[1] Biomedical and Health Informatics

[2] Computer Science and Engineering

University of Washington

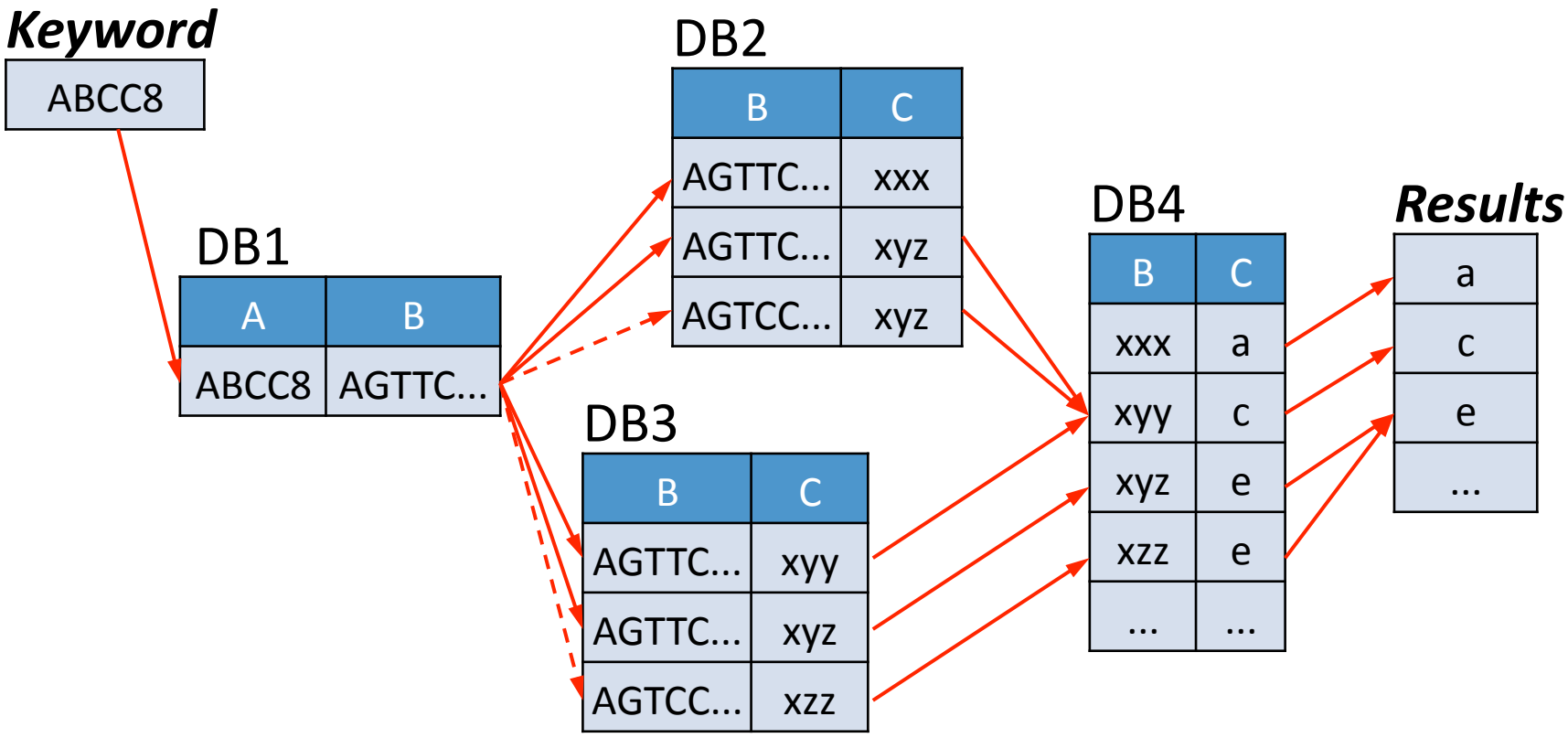
Jan 19, 2010

Integrating and Ranking Uncertain Scientific Data

Wolfgang Gatterbauer[2]

Based on joint work with: Todd Detwiler[1], Abhay Jha[2], Brent Louie[1], Dan Suciu[2], and Peter Tarczy-Hornoch[1]

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

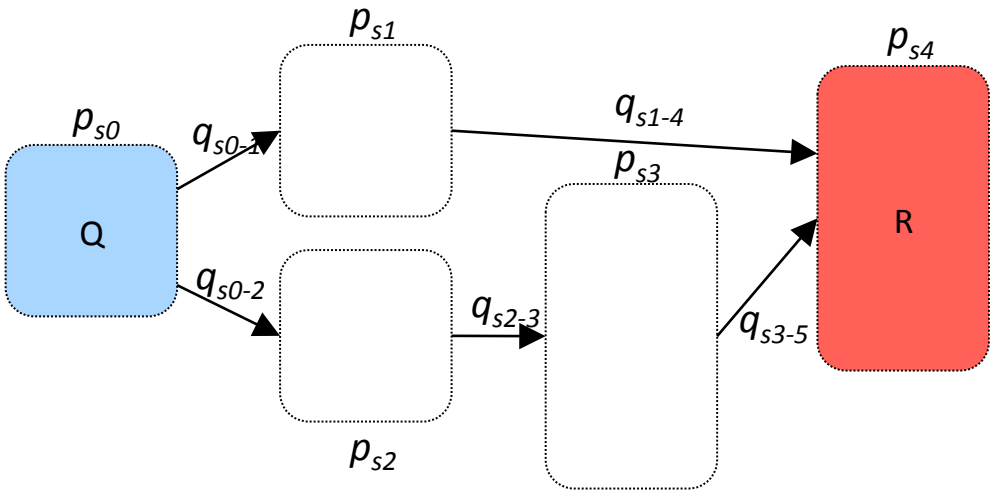
- How to model uncertainty in data integration?
- How do we rank?
- How well, how fast, how robust on real data?
- A short database research point of view

