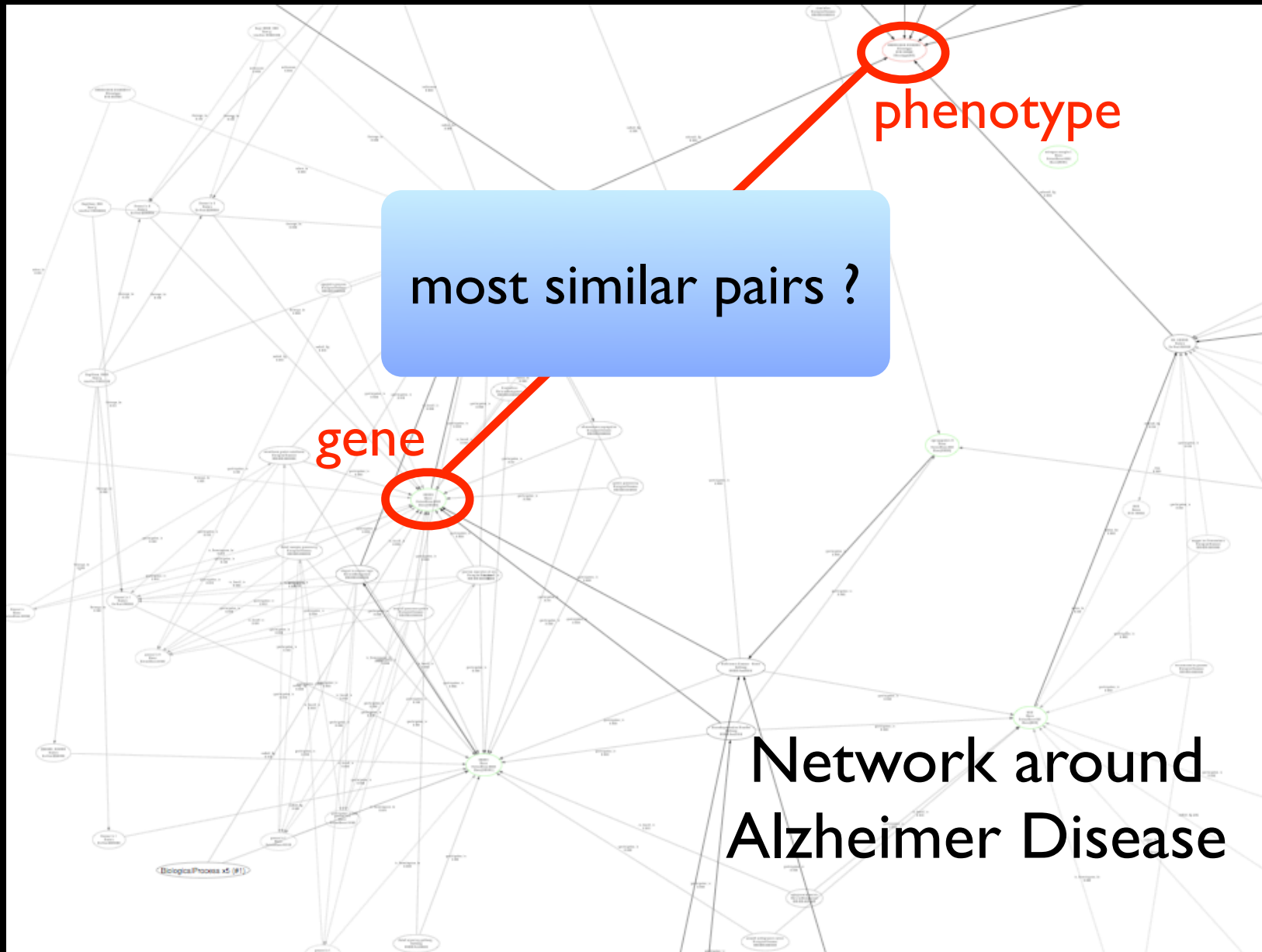
Theory:
knowledge
about
world (fire,
meat, ...)



Example:
no burned
hands

"Hey! Look what Zog do!"

