



HELSINGIN YLIOPISTO
HELSINGFORS UNIVERSITET
UNIVERSITY OF HELSINKI

Biomine search engine for probabilistic graphs

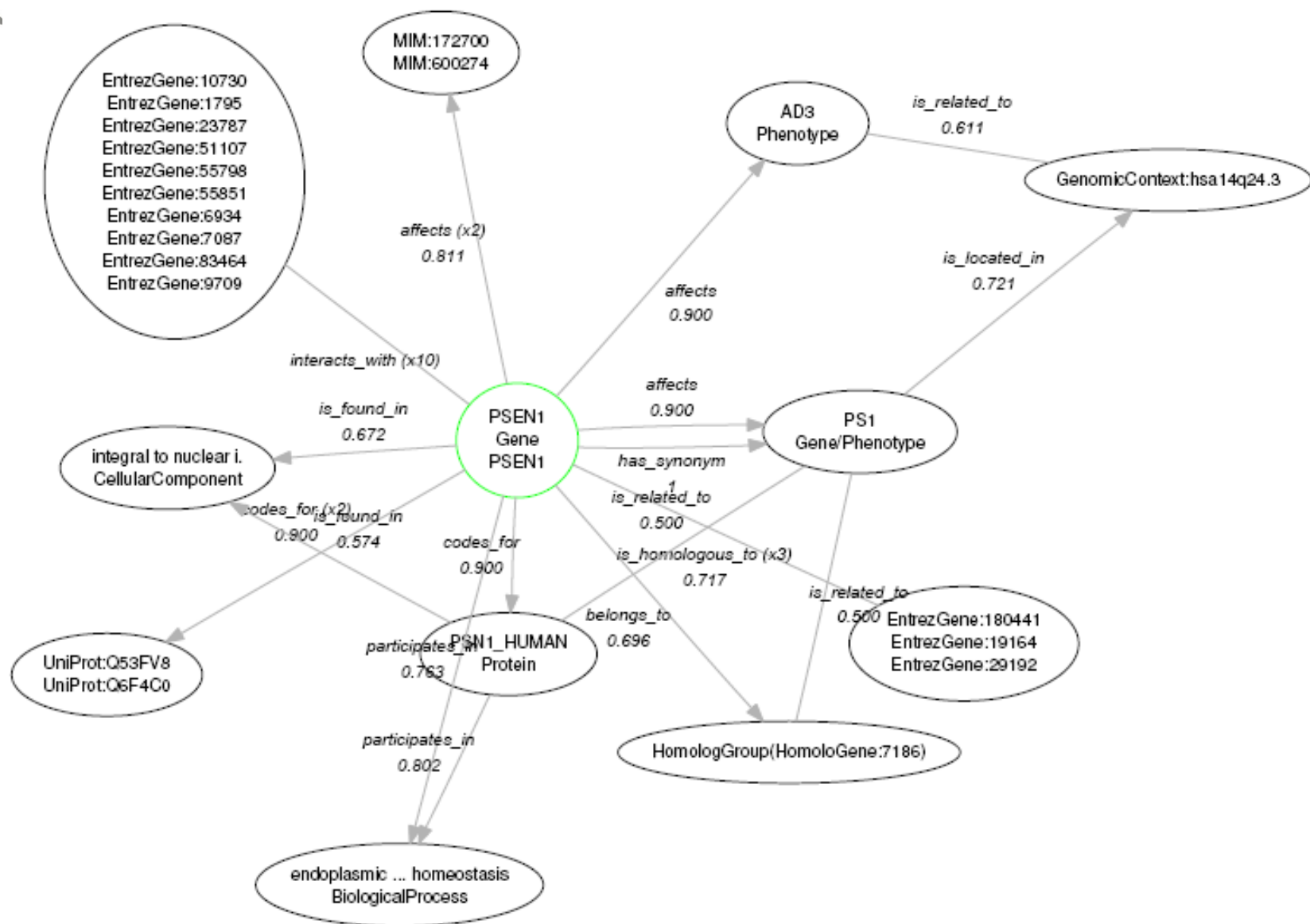
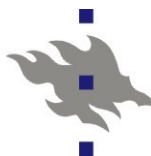
Hannu Toivonen
University of Helsinki

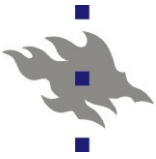
MLG, Helsinki, July 5, 2008





What is known about PSEN1 (presenilin1) gene?



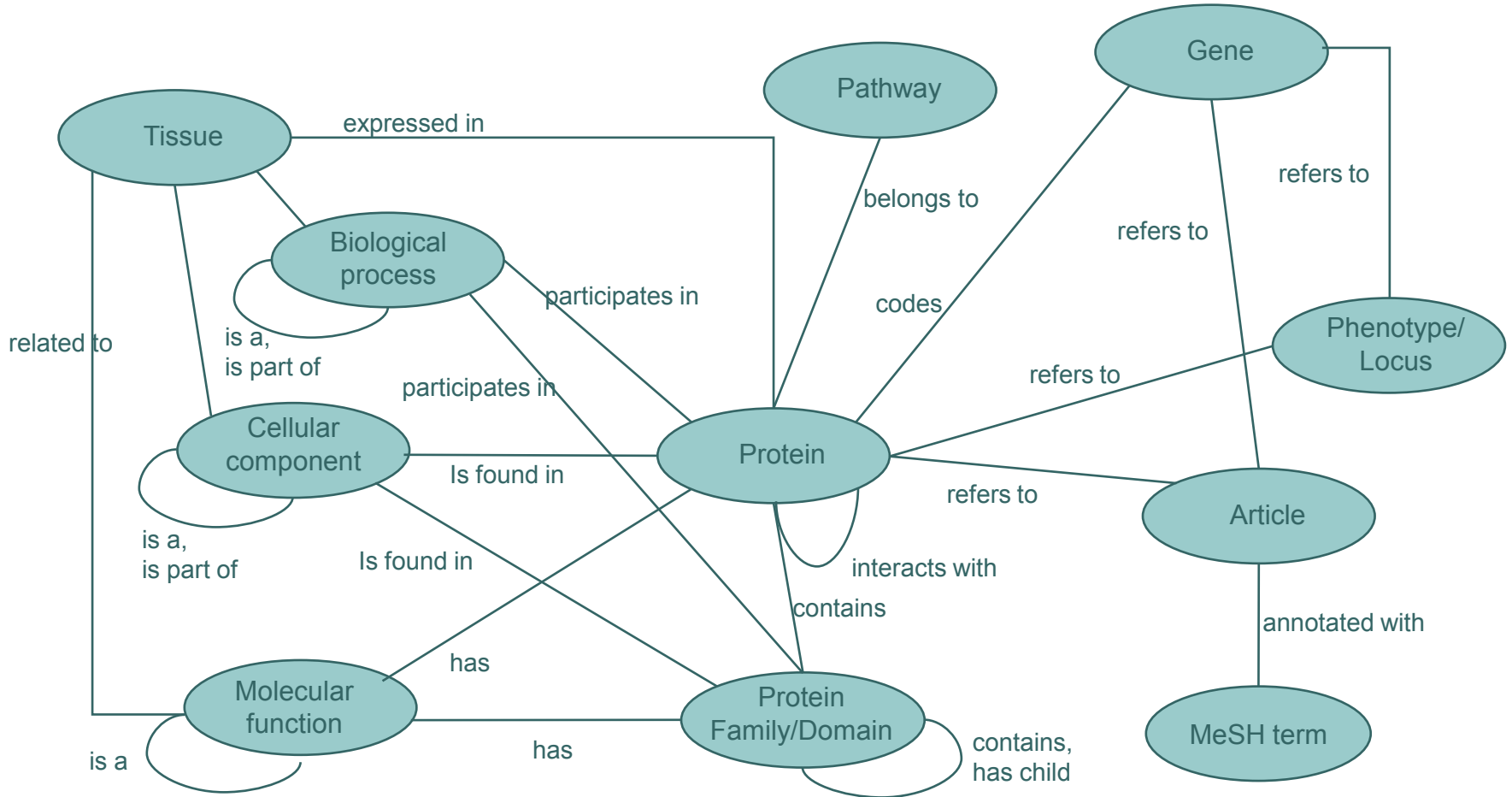


Biomine: Search in biological graphs

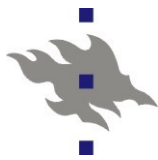
- A graphical representation of biological data
 - Nodes: genes, proteins, tissues, processes, pathways, homology groups, phenotypes, ...
 - Edges: known, reported or predicted relationships between nodes
 - Edges have weights to describe their certainty (and relevance and informativeness)
- A data mining goal: discovery of novel, non-trivial (indirect) relationships
 - E.g. possible explanations for a biological hypothesis, or discovery of new hypotheses



Biomine graph schema



■ Edge weight = probability



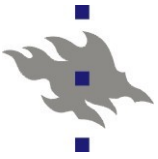
Databases and nodes indexed by Biomine

Source databases: Node types and counts:

Entrez Gene	Article	409219
Entrez Protein	Protein	355188
GO	Gene	175230
HGNC	HomologGroup	39493
HomoloGene	GO	25875
InterPro	Ligand	24149
KEGG	Compound	15003
MIM	BiologicalProcess	14919
MeSH	GenomicContext	14730
PubMed	OrthologGroup	11345
STRING	MolecularFunction	8789
UniProt	Drug	6637
	Phenotype	6331
	...	