4 Probabilistic Metrics in UII

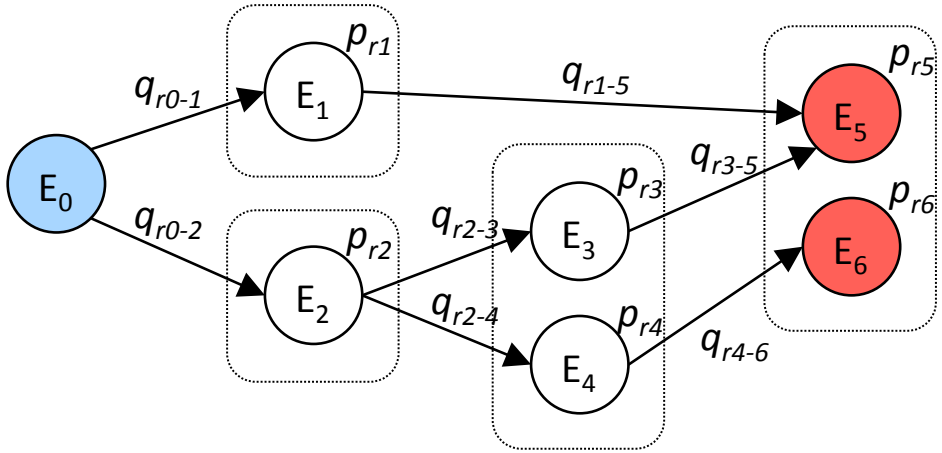
		Granularity	
		Schema	Instance
E/R	Entity	p_s	p_r
	Relationship	q_s	q_r

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

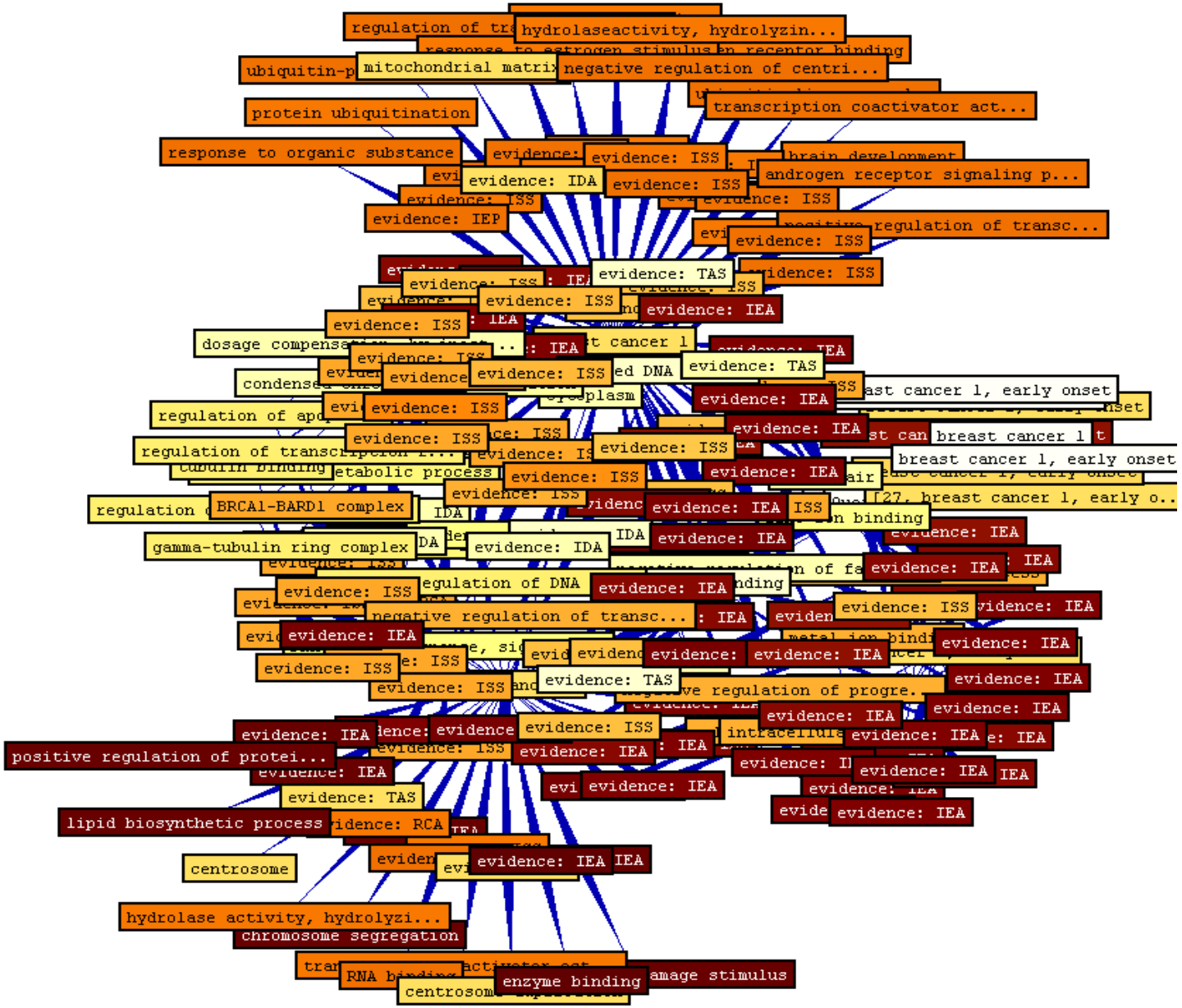
Example transformations:

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

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

How can we assign some score (here the color)...



Source: Todd

... that allows ranking?

