Repair and Prediction (under Inconsistency) in Large Biological Networks with Answer Set Programming

Martin Gebser[1] Carito Guziolowski[2] Mihail Ivanchev[1]
Torsten Schaub[1] Anne Siegel[2] Sven Thiele[1]
Philippe Veber[3]

[1]Institute for Informatics, University of Potsdam
[2]INRIA/Irisa, Rennes [3]Institut Cochin, Paris

Outline

Outline

The Problem

Molecular Biology

- Repositories of biochemical reactions and genetic regulations
  - *Often established experimentally*
- High-throughput methods for collecting experimental profiles
  - *Often incompatible with biological knowledge*

The Problem

Molecular Biology

- Repositories of biochemical reactions and genetic regulations
  - *Often established experimentally*
- High-throughput methods for collecting experimental profiles
  - *Often incompatible with biological knowledge*
- Incompatibilities due to unreliable data or missing reactions
  - *It is still a common practice to shift the task of making biological sense out of experimental profiles on human experts!*

The Problem

Molecular Biology

- Repositories of biochemical reactions and genetic regulations
  - *Often established experimentally*
- High-throughput methods for collecting experimental profiles
  - *Often incompatible with biological knowledge*
- Incompatibilities due to unreliable data or missing reactions
  - *It is still a common practice to shift the task of making biological sense out of experimental profiles on human experts!*

Qualitative Approach

- Represent regulatory networks by influence graphs
- Represent experimental profiles by observed variations

The Problem

Molecular Biology

- Repositories of biochemical reactions and genetic regulations
  - *Often established experimentally*
- High-throughput methods for collecting experimental profiles
  - *Often incompatible with biological knowledge*
- Incompatibilities due to unreliable data or missing reactions
  - *It is still a common practice to shift the task of making biological sense out of experimental profiles on human experts!*

Qualitative Approach

- Represent regulatory networks by influence graphs
- Represent experimental profiles by observed variations

- An experimental profile is consistent with a regulatory network **iff** each observed variation can be explained by some influence
  - *Inconsistencies point to unreliable data or missing reactions!*

Outline

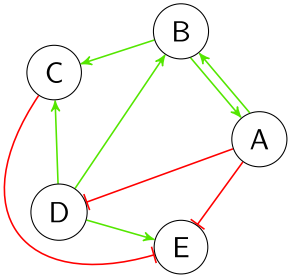
Influence Graphs

Vertices: genes, metabolites, proteins

Edges: regulations

- activation
- inhibition

Example:

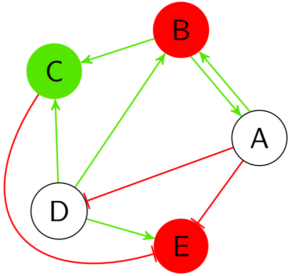


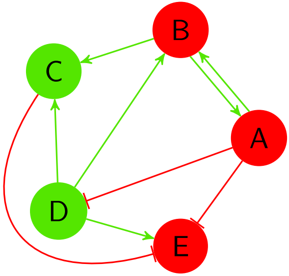
Observations

Labels: variations found in genetic profiles

- increase
- decrease

Examples:



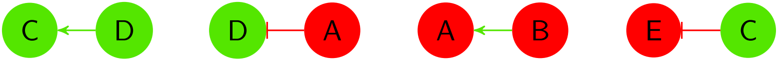


Note: Observations and regulation labelings can be partial

Sign Consistency Constraints (SCCs)

Local Consistency:

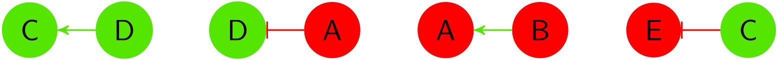
- A variation is consistent **iff** it is explained by some influence



Sign Consistency Constraints (SCCs)

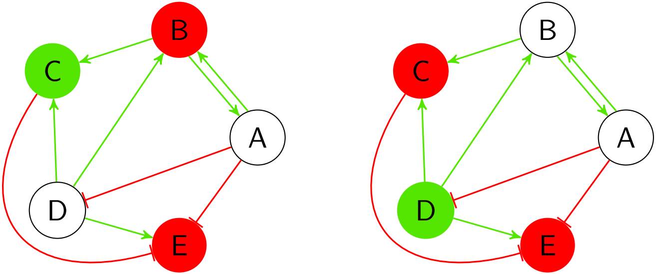
Local Consistency:

- A variation is consistent **iff** it is explained by some influence



Global Consistency:

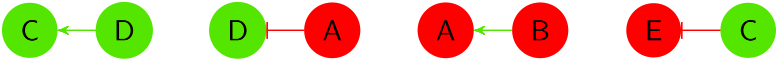
- A (partially) labeled influence graph is consistent **iff** there is a total labeling such that every variation is explained



Sign Consistency Constraints (SCCs)

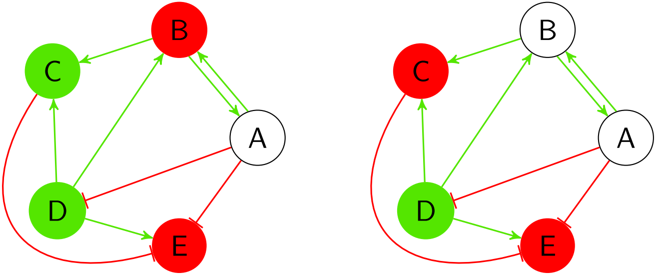
Local Consistency:

- A variation is consistent **iff** it is explained by some influence



Global Consistency:

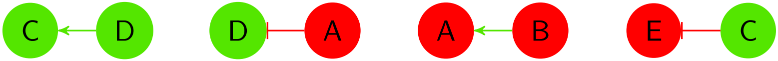
- A (partially) labeled influence graph is consistent **iff** there is a total labeling such that every variation is explained



Sign Consistency Constraints (SCCs)

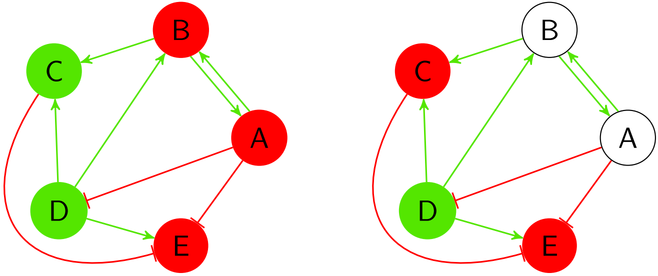
Local Consistency:

- A variation is consistent **iff** it is explained by some influence



Global Consistency:

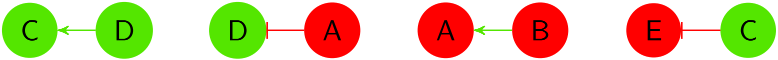
- A (partially) labeled influence graph is consistent **iff** there is a total labeling such that every variation is explained



Sign Consistency Constraints (SCCs)

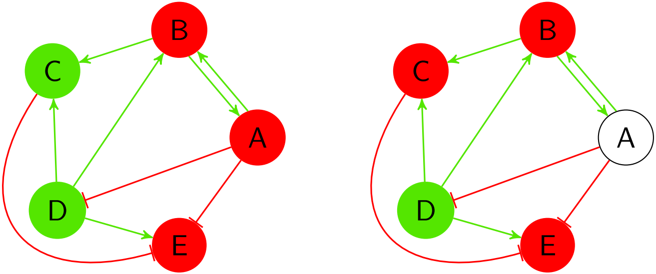
Local Consistency:

- A variation is consistent **iff** it is explained by some influence



Global Consistency:

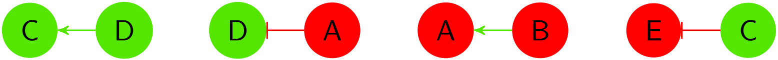
- A (partially) labeled influence graph is consistent **iff** there is a total labeling such that every variation is explained



Sign Consistency Constraints (SCCs)

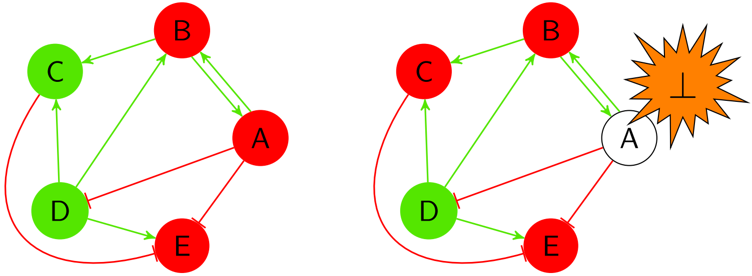
Local Consistency:

- A variation is consistent **iff** it is explained by some influence



Global Consistency:

- A (partially) labeled influence graph is consistent **iff** there is a total labeling such that every variation is explained



SCCs and Ordinary Differential Equations (ODEs)

SCCs model a rather general class of ODEs.

Theorem (Siegel et al, Biosystems)

Given a differential dynamics $\frac{dX}{dt} = F(X)$ *s.t.:*

- *Regulations with constant sign*

  $\frac{\partial F_i}{\partial X_j}$ *has a constant sign in phase space*

- *Self-degradation*

$$\exists C > 0 \ \frac{\partial F_i}{\partial X_i} < -C$$

- *Genes expressed when absent*

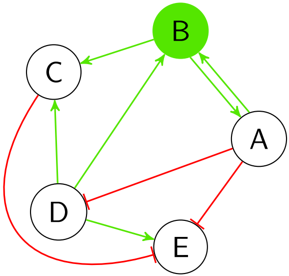
$$F(X_i = 0, X) > 0$$

Then, the SCC holds between any two steady states

Predicting Variations
under Consistency

A partially labeled influence graph may admit several solutions.

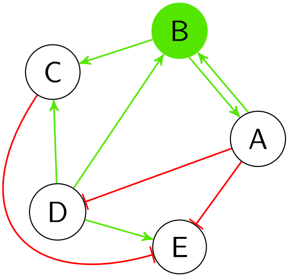
Example:

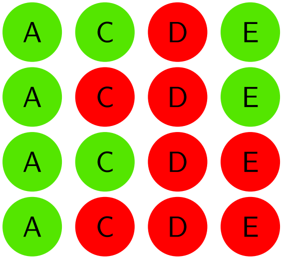


Predicting Variations
under Consistency

A partially labeled influence graph may admit several solutions.

Example:

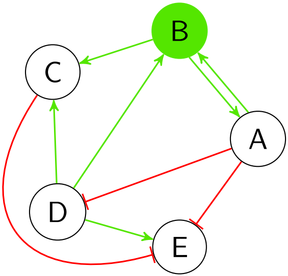


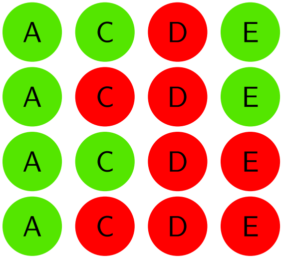


Predicting Variations
under Consistency

A partially labeled influence graph may admit several solutions.

