Pathway Logic

Application of Formal Modeling Techniques to Understanding Biological Signaling Processes

Carolyn Talcott
SRI International
October 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

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 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

BIOLOGICAL SYSTEMS

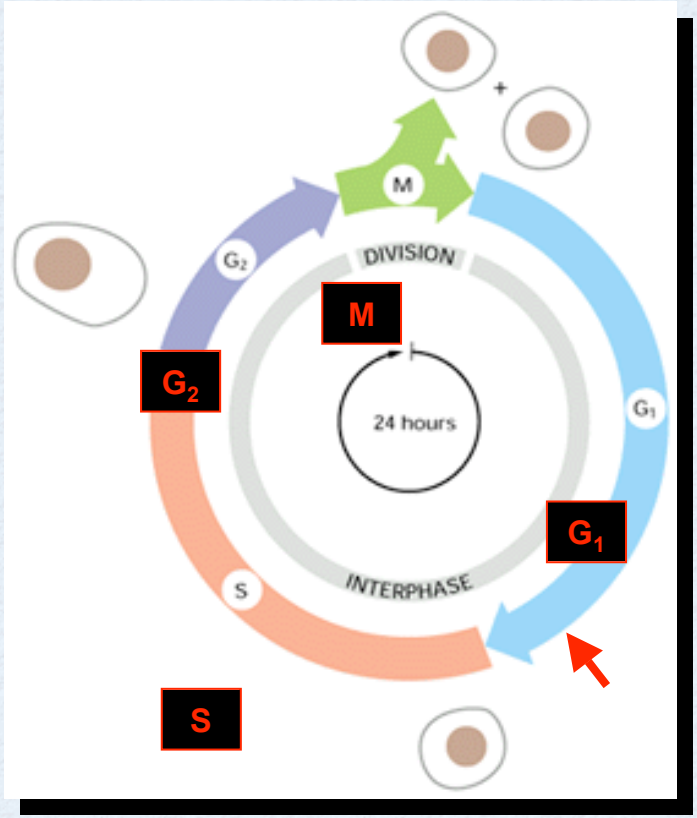
- Biological processes are complex
- Dynamics that range over huge timescales
  - microseconds to years
- Spatial scales over 12 orders of magnitude
  - single protein to cell, cell to whole organism
- Oceans of experimental biological data generated
- Important intuitions captured in mental models that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

UNDERSTANDING HOW CELLS WORK

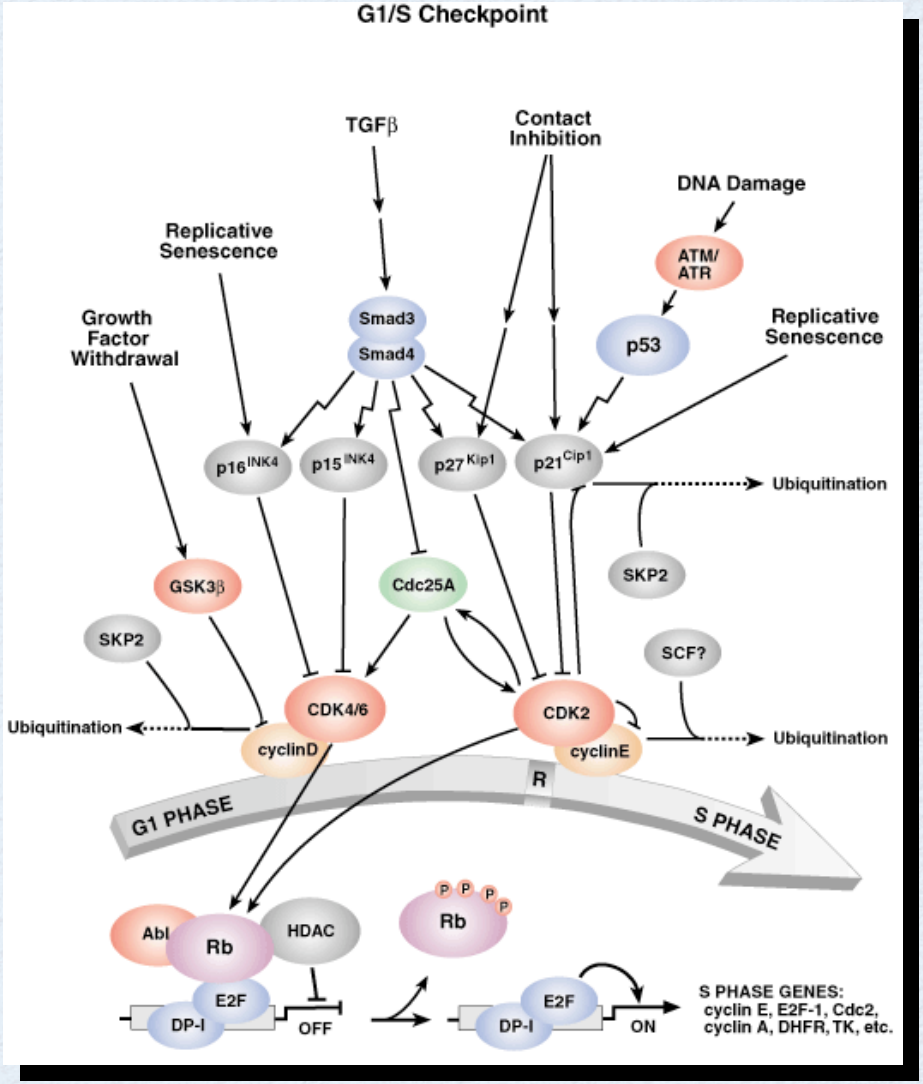
Challenges

- 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)