#Nodes: 1 083 891

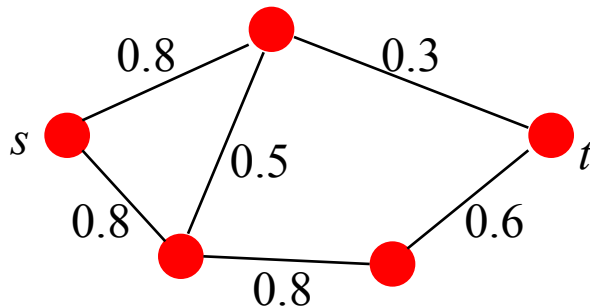
#Edges: 6 653 464



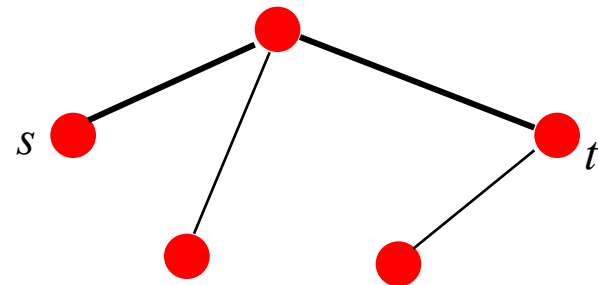
Probabilistic graphs

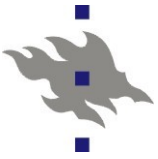
- A weighted graph $G=(V, E, P)$
- V, E as in standard graphs
- $P(e)$ is the probability of e in E
 - Edge e is true (or exists) with probability $P(e)$
 - Edges are mutually independent

Probabilistic graph G



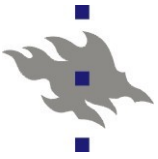
A random realization of G





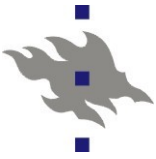
Connectivity between nodes

- An elementary question: how strongly are two nodes s and t connected?
 - Given a node s , search for nodes t that are strongly connected to s
 - Given nodes s, t_1, t_2, \dots , rank t_1, t_2, \dots by their connectivity to s



Measures of connectivity

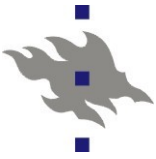
- *Reliability*: the probability that nodes s , t are connected in the probabilistic graph (i.e., that there exists a path of true edges connecting s and t)
 - Known as two-terminal network reliability from 60's
- Simple alternative: *probability of the best path* connecting s and t



Properties of reliability

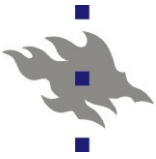
- Penalizes long paths (long chains of uncertain inference)
- Rewards parallelism (alternative explanations)
- A natural probabilistic interpretation

- Related models and measures
 - Maximum network flow
 - does not penalize path length
 - Current in resistor networks (Faloutsos et al., 2004)
 - no easy intuitive interpretation
 - Expected time to meeting/arrival in random walks (SimRank; Jeh&Widom, 2002)
 - does not reward parallelism



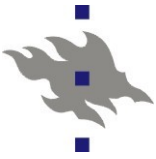
Notes on computation

- Computing the probability of the best path: trivial
- Finding the best path
 - Can be solved with shortest path algorithms
- Computing (two-terminal network) reliability
 - Investigated since the 1960's
 - NP-hard (Valiant 1970)
 - Approximation methods
 - Monte Carlo simulation
 - Exact computation (with BDDs) for a subgraph
 - Lower (and upper) bounds by exact computation
 - Series-parallel reductions
 - ...



Origin of probabilities in Biomine

- Probabilities are computed from three factors
 - Reliability of the link source
 - Method or database specific, e.g., based on sequence similarity or strength of association
 - Relevance to the user
 - Subjective view of what is interesting
 - Rarity of the link
 - Informativeness of an edge, low for nodes with a high degree
- Reliability, relevance, rarity are in $[0,1]$
- Edge probability = reliability x relevance x rarity



Two search problems

- Consider search types where
 - input consists of a node or a set of nodes
 - output is a subgraph (or a set of nodes)

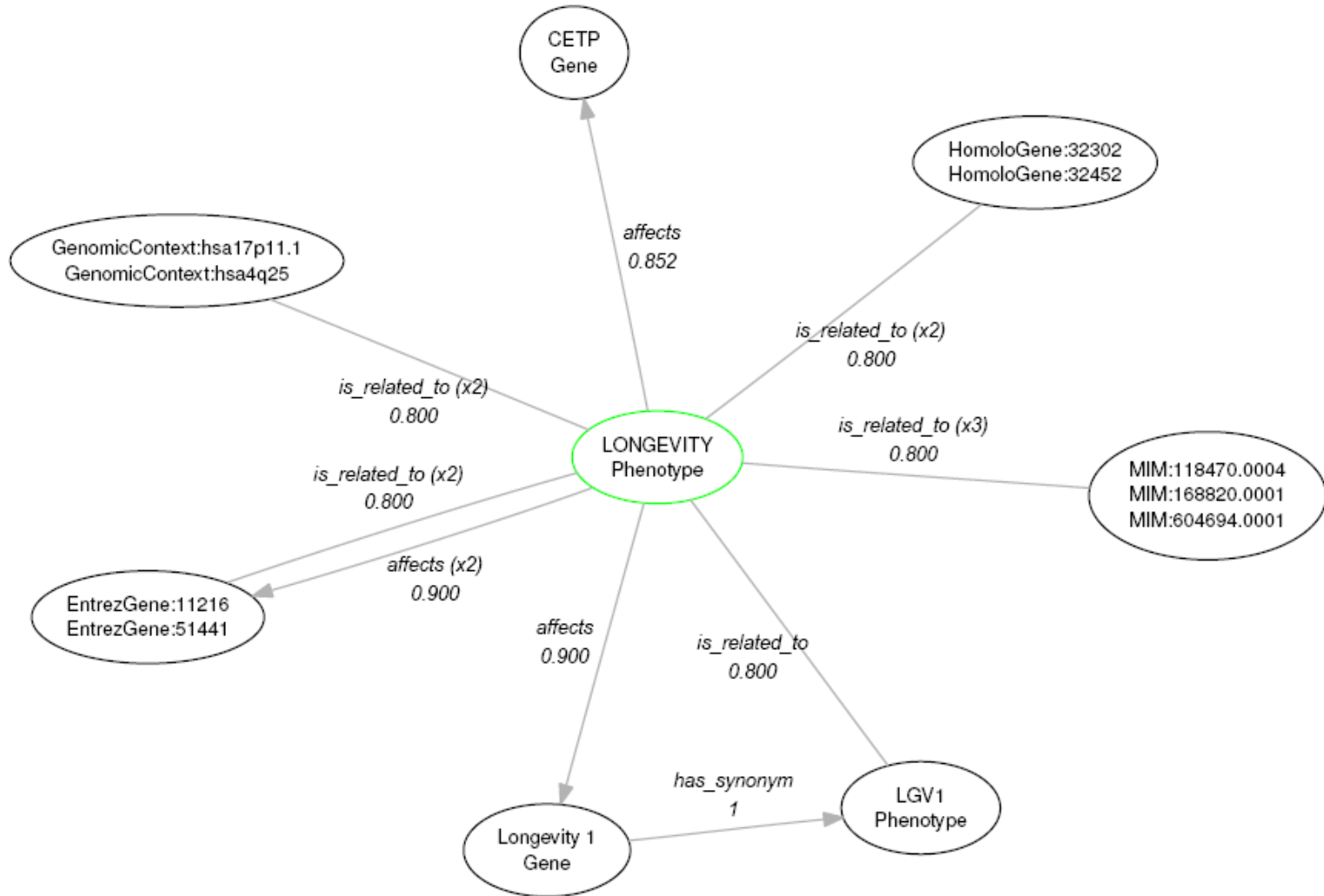
- and where the general goal is to
 - maximise the probability that nodes in the output are connected to nodes in the input (i.e., the reliability of the output graph wrt the input nodes)

