



<sup>1</sup> Biomedical and  
Health Informatics



<sup>2</sup> Computer Science  
and Engineering

University of Washington

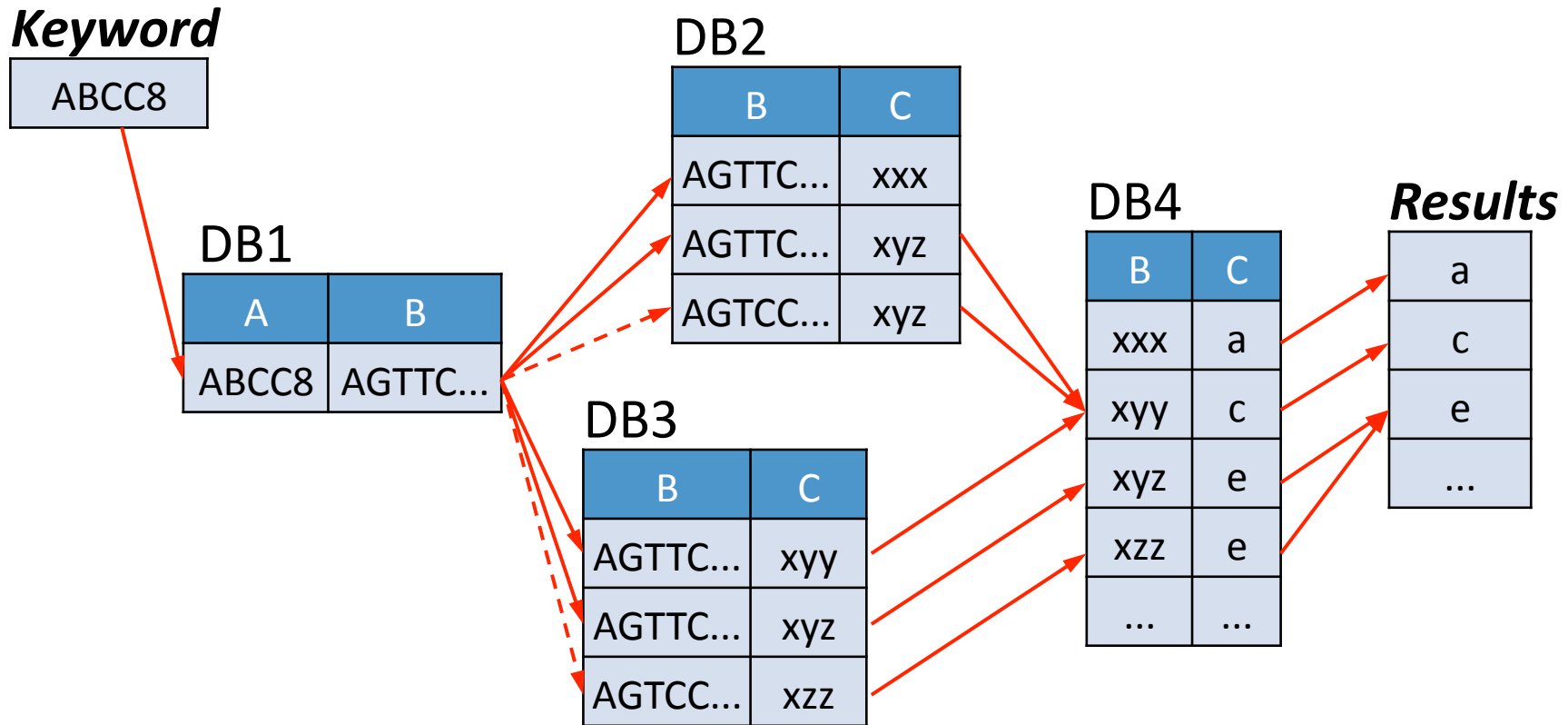
Jan 19, 2010

# Integrating and Ranking Uncertain Scientific Data

**Wolfgang Gatterbauer<sup>2</sup>**

Based on joint work with: Todd Detwiler<sup>1</sup>, Abhay Jha<sup>2</sup>,  
Brent Louie<sup>1</sup>, Dan Suciu<sup>2</sup>, and Peter Tarczy-Hornoch<sup>1</sup>

# Motivation: Retrieving relevant infos across several DBs



**Problem:** multiple expansions across different databases can quickly lead to many less relevant results.

**Question:** how can prune or rank those results?

# Agenda

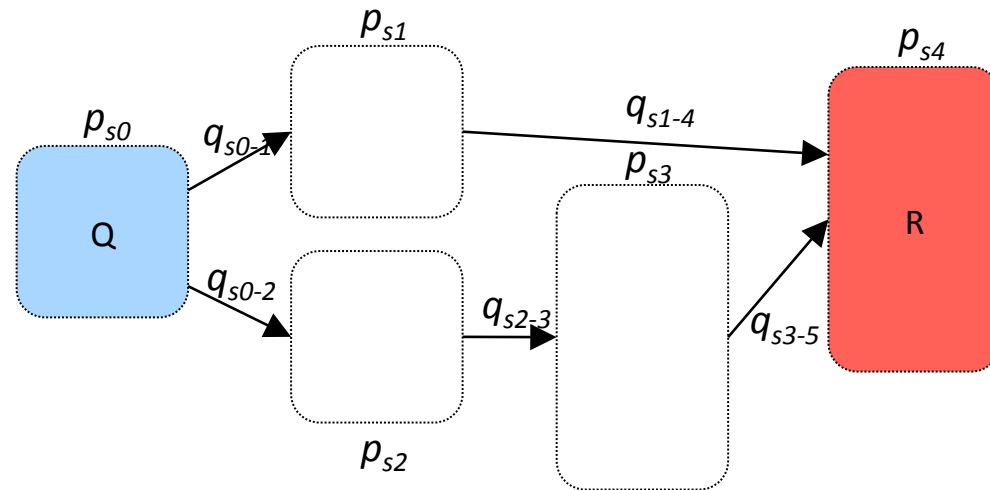
- How to model uncertainty in data integration?
- How do we rank?
- How well, how fast, how robust on real data?
- A short database research point of view