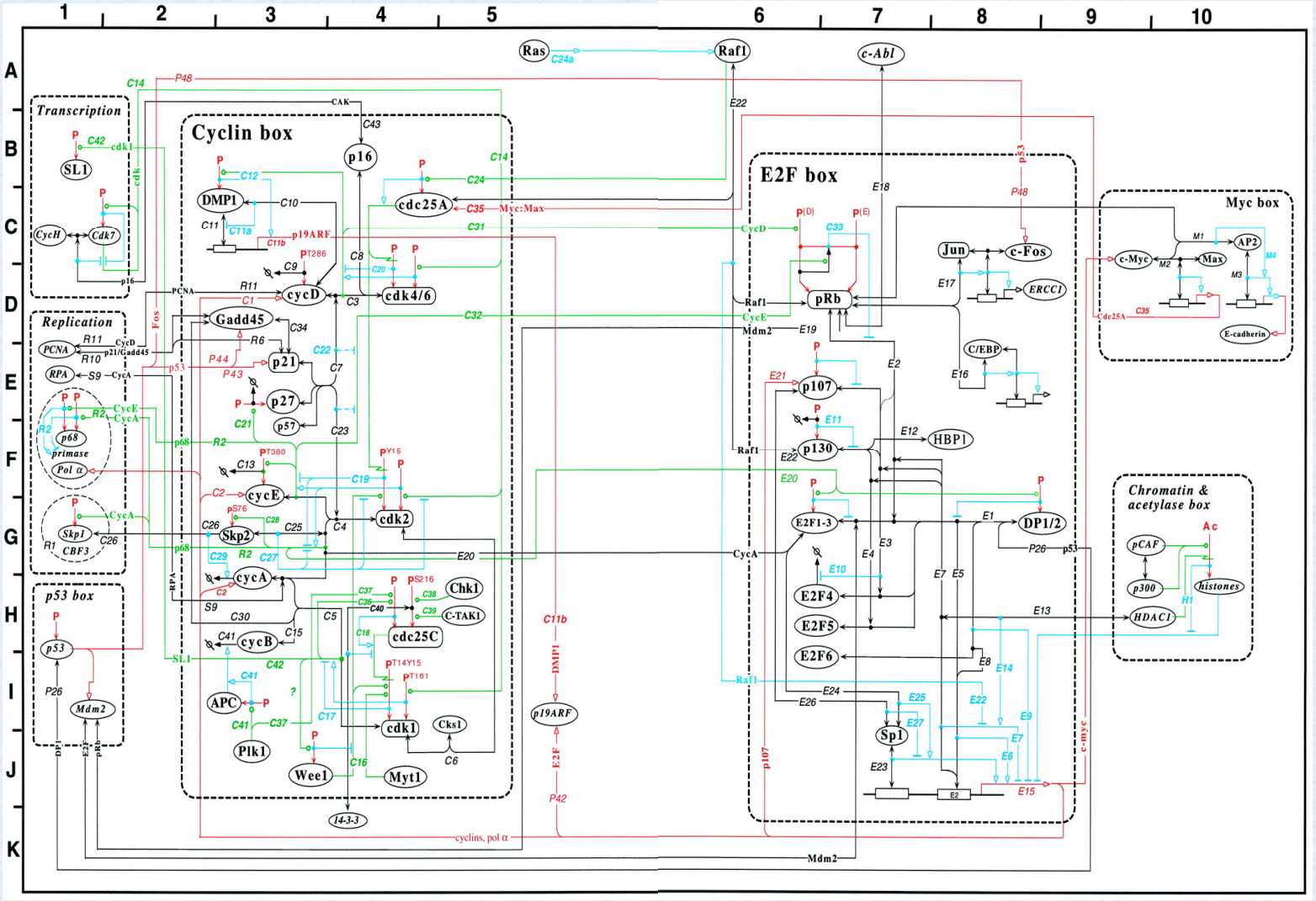
ENTRY TO CELL CYCLE



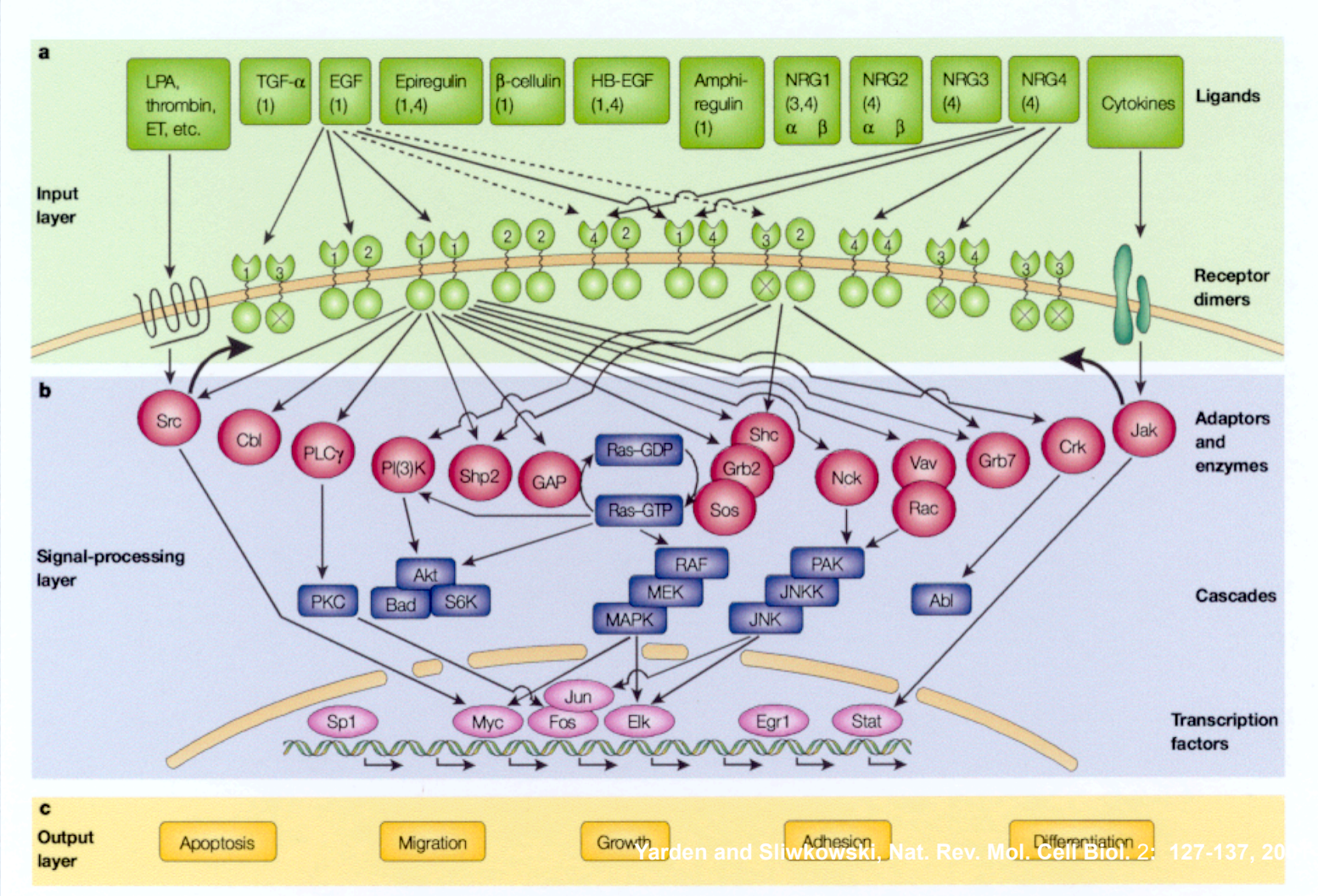
Somatic Cell Cycles Consist of Alternating DNA Synthesis (S) and mitotic (M) Phases, Separated by Gap Phases (G_1 and G_2)