Explanation: use stick



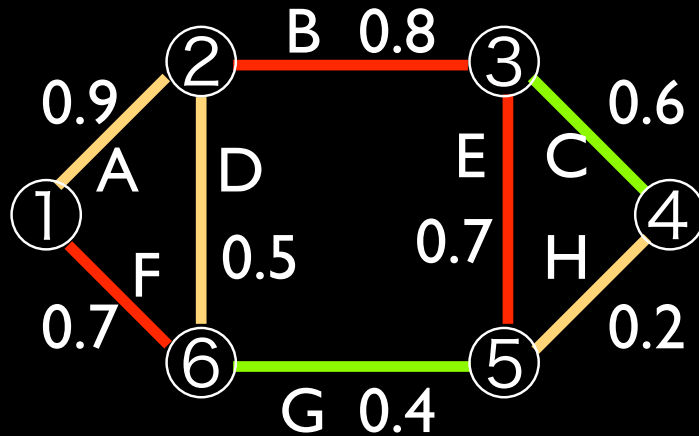
most similar pairs ?

gene

phenotype

Network around
Alzheimer Disease

Most Likely Generalized Explanation



example

① → ④

$\text{path}(x,y) \text{ :- edge}(x,y)$

$\text{path}(x,y) \text{ :- edge}(x,z), \text{path}(y,z)$

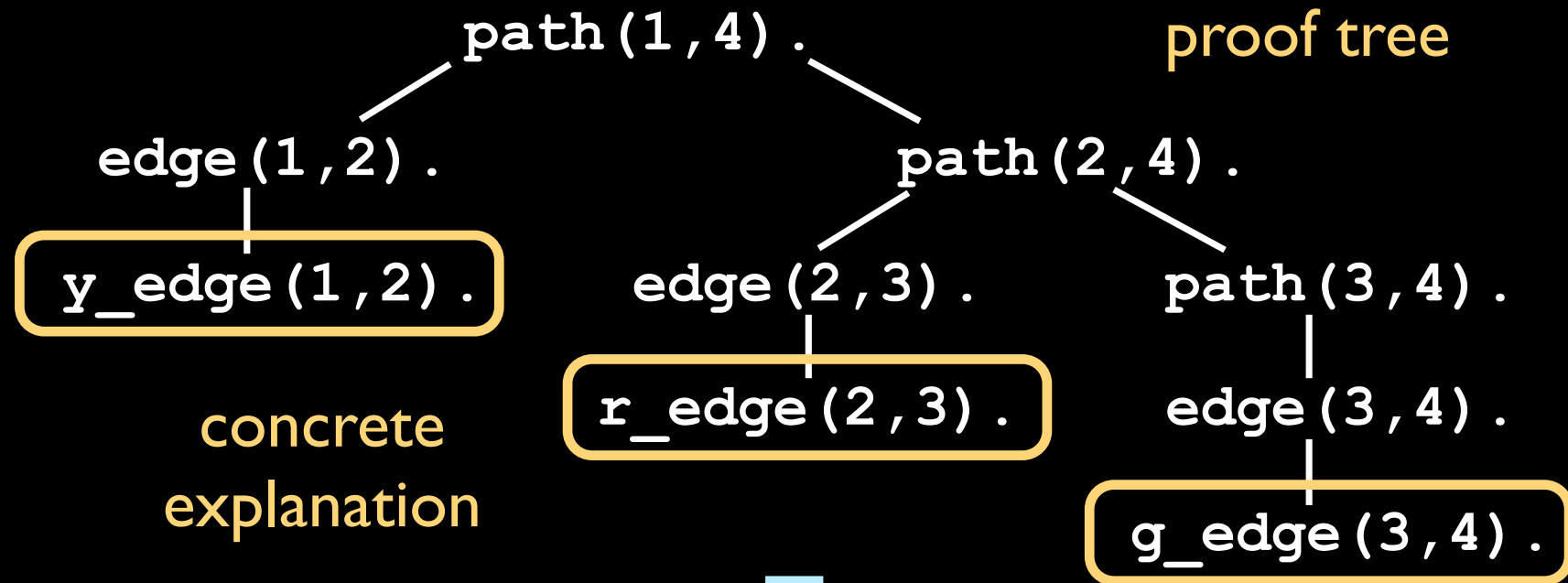


Kimmig et. al. Best Paper
Award ECML 2007

Generalize Explanation

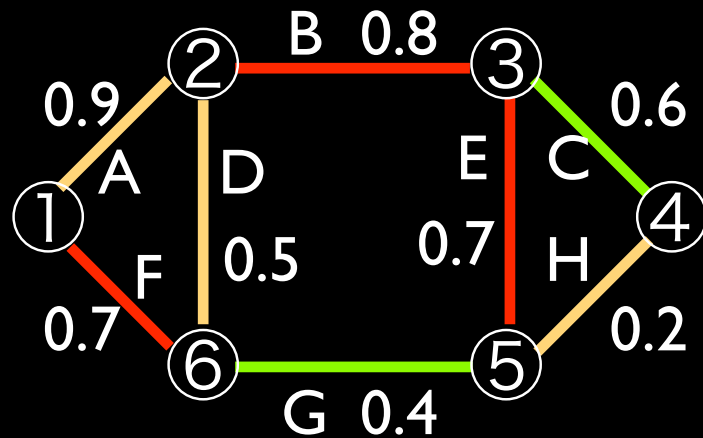


Prolog Setting



path(P,S) ←
y_edge(P,Q), r_edge(Q,R), g_edge(R,S).

Use of Generalized Explanation



Use of Generalized Explanation



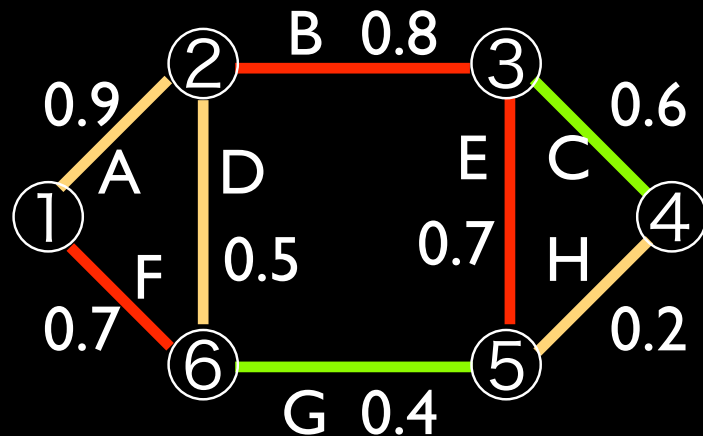
$$\textcircled{3} \rightarrow \textcircled{1} \quad 0.72$$

$$\textcircled{6} \rightarrow \textcircled{2} \quad 0.63$$

$$\textcircled{3} \rightarrow \textcircled{6} \quad 0.40$$

$$\textcircled{1} \rightarrow \textcircled{2} \quad 0.35$$

$$\textcircled{3} \rightarrow \textcircled{4} \quad 0.14$$



reasoning by similarity / analogy

Experiments

	depth	nodes	edges	ag	ng	pt	pos	neg
Alz1	4	122	259	14	15	3	182	2254
Alz2	5	658	3544	17	20	4	272	5056
Alz3	4	351	774	72	33	3	5112	27648
Alz4	5	3364	17666	130	55	6	16770	187470
Ast1	4	127	241	7	12	2	42	642
Ast2	5	381	787	11	12	2	110	902

Table 1. Graph characteristics: search depth used during graph extraction, numbers of nodes and edges, number of genes annotated resp. not annotated with the corresponding disease and number of phenotypes, number of positive and negative examples for connecting two genes and a phenotype.