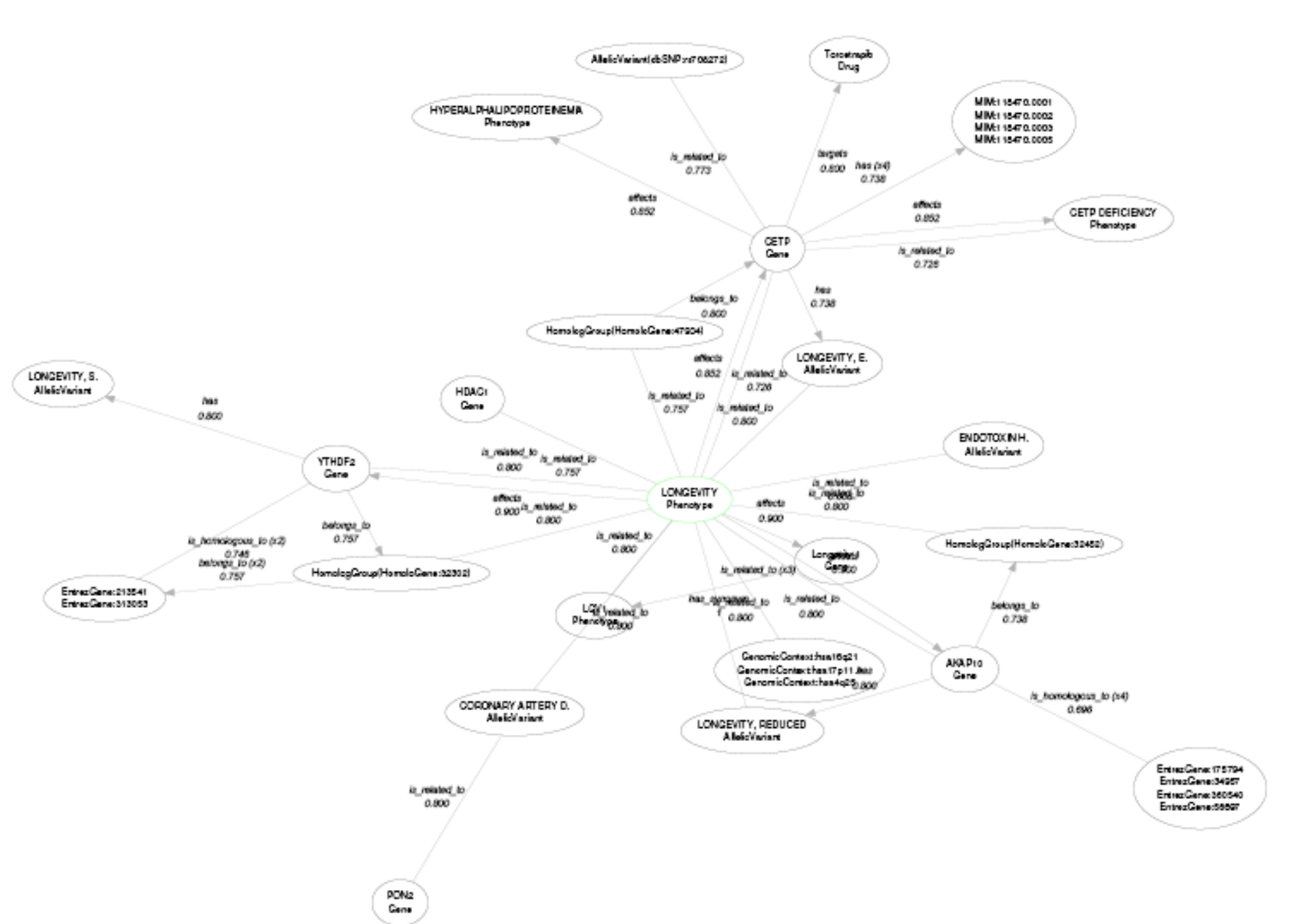


1. Neighborhood query

- Given a query node s , retrieve its neighbors
- (Or, given a set of query nodes, return the union of their neighborhoods)

- Find those k nodes that have the highest reliability of being connected with node s



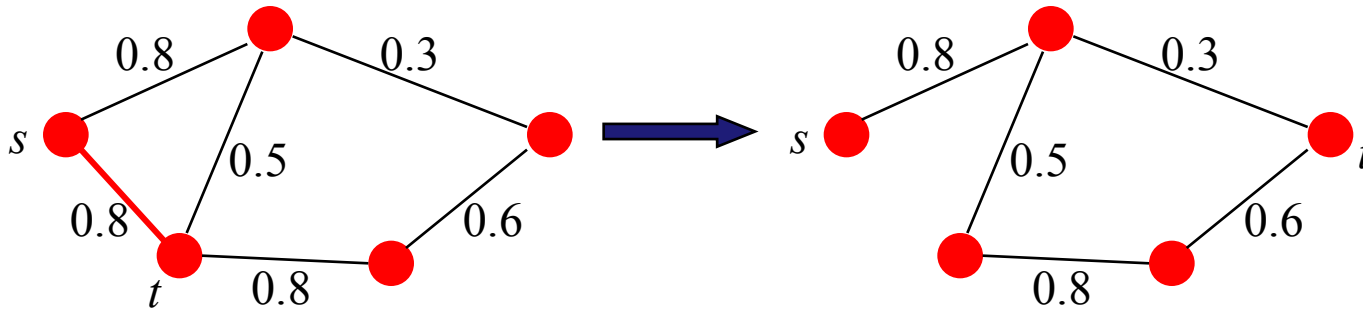




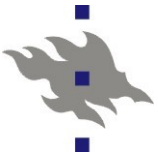
Are neighborhood queries useful?