CELL CYCLE CONTROL

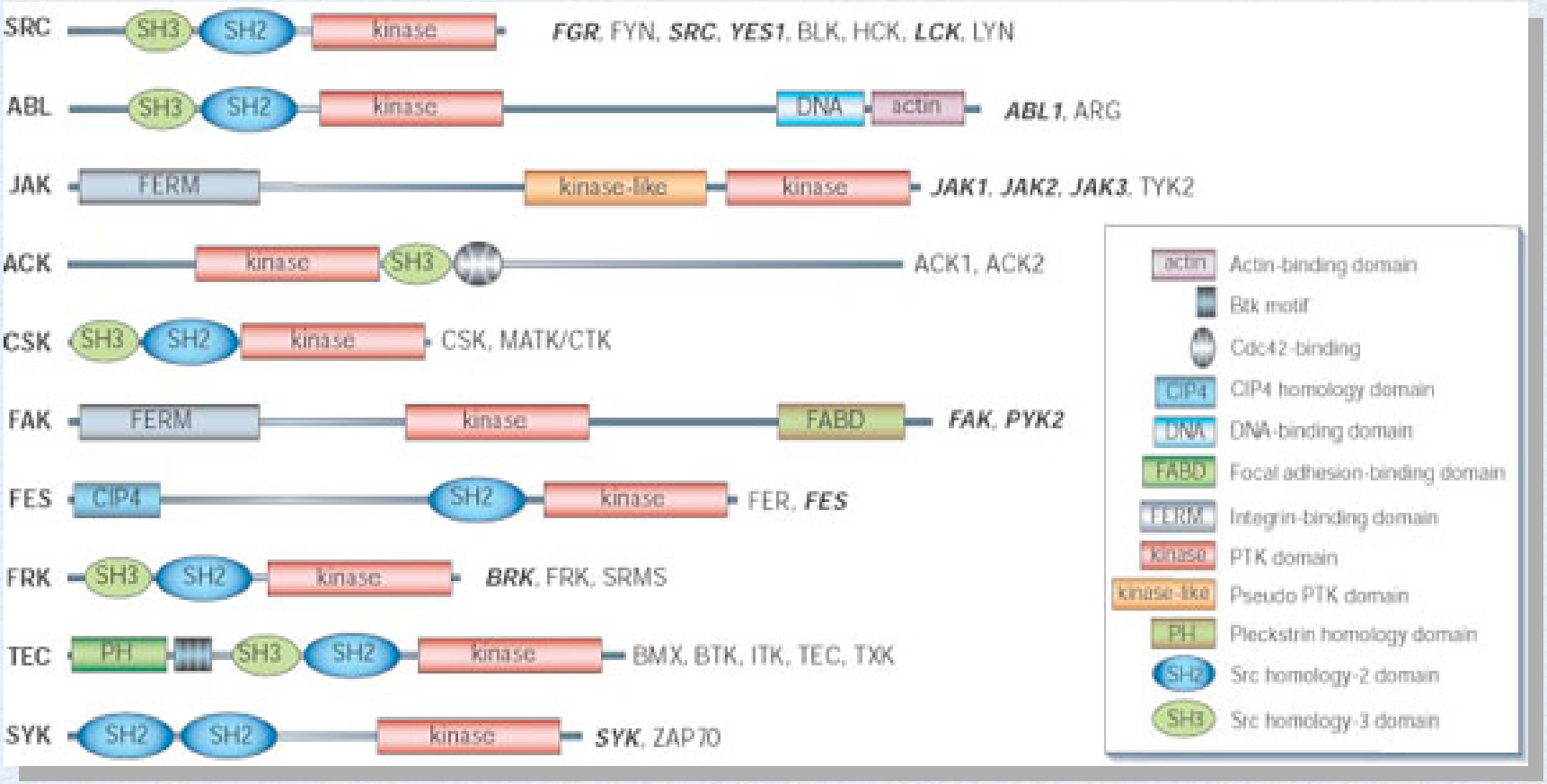


THE ERBB NETWORK



Yarden and Sliwkowski, Nat. Rev. Mol. Cell Biol. 2: 127-137, 20

PROTEINS IN MORE DETAIL



Signaling Proteins Are Collections of Domains or Modules (PFDs)

(Blume-Jensen and Hunter, Nature 411: 355-65, 2001)

How does the ErbB network work?

- What is the information flow?
- What are the controls?
- Can Jun and Fos be activated?
- What if Raf is blocked?
- How do subnets interact?
- ... a host of more detailed questions ...