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

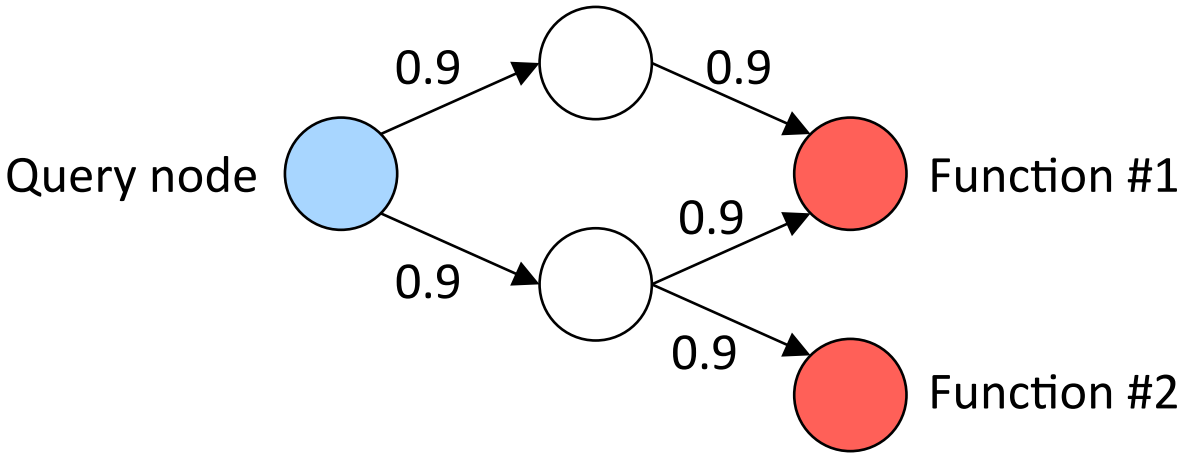
Source: Todd

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.

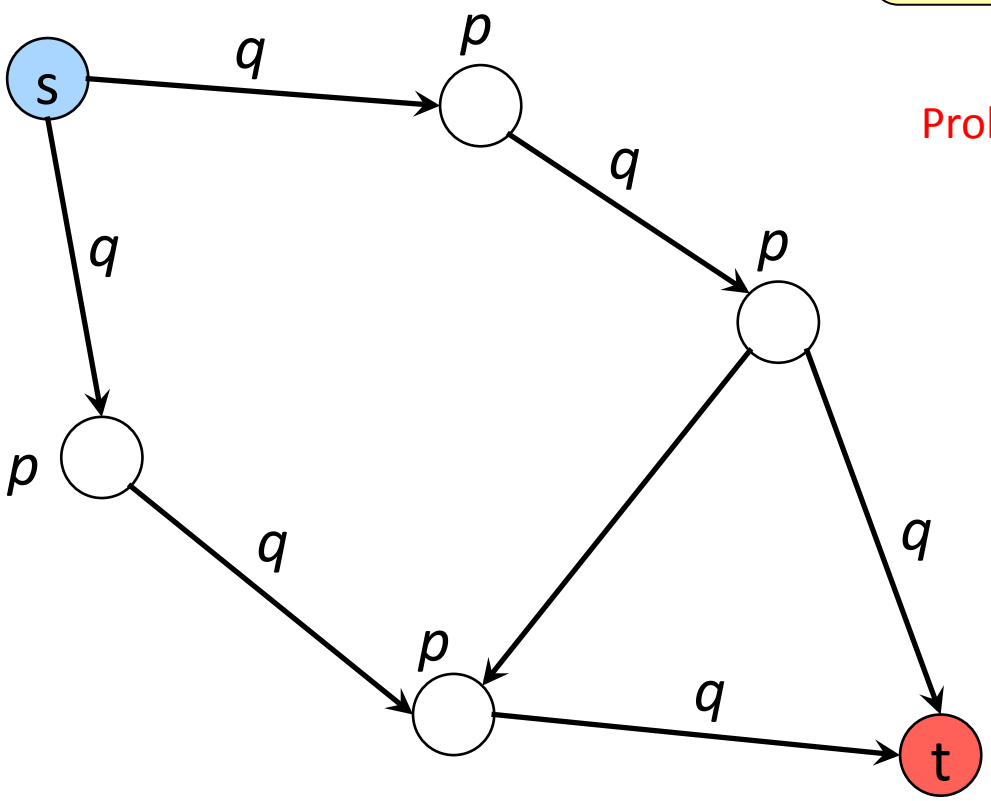


Function #2 = 0.9^2 = **0.81**

Function #1 = 1 – Prob(all paths failed)
= $1 – (1 – 0.9^2)(1 – 0.9^2)$
= **0.9639**

Incorporating Uncertainty: Network Reliability Theory

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



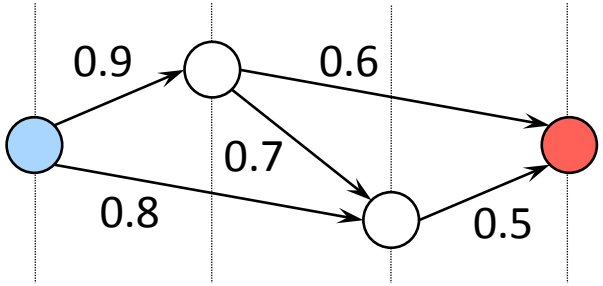
Problem: Computing U2 score is #P.

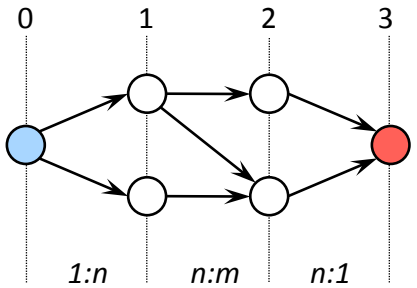
Why is reliability = reachability hard?

The following graph is nasty = hard!

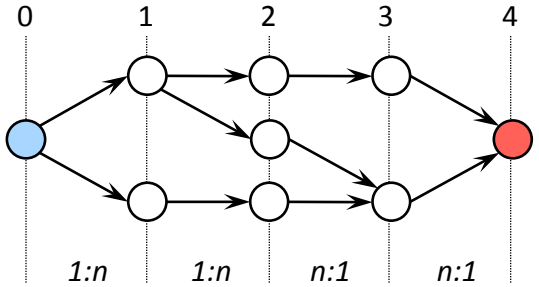
Can come in different forms:

Wheatstone Bridge

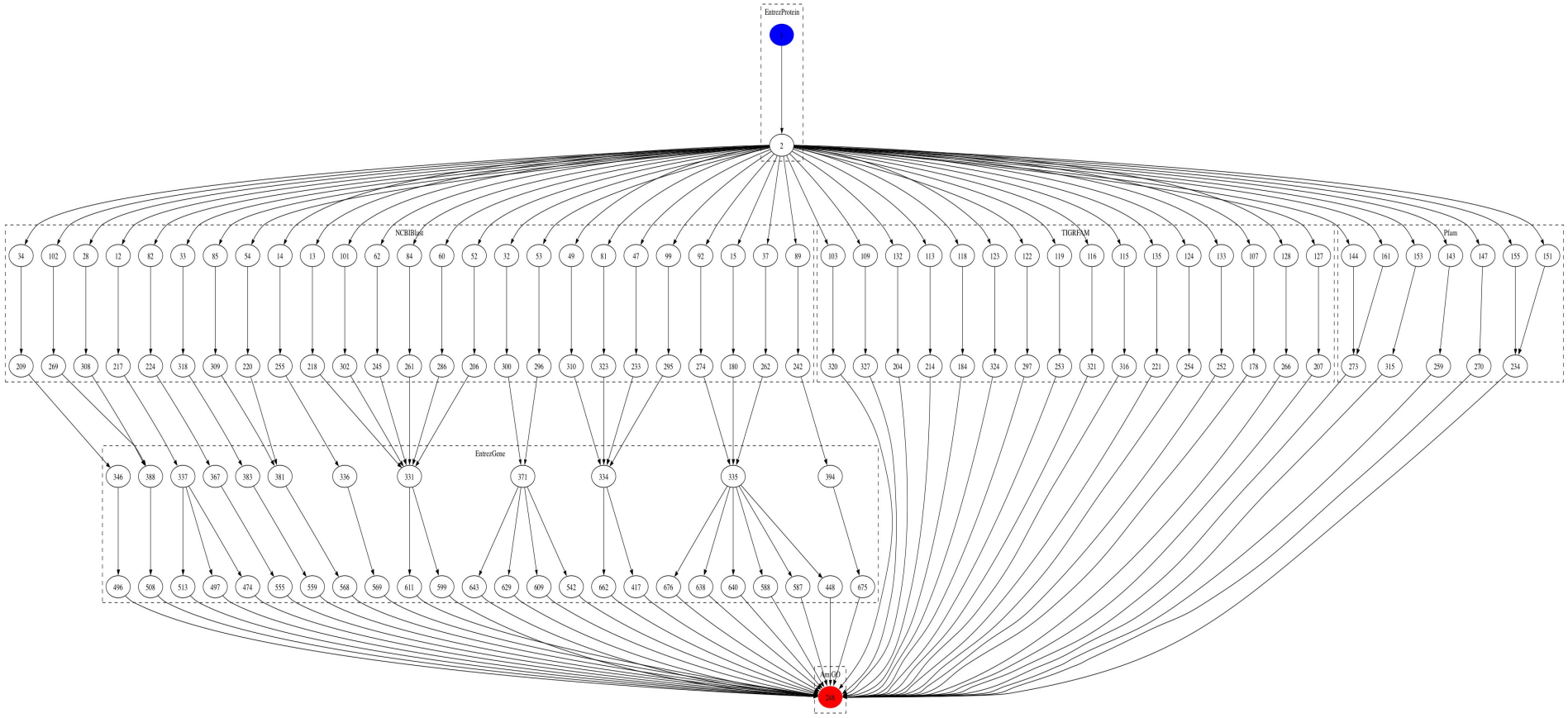




Reachability score:
0.8·0.5 + 0.9·0.7·0.5 + 0.9·0.6
-0.9·0.6·0.7·0.5
-0.9·0.6·0.8·0.5
-0.9·0.7·0.5·0.8
+0.9·0.6·0.7·0.5·0.8
= **0.7492**



Closed solution is possible sometimes



Detail: gene ABCC8, upstream node 248

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

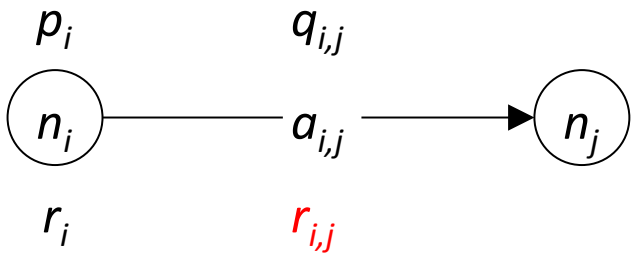
Propagation score

Deterministc counterparts

Ignoring correlations: the „relevance propagation“ model

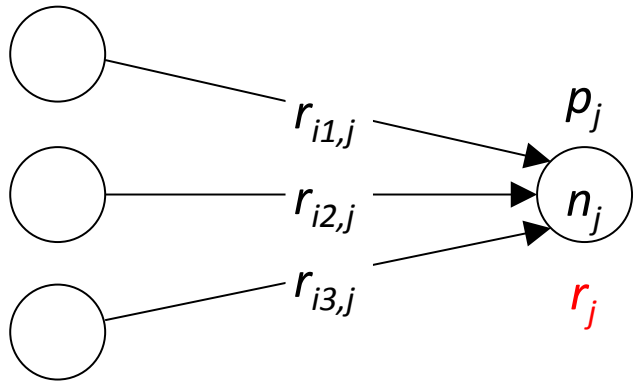
- Ignoring correlations leads to a local point of view.
- One equation for relevance r for each node n_i and each arc $a_{i,j}$
- Solve simple equation system (closed or iteratively)