■ Test setting

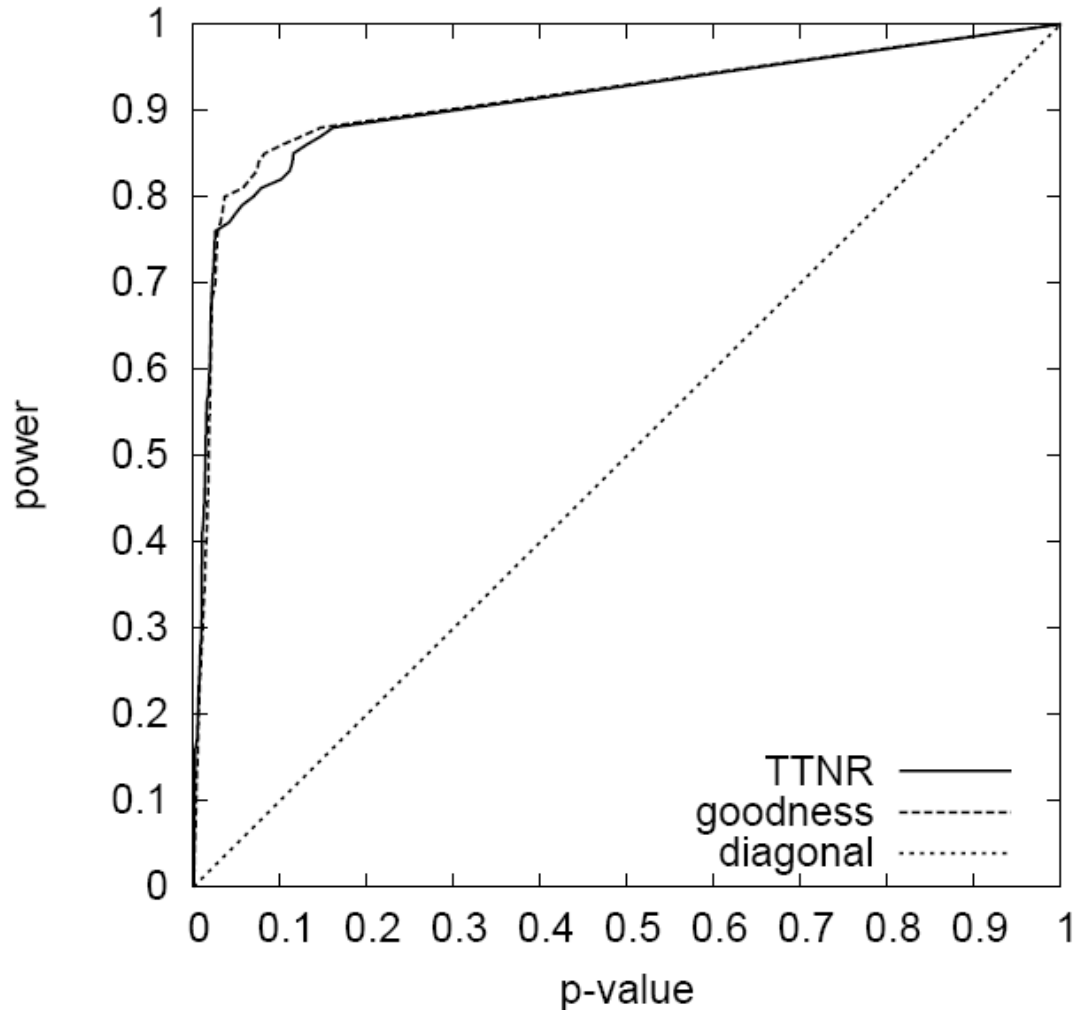
- Use a hold-out set of edges
- For each hold-out edge (s, t) , compute the reliability of the graph wrt. s and t



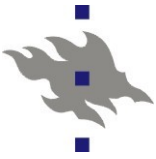
- Compute the reliability for random node pairs (chosen to be similar to s and t) (null distribution, \sim negative examples)
- Obtain a p-value for edge (s, t)



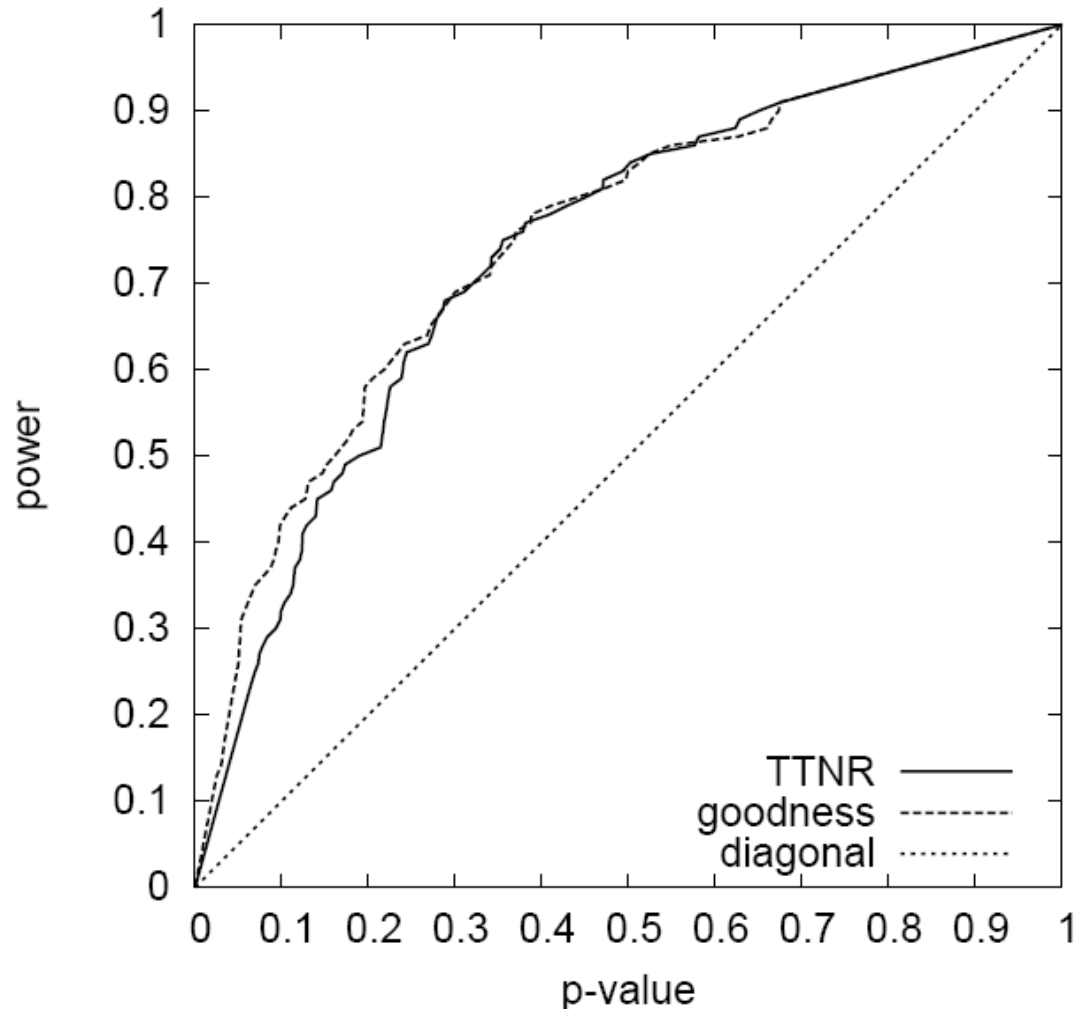
Prediction of missing protein interactions



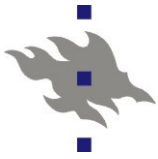
- (Gene interactions and gene-phenotype relations were also removed)



Prediction of future gene interactions

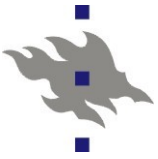


- (Note: comparison is against interactions discovered within the next six months, not true ones.)



Are neighborhood queries useful?

- Apparently yes: there is potential to predict links
 - Reliability and probability of the best path seem to perform equally well
- BTW, no machine learning so far
 - Given a training set, we could fit our model (the probabilistic graph) better to the data
 - E.g., learn data source specific reliabilities or edge type relevances, even individual edge probabilities
 - (Let's see what the next talk in this session is about...)