Experiments

	Alz1						Ast1					
	pos(1)	pos(3)	pos(5)	pos_n	pos_a	prec	pos(1)	pos(3)	pos(5)	pos_n	pos_a	prec
Alz1	0.95	2.53	3.95	6.91	16.82	0.46	1.00	3.00	4.86	6.86	10.57	0.23
Alz2	0.84	2.24	3.60	7.37	18.65	0.42	0.86	2.86	4.71	6.86	14.56	0.22
Alz3	0.99	2.64	4.09	23.20	126.09	0.48	1.00	2.71	4.14	6.86	28.00	0.24
Alz4	0.84	2.23	3.58	7.37	18.80	0.42	0.86	2.29	3.43	5.14	28.00	0.15
Ast1	0.09	0.26	0.44	2.07	2.07	0.02	1.00	3.00	4.86	17.14	17.14	0.34
Ast2	0.08	0.23	0.38	2.00	2.00	0.01	0.86	2.57	4.29	16.57	16.57	0.20

Table 2. Averaged results over all examples learned on Alz1 resp. Ast1 and evaluated on 6 different graphs: number of positives among the first k answers for $k = 1, 3, 5$, number of positives returned before the first negative, absolute number of positives returned, and precision.

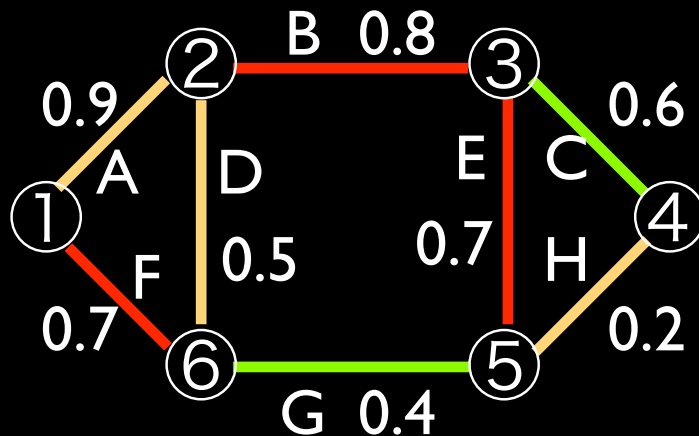
PEBL Contributions

- EBL in probabilistic context
- Multiple explanations: most likely one
- Reasoning by analogy:
background knowledge + likelihood

2. Probabilistic Pattern Mining

What are the most likely explanations the examples have in common ?

