Pathway Logic

Application of Formal Modeling Techniques to Understanding Biological Signaling Processes

Carolyn Talcott
SRI International
ISB December 2005

Pathway Logic Team

- Keith Laderoute
- Patrick Lincoln
- Carolyn Talcott

- Linda Briesemeister
- Steven Eker
- Merrill Knapp
- Ian Mason
- Andy Poggio

- Alessandro Abate
- Yu Bai

- Biology Computer Science Students

PLAN

- **Symbolic** systems biology -- setting context
- Rewriting Logic
- Pathway Logic
  - Pathway Logic Models
  - Pathway Logic Assistant
- Future challenges

Symbolic Systems Biology

Biological Systems

- Biological processes are complex
- Dynamics that range over huge timescales
  - microseconds to years
- Spatial scales over 12 orders of magnitude
  - single protein to cell, cell to whole organism
- Oceans of experimental biological data generated
- Important intuitions captured in mental models that biologists build of biological processes

Symbolic Systems Biology

The ***qualitative and*** quantitative study of biological processes as ***integrated*** systems rather than as isolated parts

Goals:

- Model causal networks of biomolecular interactions and reactions in a logical framework
- Develop formal models that are as close as possible to domain expert's mental models
- Compute with and analyze these complex networks
  - Abstract and refine logical models
  - Simulate or use deduction to check properties
  - Make predictions about possible outcomes, experiment, update model

Understanding How Cells Work

Challenges

- Choosing the right abstractions
  - protein and regulatory networks are large and diverse
  - balance complexity and level of detail
  - move between levels and combine them consistently
- Composing different views or models of different components
  - biological networks combine to produce high levels of physiological organization (e.g., circadian clock subnetworks are integrated with metabolic, survival, and growth subnetworks)

Computational Biology @ SRI

- Pathway Logic
- BioSpice
- BioCyc
- Hybrid SAL
- BioDeducta
- Hormone discovery
- Sleep disorders

Modeling Landscape

- Statistical / probabilistic analysis of LARGE data sets.
  - Correlations, dependencies, patterns
- Mathematical models of processes
  - Solving equations (linear, polynomial, differential ...)
  - Numerical simulation of individual reactions
- Formal (symbolic/logical) models
  - Aspects of system represented as logical formulas expressing both structure and process
  - Logical inference used to answer queries/make predictions
  - Executable models allow to explore system behavior

Formally Based Systems
A Sampling

- Pathway Logic
- BIOCHAM
- Membrane calculi -- spatial process calculi / logics
  - Brane calculus -- mobility of membranes
  - P Systems -- mobility of processes
- Statecharts
- BioSPI, SPIM -- stochastic pi
- Hybrid SAL -- hybrid (discrete + continuous) systems

REWRITING LOGIC

What is Rewriting Logic

- A1: A logic for executable specification and analysis of software systems, that may be concurrent, distributed, or even mobile.

- A2: A logic to specify other logics or languages

- A3: An extension of equational logic with local rewrite rules to express
  - concurrent change over time
  - inference rules

What Rewriting Logic Isn't

- A rewrite theory plus a term describes a state transition system
  - states can have rich algebraic structure
  - transitions are local and possibly concurrent
- The equational part of a rewrite theory is similar to a term rewriting system (modulo ACI axioms)
  - it is usually desirable for equations to be CR and terminating
  - rewrite rules are often non-deterministic and non-terminating

Rewriting Logic Specifications

- A specification has two parts
  - A description of the structure of possible system states (as terms in a formal language)
  - Rewrite rules describing how a system might change
    - rules have the form (t => t' if C)
    - rules apply locally and concurrently, modulo equations
- Deduction = computation = rule application (rewriting)

MAUDE

- Maude is a language and tool based on rewriting logic
- See: http://maude.cs.uiuc.edu
- Features:

  Executability -- position /rule/object fair rewriting

  High performance engine --- {ACI} matching

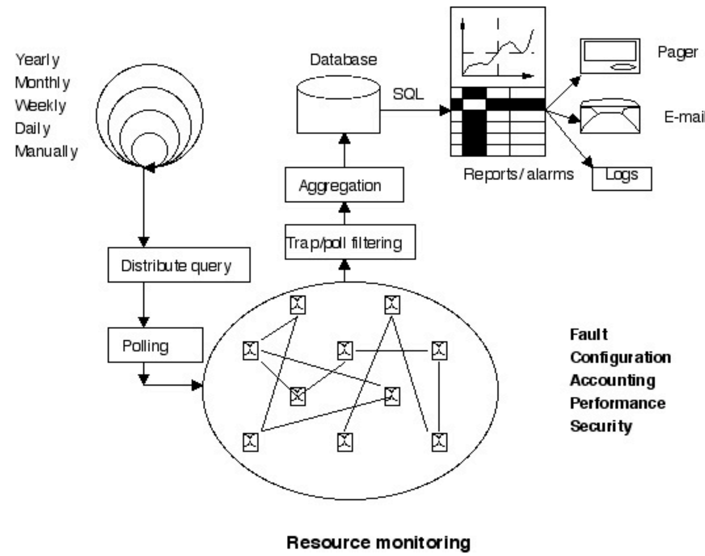
  Modularity and parameterization

  Builtins -- booleans, number hierarchy, strings

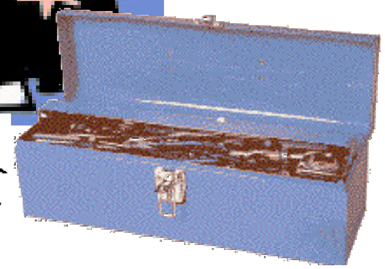
  Reflection -- using descent and ascent functions

  Search and model-checking

Maude Formal Methodology



model building

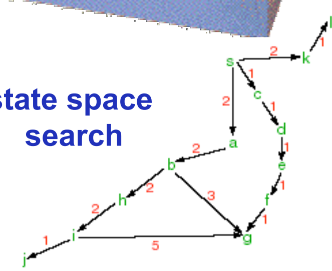


$S \models \Phi$

model checking

rapid prototyping

state space search



$S \vdash \Phi$

theorem provng

Pathway Logic (PL)

http://www.csl.sri.com/~clt/PLweb/

About Pathway Logic

Pathway Logic (PL) is an approach to modeling biological processes as executable formal specifications (in Maude)

The resulting models can be queried

- using formal methods tools: given an initial state
  - execute --- find some pathway
  - search --- find all reachable states satisfying a given property
  - model-check --- find a pathway satisfying a temporal formula
- using reflection
  - find all rules that use / produce X (for example, activated Rac)
  - find rules down stream of a given rule or component

Pathway Logic Goals

- A formal framework for developing network models that naturally express biologists intuitions.
- Integrate formal methods tools to allow working biologists interact with, query, and modify network models.
- Enable bench researchers to generate informed hypotheses about complex biological networks. For example to investigate questions such as:

“How is the network perturbed when I knockout/in gene X”.

“How does the signaling pathway activated by X interact with that activated by Y?”

PL Models

A Pathway Logic model has four parts

- Theops --- sorts and operations
- Components --- specific proteins, chemicals ...
- Rules --- signal transduction reactions
- Dishes --- initial states

THEOPS

Specifies data types used to represent cells:

- Proteins
- Complexes
- Soup --- mixtures / solutions / supernatant ...
- Post-translational modifications
- Locations --- cellular compartments refined
- Cells --- collection of locations
- Dishes --- for experiments, think Petri dish

Example Cell & Dish

```
mod CELL is inc LOCATION .
  sorts Cell CellType .
  subsort Cell < Soup .
  op [_|_] : CellType Soup -> Cell .
  op Cell : -> CellType .
  op HMEC : -> CellType .
  ...
endm
```

Example cell RRME:

[Cell | {CLi | [Hras - GTP] [Pak - act] Src }
{CLc | Raf1 1433x1 PP2a Mek [Ksr1 - phos] 1433x2 Erk }]

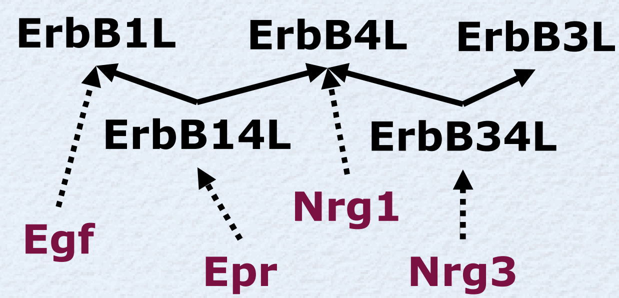
```
mod DISH is inc CELL .
  sort Dish .
  op PD : Soup -> Dish .
endm
```

Example dish: PD(Egf [Cell | {Clm | Egfr } ...])

Components: Sorts

ErbBs and their ligands

```
*** ErbBn3 is any ErbB except ErbB3
sort ErbB ErbBn3 .
subsort ErbBn3 < ErbB < Protein .
```




```
*** ErbB Ligands
sort ErbB1L ErbB4L ErbB14L ErbB34L .
subsort ErbB14L < ErbB1L < Protein .
subsort ErbB14L ErbB34L < ErbB4L < Protein .
subsort ErbB34L < ErbB3L < Protein .
```

RULES

- A PL rule specifies the change in a cell due to an enabled reaction. The rule label gives a hint as to what happens.

- In addition rules must be annotated with evidence
  - literature citations

    pubmed id (type: review, data) brief description
  - curator notes

RULE 1

A simplified description of the activation of EgfR:

If a dish contains an EgfR ligand (?ErbB1L:ErbB1L) outside a cell with EgfR in the cell membrane then the ligand binds to exterior part of the receptor and the receptor is activated.

```
rl[1.EgfR.on]: ?ErbB1L:ErbB1L
  [CellType:CellType | ct
  {CLo | clo                    } s
  {CLm | clm EgfR               } ]
  =>
  [CellType:CellType | ct
  {CLo | clo [?ErbB1L:ErbB1L - bound] }
  {CLm | clm [EgfR - act]       } ] .
  -------------------------------------
  *** 11566606(R) ErbB1Ls are AR Egf TGFa Btc Epr Hbegf
  *** 12620237(D) Crystal structure of Egf-EgfR interaction.
```

The Pathway Logic Assistant (PLA)