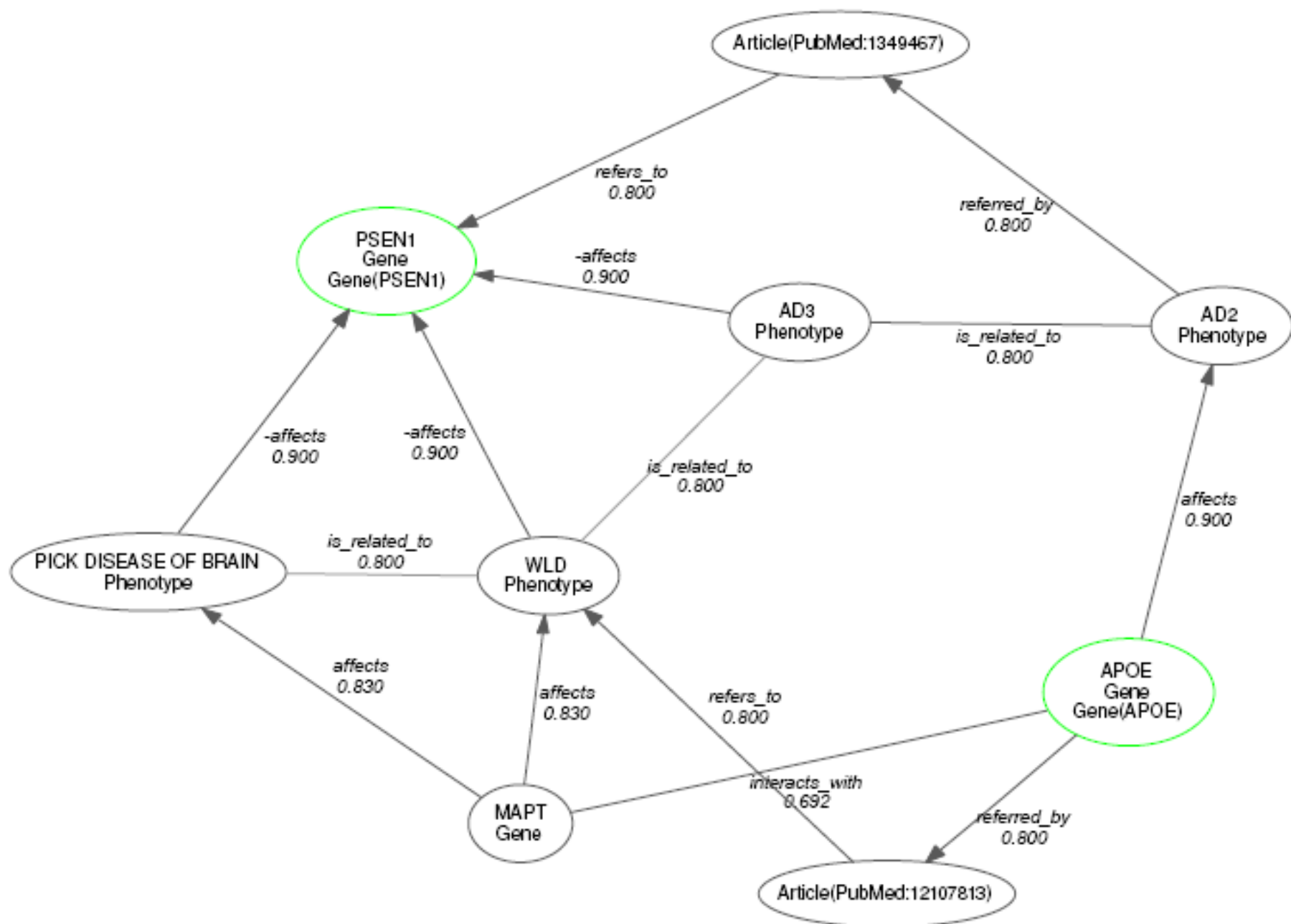


2. The most reliable subgraph problem

- Given two query nodes s and t , find a subgraph (of a limited maximum size) that connects the query nodes as strongly as possible
- Motivation
 - Visualization
 - Preprocessing for computationally intensive methods
- For a probabilistic graph: extract the most reliable subgraph (of size at most k) wrt. to s and t
 - Ensures relevance wrt to both s and t
 - Favors results with little redundancy

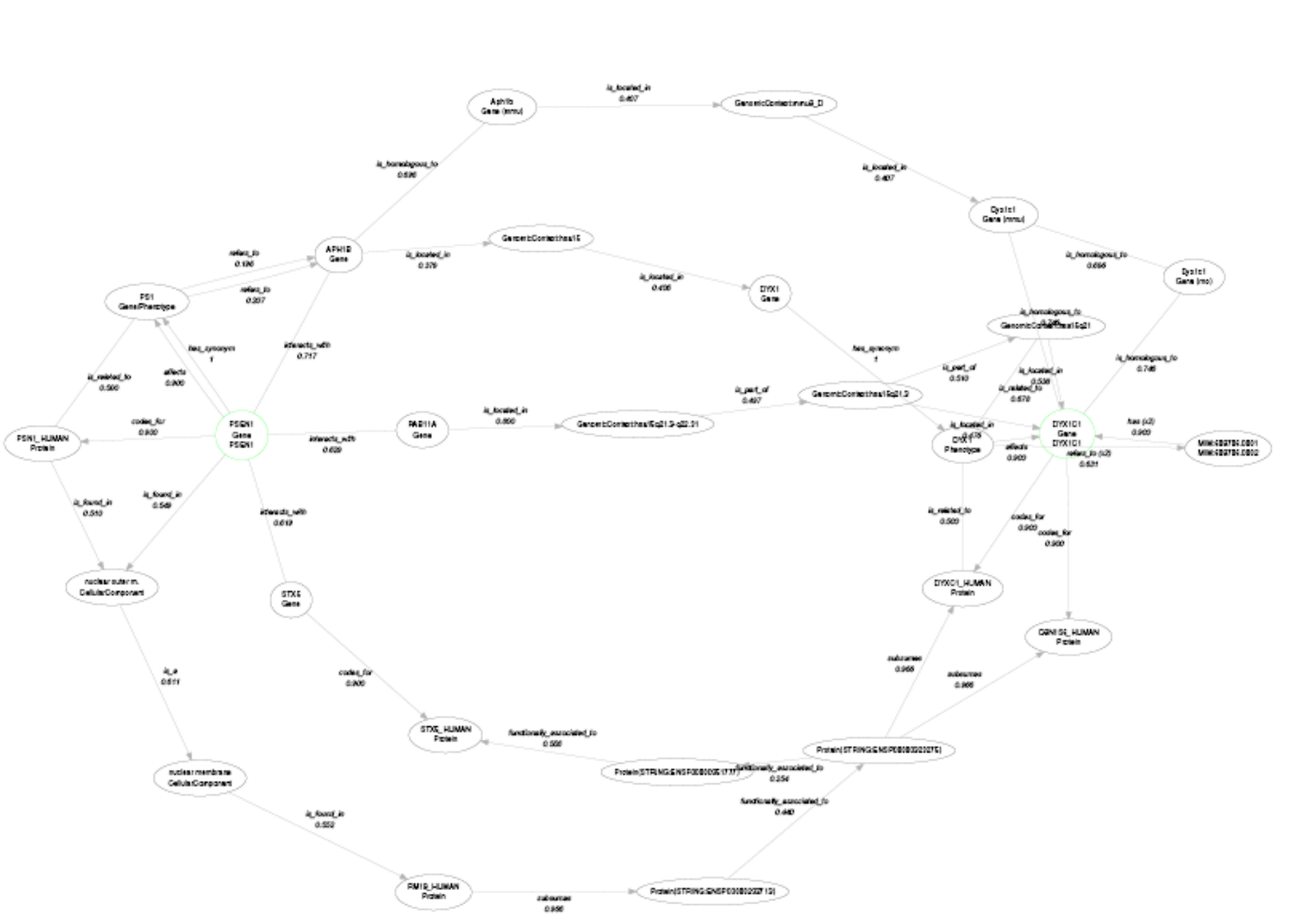


How are genes PSEN1 (presenilin1) and APOE (apolipoprotein E) related?





How are genes PSEN1 (presenilin1) and
DYX1C1 (apolipoprotein E) possibly related?





Subgraph extraction

■ Related work

- Faloutsos et al (2004): connection subgraphs
 - model: current in resistor networks
- De Raedt, Kimmig, Toivonen (2008): ProbLog theory compression
 - Similar to Biomine, but in first-order logic

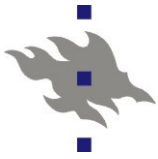
■ Two opposite heuristic approaches

- Prune the original graph until the required size is reached
 - Complexity depends on the size of the original graph
- Construct a subgraph incrementally
 - Complexity depends (more) on the size of the result