## 4 Probabilistic Metrics in UII

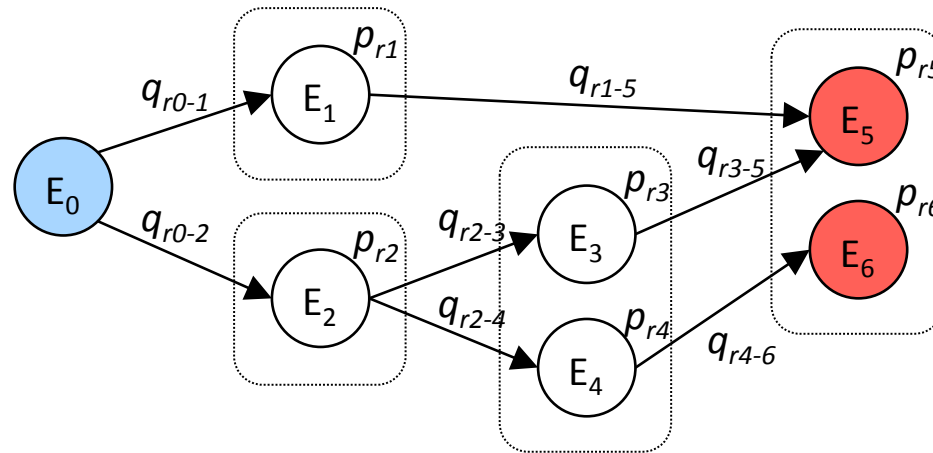
		<i>Granularity</i>	
		Schema	Instance
<i>E/R</i>	Entity	$p_s$	$p_r$
	Relationship	$q_s$	$q_r$

# Example Belief Metrics

## Schema graph



## Instance graph



## Final Scores

$$p = p_s p_r \quad q = q_s q_r$$

# Translation of Uncertainties into Probabilistic Weights

We use [domain experts](#) to quantify and transform data uncertainties into the 4 types of probabilistic weights

Example transformations:

Status code	$p_r$
Reviewed	1.0
Validated	0.8
Provisional	0.7
Predicted	0.4
Model	0.3
Inferred	0.2

Evidence code	$p_r$
IDA / TAS	1.0
IGI / IMP / IPI	0.9
IEP / ISS / RCA	0.7
IC	0.6
NAS	0.5
IEA	0.3
ND / NR	0.2

$$q_r = -\frac{1}{300} \log(\text{E-value})$$



# ... that allows ranking?

Name	Type	Database	▲ UII	Ps	Pr	Loaded
breast cancer 1, early onset	Gene	EntrezGene	1.0	1.0	1.0	<input checked="" type="checkbox"/>
breast and ovarian cancer suscep...	Gene	EntrezGene	1.0	1.0	1.0	<input checked="" type="checkbox"/>
breast cancer 1, early onset	Gene	EntrezGene	1.0	1.0	1.0	<input checked="" type="checkbox"/>
breast cancer 1	Gene	EntrezGene	1.0	1.0	1.0	<input checked="" type="checkbox"/>
Query1	UserQuery	User	1.0	1.0	1.0	<input checked="" type="checkbox"/>
evidence: TAS	ClassificationEvidence	EntrezGene	0.9213	1.0	1.0	<input checked="" type="checkbox"/>
protein binding	Classification	EntrezGene	0.9213	1.0	1.0	<input checked="" type="checkbox"/>
evidence: TAS	ClassificationEvidence	EntrezGene	0.9123	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IDA	ClassificationEvidence	EntrezGene	0.9123	1.0	1.0	<input checked="" type="checkbox"/>
damaged DNA binding	Classification	EntrezGene	0.9123	1.0	1.0	<input checked="" type="checkbox"/>
nucleus	Classification	EntrezGene	0.8711	1.0	1.0	<input checked="" type="checkbox"/>
dosage compensation, by inactiva...	Classification	EntrezGene	0.8706	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IDA	ClassificationEvidence	EntrezGene	0.8706	1.0	1.0	<input checked="" type="checkbox"/>
negative regulation of fatty acid b...	Classification	EntrezGene	0.8686	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IDA	ClassificationEvidence	EntrezGene	0.8679	1.0	1.0	<input checked="" type="checkbox"/>
condensed chromosome	Classification	EntrezGene	0.8679	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IDA	ClassificationEvidence	EntrezGene	0.8663	1.0	1.0	<input checked="" type="checkbox"/>
cytoplasm	Classification	EntrezGene	0.8663	1.0	1.0	<input checked="" type="checkbox"/>
evidence: TAS	ClassificationEvidence	EntrezGene	0.8531	1.0	1.0	<input checked="" type="checkbox"/>
DNA repair	Classification	EntrezGene	0.8531	1.0	1.0	<input checked="" type="checkbox"/>
centrosome cycle	Classification	EntrezGene	0.839	1.0	1.0	<input checked="" type="checkbox"/>
DNA replication	Classification	EntrezGene	0.8341	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IGI	ClassificationEvidence	EntrezGene	0.7804	1.0	0.9	<input checked="" type="checkbox"/>
zinc ion binding	Classification	EntrezGene	0.7791	1.0	1.0	<input checked="" type="checkbox"/>
evidence: IMP	ClassificationEvidence	EntrezGene	0.7758	1.0	0.9	<input checked="" type="checkbox"/>
DNA damage response, signal tra...	Classification	EntrezGene	0.7758	1.0	1.0	<input checked="" type="checkbox"/>
DNA binding	Classification	EntrezGene	0.7529	1.0	1.0	<input checked="" type="checkbox"/>
carbohydrate metabolic process	Classification	EntrezGene	0.7477	1.0	1.0	<input checked="" type="checkbox"/>
DNA damage response, signal tra...	Classification	EntrezGene	0.7461	1.0	1.0	<input checked="" type="checkbox"/>
regulation of transcription from R...	Classification	EntrezGene	0.7441	1.0	1.0	<input checked="" type="checkbox"/>
regulation of apoptosis	Classification	EntrezGene	0.7431	1.0	1.0	<input checked="" type="checkbox"/>
tubulin binding	Classification	EntrezGene	0.7422	1.0	1.0	<input checked="" type="checkbox"/>

Source: Todd

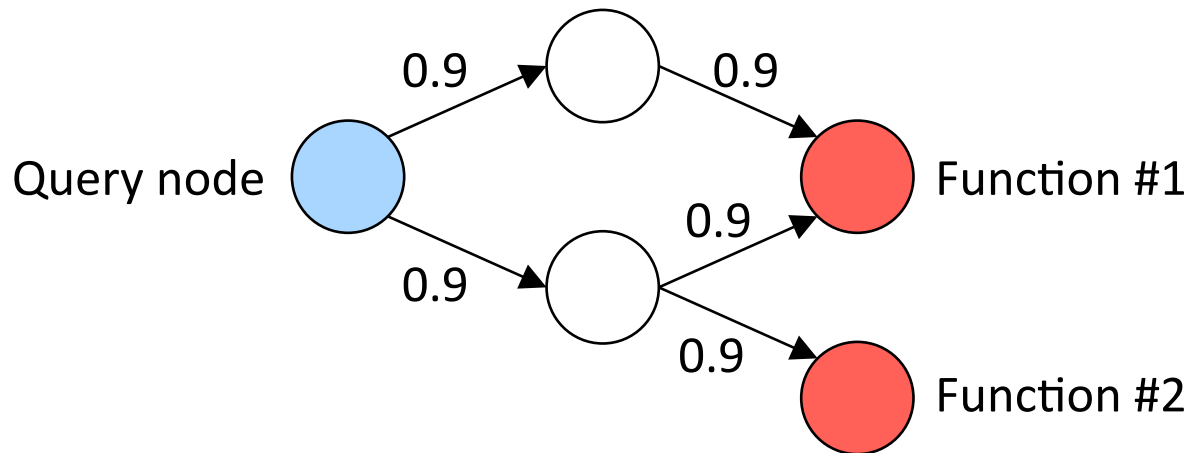


# Agenda

- How to model uncertainty in data integration?
- How do we rank?
- How well, how fast, how robust on real data?
- A short database research point of view

# Network Reliability Theory (“source-target reachability”)

Source-target-reachability: probability that a node is reachable from the start (query) node.