ARC $a_{i,j}$



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

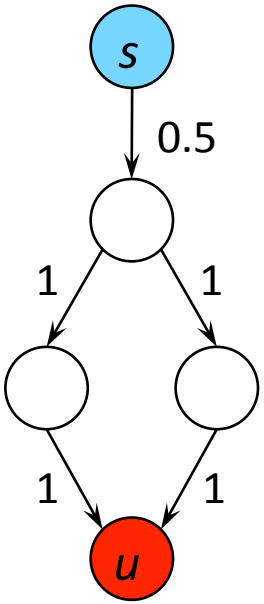
NODE $e_{i,j}$



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

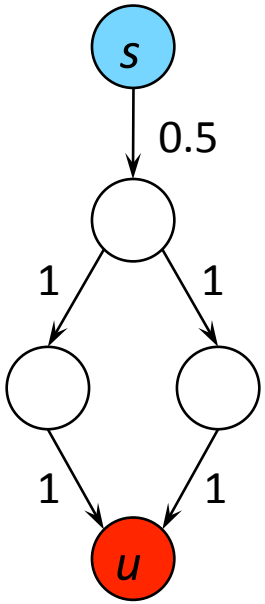
Example: reliability vs. propagation

Reliability



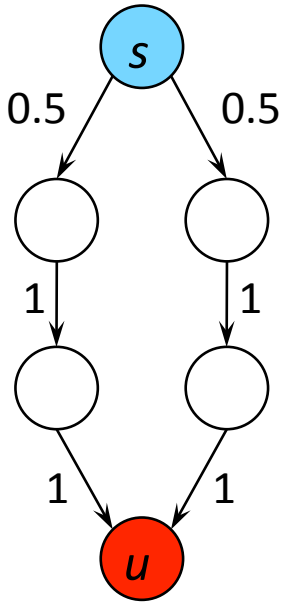
$r = 0.5$

Propagation



$r = 0.75$

Reliability = Propagation



$r = 0.75$

Comparing reliability and propagation: complexity

Reliability	Propagation
• global measure	• local measure
• combinatorial problem	• continuous state space
• P# = hard	• P = not hard
• Mone Carlo estimates	• 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)

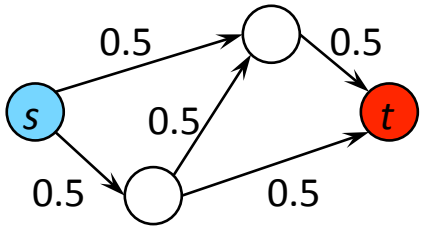
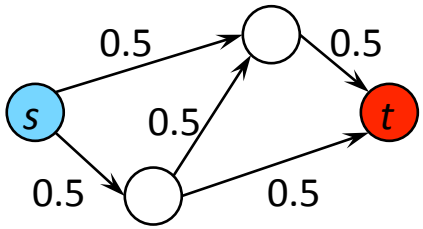
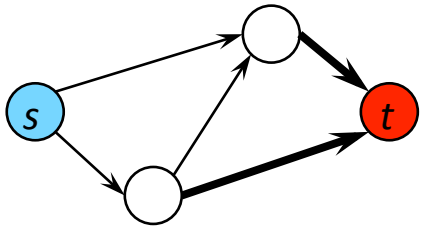
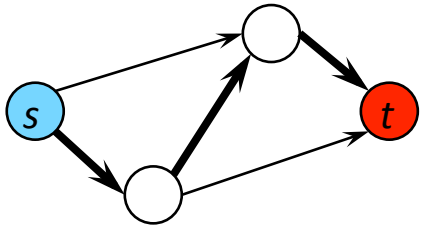
1. How well (1/3): Average Precision

Assume 4 out of 10 items are "relevant"

	Ranking method 1		Ranking method 2		
Rank	relevant	precision@k	relevant	precision@k	**Random AP**
1	x	1.00 (=1/1)	x	1.00 (=1/1)	*Averaged*
2	x	1.00 (=2/2)			*over all* $\binom{10}{4}$
3			x	0.67 (=2/3)	*permutations*
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					
AP		**0.83**		**0.73**	**0.53**

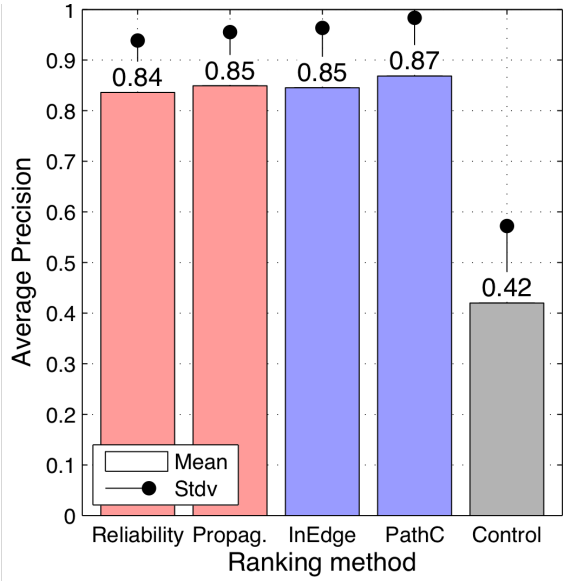
AP as measure for the quality of the ranking semantics with regard to "ground truth"

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

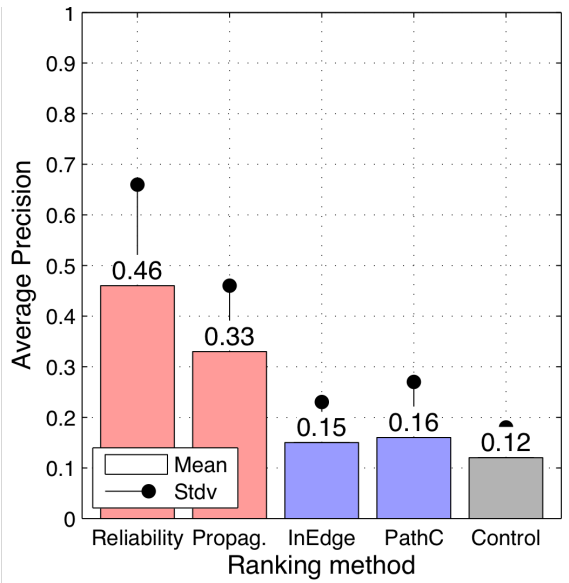
Scoring function	Example graph	Example score
Reliability		0.469
Propagation		0.484
InEdge		2 incoming edges
PathCount		3 paths *(1 shown)*
Random AP		*no score: AP averaged over all ranking permutations*

