Inferring Executable Models from Formalized Experimental Evidence

Vivek Nigam, Robin Donaldson, Merrill Knapp, Tim McCarthy,
Carolyn Talcott
CMSB
September 2015

Executive Summary

- Executable models of signal transduction provide
  - insights into how cells work
  - explanations of observed outcomes
  - a means to understand and predict the effects of perturbations and mutations

- Developing such models from experimental findings is low throughput and requires substantial expertise. Automation can help.

- Aspect of automation:
  (1) formal representation of experimental findings,
  (2) formal representation of rules as elements of executable models,
  (3) extracting findings from papers,
  (4) algorithms for inferring rules from findings and
  (5) algorithms for assembly of executable models.

- This talk addresses aspects (1), (2) and (4).

Contributions

- A formal representation of experimental evidence called *datums*.
- A language of logical assertions that formalize the elements of a datum.
- A translation from datum syntax to logical assertions.
- A set of axioms that capture the semantics of datums interpreted as constraints on signal transduction rule patterns.
- Viewing the axioms and assertions as Answer Set Programs, minimal models are inferred, and reaction rules extracted.

Plan

- Pathway Logic in a nutshell
- Rule inference informally
- Formalizing rule Inference
- Hras case study
- Concluding

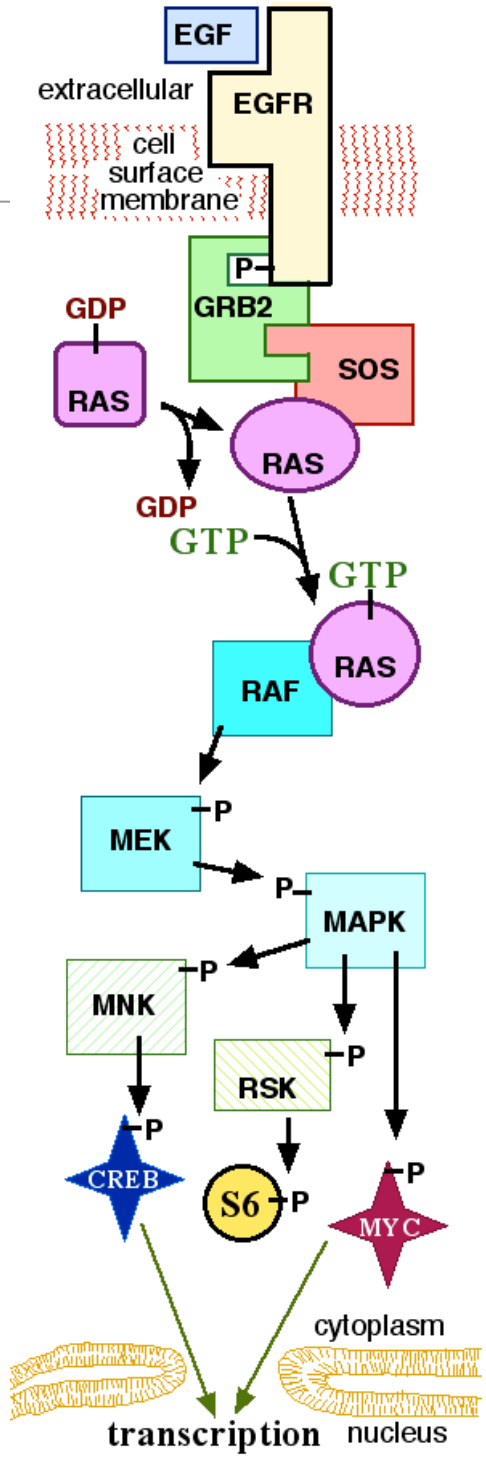
Pathway Logic

Executable models of cellular processes

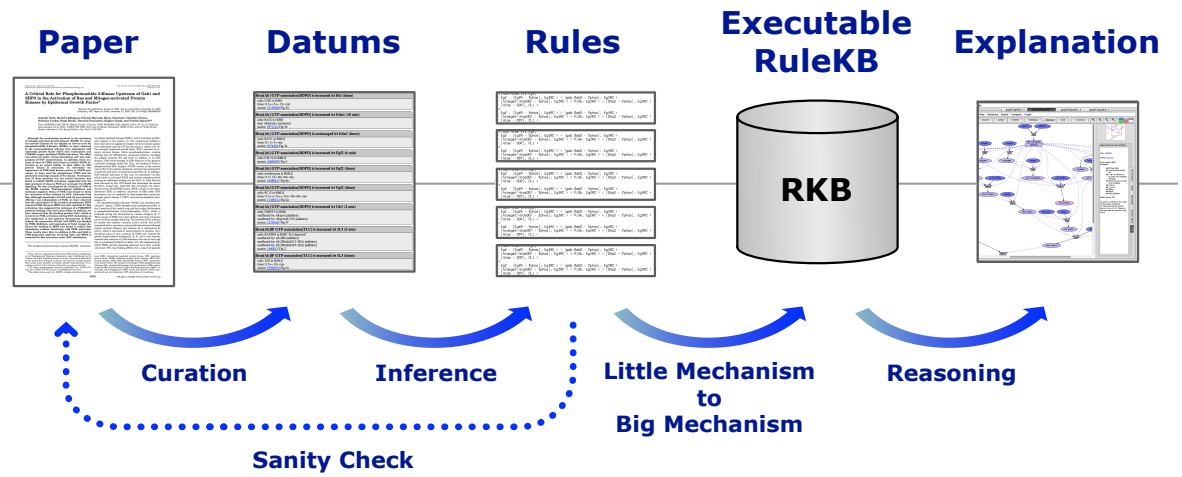
http://pl.csl.sri.com

Pathway Logic (PL) Goals

- Understanding how cells work
- Formal models of biomolecular processes that
  - capture biologist intuitions
  - can be executed
- Tools to
  - organize and analyze experimental findings
  - carry out gedanken experiments
  - discover/assemble execution pathways
- New insights into the inner workings of a cell.
- A new kind of review