Example:



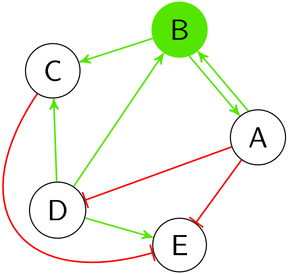


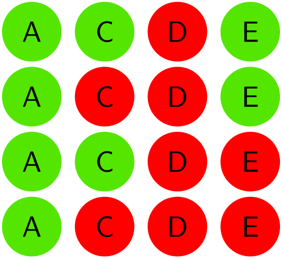
Predicted Variations:

Predicting Variations
under Consistency

A partially labeled influence graph may admit several solutions.

Example:



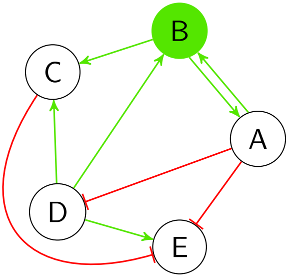


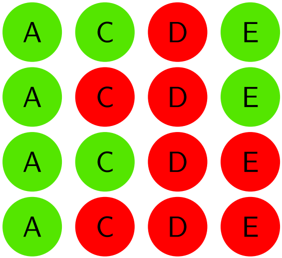
Predicted Variations:

Predicting Variations
under Consistency

A partially labeled influence graph may admit several solutions.

Example:



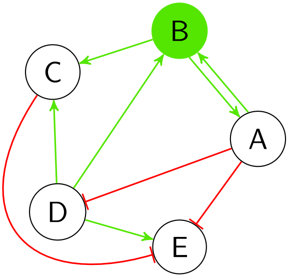


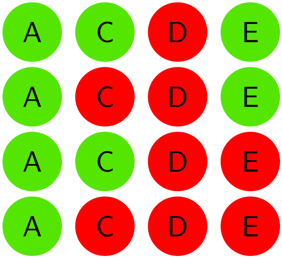
Predicted Variations:

Predicting Variations
under Consistency

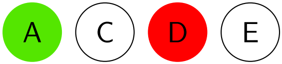
A partially labeled influence graph may admit several solutions.

Example:





Predicted Variations:



Outline

Answer Set Programming (ASP)
in a Nutshell

Answer Set Programming (ASP) *in a Nutshell*

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities

  tailored to Knowledge Representation and Reasoning

Answer Set Programming (ASP) *in a Nutshell*

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities

  tailored to Knowledge Representation and Reasoning
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way (being more compact than SAT)

Answer Set Programming (ASP) *in a Nutshell*

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities

  tailored to Knowledge Representation and Reasoning
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way (being more compact than SAT)
- The versatility of ASP is reflected by the ASP solver `clasp`, winning first places at ASP'07/09/11, PB'09/11, and SAT'09/11
  - `http://potassco.sourceforge.net`

Answer Set Programming (ASP) *in a Nutshell*

- ASP is an approach to declarative problem solving, combining
  - a rich yet simple modeling language
  - with high-performance solving capacities

  tailored to Knowledge Representation and Reasoning
- ASP allows for solving all search problems in NP (and NP^{NP}) in a uniform way (being more compact than SAT)
- The versatility of ASP is reflected by the ASP solver `clasp`, winning first places at ASP'07/09/11, PB'09/11, and SAT'09/11
  - `http://potassco.sourceforge.net`
- ASP embraces many emerging application areas

Overview on Answer Set Programming

- A logic program is a set of rules

$$a \leftarrow b_1, \ldots, b_m, not\ c_{m+1}, \ldots, not\ c_n.$$

- It is used to specify sets of (ground) atoms, its answer sets
- An answer set
  - satisfies each of the rules
  - satisfies the stability criterion
  - which implies derivability of its atoms
- Particular cases

Facts	e.g.:	$a.$
Integrity rules	e.g.:	$\leftarrow b, not\ c.$
Choice rules	e.g.:	$1\{a_1, a_2\}1 \leftarrow b, not\ c.$
(used as shorthands)		

Influence Graphs and Variations

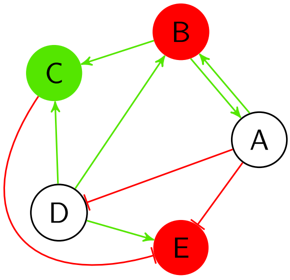
Vertices: *vertex*(i).

Edges: *edge*(j, i).

- *observedE*($j, i, +1$).
- *observedE*($j, i, -1$).

Variations:

- *observedV*($i, +1$).
- *observedV*($i, -1$).



Influence Graphs and Variations

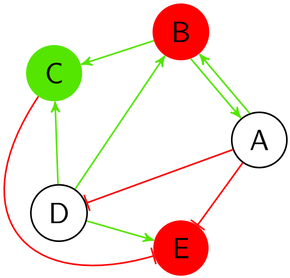
Vertices: $vertex(i)$.

Edges: $edge(j, i)$.

- (green line) $observedE(j, i, +1)$.
- (red line) $observedE(j, i, -1)$.

Variations:

- (green dot) $observedV(i, +1)$.
- (red dot) $observedV(i, -1)$.



Example:

$vertex(\mathrm{A})$. ... $vertex(\mathrm{E})$.

$edge(\mathrm{A}, \mathrm{B})$. $edge(\mathrm{A}, \mathrm{D})$. ... $edge(\mathrm{D}, \mathrm{C})$. $edge(\mathrm{D}, \mathrm{E})$.