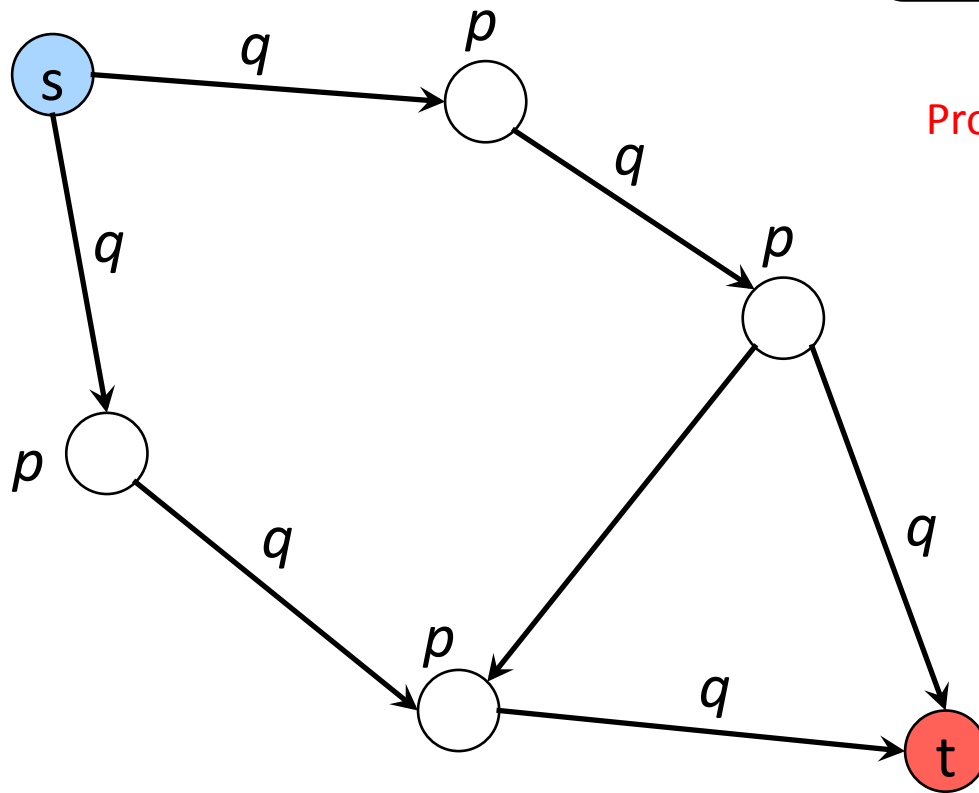


$$\text{Function \#2} = 0.9^2 = \mathbf{0.81}$$

$$\begin{aligned} \text{Function \#1} &= 1 - \text{Prob}(\text{all paths failed}) \\ &= 1 - (1 - 0.9^2)(1 - 0.9^2) \\ &= \mathbf{0.9639} \end{aligned}$$

# Incorporating Uncertainty: Network Reliability Theory

score = probability that an answer node is reachable from the start (query) node.



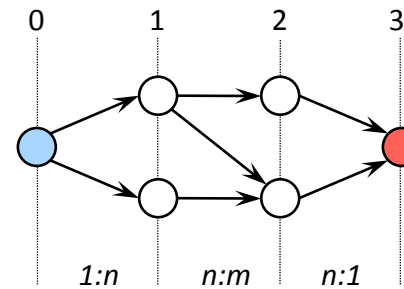
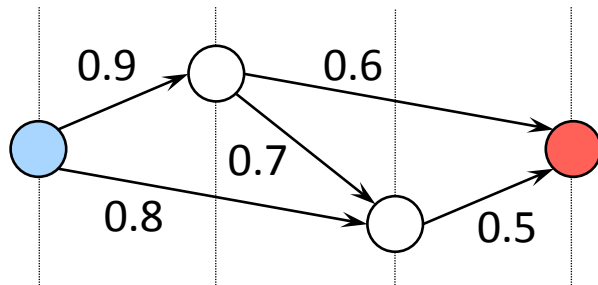
Problem: Computing U2 score is #P.

# Why is reliability = reachability hard?

The following graph is nasty = hard!

Can come in different forms:

## Wheatstone Bridge



Reachability score:

$$0.8 \cdot 0.5 + 0.9 \cdot 0.7 \cdot 0.5 + 0.9 \cdot 0.6$$

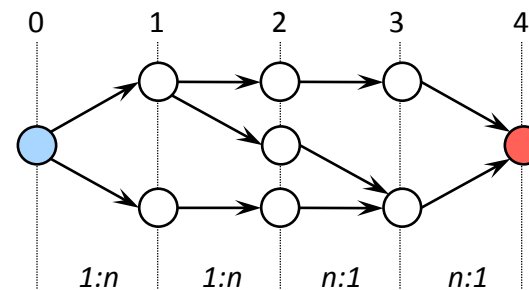
$$- 0.9 \cdot 0.6 \cdot 0.7 \cdot 0.5$$

$$- 0.9 \cdot 0.6 \cdot 0.8 \cdot 0.5$$

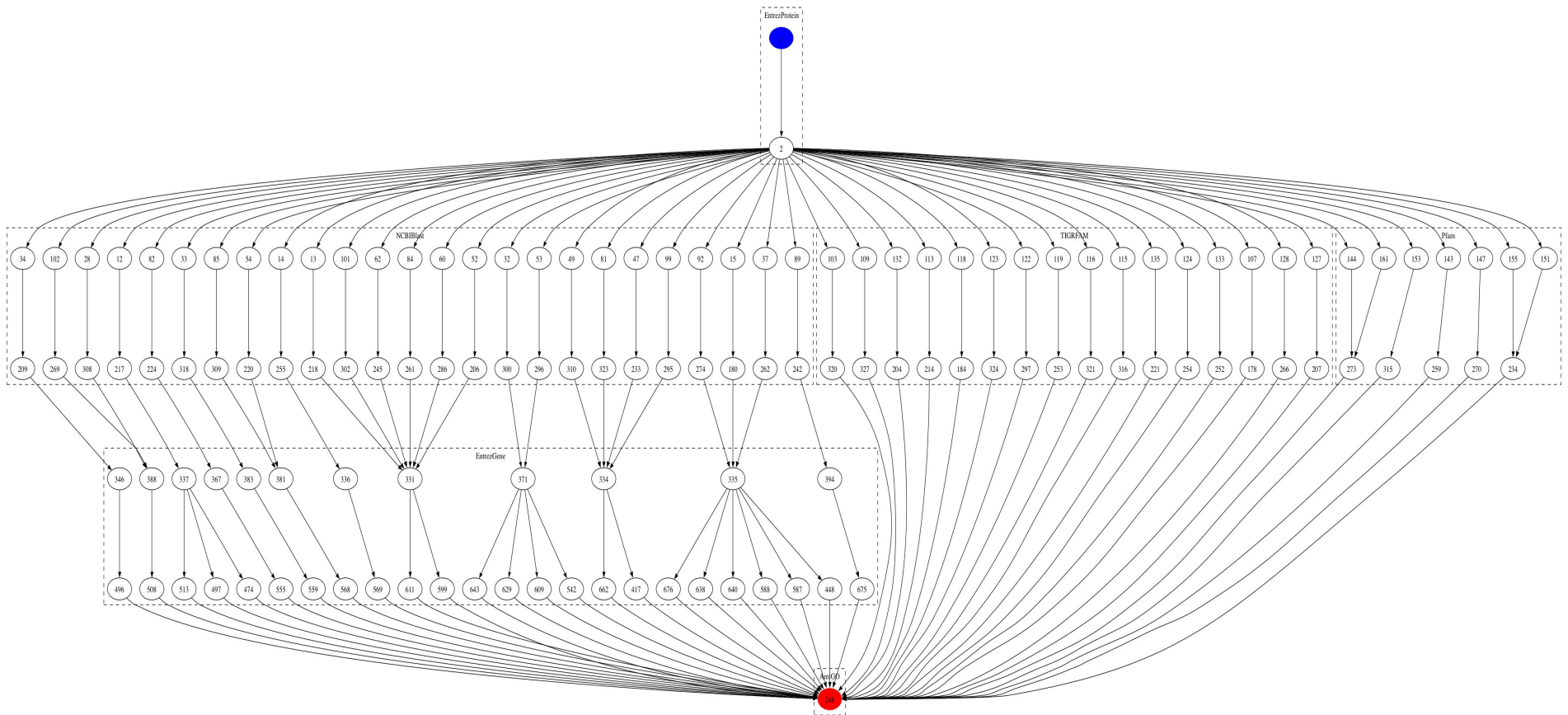
$$- 0.9 \cdot 0.7 \cdot 0.5 \cdot 0.8$$

$$+ 0.9 \cdot 0.6 \cdot 0.7 \cdot 0.5 \cdot 0.8$$

$$= \mathbf{0.7492}$$



# Closed solution is possible sometimes



# Techniques to perform probabilistic scoring

Naive Monte Carlo simulation

Improved Monte Carlo simulation

Analyze the necessary number of simulations

Graph reductions (Parallel-serial reductions)

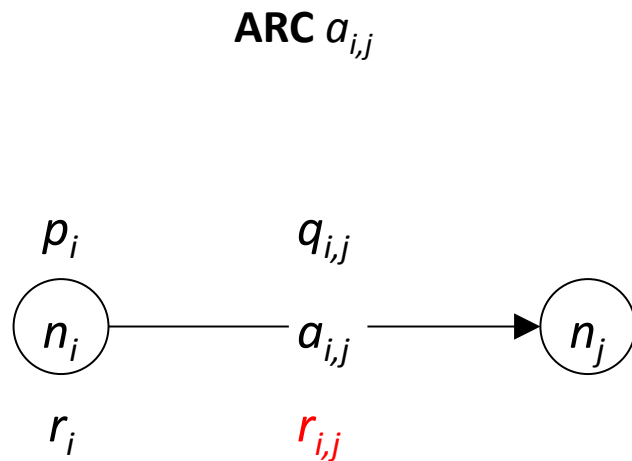
Closed solution for subgraphs

## **Propagation score**

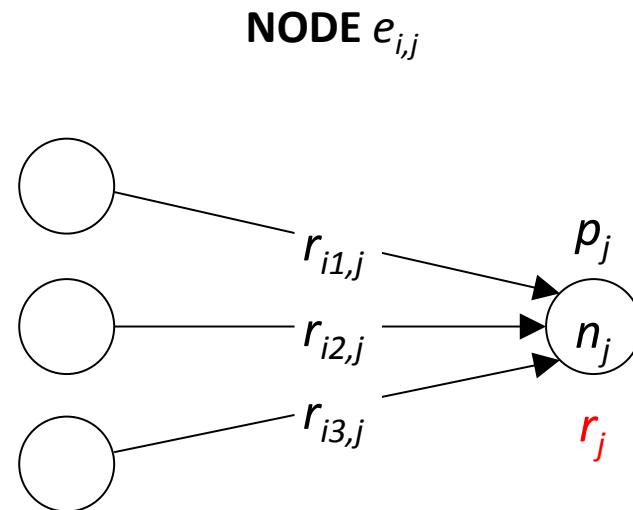
Deterministic counterparts

# Ignoring correlations: the „relevance propagation“ model

- Ignoring correlations leads to a local point of view.
- One equation for relevance  $r$  for each node  $n_i$  and each arc  $a_{i,j}$
- Solve simple equation system (closed or iteratively)



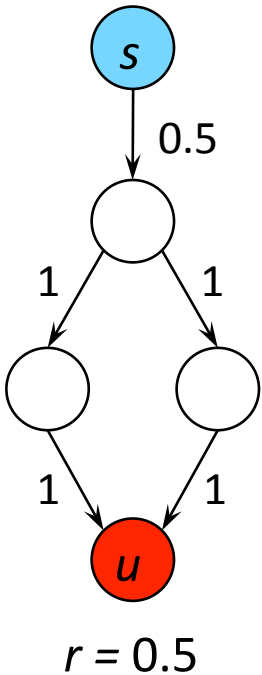
$$r_{i,j} = r_i \cdot q_{i,j}$$



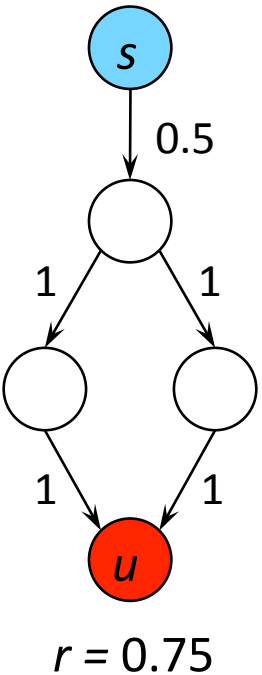
$$r_j = \left(1 - \prod_i (1 - r_{i,j})\right) \cdot p_j$$