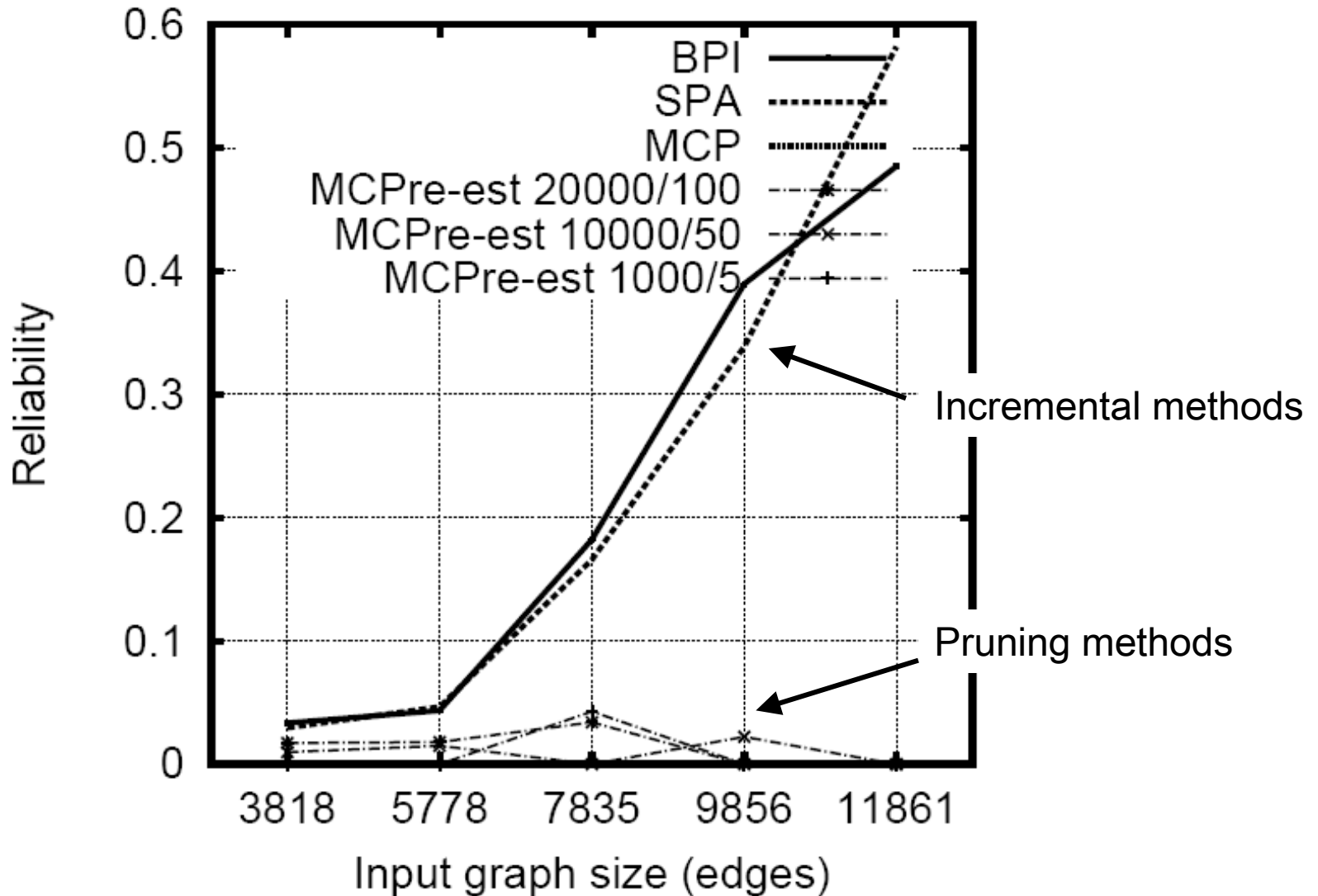


Two new incremental methods

- Parameter k : upper limit for the size of the result
- BPI, Best Paths Incremental
 - Take K best paths, such that they span a graph of size k
 - Very simple (not even greedy)
 - A greedy variant would require repetitive evaluations of the reliability, which is computationally demanding
- SPA, Series-Parallel Augmentation
 - Greedily builds a series-parallel of size at most k
 - Series-parallel graphs can be evaluated efficiently
 - A greedy method makes optimal additions
 - ...but optimal only in the restricted class of S-P graphs
- Hintsanen and Toivonen (PKDD/DAMI 2008)

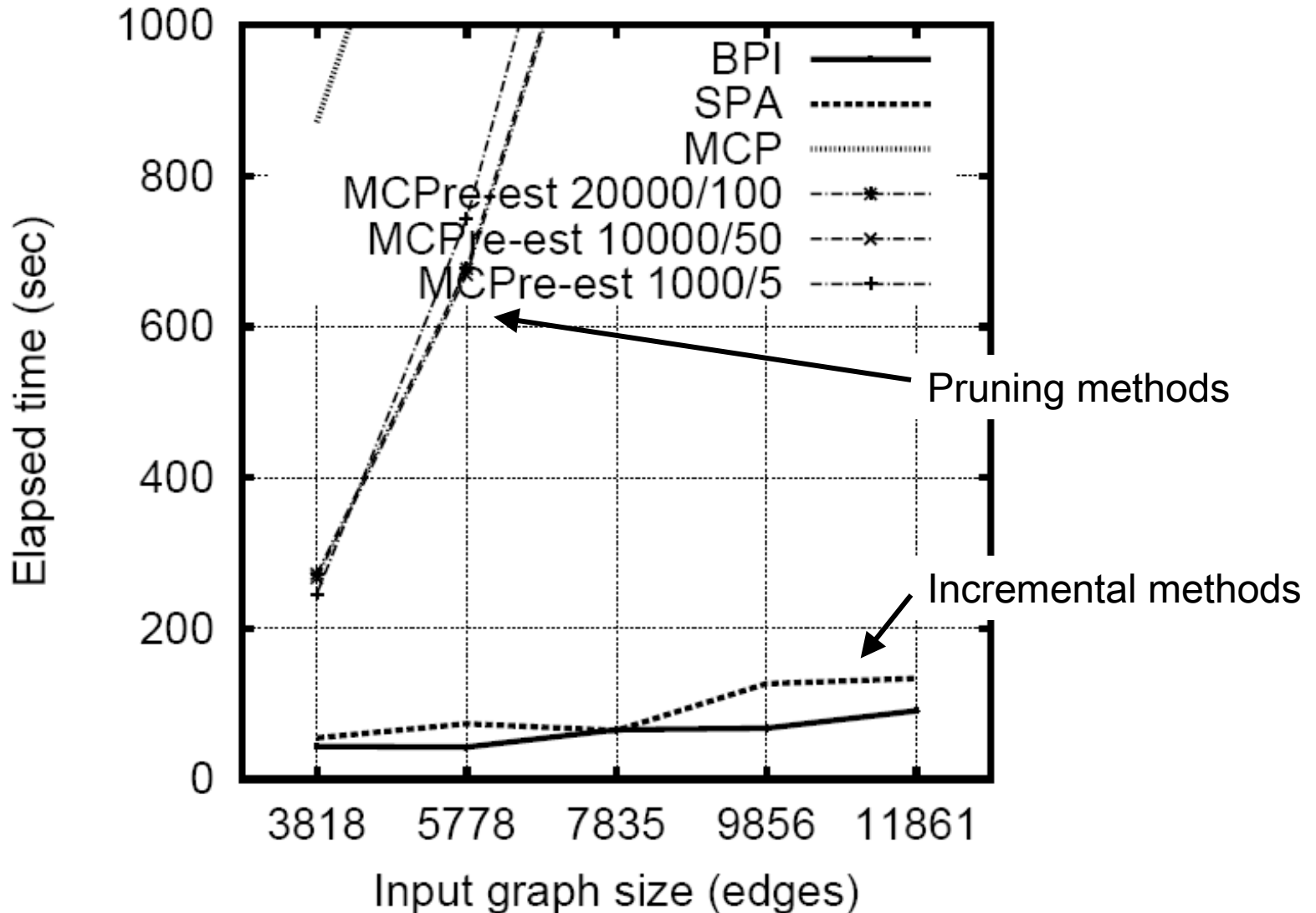


Quality of the extracted subgraph (as a function of the size of the input)