PL from 1k feet



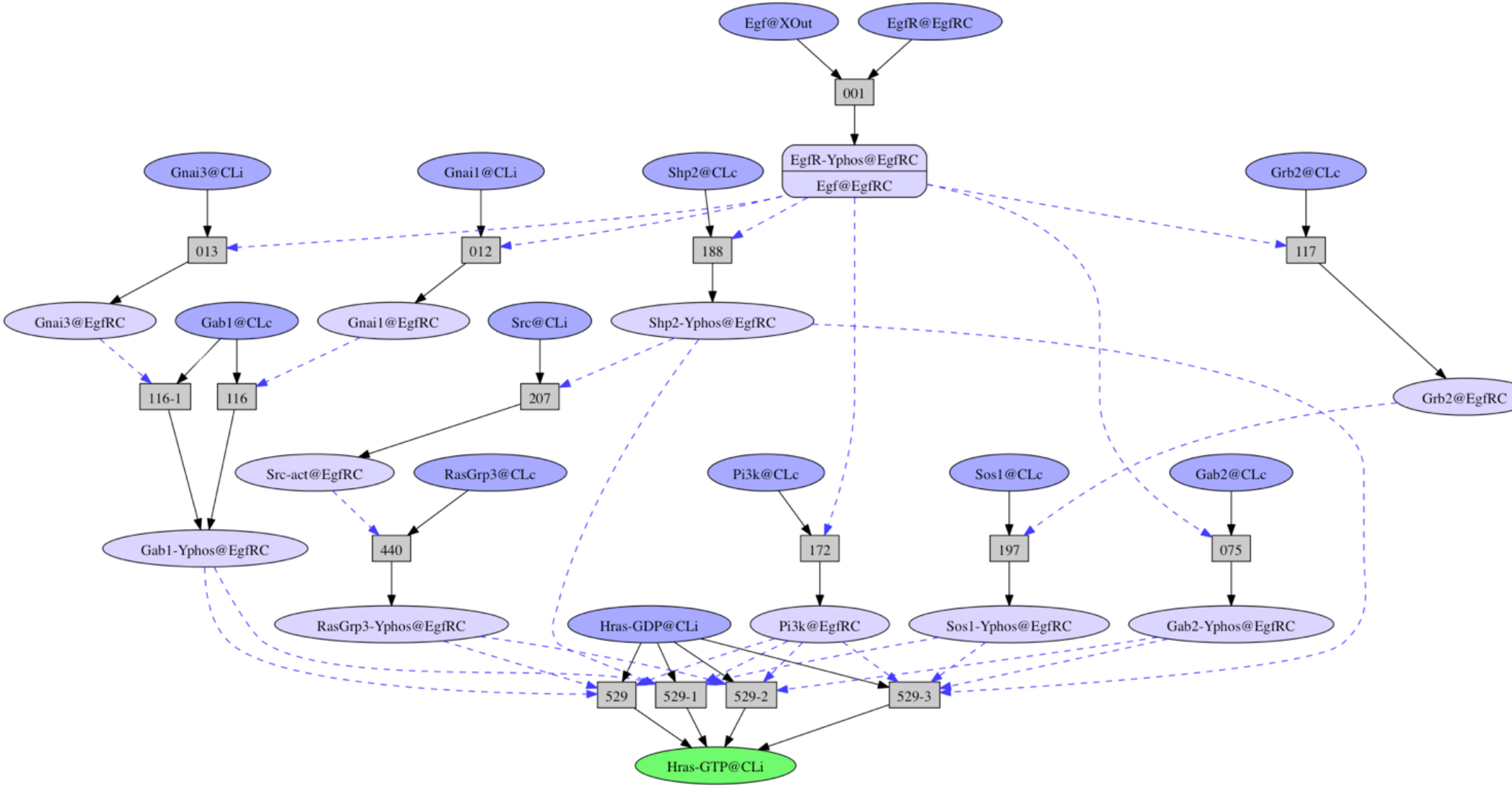
Key components

- Representation system
  - controlled vocabulary
  - datums (formalized experimental results)
  - rules describing local change/interactions
- Curated datum knowledge base (DKB) and search tool
- Evidence based rule knowledge bases (RKB)
  - STM, Protease, Mycolate, GlycoSTM …
- Executable models
  - generated by specifying initial conditions and constraints
  - queried using formal reasoning techniques
- PLA to visualize and browse subnets

Deriving the Hras rule informally

The subnet of the Egf model for activating (GTPing) **Hras**.

(Gold Standard for the running example.)



The Hras Rule formally

---

rl[529.Hras.irt.Egf]:

< Egf : [EgfR - Yphos], EgfRC > < [gab:GabS - Yphos], EgfRC >

< [hrasgef:HrasGEF - Yphos], EgfRC > < Pi3k, EgfRC > < [Shp2 - Yphos], EgfRC >

< [Hras - GDP], CLi >

=>

< Egf : [EgfR - Yphos], EgfRC > < [gab:GabS - Yphos], EgfRC >

< [hrasgef:HrasGEF - Yphos], EgfRC > < Pi3k, EgfRC > < [Shp2 - Yphos], EgfRC >

< [Hras - GTP], CLi >

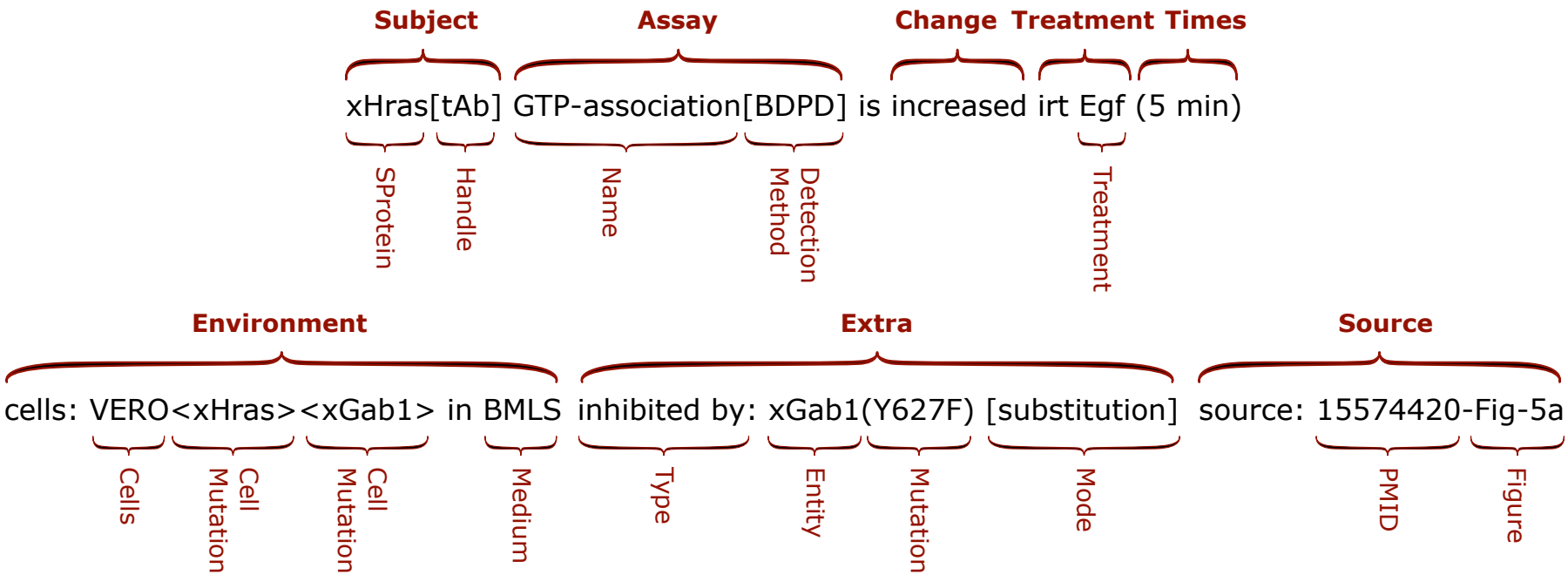
*** ~/evidence/Egf-Evidence/Hras.irt.Egf.529.txt

The rule says that GDP will be exchanged for GTP if, in addition to the EgfR complex, there is tyrosine phosphorylated Gab1 or Gab2 ([gab:GabS - Yphos]), a tyrosine phosphorylated HrasGef ([hrasgef:HrasGEF - Yphos]), Pi3k, and tyrosine phosphorylated Shp2 all recruited to the EgfR complex (EgfRC).

Where do rules come from?

- They are inferred from experimental findings.
- These are collected using a formal data structure call datums
  - datums are available in text (readable) or json (computable)
- The datum below says that the amount of GTP bound to Hras is increased after addition of Egf (Epidermal Growth Factor) to VERO cells for 5 minutes.

The Elements of a Datum



Inferring the Hras rule: the basic pattern

The `first line' of the previous Hras datum:

xHras[tAb] GTP-association[BDPD] is increased irt Egf (5 min)

can be represented by a rule pattern:

EgfTC C < [G - gmods act], Lg > < [Hras - GDP pmods], CLi >

=>

EgfTC C < [G - gmods act], Lg > < [Hras - GTP pmods], CLi >

- EgfTC is the treatment complex formed when Egf binds to the Egf Receptor
- G is a variable ranging over Hras GEFs, representing the general knowledge that exchange of GDP for GTP requires a GEF (Guanine exchange factor).
- gmods, pmods are variables indicating that we don't know the exact state of G or Hras.
- C is a variable standing for currently unknown requirements

Inferring that Sos1 is a candidate GEF

The datum

rHras GDP-dissociation[3H-GDP] is increased by xSos1[tAb]IP
cells: none
IPfrom: HEK293 in BMS
source: 15039778-Fig-2c

reports direct GEF action of Sos1 in a test tube,

while the datum

xHras[tAb]IP GTP-association[TLC] is increased itpo xSos1
cells: HEK293 in BMS
source: 10896938-Fig-1c

reports interaction in a live cell.

The combination tells us that Sos1 is a candidate GEF for Hras.

Inferring the requirement for a Gab

The datum

Hras[Ab] GTP-association[BDPD] is increased irt Egf (times)
cells: mEFs in BMLS
times: 0 1++ 2++ 5+ min
partially reqs: Gab1 [KO]
source: 12629518(D)

tells us that Gab1 plays a role.
"Partially" indicates that Gab1 is not the only player of that role.