# Example: reliability vs. propagation

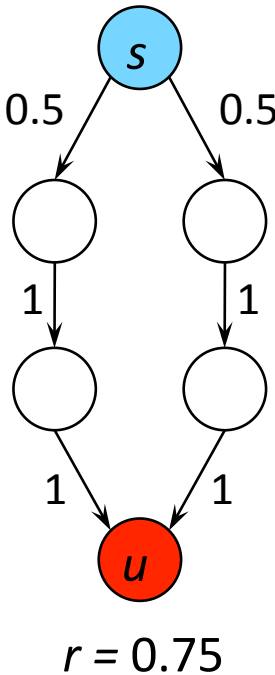
Reliability



Propagation



Reliability = Propagation





# Comparing reliability and propagation: complexity

## Reliability

- global measure
- combinatorial problem
- $P\# = \text{hard}$
- Monte Carlo estimates

## Propagation

- local measure
- continuous state space
- $P = \text{not hard}$
- Iterative algorithm

# Agenda

- How to model uncertainty in data integration?
- How do we rank?
- How well, how fast, how robust on real data?
- A short database research point of view

# Experiments: Functional gene annotation

## 3 questions

- 1) How well do different approaches perform? [Average precision (AP)]
- 2) How fast is probabilistic query evaluation? [Focus on reliability]
- 3) Where do you get the probabilities from? → How robust is our system to variations in the input probabilities? [Sensitivity analysis]

## 6 data sources:

Pfam, TIGRFAM, NCBIblast, EntrezProtein, EntrezGen, AmiGo

## 3 scenarios

- 1) Well-known functions for well-studied proteins (306/20)
- 2) Less-known functions for well-studied proteins (7/3)
- 3) Unknown functions for less-studied proteins (11/11)

# 1. How well (1/3): Average Precision

Assume 4 out of 10 items are “relevant”

<u>Rank</u>	<u>Ranking method 1</u>		<u>Ranking method 2</u>		<u>Random AP</u>
	<u>relevant</u>	<u>precision@k</u>	<u>relevant</u>	<u>precision@k</u>	
1	x	1.00 (=1/1)	x	1.00 (=1/1)	<i>Averaged over all <math>\binom{10}{4}</math> permutations</i>
2	x	1.00 (=2/2)			
3			x	0.67 (=2/3)	
4	x	0.75 (=3/4)	x	0.75 (=3/4)	
5					
6					
7	x	0.57 (=4/7)			
8			x	0.50 (=4/8)	
9					
10					
<b>AP</b>		<b>0.83</b>		<b>0.73</b>	<b>0.53</b>

AP as measure for the quality of the ranking semantics with regard to “ground truth”

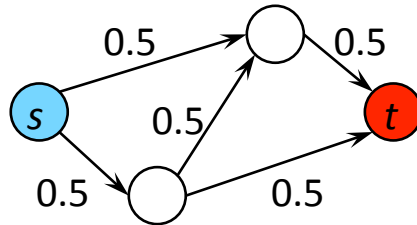
# 1. How well (2/3): Scoring functions

## Scoring function

## Example graph

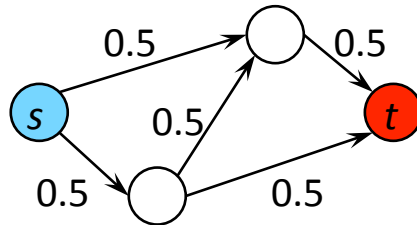
## Example score

Reliability



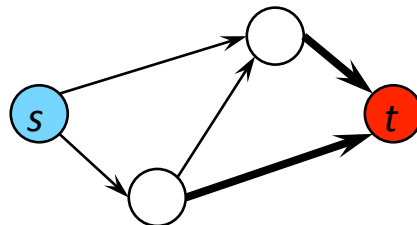
0.469

Propagation



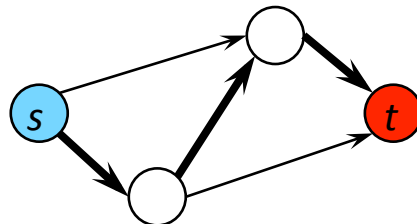
0.484

InEdge



2 incoming edges

PathCount



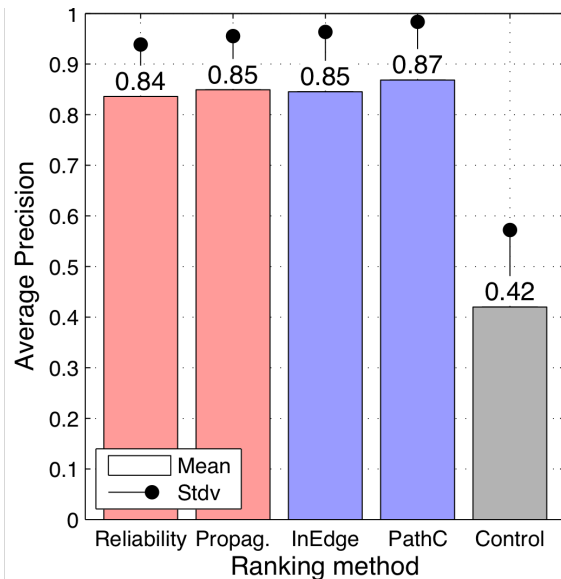
3 paths (1 shown)

Random AP

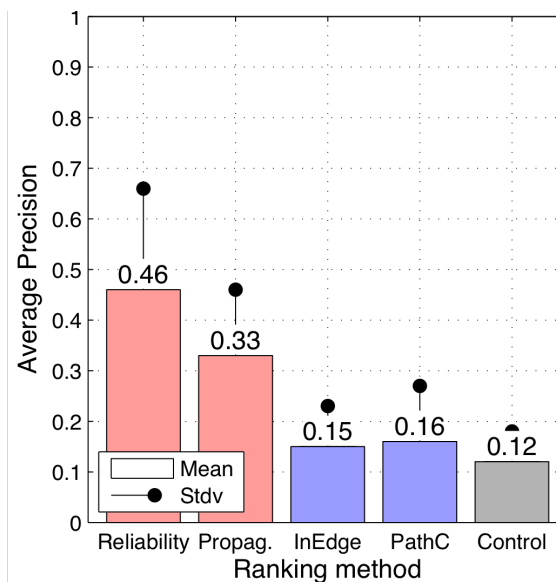
*no score: AP averaged over all ranking permutations*

# 1. How well (3/3): AP across 3 scenarios

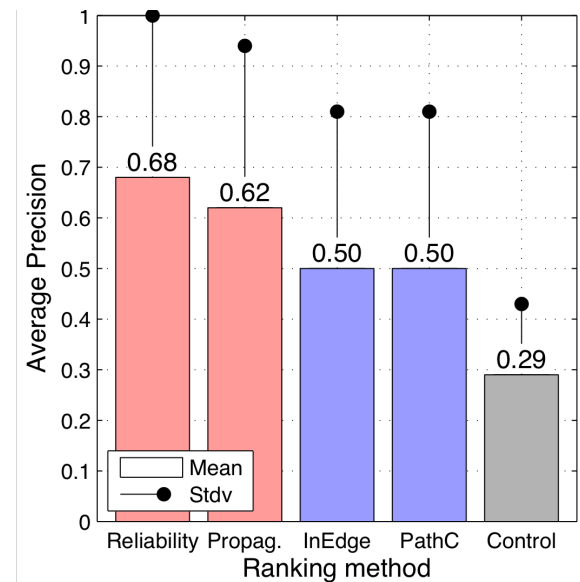
Scenario 1:  
306 well-known function,  
20 well-studied proteins