$observedE(\mathrm{A}, \mathrm{B}, +1)$. $observedE(\mathrm{A}, \mathrm{D}, -1)$. ...
$observedE(\mathrm{D}, \mathrm{C}, +1)$. $observedE(\mathrm{D}, \mathrm{E}, +1)$.

$observedV(\mathrm{B}, -1)$. $observedV(\mathrm{C}, +1)$. $observedV(\mathrm{E}, -1)$.

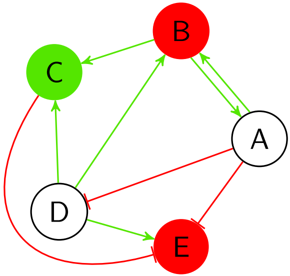
Generating Total Labelings

Edge Labels:

$$1\{labelE(J, I, +1), labelE(J, I, -1)\}1 \leftarrow edge(J, I).$$
$$labelE(J, I, S) \leftarrow observedE(J, I, S).$$

Vertex Labels:

$$1\{labelV(I, +1), labelV(I, -1)\}1 \leftarrow vertex(I).$$
$$labelV(I, S) \leftarrow observedV(I, S).$$



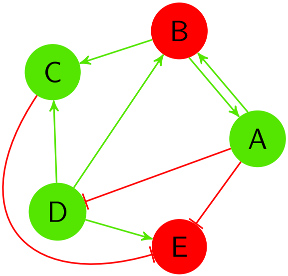
Generating Total Labelings

Edge Labels:

$$1\{labelE(J, I, +1), labelE(J, I, -1)\}1 \leftarrow edge(J, I).$$
$$labelE(J, I, S) \leftarrow observedE(J, I, S).$$

Vertex Labels:

$$1\{labelV(I, +1), labelV(I, -1)\}1 \leftarrow vertex(I).$$
$$labelV(I, S) \leftarrow observedV(I, S).$$



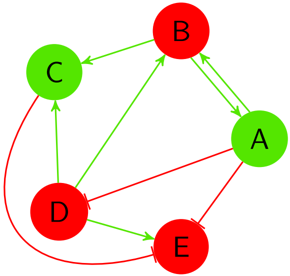
Generating Total Labelings

Edge Labels:

$$1\{labelE(J, I, +1), labelE(J, I, -1)\}1 \leftarrow edge(J, I).$$
$$labelE(J, I, S) \leftarrow observedE(J, I, S).$$

Vertex Labels:

$$1\{labelV(I, +1), labelV(I, -1)\}1 \leftarrow vertex(I).$$
$$labelV(I, S) \leftarrow observedV(I, S).$$



Generating Total Labelings

Edge Labels:

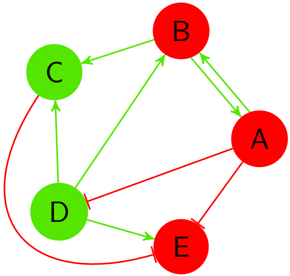
$$1\{labelE(J, I, +1), labelE(J, I, -1)\}1 \leftarrow edge(J, I).$$

$$labelE(J, I, S) \leftarrow observedE(J, I, S).$$

Vertex Labels:

$$1\{labelV(I, +1), labelV(I, -1)\}1 \leftarrow vertex(I).$$

$$labelV(I, S) \leftarrow observedV(I, S).$$



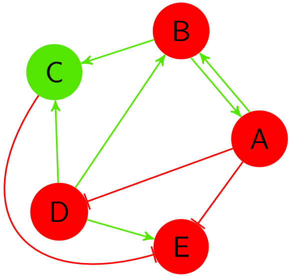
Generating Total Labelings

Edge Labels:

$$1\{labelE(J, I, +1), labelE(J, I, -1)\}1 \leftarrow edge(J, I).$$
$$labelE(J, I, S) \leftarrow observedE(J, I, S).$$

Vertex Labels:

$$1\{labelV(I, +1), labelV(I, -1)\}1 \leftarrow vertex(I).$$
$$labelV(I, S) \leftarrow observedV(I, S).$$



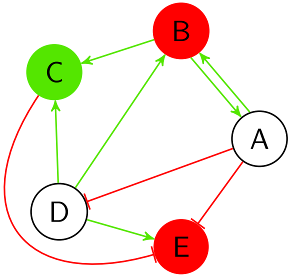
Testing Total Labelings

Influences:

$receive(I, S * T) \leftarrow labelE(J, I, S), labelV(J, T).$

Sign Consistency:

$\leftarrow labelV(I, S), not\ receive(I, S).$



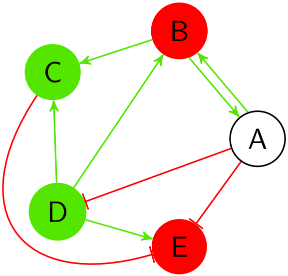
Testing Total Labelings

Influences:

$$receive(I, S * T) \leftarrow labelE(J, I, S), labelV(J, T).$$

Sign Consistency:

$$\leftarrow labelV(I, S), not\ receive(I, S).$$



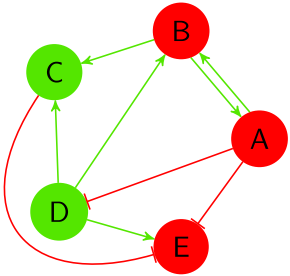
Testing Total Labelings

Influences:

$receive(I, S * T) \leftarrow labelE(J, I, S), labelV(J, T).$

Sign Consistency:

$\leftarrow labelV(I, S), not\ receive(I, S).$



Outline

Motivation

Observation: Regulatory networks and experimental profiles are often inconsistent with each other!

Question: How to predict unobserved variations in this case?