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

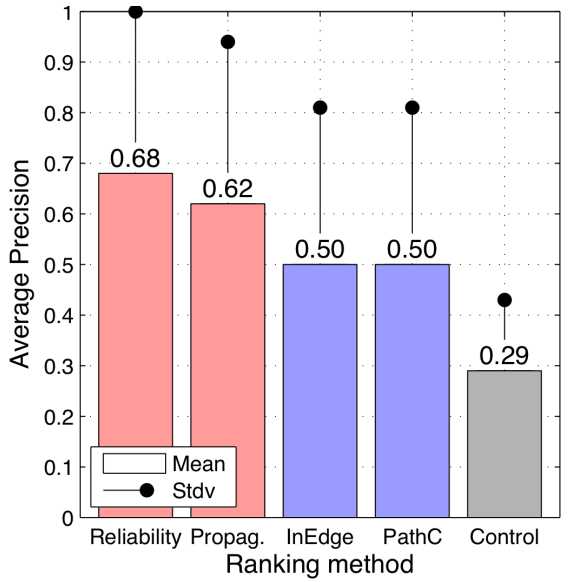
Scenario 1:
306 well-known function,
20 well-studied proteins



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



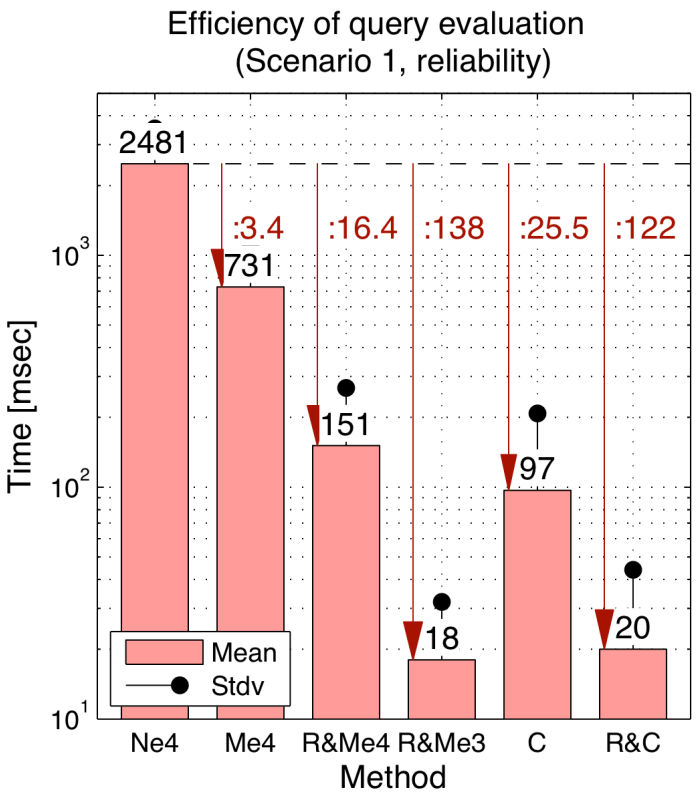
Scenario 1:
11 unknown functions,
11 less-studied proteins

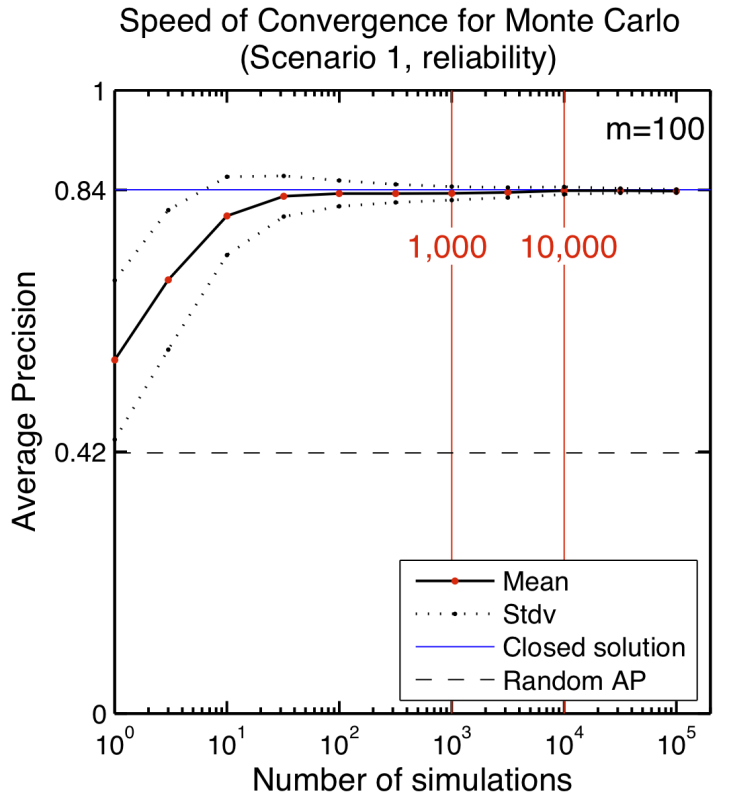


Observation 1: Probabilistic methods perform better for predicting less-known or previously unknown functions!

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 ~20ms (propagation ~5ms, InEdge and Pathcount ~1ms)

3. How robust: sensitivity analysis

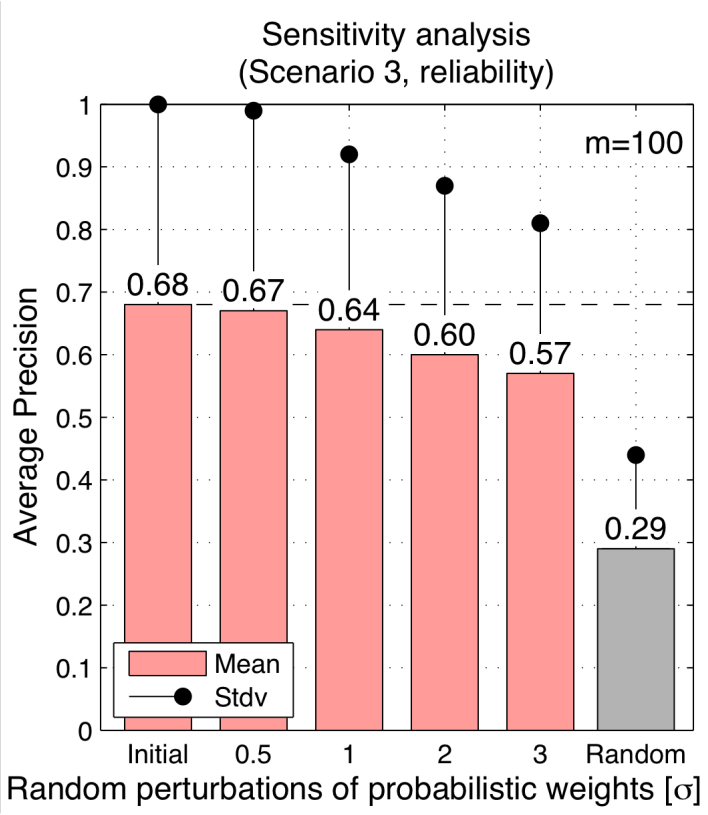
Our approach depends on transforming uncertainty into probabilistic weights. How robust is the performance to systematic variations in these input parameters?