Scenario 2  
7 less-known functions,  
3 well-studied proteins



Scenario 1:  
11 unknown functions,  
11 less-studied proteins

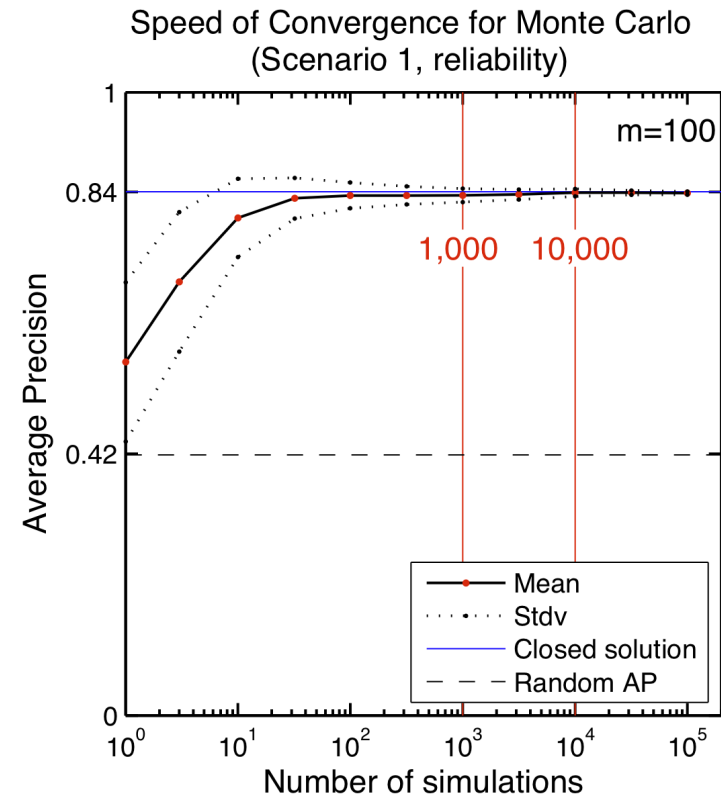
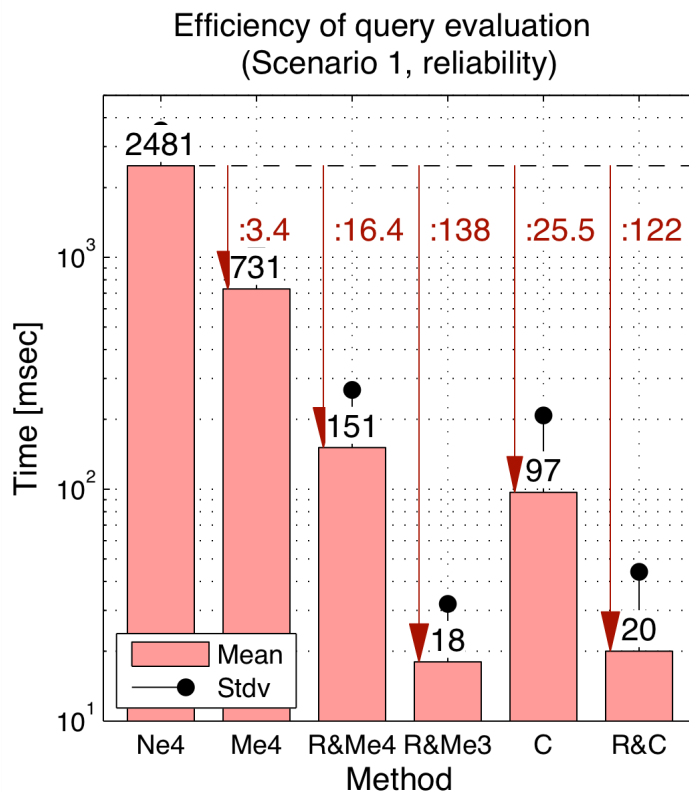


**Observation 1: Probabilistic methods perform better for predicting less-known or previously unknown functions!**

## 2. How fast: Several techniques for speeding up reliability

Techniques (not discussed in detail):

naive Monte Carlo (**N**), efficient Monte Carlo (**M**), 1.000 instead of 10.000 simulations (**e4**, **e5**), graph reductions (**R**), closed solution (**C**)



Observation 2: Several techniques allowed us to evaluate the reliability semantics in  $\sim 20$ ms (propagation  $\sim 5$ ms, InEdge and Pathcount  $\sim 1$ ms)

### 3. How robust: sensitivity analysis

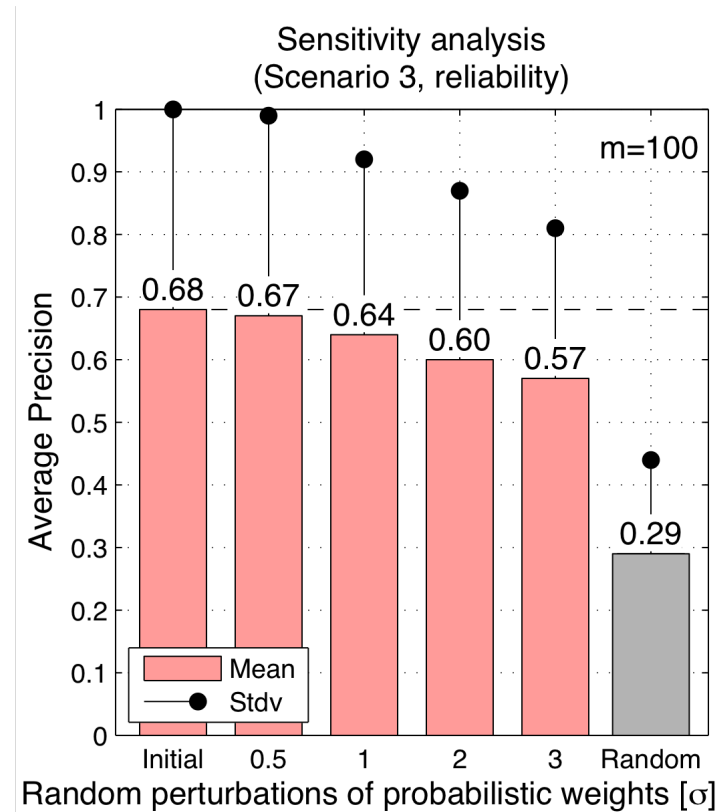
*Our approach depends on transforming uncertainty into probabilistic weights. How robust is the performance to systematic variations in these input parameters?*