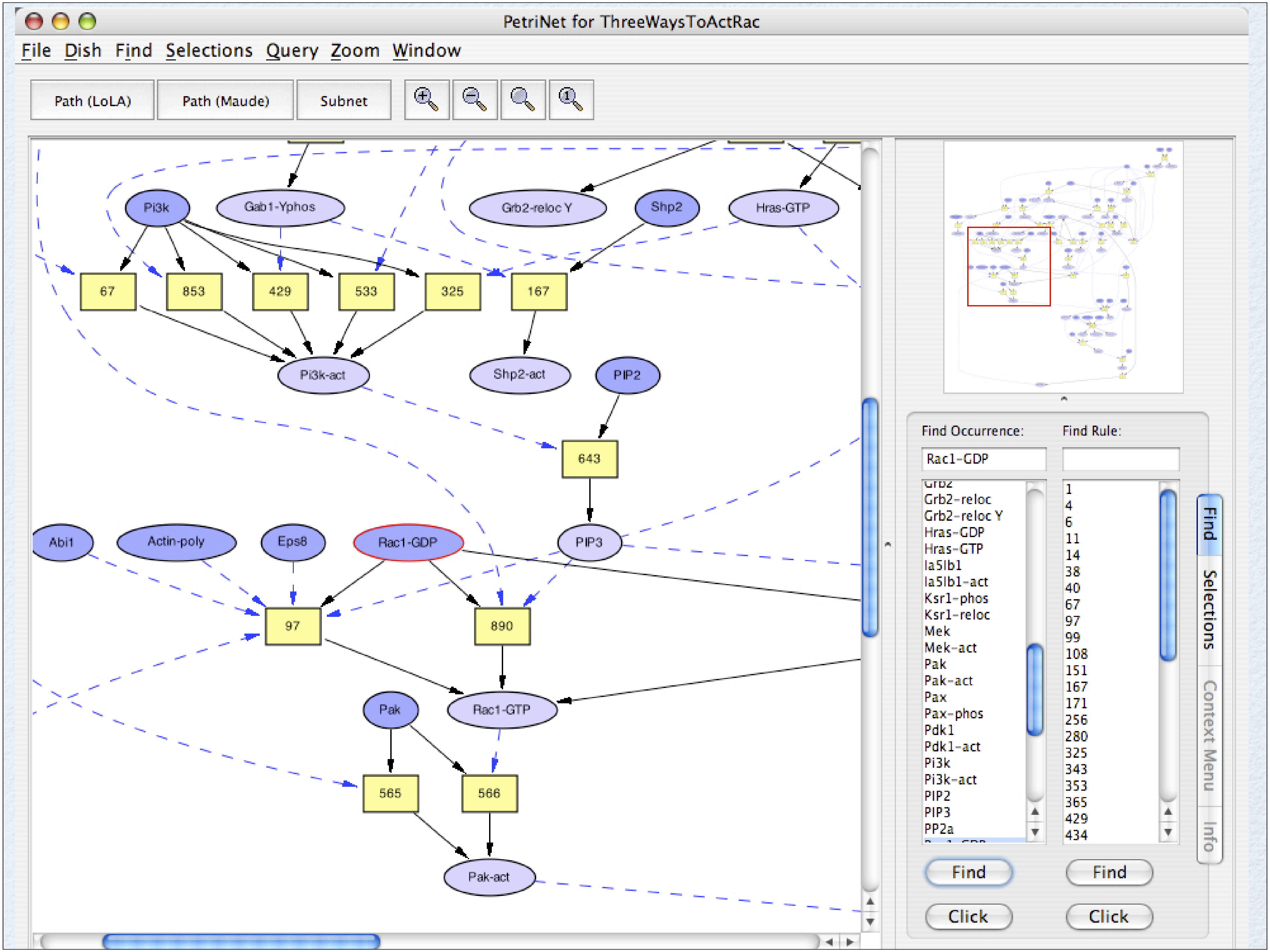
PLA I

- Provides a means to interact with a PL model
  - Inspect, Modify, Query
- Manages multiple representations
  - Maude module (logical representation)
  - PetriNet (process representation for efficient query)
  - Graph (for interactive visualization)

PLA II

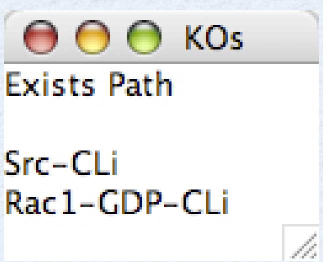
- Exports Representations to other tools
  - Lola
  - Dot -- graph layout
  - Graphics2d --- interactive visualization
  - SBML
- Imports (some) SBML based models