Modeling Landscape

- Statistical/probablistic 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

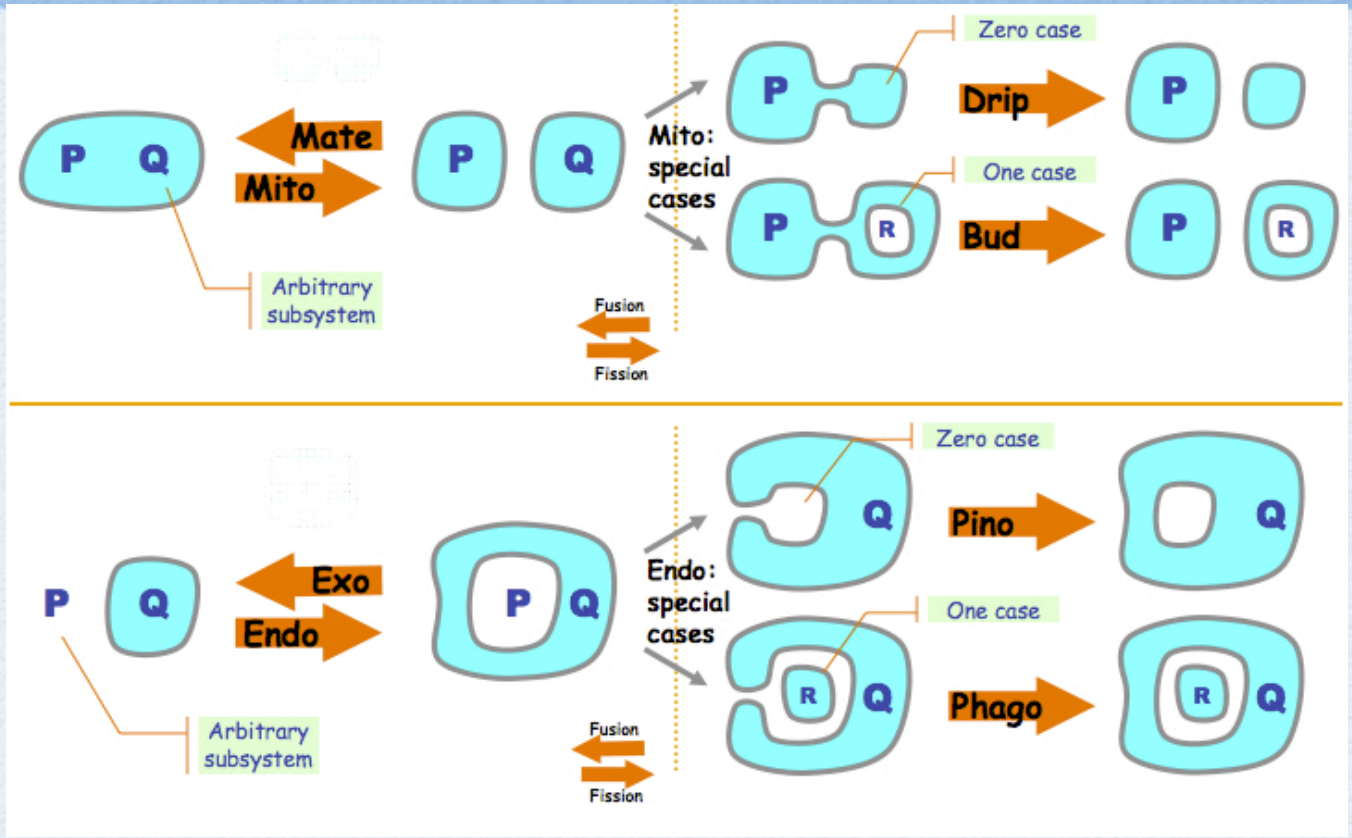
FORMALLY BASED SYSTEMS
A SAMPLING

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

BIOCHAM

- Modeling language similar to PL
  - Chemical abstract machine vs Rewriting Logic
- Multiple interpretations
  - boolean (ala PL)
  - quantitative
  - stochastic
- Richer Query Language
  - Computation tree logic: CTL vs LTL

MEMBRANE MACHINE



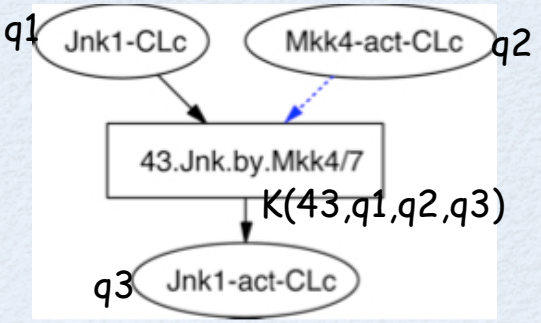
Membrane algorithms

- protein production
- viral reproduction
- LDL cholesterol degration

2005 TCSB. Cardelli. Abstract machines of systems biology

PSYSTEMS

- Elaborates PL models with quantity and rate information
- Can reason about feedback loops, effects of concentration change.
- Simulation using CLIPS
- Applied to study of EGF-ERK signalling
- Problem: numbers come from multiple cell-types and conditions



2005 CMSB. Perez-Jimenez, Romero-Campero
Modelling EGFR signalling cascade using continuous membrane systems.

HYBRID SAL

- Based on hybrid/embedded systems modeling techniques
- Abstract given quantitative model into discrete regions, based on
  - Input model
  - Property of interest
  - First and higher derivatives
- Apply variety of model checking tools
- Concretize output back to biological quantitative domain

HYBRID SAL APPLICATION

- Advantage: handles
  - uncertainty
  - multiple scales
  - quantitative/qualitative
- Need to decompose to manage complexity
- Case studies
  - Delta Notch
  - B Subtilis Sporulation
  - Insulin/glucose metabolism (whole organism)