Motivation

Observation: Regulatory networks and experimental profiles are often inconsistent with each other!

Question: How to predict unobserved variations in this case?

Idea:

1. Repair inconsistencies
2. Predict from repaired networks and/or profiles

Repairing Networks and/or Profiles

Network Repair:

- Adding edges completes an incomplete network (w.r.t. profiles)
- Flipping edge labels curates an improper network
- Making vertices input indicates incompleteness or oscillations

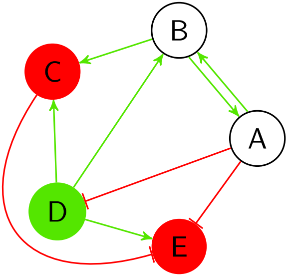
Profile Repair:

- Flipping vertex labels indicates aberrant experimental data

Repair Operations

Adding Edges

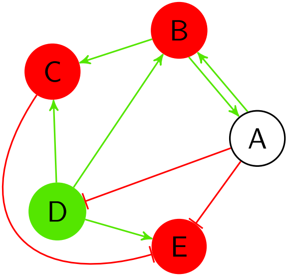
$$rep(add\_e(U,V)) \leftarrow vertex(U), vertex(V), U \neq V, not\ edge(U,V).$$



Repair Operations

Adding Edges

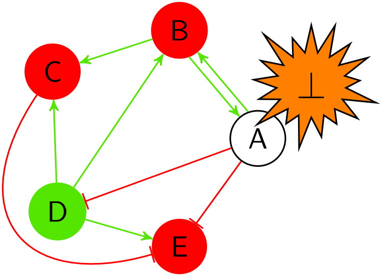
$$rep(add\_e(U,V)) \leftarrow vertex(U), vertex(V), U \neq V, not\ edge(U,V).$$



Repair Operations

Adding Edges

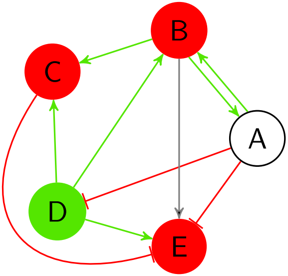
$$rep(add\_e(U,V)) \leftarrow vertex(U), vertex(V), U \neq V, not\ edge(U,V).$$



Repair Operations

Adding Edges

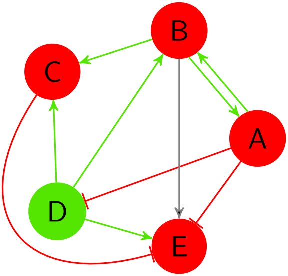
$$rep(add\_e(U, V)) \leftarrow vertex(U), vertex(V), U \neq V, not\ edge(U, V).$$



Repair Operations

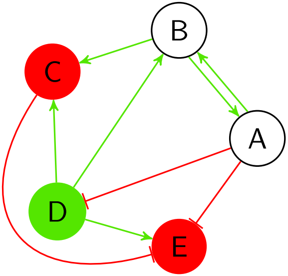
Adding Edges

$$rep(add\_e(U,V)) \leftarrow vertex(U), vertex(V), U \neq V, not\ edge(U,V).$$



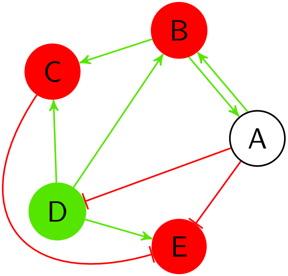
Repair Operations
Flipping Edge Labels

$$rep(flip\_e(U,V,S)) \leftarrow observedE(U,V,S).$$



Repair Operations
Flipping Edge Labels

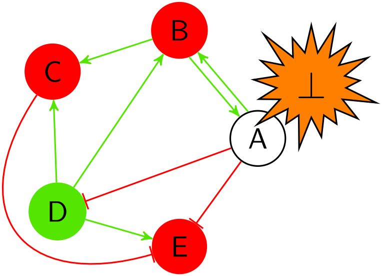
$$rep(flip\_e(U, V, S)) \leftarrow observedE(U, V, S).$$



Repair Operations

Flipping Edge Labels

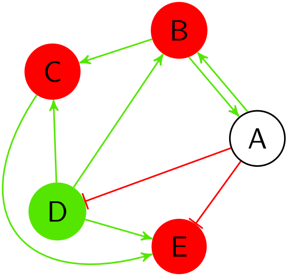
$$rep(flip\_e(U, V, S)) \leftarrow observedE(U, V, S).$$



Repair Operations

Flipping Edge Labels

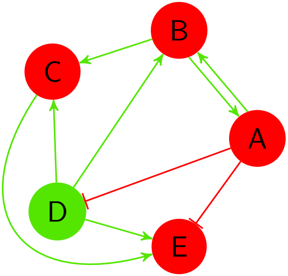
$$rep(flip\_e(U, V, S)) \leftarrow observedE(U, V, S).$$



Repair Operations

Flipping Edge Labels

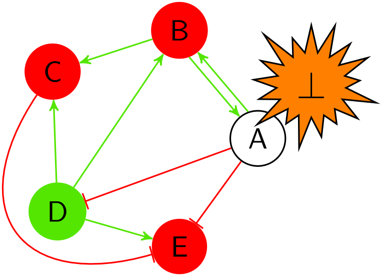
$$rep(flip\_e(U, V, S)) \leftarrow observedE(U, V, S).$$