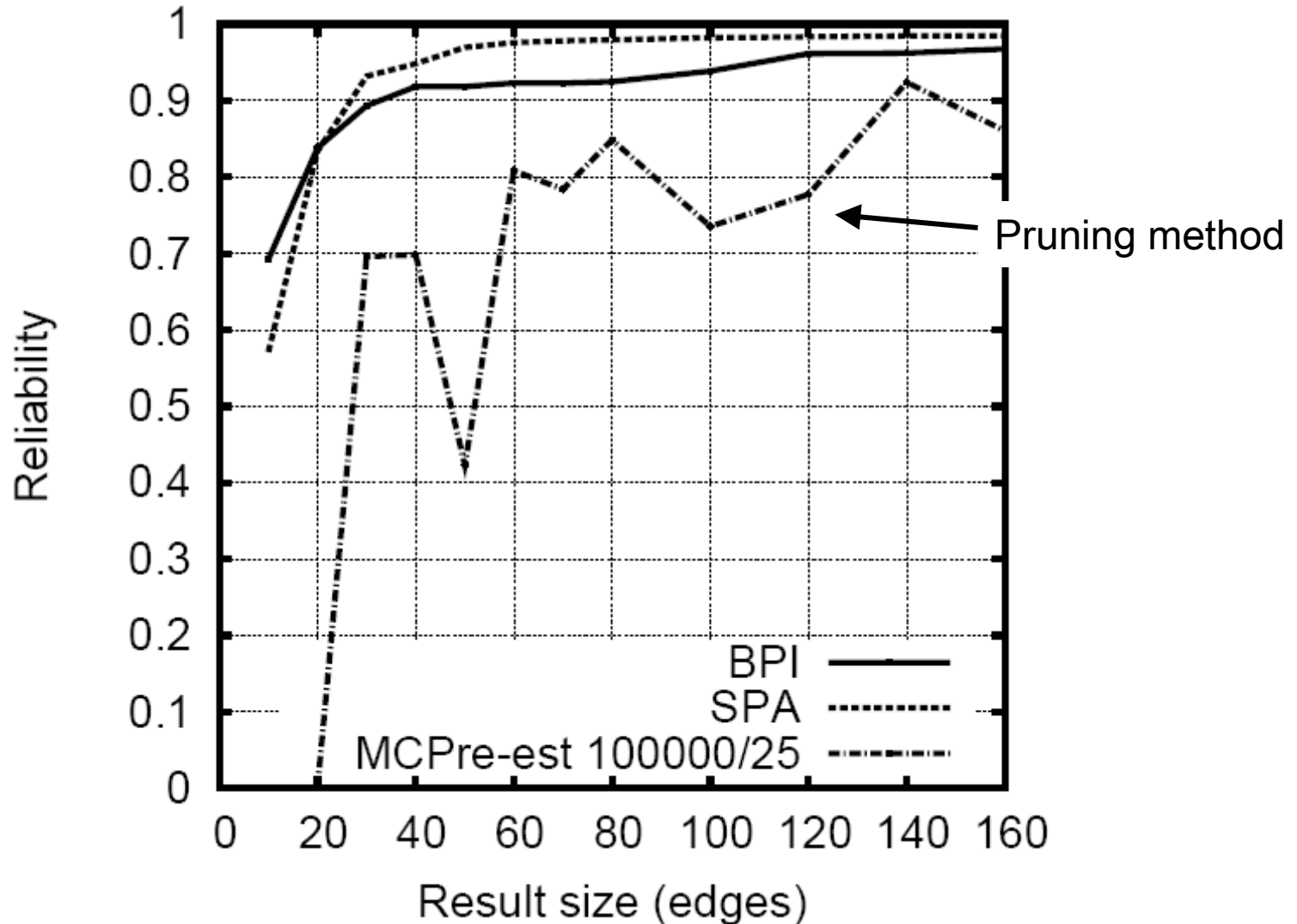


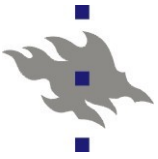
Time to extract a subgraph (as a function of the size of the input)



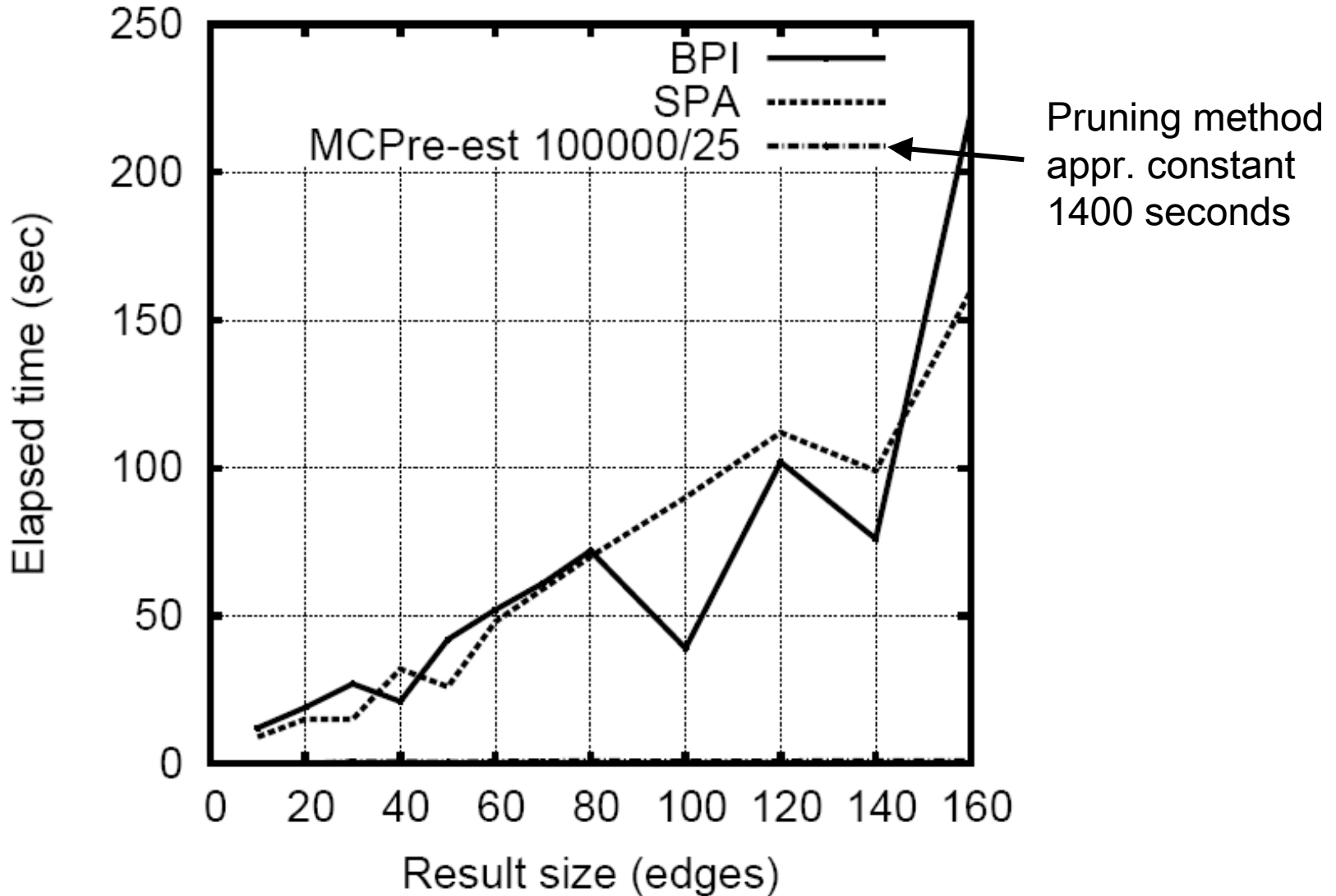


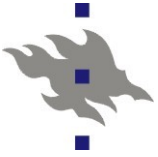
Quality of the extracted subgraph (as a function of the size of the output)





Time to extract a subgraph (as a function of the size of the input)

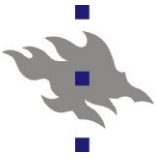




Slide 33/44

- We have now looked at
 - Probabilistic graphs
 - Reliability and path probability as measures of connectedness
 - The most reliable subgraph extraction problem

- Coming up next: different views to subgraph extraction
 - Context-free grammars as a qualitative query tool
 - ProbLog: a probabilistic Prolog

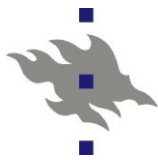


Subgraph extraction problem

- The most reliable subgraph problem is quantitative

- Consider a qualitative variant:
 - The user specifies relevant path types
 - The task is to find all paths between s, t , of the given types
 - The method returns the subgraph induced by the set of accepted paths

- Sevón & Eronen (2008)