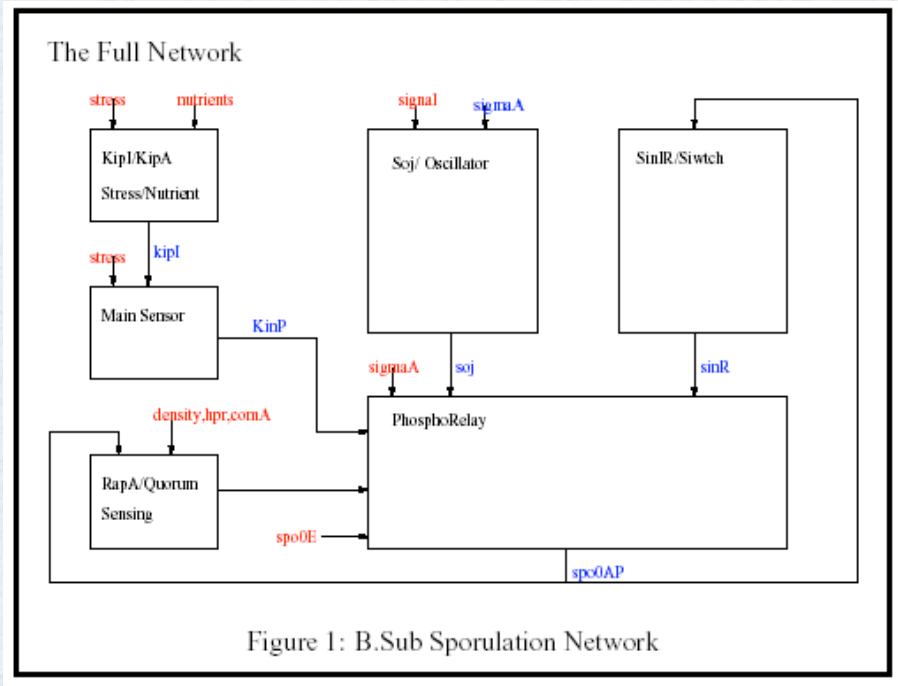


Figure 1: B.Sub Sporulation Network

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

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)

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

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

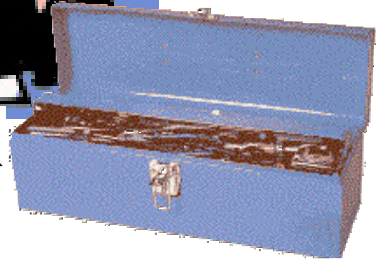
  Reflection -- using descent and ascent functions

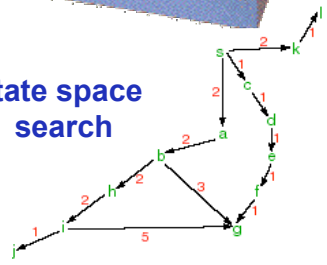
  Search and model-checking

Maude Formal Methodology

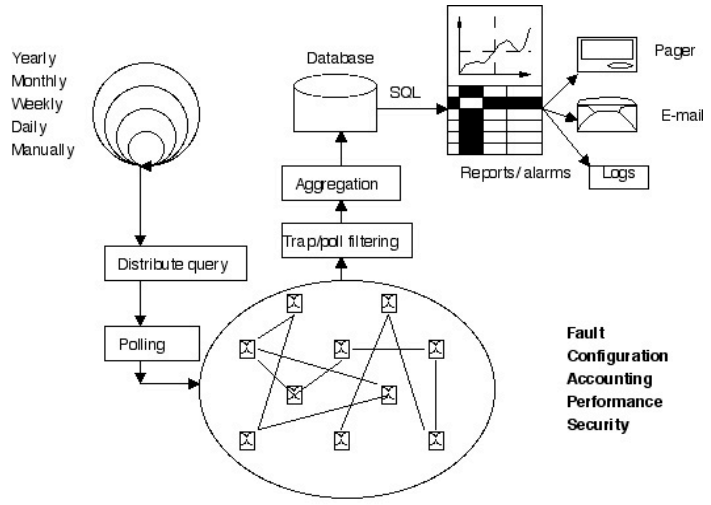
model building

rapid prototyping





state space search



Resource monitoring

$S \models \Phi$

model checking

$S \vdash \Phi$

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 based on **rewriting logic**.

- Using PL signal transduction processes can be modeled at different levels of abstraction involving:
  - the overall state of proteins, or
  - protein functional domains and their interactions
- The resulting signaling networks can be queried using formal methods tools: given an initial state
  - execute --- find some pathway
  - search --- find all pathways
  - model-check --- find a pathway satisfying a temporal formula

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

POST-TRANSLATIONAL MODIFICATIONS

```
fmod MODIFICATION is pr SOUP .
  sorts Modification ModSet .
  subsort Modification < ModSet .
  ops act bound  : -> Modification .
  ops phos Yphos : -> Modification .
    ...
  op none : -> ModSet .
  op __ : ModSet ModSet -> ModSet [assoc comm id: none] .
  op [_-_] : Protein ModSet -> Protein [right id: none ]
  op _contains_ : ModSet Modification -> Bool .
  ...
endfm
```

- Example modifications: [Raf1 - act] [Egf - bound] [Cbl - Yphos]
- Computing containment

  (bound phos) contains phos = true

  (bound phos) contains act = false

LOCATIONS

```
mod LOCATION is inc MODIFICATION .
  sorts Location LocName .
  subsort Location < Soup .
  op {_|_} : LocName Soup -> Location .

  ops CLo CLm CLi CLc : -> LocName .   *** Cell:  out,mem,in,cytosol
  ops NUo NUm NUi NUc : -> LocName .   *** Nucleus: out,mem,in,cytosol
  ....
endm
```