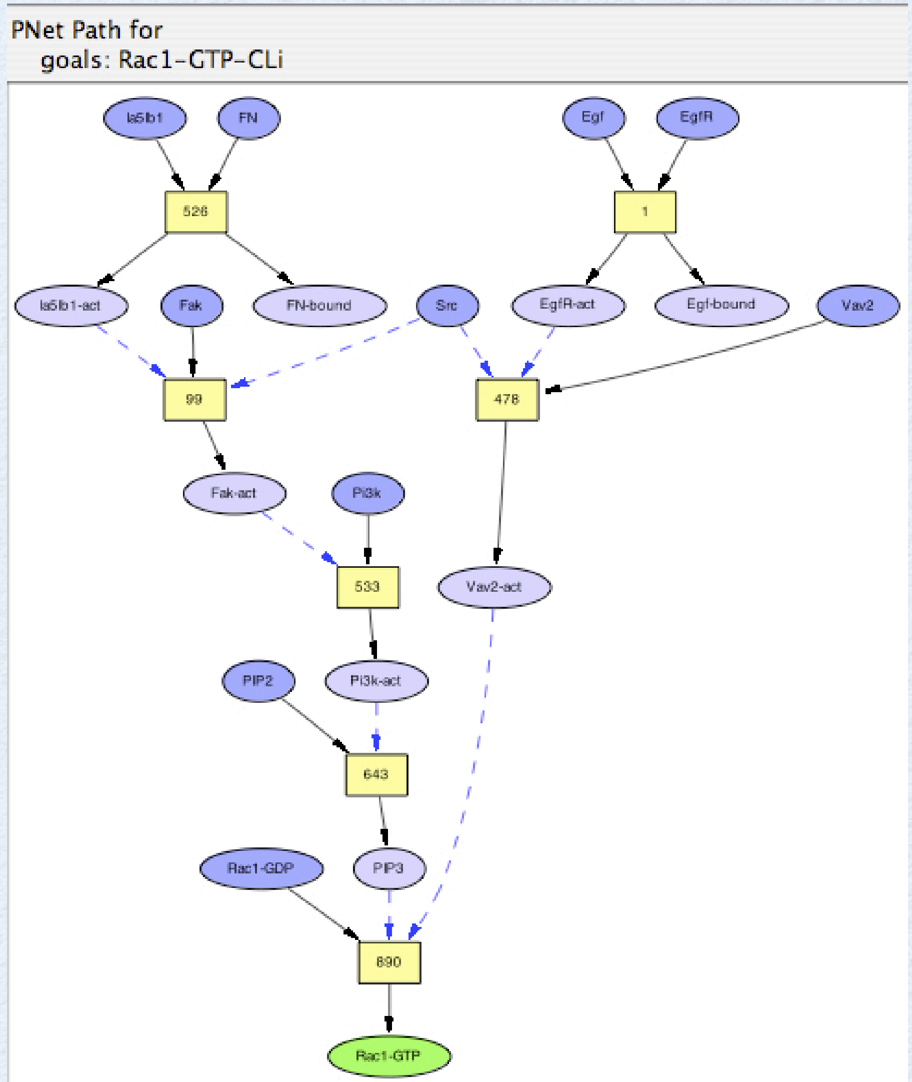
Three ways to activate RAC


PetriNet for ThreeWaysToActRac
File Dish Find Selections Query Zoom Window
Path (LoLA)
Path (Maude)
Subnet
Pi3k
Gab1-Yphos
Grb2-reloc Y
Shp2
Hras-GTP
67
853
429
533
325
167
Pi3k-act
Shp2-act
PIP2
643
Abi1
Actin-poly
Eps8
Rac1-GDP
PIP3
97
890
Pak
Rac1-GTP
565
566
Pak-act
Find Occurrence:
Find Rule:
Rac1-GDP
Grb2-reloc
Grb2-reloc Y
Hras-GDP
Hras-GTP
Ia5Ib1
Ia5Ib1-act
Ksr1-phos
Ksr1-reloc
Mek
Mek-act
Pak
Pak-act
Pax
Pax-phos
Pdk1
Pdk1-act
Pi3k
Pi3k-act
PIP2
PIP3
PP2a
1
4
6
11
14
38
40
67
97
99
108
151
167
171
256
280
325
343
353
365
429
434
Find
Find
Click
Click
Find
Selections
Context Menu
Info