The extra from the previous Hras datum

inhibited by: xGab1(Y627F) [substitution]

says that some function of Gab1 that relies on Y627 is required.

A plausible conjecture is that phosphorylation on Y627 is required.

Formalizing the Datum Logic

Answer Set Programing (ASP) in one slide

An ASP is a collection of clauses of one of three forms:

(1) D. (2) D :- b1,...,bn. (3) :- b1,...,bn.

D is a ground fact or a disjunction (the D in DLV)
b is a ground fact or negated ground fact

The meaning of an ASP is a collection of Answer Sets.

Each answer set is a set of ground facts that are minimal subject to making all clauses in the program true.

We use the DLV (DataLog with Disjunction V) engine for finding answer sets by constraint solving.

Datum assertions

- datum(Dt) -- declares Dt as a datum identifier
- subject(S,Dt)
  - S is the subject of the experiment recorded by datum Dt
- assay(Aname,Aux,Dt)
  - Aname is the assay name,
  - Aux collects assay parameters, possibly none.
  - Examples: modification site, hook, substrate
- treatment(T,Dt)
  - the treatment used in the experiment, if any
- increased(Dt), decreased(Dt), unchanged(Dt)
  - the change observed. (We currently reason only about increased)
- irt(Dt), itpo(Dt), by(Dt)
  - the treatment type
- reqs(Q)
  - entity Q is required for the experimental outcome

Mapping datums (json) to assertions (in DLV)

- Equivalent datums are merged into one super datum.
  - The merged datums have the same subject, assay, treatment/treatment type, and change
  - Extras are joined
- The shared parts of merged datums map directly to assertions
- Extras require some reasoning.
- The mapping function also reports conflicts for examination by an expert.

- Mapping the two Hras datums produces:

```
datum(“Dt1-Dt2-").
subject("Hras", "Dt1-Dt2").
assay("GTP-association", none, "Dt1-Dt2").
increased("Dt1-Dt2").
irt("Dt1-Dt2").
treatment("Egf", "Dt1-Dt2").
reqs("Gab1", "Dt1-Dt2").
```

- The actual merged datum for GTP-association assays combines **51** datums from the input datum collection.

Datum Logic organization

- There are two kinds of rule
  reactRule – a subject changes state
  moveRule – a subject changes location

- A rule template is associated to relevant assay / treatment type pairs.

- For GTP-association the template is
  C < [G - gmods act], Lg > < [S - GDP pmods], Lp >
  =>
  C < [G - gmods act], Lg > < [S - GTP pmods], Lp >

- C for rule context, G for GEF, S for subject
- gmods, pmods are modification set variables
- Lg,Lp are location variables

Sample Clauses

- Interpreting datum assertions

```
inBf(X - mods(X) - GDP) :- irt(Dt), increased(Dt),
        assay(GTP-association, none, Dt), subject(X, Dt), useM(Dt).
inAf(X, mods(X) - GTP) :- irt(Dt), increased(Dt),
        assay(GTP-association, none, Dt), subject(X, Dt), useM(Dt).
in(X) :- tc(X, Dt), useM(Dt).
```

- inBf, inAf refer to the subject (S) before and after the change, in refers to the context (C)

- Connecting assertions to template variables

```
occBf(X,L(X)) :- inBf(X), reactRule .
occAf(X,L(X)) :- inAf(X), reactRule .
occ(X, L(X))  :-  in(X), not hasLocation(X).   % use a variable if unknown
occBf(X - mods(X),L) :- subject(X,Dt),useM(Dt),locBf(X,L),moveRule.
occBf(X - mods(X),L(X)) :- subject(X,Dt),useM(Dt),not hasLocBf(X), moveRule.
occAf(modBy(X - mods(X),L) :- subject(X,Dt),useM(Dt), locAf(X,L), moveRule .
```

- occBf, occAf refer to the subject part of the rule template, occ refers to the context (C)

occ is for occurrence, the PL analog to BioPax physical entities: the abstract entity, its features/modifications and its location.