Idea: multi-way sensitivity analysis

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

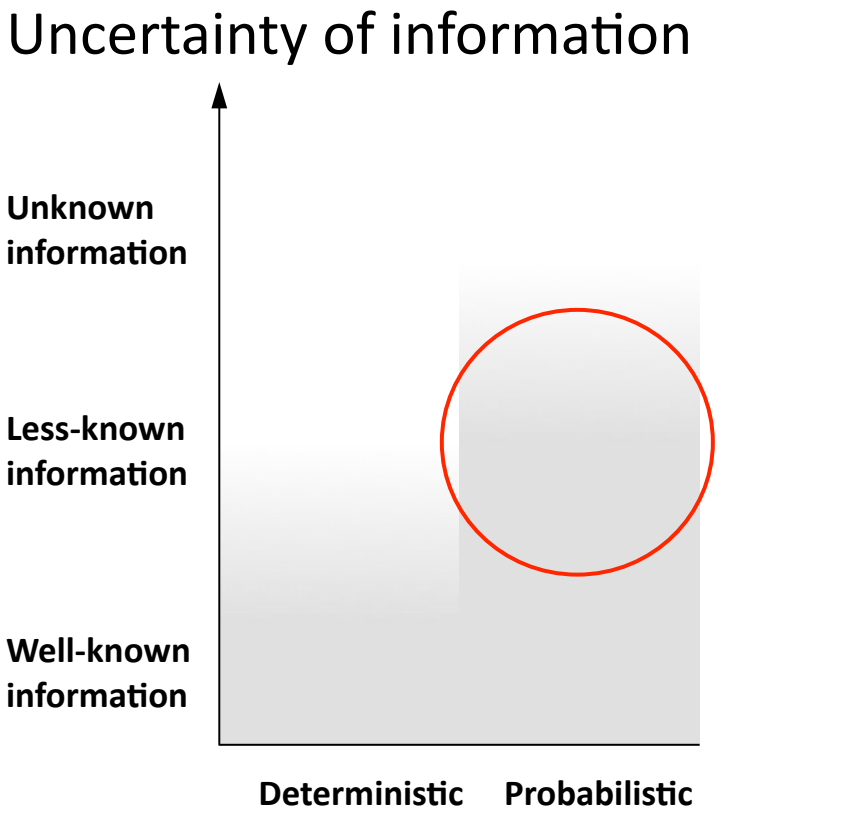
$$\varepsilon = \mathrm{N}(0, \sigma^2)$$

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



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



- 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
```

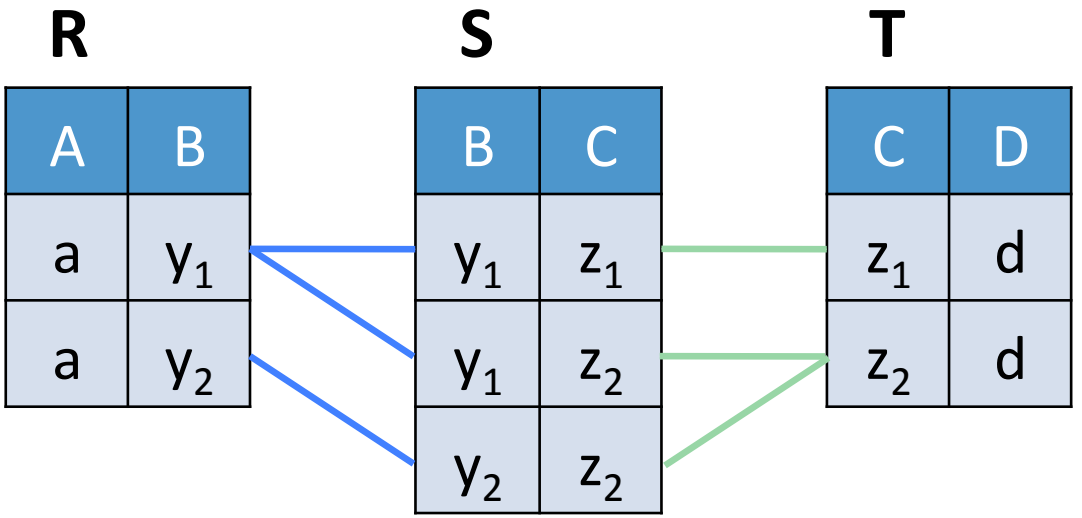
Datalog

```
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)
```

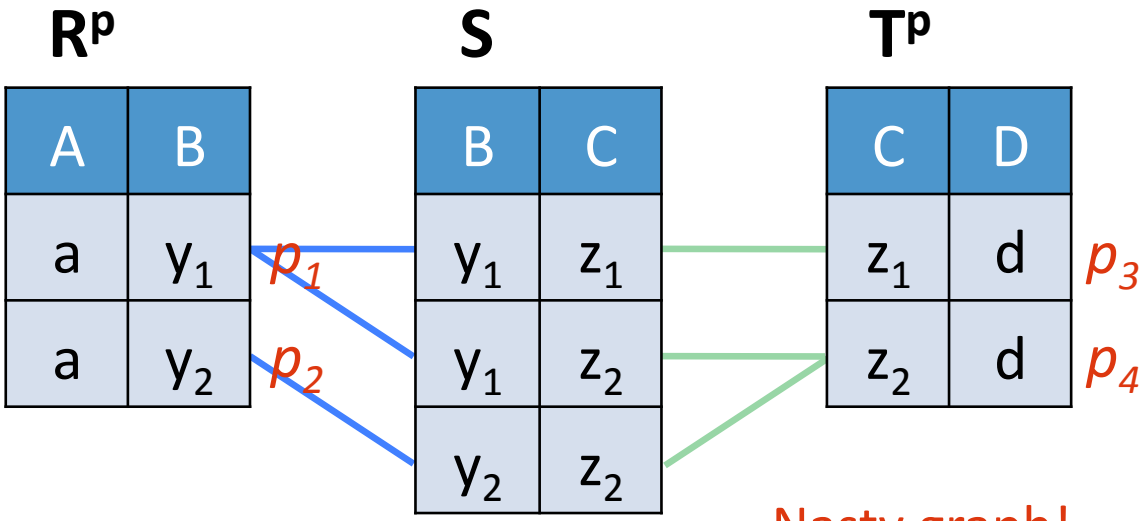


Which tuples?