Using Rule 890

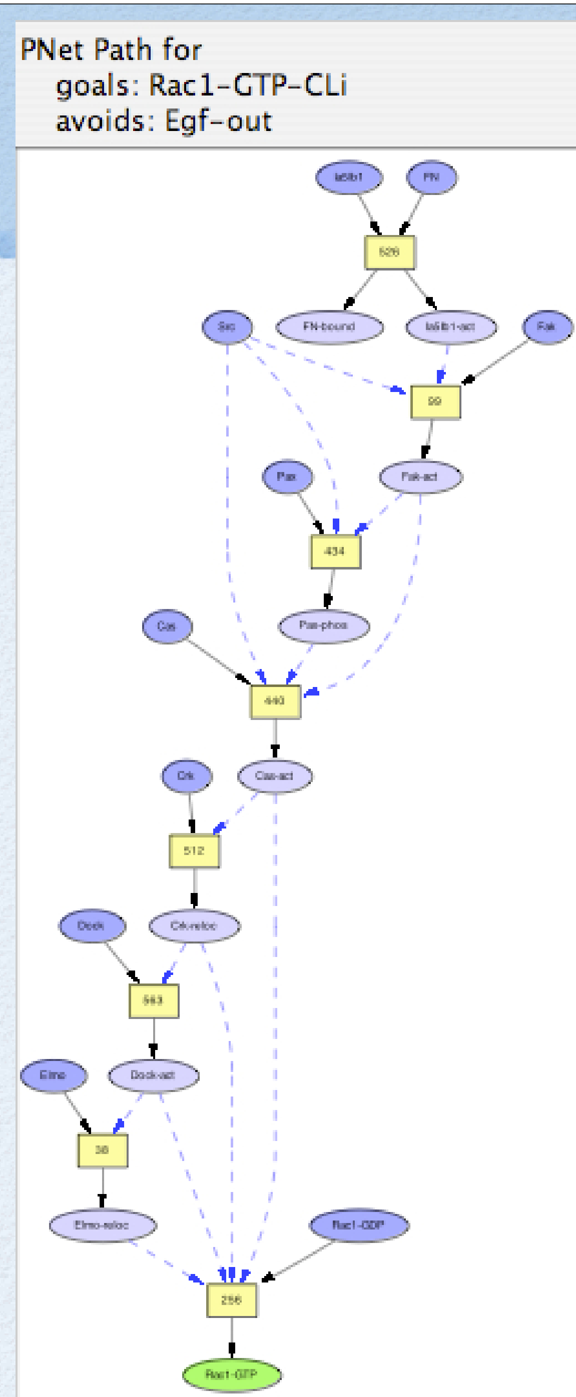
- Find Rac1-GTP
- Make it a goal
- Find path
- Compute Knockouts