criterion: average probability is higher than threshold



③ ①

⑥ ②

③ ⑥

① ②

③ ④

no definition of path

Probabilistic Pattern Mining



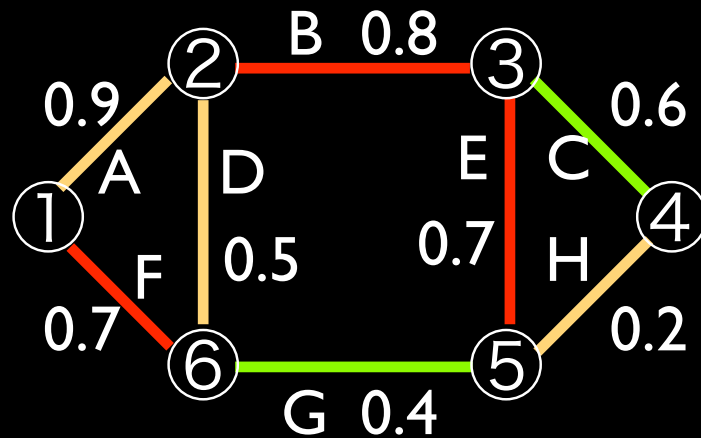
③ ①

⑥ ②

③ ⑥

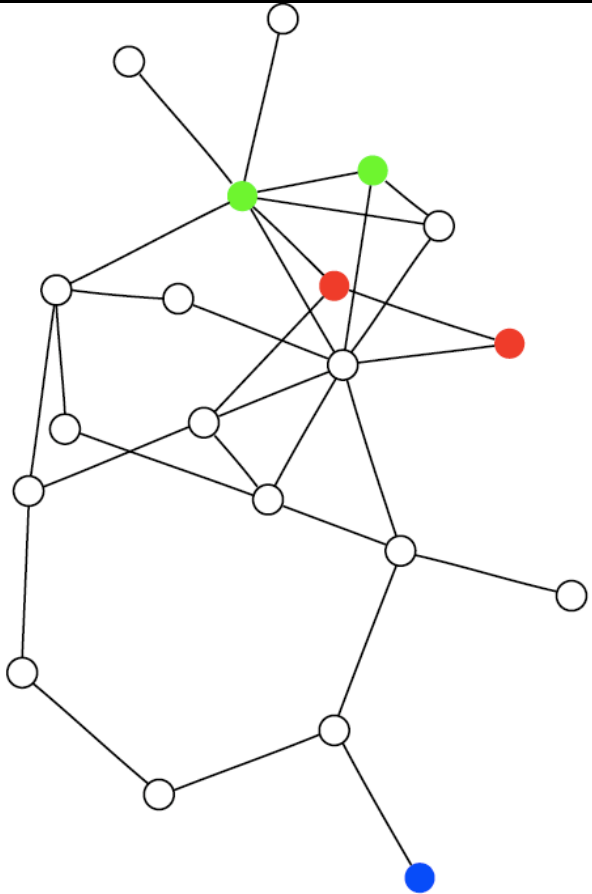
① ②

③ ④



no definition of path

3. Probabilistic Theory Compression/ Revision

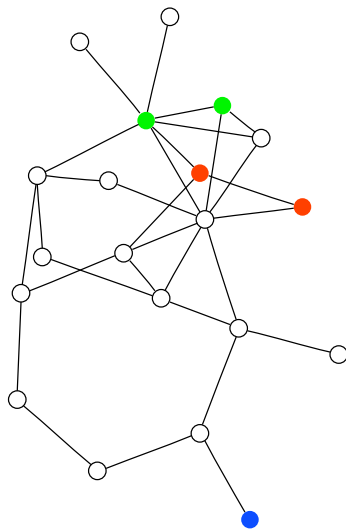


- Given
 - pos / neg interactions
 - Say (green, blue) / (red, blue)
- Find small network (k links) that maximizes prob positives and minimized prob negatives

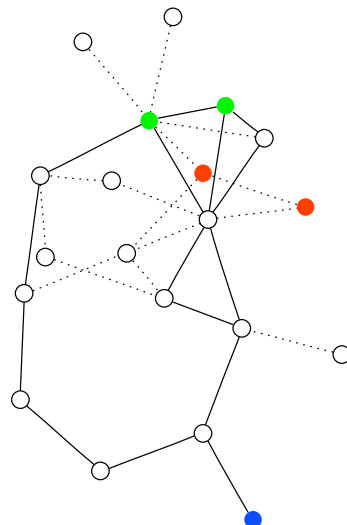
De Raedt et al. MLJ 08

Probabilistic Theory Compression

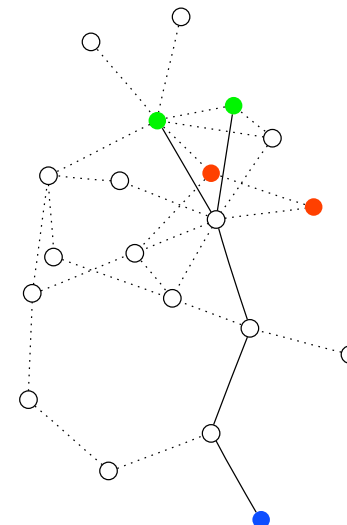
- Reduce to at most k edges (greedy approach, reusing BDDs for scoring)
- Example: Green and blue should be connected, red and blue not (all edges have probability 0.5)