- Example locations
  - {Clm | Egfr }
  - {Clo | [Egf - bound] }
  - {Clc | Erk Mek (Raf1 : Rkip) }

CELLS AND DISHES

```
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 }]

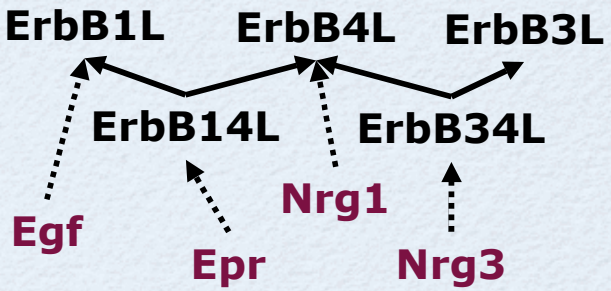
```
mod DISH is inc CELL .
  sort Dish .
  op PD : Soup -> Dish .
endm
```

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 .
```

COMPONENTS: PROTEINS

A protein declaration includes the name, subsort, and metadata specifying the SwissProt name and identifier, the Hugo name, possibly category information (for organizing menus ...) and assorted synonyms.

```
op Egf : -> ErbB1L [metadata "(\
  (spname EGF_HUMAN)(spnumber P01133)\
  (hugosym EGF) (category Ligand)\
  (synonyms \"Pro-epidermal growth factor precursor, EGF\" \
     \"Contains: Epidermal growth factor, Urogastrone\"))"] .

op EgfR : -> ErbBn3 [metadata "(\
 (spname EGFR_HUMAN)(spnumber P00533)\
 (hugosym EGFR)  (category Receptor)\
 (synonyms \"Epidermal growth factor receptor precursor\" \
    \"Receptor tyrosine-protein kinase ErbB-1, ERBB1\"))"] .
```

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.
```

RULE 280: ACTIVATION OF RAF1

```
rl[280.Raf1.on]:
  {CLi | cli [?Ras:Ras - GTP] [?Pak:Pak - act] Src  }
  {CLc | clc  PP2A Raf1   ?1433:1433                }
  =>
  {CLi | cli [?Ras:Ras - GTP] [?Pak:Pak - act] Src
             [Raf1 - act] ?1433:1433                }
  {CLc | clc  PP2A                                  } .