Repair Operations

Flipping Vertex Labels

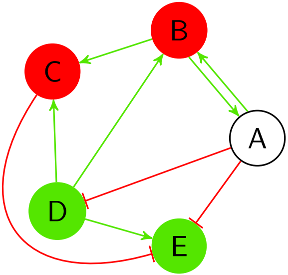
$$rep(flip\_v(V, S)) \leftarrow observedV(V, S).$$



Repair Operations

Flipping Vertex Labels

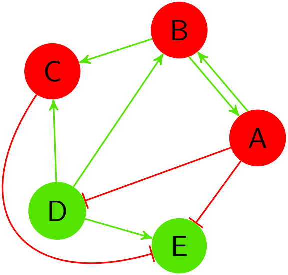
$$rep(flip\_v(V, S)) \leftarrow observedV(V, S).$$



Repair Operations

Flipping Vertex Labels

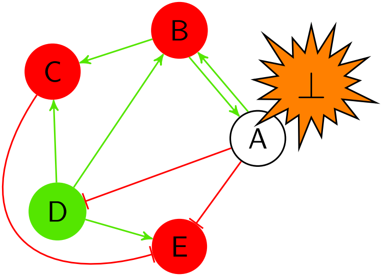
$$rep(flip\_v(V, S)) \leftarrow observedV(V, S).$$



Repair Operations

Making Vertices Input

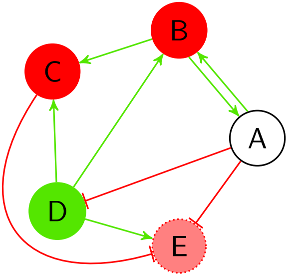
$$rep(inp\_v(V)) \leftarrow vertex(V), not\ input(V).$$



Repair Operations

Making Vertices Input

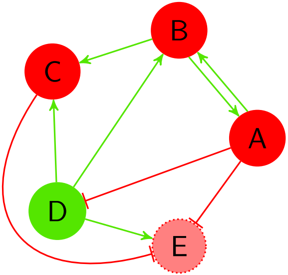
$$rep(inp\_v(V)) \leftarrow vertex(V), not\ input(V).$$



Repair Operations

Making Vertices Input

$$rep(inp\_v(V)) \leftarrow vertex(V), not\ input(V).$$



Generating Total Labelings under Repair

Applying Repair Operations:

$0\{app(R)\}1 \leftarrow rep(R).$

Generating Edge Labelings:

$1\{labelE(U, V, +1), labelE(U, V, -1)\}1 \leftarrow edge(U, V).$

$1\{labelE(U, V, +1), labelE(U, V, -1)\}1 \leftarrow app(add\_e(U, V)).$

$labelE(U, V, S) \leftarrow observedE(U, V, S), not\ app(flip\_e(U, V, S)).$

$labelE(U, V, -S) \leftarrow app(flip\_e(U, V, S)).$

Generating Vertex Labelings:

$1\{labelV(V, +1), labelV(V, -1)\}1 \leftarrow vertex(V).$

$labelV(V, S) \leftarrow observedV(V, S), not\ app(flip\_v(V, S)).$

$labelV(V, -S) \leftarrow app(flip\_v(V, S)).$

Generating Total Labelings under Repair

Applying Repair Operations:

$0\{app(R)\}1 \leftarrow rep(R).$

Generating Edge Labelings:

$1\{labelE(U,V,+1), labelE(U,V,-1)\}1 \leftarrow edge(U,V).$

$1\{labelE(U,V,+1), labelE(U,V,-1)\}1 \leftarrow app(add\_e(U,V)).$

$labelE(U,V,S) \leftarrow observedE(U,V,S), not\ app(flip\_e(U,V,S)).$

$labelE(U,V,-S) \leftarrow app(flip\_e(U,V,S)).$

Generating Vertex Labelings:

$1\{labelV(V,+1), labelV(V,-1)\}1 \leftarrow vertex(V).$

$labelV(V,S) \leftarrow observedV(V,S), not\ app(flip\_v(V,S)).$

$labelV(V,-S) \leftarrow app(flip\_v(V,S)).$

Testing Total Labelings under Repair

Enforcing Sign Consistency Constraints:

$$
\begin{aligned}
receive(I, S*T) &\leftarrow labelE(J,I,S), labelV(J,T).\\
&\leftarrow labelV(I,S), not\ receive(I,S),\\
&\quad\ not\ input(V), not\ app(inp\_v(V)).
\end{aligned}
$$

Testing Total Labelings under Repair

Enforcing Sign Consistency Constraints:

$$\begin{aligned} receive(I, S * T) &\leftarrow labelE(J, I, S), labelV(J, T). \\ &\leftarrow labelV(I, S), not\ receive(I, S), \\ &\quad\ not\ input(V), not\ app(inp\_v(V)). \end{aligned}$$

Testing Total Labelings under Repair

Enforcing Sign Consistency Constraints:

$$receive(I, S*T) \leftarrow labelE(J, I, S), labelV(J, T).$$
$$\leftarrow labelV(I, S), not\ receive(I, S), not\ input(V), not\ app(inp\_v(V)).$$

Minimal Repair

Goal:

Minimal change of networks/profiles
(re)establishing consistency

Implementation (cardinality minimality):

$$\#minimize\{app(R) : rep(R)\}.$$

(see paper for subset minimality)

Predicting under Repair

Two Phase Approach:

1. Compute minimal number of required repair operations
2. Intersect consistent labelings under minimal repair
   - Cautious reasoning (supported by answer set solver `clasp`)