*Idea: multi-way sensitivity analysis*

$$p' = \text{Lo}^{-1}(\text{Lo}(p) + \varepsilon)$$

$$\varepsilon = \text{N}(0, \sigma^2)$$

$$\text{Lo}(p) = \log\left(\frac{p}{1-p}\right)$$

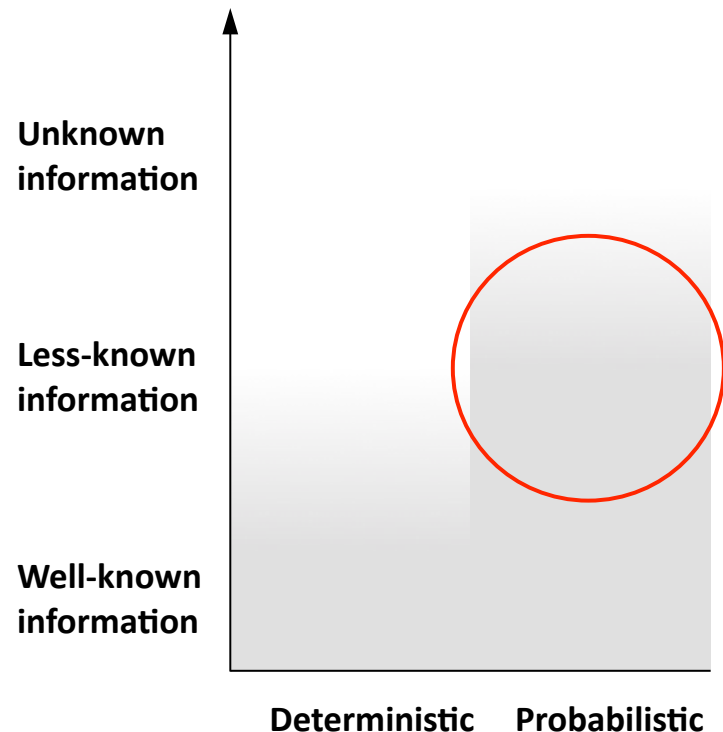


**Observation 3: Small random perturbations to the initial parameters do not negatively affect the quality of rankings. The approach is robust!**



# Take-way from experiments on real data

## Uncertainty of information



## Information integration approach

- Explicit modeling of uncertainties as probabilities increases our ability to predict less-known or previously unknown protein functions. This suggests that uncertainty models offer **utility for knowledge discovery**.
- Small perturbations in the input probabilities (parameters) tend to produce only minor changes in the quality of our result rankings. This suggests that probabilistic methods are **robust against variations in the way uncertainties are transformed into probabilities**.
- Several techniques allow us to evaluate probabilistic rankings efficiently. This suggests that **probabilistic query evaluation is not as hard** for real-world problems as theory indicates.

# Agenda

- How to model uncertainty in data integration?
- How do we rank?
- How well, how fast, how robust on real data?
- **A short database research point of view**

## Short database background (1/2)

### Schema

ATTEND(student, class)

TEACH(class, prof)

DEP(prof, department)

### SQL query

```
select  A.student, T.department
from    ATTEND A, TEACH T, DEP D
where   ATTEND.class=TEACH.class
and     TEACH.prof=DEP.prof
```

## Short database background (2/2)

### Schema

R(A, B)

S(B, C)

T(C, D)

### SQL query

```
select  R.A, T.D
from    R, S, T
where   R.B=S.B
and     S.C=T.C
```

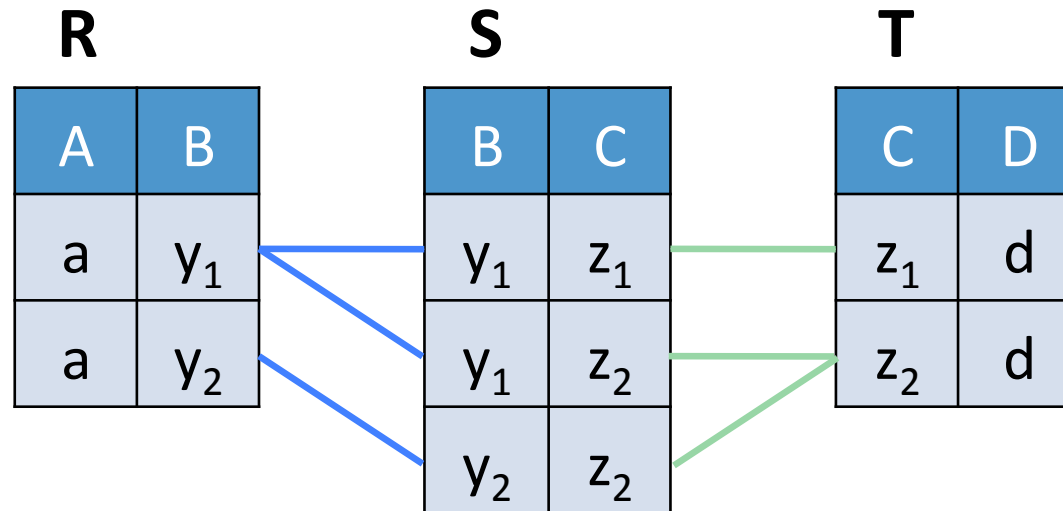
### Datalog

$q(x, u) :- R(x, y), S(y, z), T(z, u)$

Conjunctive queries: very efficient!

## Probabilistic databases (1/3)

$$q(x, u) :- R(x, y), S(y, z), T(z, u)$$

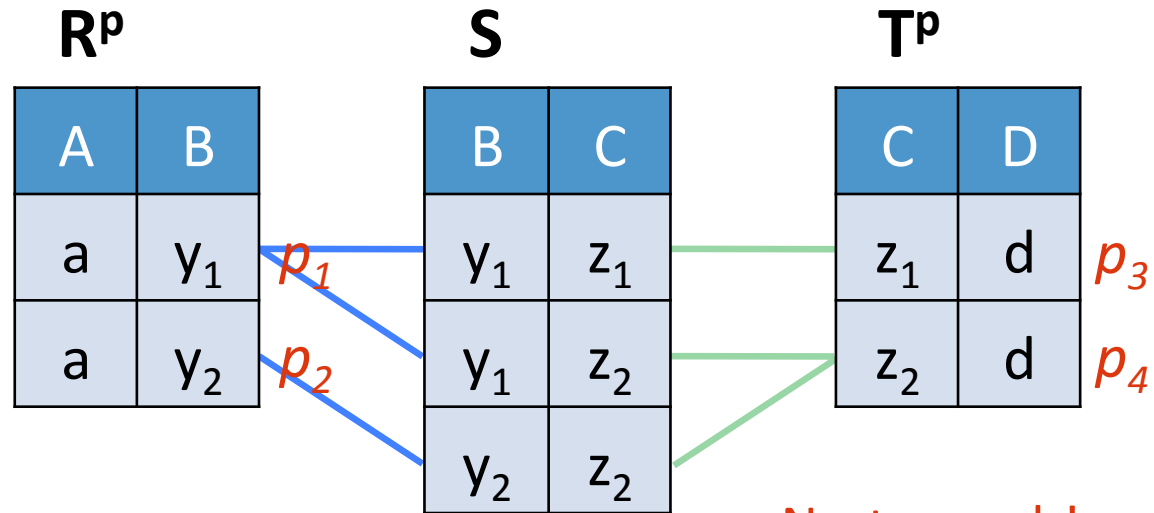


Which tuples?