Example: paths from ACHB3_HUMAN to AD

QUERY

SEQUENCE

PROTEIN

IS_SIMILAR_TO

IS_ASSOCIATED_TO

PROTEIN

BIOLOGICALPROCESS

GENE

ARTICLE

PHENOTYPE

Protein

belongs_to

BiologicalProcess

-belongs_to Protein -codes_for

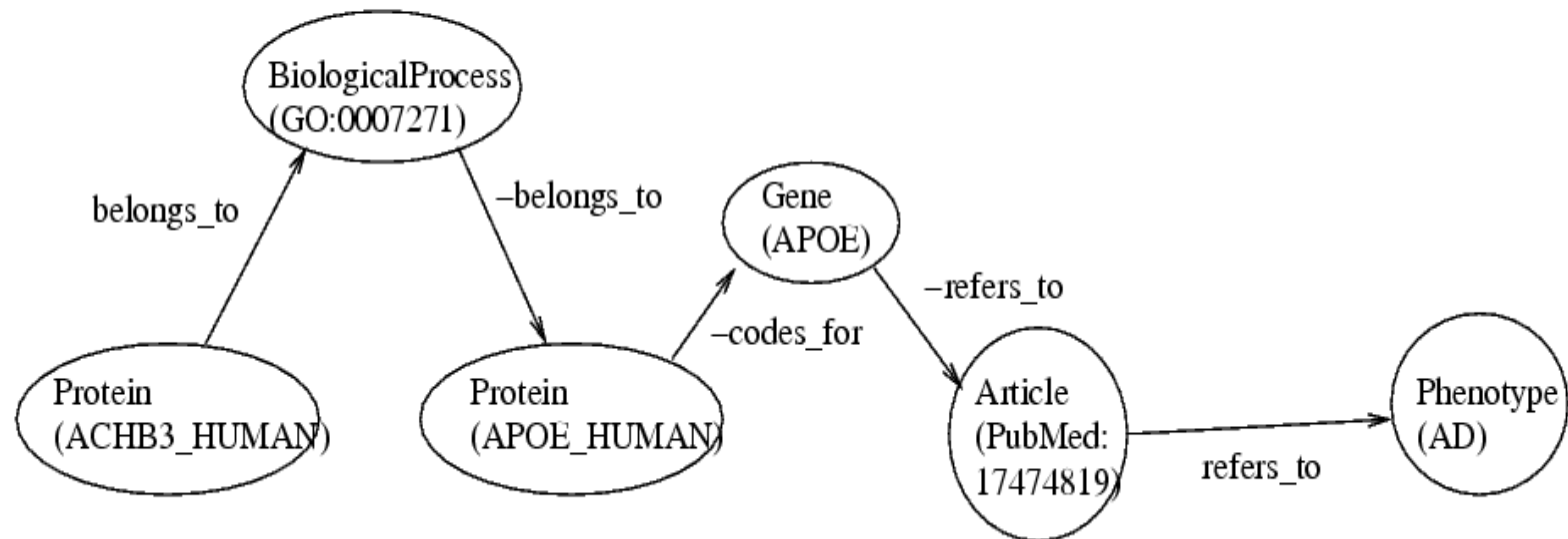
Gene

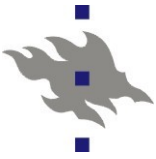
-refers_to

Article

refers_to

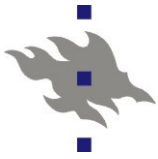
Phenotype





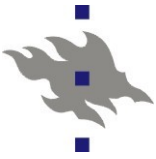
Subgraph queries with context-free grammars

- Path type = the alternating sequence of node and edge types on a path
 - e.g., Gene participates_in Pathway is_related_to Phenotype
- Use CFG to specify the class of acceptable path types
 - terminal symbols = node and edge types
 - nonterminal symbols = path classes
 - starting nonterminal = class of acceptable paths
- Path classes are pre-defined in a background CFG, queries are formulated by specifying the root level production rules



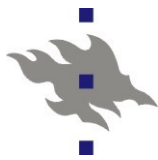
Subgraph queries with context-free grammars

- Key idea: extract a subgraph, spanned by admissible paths
 - The grammar controls what is useful, relevant, or plausible
- Algorithmic issues
 - How to parse all graphs from a graph efficiently
 - Sevón & Eronen (IB 2008)
- Combining qualitative and quantitative approaches: probabilistic grammars
 - Paths (strings) have probabilities, derived from the edges
 - Productions of the grammar can have probabilities, too



ProbLog

- ProbLog = Prolog + probabilities of clauses
 - `0.3779:edge('EntrezProtein_4885045','HGNC_620').`
`0.4928:edge('HGNC_620','PubMed_12653567').`
`0.6054:edge('EntrezProtein_4885045','HGNC_12850')`
`.`
`0.9022:edge('PubMed_2322535','HGNC_983').`
`0.8750:edge('HomoloGene_20065','HGNC_983').`
`...`
`1.0:path(X,Y):-edge(X,Y).`
`1.0:path(X,Y):-edge(X,Z),path(Z,Y).`
- Each clause has a probability to be in a Prolog program
- Clauses are mutually independent
- Suitable for representing and querying probabilistic graphs
- [De Raedt, Kimmig, Toivonen, IJCAI 07, PKDD 07]



ProbLog semantics

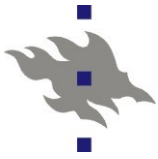
- A ProbLog program $\{p_1 : c_1, \dots, p_n : c_n\}$ defines a probability distribution over Prolog programs $L \subseteq \{c_1, \dots, c_n\}$

$$P(L|Program) = \prod_{c_i \in L} p_i \prod_{c_j \notin L} (1 - p_j)$$

- The probability of a goal q :

$$P(q|Program) = \sum_{L \subseteq \{c_1, \dots, c_n\}} \overbrace{P(q|L)} \times P(L|Program)$$

$$\begin{aligned} \underbrace{P(q|L)} &= 1 \text{ if } L \models q \\ &= 0 \text{ otherwise} \end{aligned}$$



ProbLog inference

- Given a ProbLog program T and a query q , $P(q|T)$ gives the probability that a proof exists for q in T
- Obvious application to graphs
 - $P(\text{path}(s,t) | T)$ is the probability that nodes s and t are connected in graph T
- A generalization of network reliability
- How to compute $P(q|T)$?
 - [De Raedt, Kimmig, Toivonen, IJCAI 07]



Compression of ProbLog programs

- A generalization of the most reliable subgraph problem to ProbLog
- Given
 - ProbLog program T
 - positive and negative example queries Pos and Neg
 - constant k
- find the program $T' \subseteq T$ of size at most k that maximizes

$$Likelihood(T'|Pos, Neg) = \prod_{p \in Pos} P(p|T') \prod_{n \in Neg} (1 - P(n|T'))$$

- [De Raedt et al., Machine Learning, 2008]



Conclusions

- Search in large probabilistic graphs
- Probabilistic definitions of search objectives
 - Input: a set of nodes; output: a subgraph
 - Maximise reliability of output, minimize its size
- Grammars, ProbLog: also qualitative criteria
- Computationally non-trivial solutions (skipped)

- Potential for discoveries/hypothesis generation

- An experimental search engine is available at **<http://biomine.cs.helsinki.fi>**



Thanks

- Biomine (Helsinki): Petteri Sevon, Lauri Eronen, Petteri Hintsanen, Kimmo Kulovesi, Laura Langohr
- Problog (Leuven and Freiburg): Luc De Raedt, Angelika Kimmig, Kristian Kersting, Kate Revorado



Publications

- **Link discovery in graphs derived from biological databases**, Petteri Sevon, Lauri Eronen, Petteri Hintsanen, Kimmo Kulovesi, Hannu Toivonen. *Data Integration in the Life Sciences (DILS) 2006*.
- **ProbLog: A Probabilistic Prolog and its Application in Link Discovery**, Luc De Raedt, Angelika Kimmig, Hannu Toivonen. *Twentieth International Joint Conference on Artificial Intelligence (IJCAI) 2007*.
- **Probabilistic Explanation Based Learning**, Luc De Raedt, Angelika Kimmig, Hannu Toivonen. *18th European Conference on Machine Learning (ECML) 2007*
- **The Most Reliable Subgraph Problem**, Petteri Hintsanen. *11th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD) 2007*.
- **Compressing Probabilistic Prolog Programs**, Luc De Raedt, Kristian Kersting, Angelika Kimmig, Kate Revoredo, Hannu Toivonen. *Machine Learning 2008*.
- **Finding Reliable Subgraphs from Large Probabilistic Graphs**, Petteri Hintsanen and Hannu Toivonen, *Data Mining and Knowledge Discovery (Special issue on PKDD) 2008*