Outline

Predicting Variations
under Inconsistency

- Transcriptional network of *Escherichia coli*, obtained from RegulonDB by Gama-Castro *et al.* [2008], consisting of
  - 5150 interactions between 1914 genes
- Two datasets
  - Exponential-Stationary growth shift by Bradley *et al.* [2007]
  - Heatshock by Allen *et al.* [2003]
- The data of both experiments is highly noisy and inconsistent with the (well-curated) RegulonDB model

Predicting Variations
under Inconsistency

- Transcriptional network of *Escherichia coli*, obtained from RegulonDB by Gama-Castro *et al.* [2008], consisting of
  - 5150 interactions between 1914 genes
- Two datasets
  - Exponential-Stationary growth shift by Bradley *et al.* [2007]
  - Heatshock by Allen *et al.* [2003]
- The data of both experiments is highly noisy and inconsistent with the (well-curated) RegulonDB model
- For enabling prediction rate and accuracy assessment, we randomly select samples of significantly expressed genes (3%,6%,9%,12%,15% of the whole data, 200 samples each) and use them for testing both our repair modes and prediction

Repair and Prediction Times

Repair	Exponential-Stationary					Heatshock				
	3%	6%	9%	12%	15%	3%	6%	9%	12%	15%

'e': flipping edge labels 'i': making vertices input 'v': flipping vertex labels

Repair and Prediction Times

Repair

Repair			Exponential-Stationary 3%	6%	9%	12%	15%	Heatshock 3%	6%	9%	12%	15%
e			6.58	8.44	11.60	14.88	26.20	25.54	42.76	50.46	69.23	84.77
	i		2.18	2.15	2.21	2.23	2.21	2.10	2.13	2.13	2.05	2.08
		v	1.41	1.40	1.40	1.41	1.37	1.41	1.47	1.42	1.37	1.39
e	i		73.16	202.66	392.97	518.50	574.85	120.91	374.69	553.00	593.20	595.99
e		v	28.53	85.17	189.27	327.98	470.48	67.92	236.05	465.92	579.88	596.17
	i	v	2.09	2.14	2.45	3.08	6.06	2.27	4.94	60.63	257.68	418.93
e	i	v	133.84	391.60	538.93	593.33	600.00	232.29	542.48	593.88	600.00	600.00

‘e’: flipping edge labels ‘i’: making vertices input ‘v’: flipping vertex labels

Repair and Prediction Times

	Repair	Exponential-Stationary 3%	6%	9%	12%	15%	Heatshock 3%	6%	9%	12%	15%
Repair	e	6.58	8.44	11.60	14.88	26.20	25.54	42.76	50.46	69.23	84.77
	i	2.18	2.15	2.21	2.23	2.21	2.10	2.13	2.13	2.05	2.08
	v	1.41	1.40	1.40	1.41	1.37	1.41	1.47	1.42	1.37	1.39
	e i	73.16	202.66	392.97	518.50	574.85	120.91	374.69	553.00	593.20	595.99
	e v	28.53	85.17	189.27	327.98	470.48	67.92	236.05	465.92	579.88	596.17
	i v	2.09	2.14	2.45	3.08	6.06	2.27	4.94	60.63	257.68	418.93
	e i v	133.84	391.60	538.93	593.33	600.00	232.29	542.48	593.88	600.00	600.00
Prediction	e	13.27	12.19	14.76	15.34	25.90	25.77	37.18	29.09	36.23	41.88
	i	6.18	5.26	4.77	4.60	4.42	6.57	5.93	5.17	4.86	4.54
	v	4.64	4.45	4.39	4.40	4.30	4.86	5.06	5.34	5.42	5.52
	e i	35.25	97.66	293.80	456.55	550.33	85.47	293.28	524.19	591.81	594.74
	e v	14.35	26.17	90.17	200.25	363.36	23.32	111.99	338.95	545.56	591.23
	i v	6.43	5.75	6.27	6.69	8.61	6.91	6.63	30.33	176.14	371.95
	e i v	42.51	248.30	468.71	579.58	—	101.82	466.91	585.64	—	—

'e': flipping edge labels 'i': making vertices input 'v': flipping vertex labels

Repair and Prediction Times

It's feasible!

		Exponential-Stationary					Heatshock				
		3%	6%	9%	12%	15%	3%	6%	9%	12%	15%
Repair	e	6.58	8.44	11.60	14.88	26.20	25.54	42.76	50.46	69.23	84.77
Repair	i	2.18	2.15	2.21	2.23	2.21	2.10	2.13	2.13	2.05	2.08
Repair	v	1.41	1.40	1.40	1.41	1.37	1.41	1.47	1.42	1.37	1.39
Repair	e i	73.16	202.66	392.97	518.50	574.85	120.91	374.69	553.00	593.20	595.99
Repair	e v	28.53	85.17	189.27	327.98	470.48	67.92	236.05	465.92	579.88	596.17
Repair	i v	2.09	2.14	2.45	3.08	6.06	2.27	4.94	60.63	257.68	418.93
Repair	e i v	133.84	391.60	538.93	593.33	600.00	232.29	542.48	593.88	600.00	600.00
Prediction	e	13.27	12.19	14.76	15.34	25.90	25.77	37.18	29.09	36.23	41.88
Prediction	i	6.18	5.26	4.77	4.60	4.42	6.57	5.93	5.17	4.86	4.54
Prediction	v	4.64	4.45	4.39	4.40	4.30	4.86	5.06	5.34	5.42	5.52
Prediction	e i	35.25	97.66	293.80	456.55	550.33	85.47	293.28	524.19	591.81	594.74
Prediction	e v	14.35	26.17	90.17	200.25	363.36	23.32	111.99	338.95	545.56	591.23
Prediction	i v	6.43	5.75	6.27	6.69	8.61	6.91	6.63	30.33	176.14	371.95
Prediction	e i v	42.51	248.30	468.71	579.58	—	101.82	466.91	585.64	—	—