initially



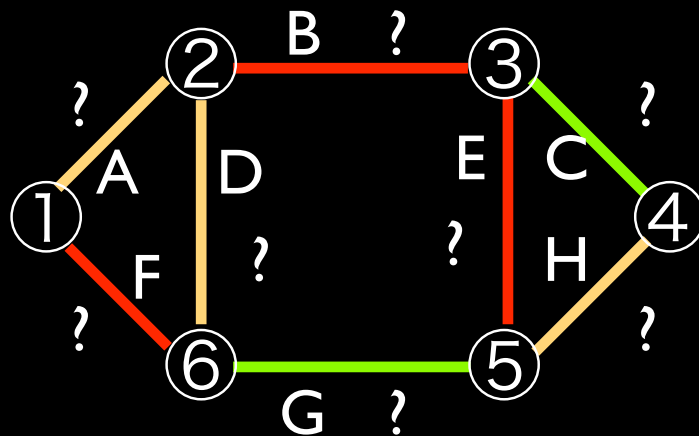
$k = 15$



$k = 5$

4. Parameter Estimation

using least
squares and
gradient



$$\textcircled{3} \rightarrow \textcircled{1} \quad 0.72$$

$$\textcircled{6} \rightarrow \textcircled{2} \quad 0.63$$

$$\textcircled{3} \rightarrow \textcircled{6} \quad 0.40$$

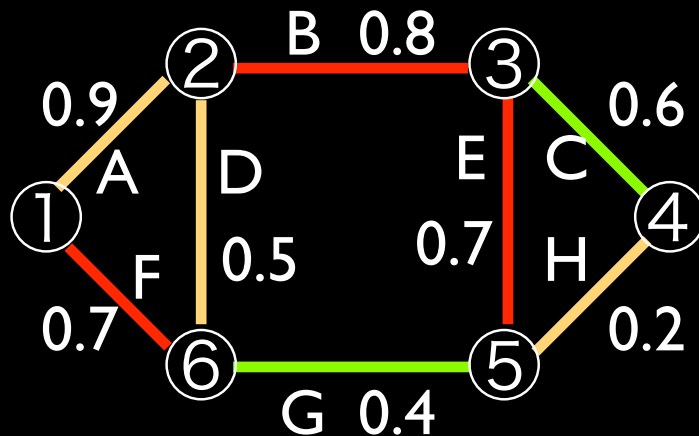
$$\textcircled{1} \rightarrow \textcircled{2} \quad 0.35$$

$$\textcircled{3} \rightarrow \textcircled{4} \quad 0.14$$

Gutmann et al. ECML 08

Parameter Estimation

using least
squares and
gradient



$$\textcircled{3} \rightarrow \textcircled{1} \quad 0.72$$

$$\textcircled{6} \rightarrow \textcircled{2} \quad 0.63$$

$$\textcircled{3} \rightarrow \textcircled{6} \quad 0.40$$

$$\textcircled{1} \rightarrow \textcircled{2} \quad 0.35$$

$$\textcircled{3} \rightarrow \textcircled{4} \quad 0.14$$

Gutmann et al. ECML 08

Experiments

- For all of the settings specified, we did set up experiments that show that meaningful links can be (re)-discovered

D. Challenge

Welcome to *Travian*



- *Travian*: A massively multiplayer real-time strategy game
 - Commercial game run by TravianGames GmbH
 - ~3.000.000 players spread over different “worlds”
 - ~25.000 players in one world



World Dynamics

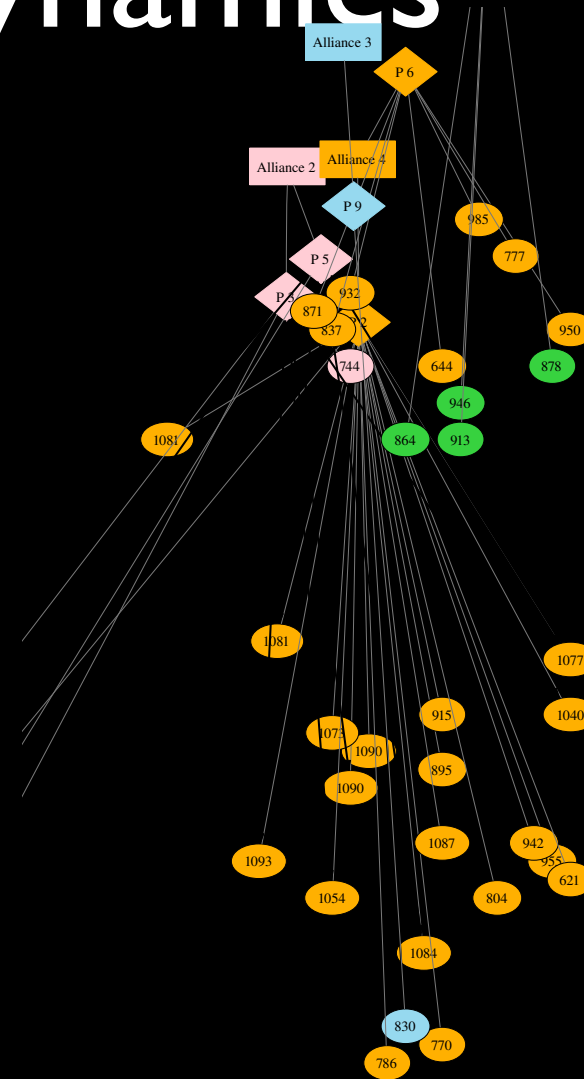
Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