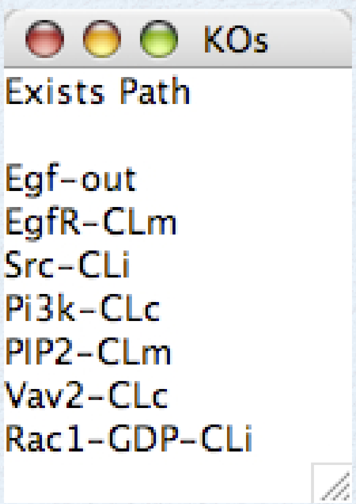
Using Rule 256

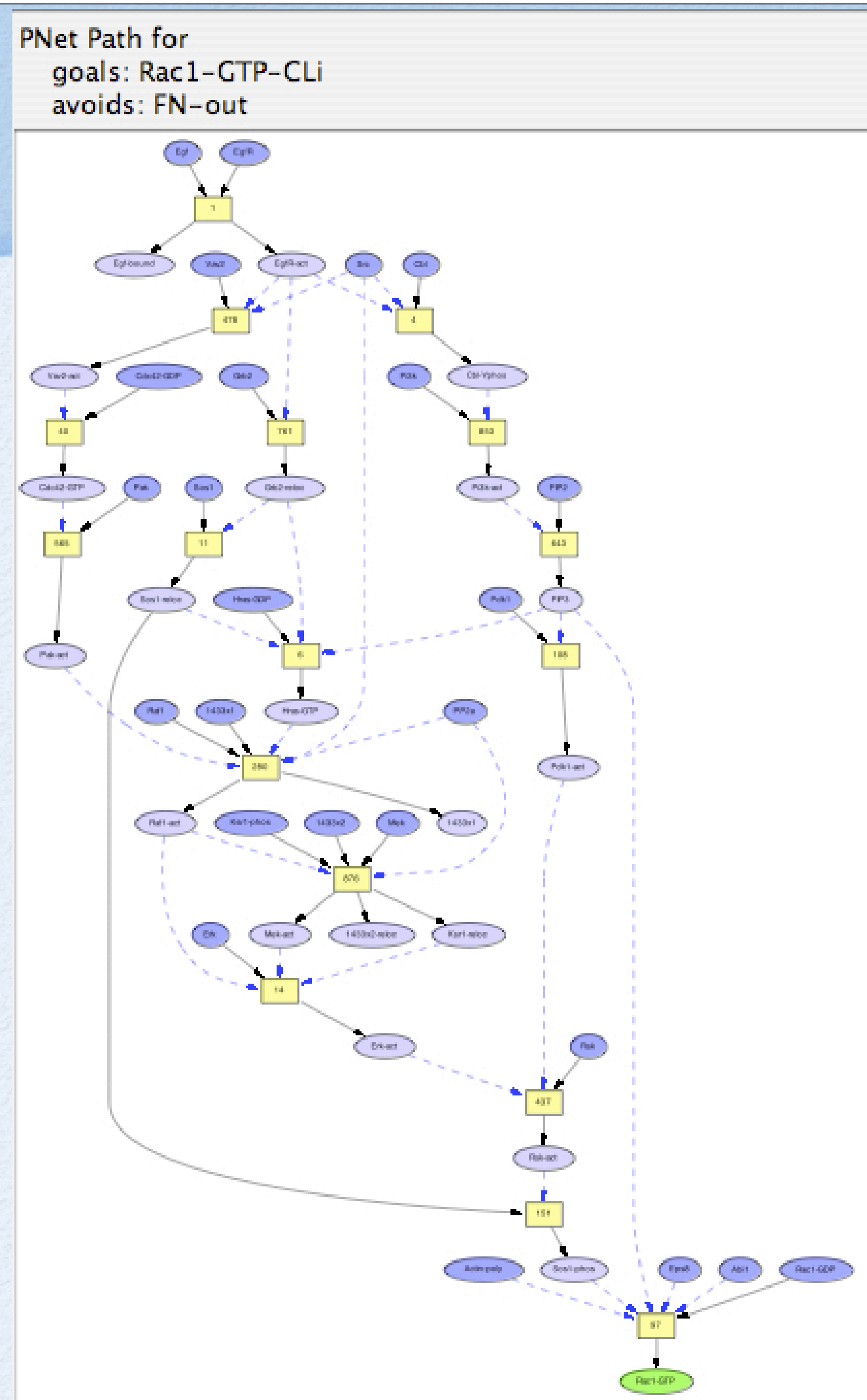
- Find Egf
- Knock it out (avoid)
- Find path



Using Rule 97

- Restore Egf
- Avoid FN and rule 890
- Find path
- Compute knockouts with no FN





FUTURE DIRECTIONS

Future Challenges I

- Scale to bigger models
  - optimize Petri net generation
  - property preserving abstractions
  - hierarchical networks
- Richer model
  - semi-quantitative information
  - more detailed representation of interactions
  - multi-cell systems
- More functionality
  - incremental path exploration
  - path relations, cross talk

Future Challenges II

- Integration of models
  - regulation / transduction / metabolism
  - quantitative and qualitative
  - time scales
  - spatial scales
- Modeling issues
  - choice/conflict -- internal, external, probablistic
  - reasoning about causality (TL not adequate)
  - representing and reasoning about what you don't know