'e': flipping edge labels 'i': making vertices input 'v': flipping vertex labels

Prediction Rate and Accuracy in Percent

Repair	Exponential-Stationary					Heatshock				
	3%	6%	9%	12%	15%	3%	6%	9%	12%	15%

‘e’: flipping edge labels ‘i’: making vertices input ‘v’: flipping vertex labels

Prediction Rate and Accuracy in Percent

Rate

Repair			Exponential-Stationary 3%	6%	9%	12%	15%	Heatshock 3%	6%	9%	12%	15%
e			15.00	18.51	20.93	22.79	23.94	15.47	19.54	21.87	23.17	24.78
	i		15.00	18.51	20.93	22.79	23.93	15.48	19.62	21.89	23.20	24.80
		v	14.90	18.37	20.86	22.73	23.77	15.32	19.59	21.37	22.13	23.79
e	i		14.92	18.61	20.55	21.96	22.80	15.37	19.62	22.83	23.44	24.05
e		v	14.89	18.33	21.07	22.52	23.74	15.33	19.21	21.00	22.65	24.90
	i	v	14.89	18.33	20.79	22.59	23.66	15.41	19.47	21.36	21.81	23.55
e	i	v	14.58	19.00	20.29	21.13	—	15.01	19.11	22.52	—	—

'e': flipping edge labels 'i': making vertices input 'v': flipping vertex labels

Prediction Rate and Accuracy in Percent

	Repair			Exponential-Stationary 3%	6%	9%	12%	15%	Heatshock 3%	6%	9%	12%	15%
Rate	e			15.00	18.51	20.93	22.79	23.94	15.47	19.54	21.87	23.17	24.78
		i		15.00	18.51	20.93	22.79	23.93	15.48	19.62	21.89	23.20	24.80
			v	14.90	18.37	20.86	22.73	23.77	15.32	19.59	21.37	22.13	23.79
	e	i		14.92	18.61	20.55	21.96	22.80	15.37	19.62	22.83	23.44	24.05
	e		v	14.89	18.33	21.07	22.52	23.74	15.33	19.21	21.00	22.65	24.90
		i	v	14.89	18.33	20.79	22.59	23.66	15.41	19.47	21.36	21.81	23.55
	e	i	v	14.58	19.00	20.29	21.13	—	15.01	19.11	22.52	—	—
Accuracy	e			90.93	91.98	92.42	92.70	92.81	91.87	92.93	92.92	92.83	92.71
		i		90.93	91.98	92.42	92.70	92.81	91.93	92.90	92.94	92.87	92.76
			v	90.99	92.05	92.44	92.73	92.89	92.29	93.27	93.88	94.27	94.36
	e	i		91.09	91.90	92.57	93.03	93.19	91.99	92.49	91.16	93.62	94.44
	e		v	90.99	92.03	92.50	92.82	92.94	92.30	93.37	93.66	94.36	94.35
		i	v	90.99	92.03	92.42	92.71	92.87	92.24	93.34	93.90	94.26	94.38
	e	i	v	91.35	92.29	92.52	93.04	—	92.26	93.04	91.78	—	—

'e': flipping edge labels 'i': making vertices input 'v': flipping vertex labels

Prediction Rate and Accuracy in Percent

	Repair			Exponential-Stationary 3%	6%	9%	12%	15%	Heatshock 3%	6%	9%	12%	15%
Rate	e			15.00	18.51	20.93	22.79	23.94	15.47	19.54	21.87	23.17	24.78
		i		15.00	18.51	20.93	22.79	23.93	15.48	19.62	21.89	23.20	24.80
			v	14.90	18.37	20.86	22.73	23.77	15.32	19.59	21.37	22.13	23.79
	e	i		14.92	18.61	20.55	21.96	22.80	15.37	19.62	22.83	23.44	24.05
	e		v	14.89	18.33	21.07	22.52	23.74	15.33	19.21	21.00	22.65	24.90
		i	v	14.89	18.33	20.79	22.59	23.66	15.41	19.47	21.36	21.81	23.55
	e	i	v	14.58	19.00	20.29	21.13	—	15.01	19.11	22.52	—	—
Accuracy	e			90.93	91.98	92.42	92.70	92.81	91.87	92.93	92.92	92.83	92.71
		i		90.93	91.98	92.42	92.70	92.81	91.93	92.90	92.94	92.87	92.76
			v	90.99	92.05	92.44	92.73	92.89	92.29	93.27	93.88	94.27	94.36
	e	i		91.09	91.90	92.57	93.03	93.19	91.99	92.49	91.16	93.62	94.44
	e		v	90.99	92.03	92.50	92.82	92.94	92.30	93.37	93.66	94.36	94.35
		i	v	90.99	92.03	92.42	92.71	92.87	92.24	93.34	93.90	94.26	94.38
	e	i	v	91.35	92.29	92.52	93.04	—	92.26	93.04	91.78	—	—

‘e’: flipping edge labels ‘i’: making vertices input ‘v’: flipping vertex labels

Accuracy over 90%!

Outline

Summary

- We introduced repair-based reasoning techniques for computing minimal modifications of
  - biological networks and
  - experimental profiles

  in order to make them mutually consistent.
- Using Answer Set Programming, we demonstrated on real data that predictions after repair are
  - feasible and
  - highly accurate.
- Answer Set Programming provided a
  - declarative,
  - succinct, and
  - highly efficient

  solution to a knowledge-intense yet error-prone application.