```
q(a,d)
```

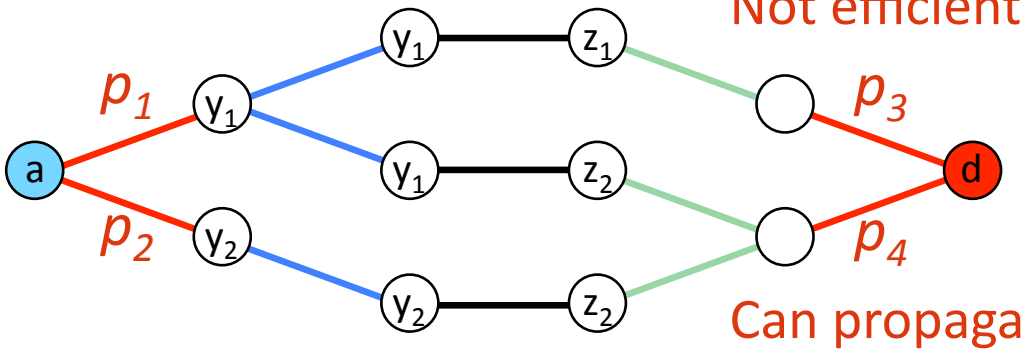
Probabilistic databases (2/3)

```
q(x,u):-Rᵖ(x,y), S(y,z), Tᵖ(z,u)
```



Nasty graph!
Not efficient!

Which tuples
& how likely?

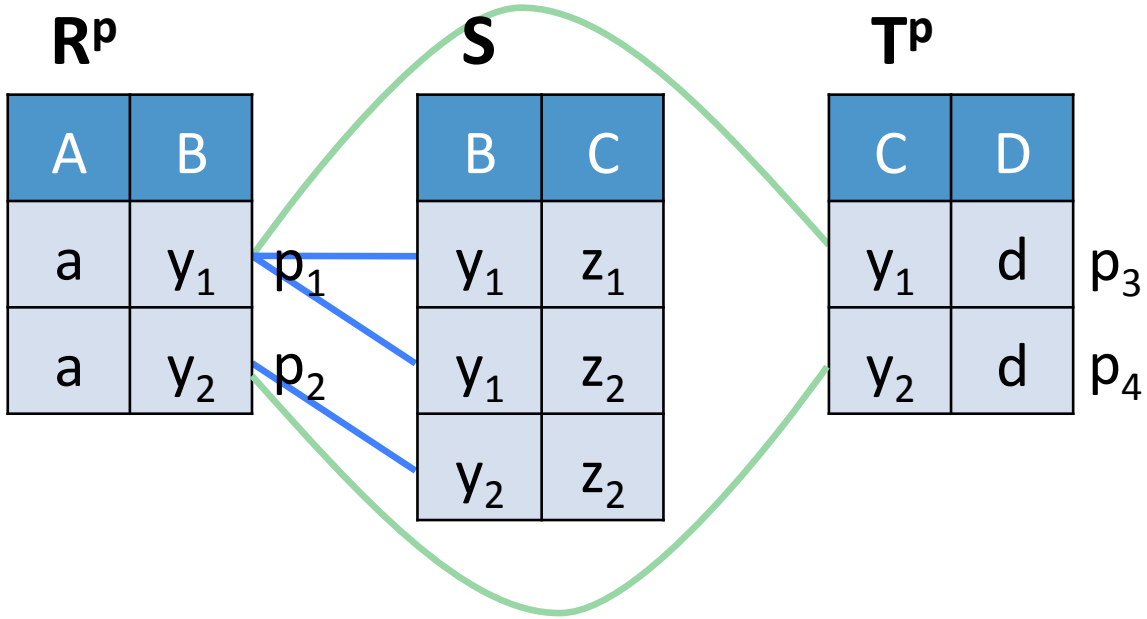


Can propagation help ?

P[q(a,d)] = $p_1p_3 \vee p_1p_4 \vee p_2p_4$ = reachability a → 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?

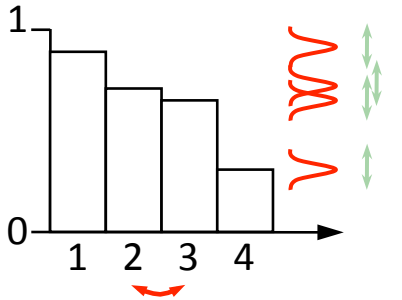
Input (probabilistic) data —?→ **Ouput ranked results**

1. Hard in general

R^p

A	B	
a	a	p_1
a	e	p_2
b	c	p_3
d	c	p_4
d	a	p_5
e	a	p_6
e	c	p_7
e	d	p_8

Possible world semantics
~ *reliability*

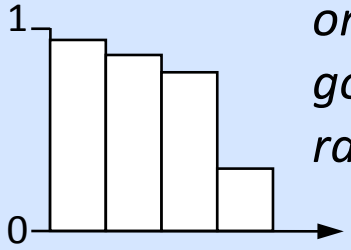


2. Sensitivity of ranking with respect to accuracy of input probabilit

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

3. Decrease in ranking quality due to approximation

Alternative ranking semantics
~ *propagation*



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

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

THANKS!