World Dynamics

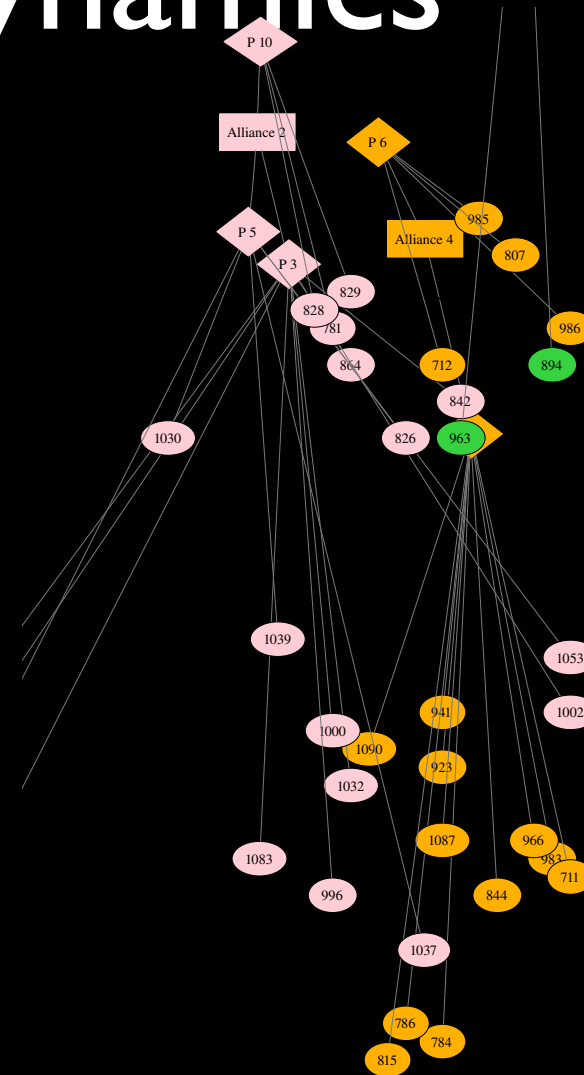
Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