Example answer set and associated rule

Answer Set

reactRule
occBf(Hras - mods(Hras) - GDP,L(Hras)),
occAf(Hras - mods(Hras) - GTP,L(Hras))
occ(Egf:EgfR-Yphos,EgfRC)
occ(Sos1 – act - mods(Sos1),L(Sos1))
occ(Gab1 - mods(Gab1),L(Gab1))

Rule

< [Hras - mods(Hras) GDP], L(Hras) > < Egf : [EgfR - Yphos], EgfRC >
< [Sos1 - act mods(Sos1)], L(Sos1) > < [Gab1 - mods(Gab1)], L(Gab1) >
=>
< [Hras - mods(Hras) GTP], L(Hras) > < Egf : [EgfR - Yphos], EgfRC >
< [Sos1 - act mods(Sos1)], L(Sos1) > < [Gab1 - mods(Gab1)], L(Gab1) >

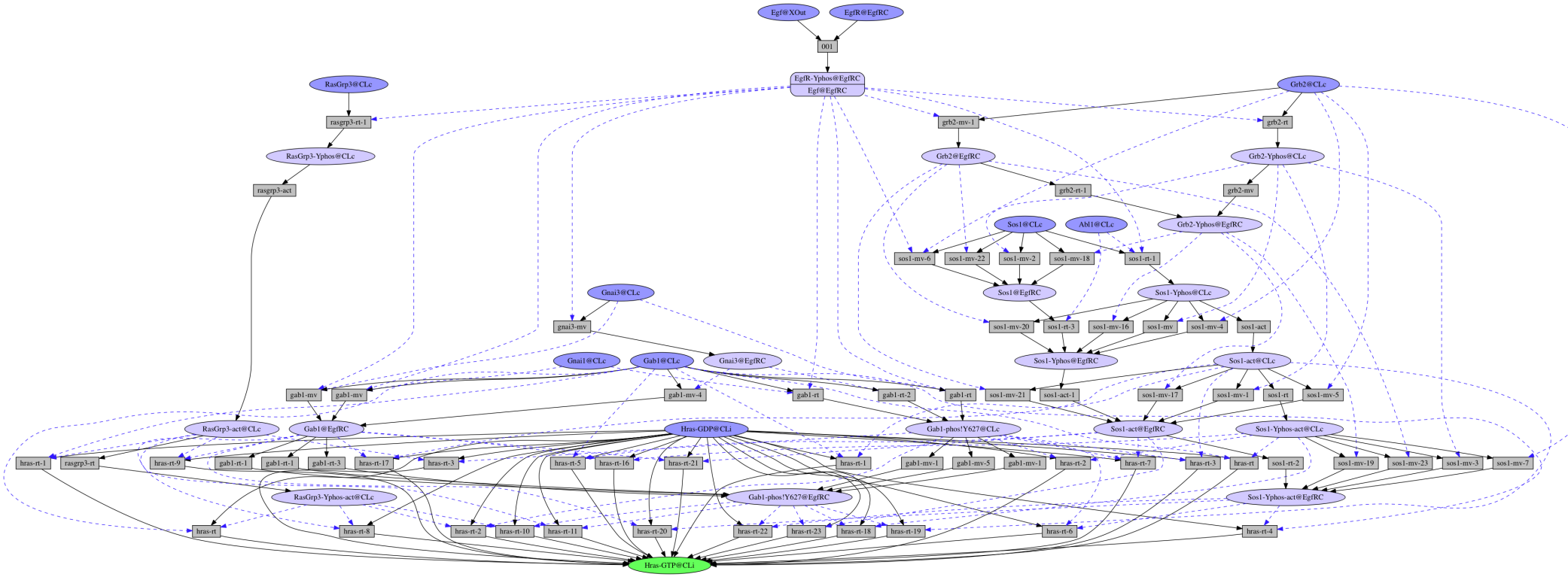
Application to Hras network datums

Inferring the Hras irt Egf model from Datums: Process

1. Select datums: evidence files for the rules in our gold standard Hras subnet plus files containing evidence for Hras GEFs.
2. Map datums to DLV assertions
3. Run DLV with assertions + core clauses to get answer sets
4. Translate answer sets to PL rules
5. Assemble PL model – non trivial
   - normalize mismatches such as Yphos => act
   - reduce combinatorial explosion due to modification variables and combinations of phos, Yphos, phos(Y 627), phos(Y 301), …
   - use PLA for derivation of concrete model from symbolic rules and a concrete initial state (dish)