-------------------------------------------------------
  ***  7811320(DA) Activated Ras recruits the RAf1 complex to the CLm
  ***  9234708(D)  Raf1 requires phos on S338,S339 for act
  ***  9823899(D)  Pak3 phoses Raf1-S338
  *** 10801448(D)  Pak1 phoses Raf1-S338
  *** 10801448(D)  Raf1 reqs phos on S338 S621 not S339 for Raf1 act
  *** 10801873(D)  PP2AA and PP2AC bind to Raf1
  *** 10998357(D)  Src is needed for Raf1 act
  *** 11756411(D)  Activated Ras recruits the RAf1 complex to the CLm
  *** 12932319(D)  Activated Ras acts PP2A by recruiting PP2AB to Raf1
  ***  8999998(DA) Ras = Hras not RRas
  *** 10246825(DA) Ras = RRas
  *** 15143186(D)  Ras = Hras Nras Kras > Tc21 RRas3 Rit
  *** 15143186(D)  Ras is not RRas Rap1 Rap2 Rin Rheb
   ....
```

COMPLEXING AND TRANSLOCATION

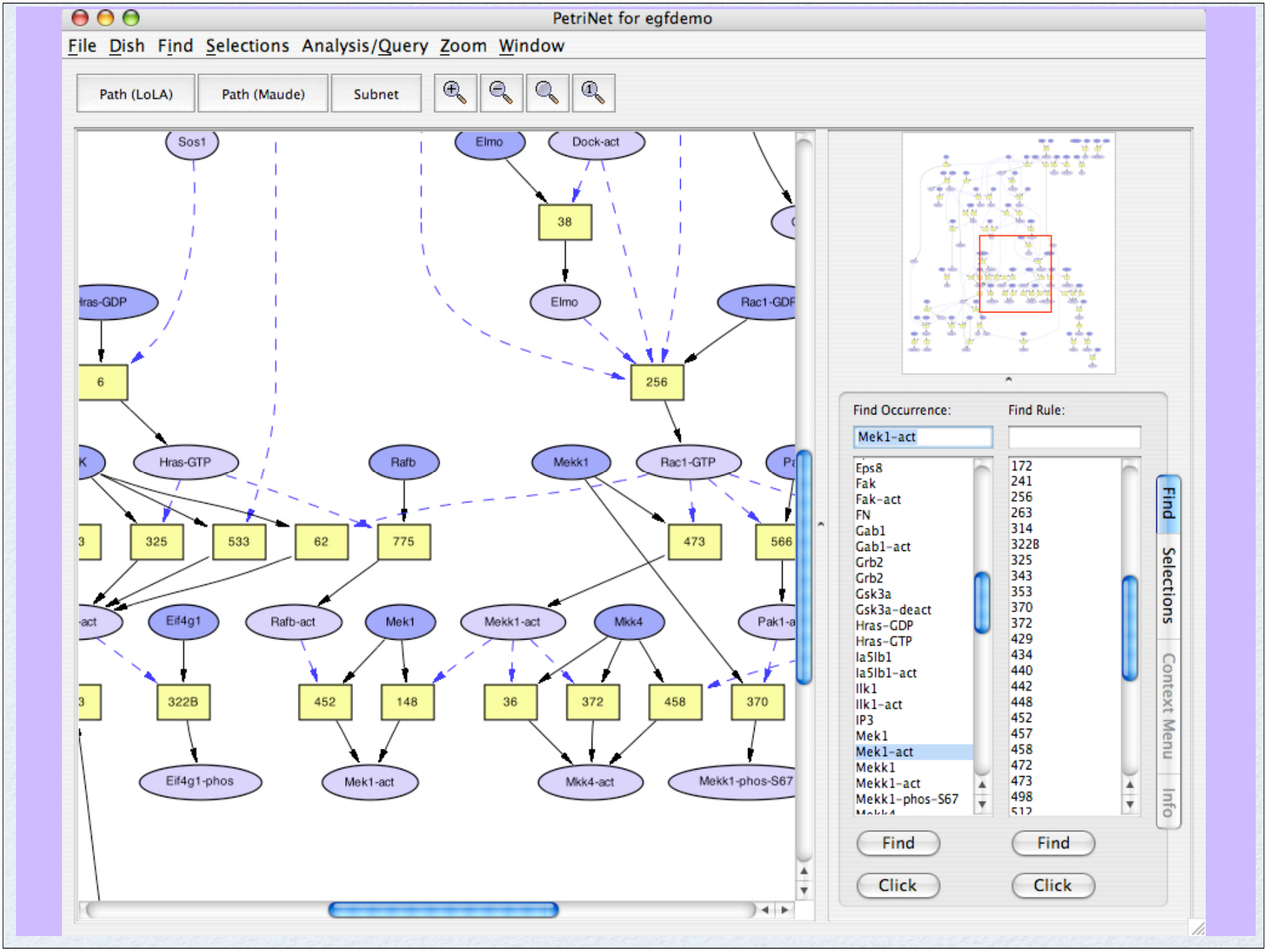
```
rl[228.Raf1.-.Rkip]:
  {CLc | clc  Raf1   Rkip  }
  =>
  {CLc | clc (Raf1 : Rkip) } .
  ----------------------------
  *** 10490027(D) 10757792(D)
  *** Rkip binds to and inhibits Raf1


rl[410.Erk.to.nuc]:
  {CLi | cli [Erk - act]  }
  {NUc | nuc              }
  =>
  {CLi | cli              }
  {NUc | nuc [Erk - act]  } .
  ----------------------------
```

*** Unregulated translocation -- currently suppressed.

The Pathway Logic Assistant (PLA)

PLA 1

- 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)


PetriNet for egfdemo
File Dish Find Selections Analysis/Query Zoom Window
Path (LoLA)
Path (Maude)
Subnet
Sos1
Elmo
Dock-act
38
Elmo
Rac1-GDP
6
256
Hras-GTP
Rafb
Mekk1
Rac1-GTP
325
533
62
775
473
566
Eif4g1
Rafb-act
Mek1
Mekk1-act
Mkk4
322B
452
148
36
372
458
370
Eif4g1-phos
Mek1-act
Mkk4-act
Mekk1-phos-S67
Find Occurrence:
Find Rule:
Mek1-act
Eps8
Fak
Fak-act
FN
Gab1
Gab1-act
Grb2
Grb2
Gsk3a
Gsk3a-deact
Hras-GDP
Hras-GTP
Ia5Ib1
Ia5Ib1-act
Ilk1
Ilk1-act
IP3
Mek1
Mek1-act
Mekk1
Mekk1-act
Mekk1-phos-S67
172
241
256
263
314
322B
325
343
353
370
372
429
434
440
442
448
452
457
458
472
473
498
Find
Selections
Context Menu
Info
Find
Find
Click
Click

PLA DEMO

FUTURE DIRECTIONS

FUTURE CHALLENGES I

- Scale to bigger models
  - optimize Petri net generation
  - property preserving abstractions
  - hierarchical networks
- Richer model
  - express semi-quantitative information
    - n-fold up/down regulation
  - integrate kinetic information
  - more detailed representation of interactions

FUTURE CHALLENGES II

- Integration of models
  - quantitative and qualitative
  - time scales
  - spatial scales
- Understanding data from diverse sources
  - genomic/proteomic
- Integration of skills
  - computational/logical models informed by biological intuitions
  - biological intuitions informed by computational/logical models