World Dynamics

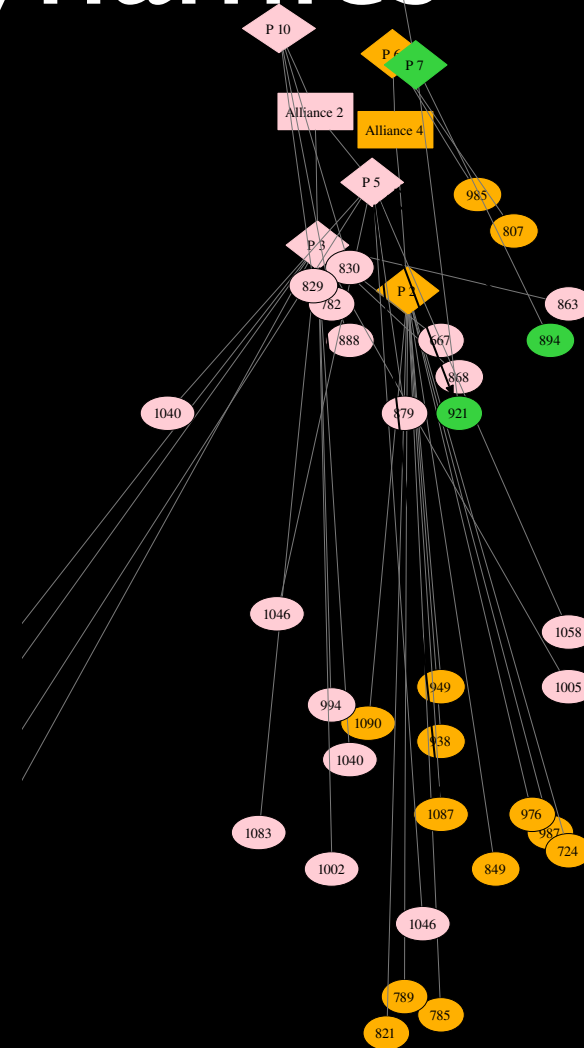
Fragment of world with

~10 alliances
~200 players
~600 cities

alliances color-coded

Can we build a model
of this world ?
Can we use it for playing
better ?

[Thon, Landwehr, De Raedt, ECML08]



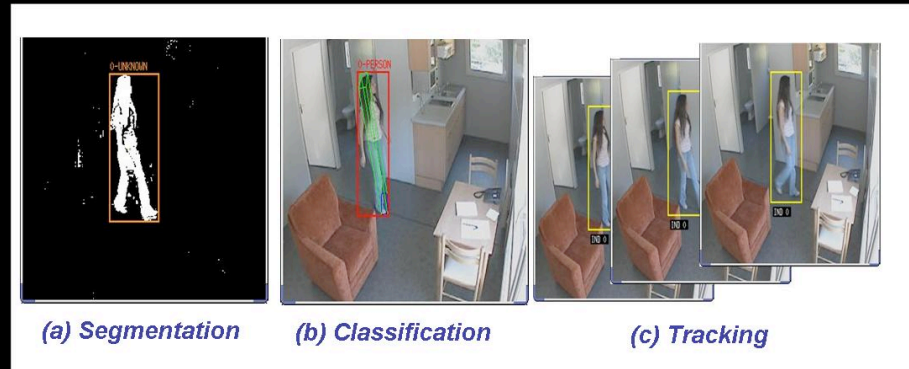
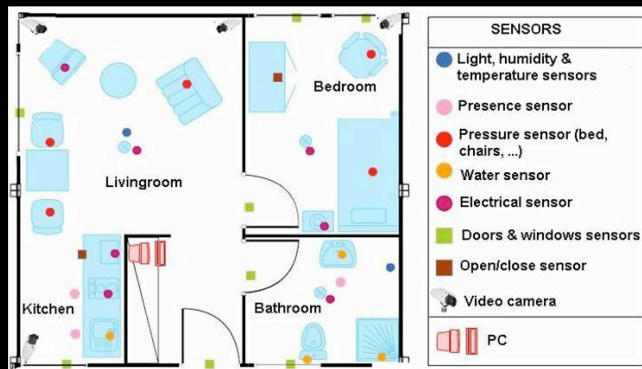
Emerging Data Sets

In many application areas :

- vision, surveillance, activity recognition, robotics, ...
- data in relational format are becoming available
- use of knowledge and reasoning is essential
- in Travian -- ako STRIPS representation

GerHome Example

Action and Activity Learning



(courtesy of Francois Bremond, INRIA-Sophia-Antipolis)

<http://www-sop.inria.fr/orion/personnel/Francois.Bremond/topicsText/gerhomeProject.html>

Conclusions

Logic and relational learning toolbox (take what you need)

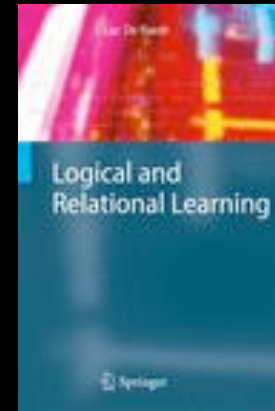
- rules & background knowledge
- generality & operators
- upgrading & downgrading
- graphs & relational database & logic
- propositionalization & aggregation
- probabilistic & logic

Further Reading

Luc De Raedt

Logical and Relational Learning

Springer, 2008, 401 pages, in print.



(should be on display at the Springer booth)