$q(a, d)$

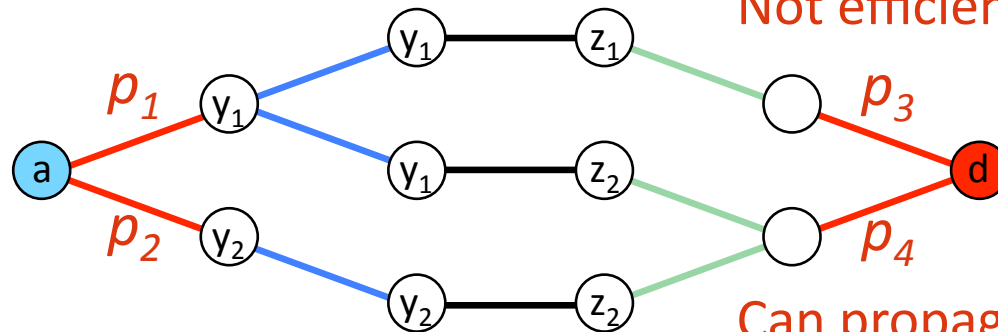
# Probabilistic databases (2/3)

$$q(x, u) : -R^p(x, y), S(y, z), T^p(z, u)$$



Nasty graph!  
Not efficient!

Which tuples  
& how likely?

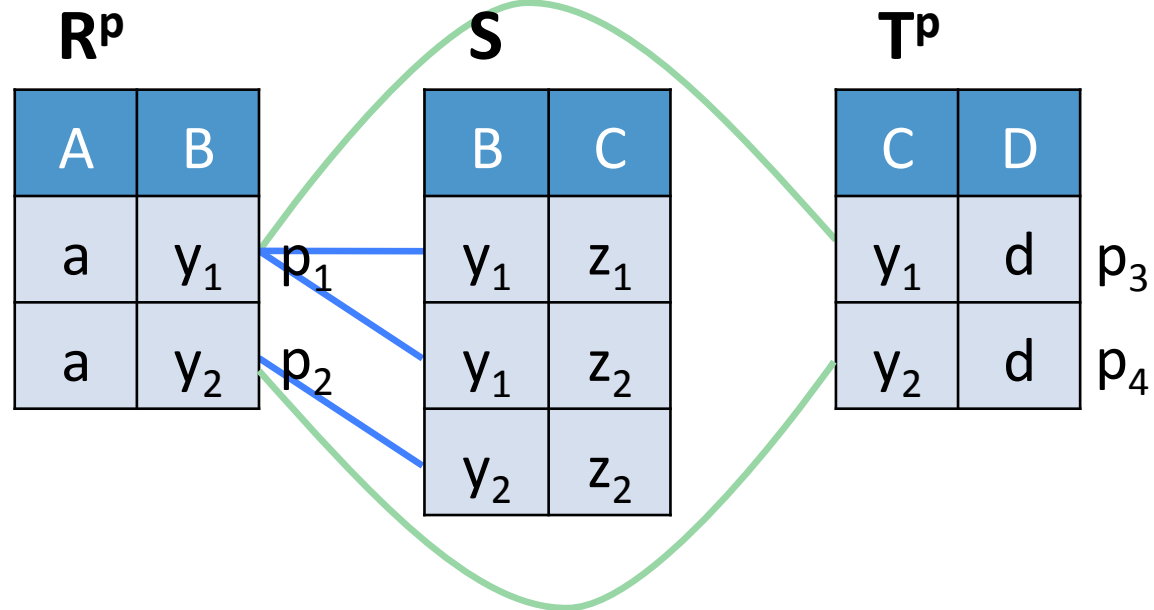


Can propagation help?

$$\mathbf{P}[q(a, d)] = p_1 p_3 \vee p_1 p_4 \vee p_2 p_4 = \text{reachability } a \rightarrow d$$

## Probabilistic databases (3/3)

$$q(x, u) : -R^p(x, y), S(y, z), T^p(y, u)$$



$$q(x, y) : -R^p(x, y), R^p(x, z), T^p(z, u)$$

Non-linear “chain queries” / self joins.

How to define a propagation semantics?

# Which ranking semantics is appropriate for real data?

Input (probabilistic) data     $\xrightarrow{?}$     Output ranked results

$R^P$

A	B
a	a
a	e
b	c
d	c
d	a
e	a
e	c
e	d

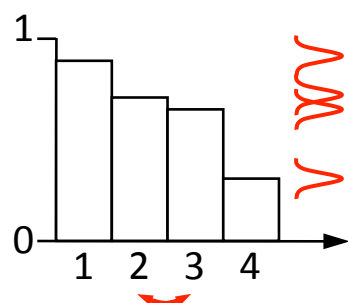
Possible world semantics  
 $\sim$  reliability

$p_1$   $\wedge$   
 $p_2$   $\wedge$   
 $p_3$   $\wedge$   
 $p_4$   $\wedge$   
 $p_5$   $\wedge$   
 $p_6$   $\wedge$   
 $p_7$   $\wedge$   
 $p_8$   $\wedge$

4. Hidden dependencies in the input data in the first place

Alternative ranking semantics  
 $\sim$  propagation

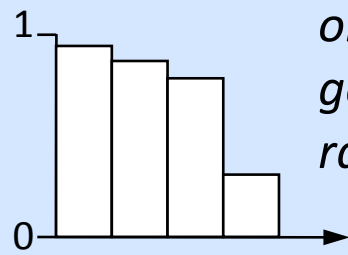
Can we get good ranking results for arbitrary queries on real data or at least a good trade-off speed / ranking accuracy?



1. Hard in general

2. Sensitivity of ranking with respect to accuracy of input probability

3. Decrease in ranking quality due to approximation





## Further information

### PAPER

L. Detwiler, W. Gatterbauer, B. Louie, D. Suciu and P. Tarczy-Hornoch.  
[Integrating and Ranking Uncertain Scientific Data](#). In Proceedings of the 25th International Conference on Data Engineering, 2009.

### PROJECT WEB PAGE

<http://www.biomediator.org>

### DATABASE RESEARCH GROUP

<http://db.cs.washington.edu/>

### CONTACT

[Wolfgang Gatterbauer: gatter@cs](mailto:gatter@cs)

THANKS!