2-4 are automated (this paper)
1,5 done by hand

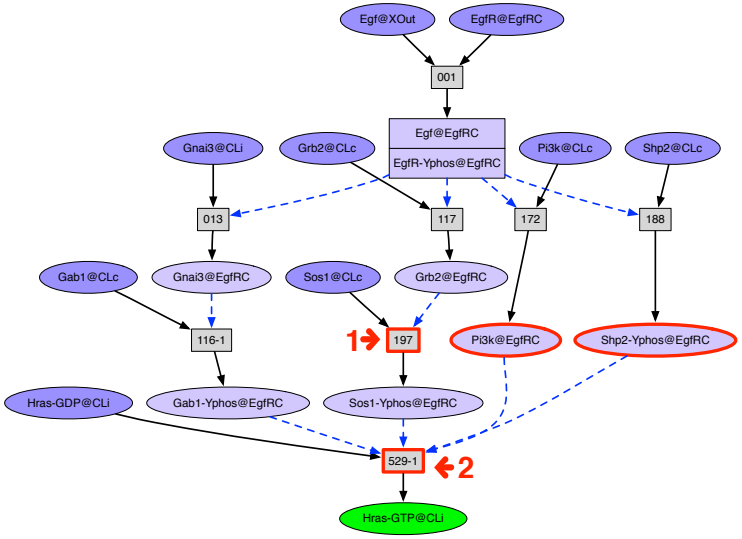
Impression of the inferred network



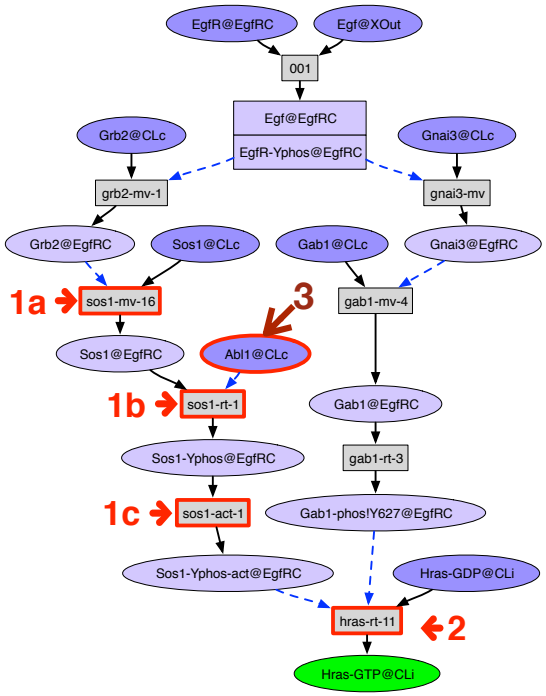
The inferred model recapitulates key properties:

- reachability
- multiple paths to the Hras-GTP goal
- (Sos1,RasGref3) as a double knockout

Pathway in gold standard model (GSM) compared to inferred net (INF)



GSM



INF

Some Differences

- Complexity due to separation of modifiation and move rules **[1]**
- Missing requirements - come from parts of datum not yet interpreted **[2]**
  inhibited by: xPik3r?(mnr)"DN" [addition]
  inhibited by: xShp2(mnr)"CIA" [addition]
- Requirement for Abl1 in inferred rule set – based on single datum **[3]**

Conclusion and Future Work

- We presented an inference system for deriving signal transduction rules from formally represented experimental findings (datums), illustrated by derivation of rules for a model of Hras activation.

- This is a step towards (partial) automation of the process of building models of cellular processes.

- There is much more todo!
  - Capture more from datums
    - reasoning about inhibition and mutation effects
    - what does decrease tell us
  - Scaling to larger models developing queries to find relevant datums
  - Automation of datum collection (NLP)
  - Automation of model assembly (from RKBs)

Questions ???