

Pathway Logic Tutorial Part II

Writing and Querying Models of Signal Transduction

<http://www.csl.sri.com/~clt/PIWeb>



The Pathway Logic Team

Linda Briesemeister

Steven Eker

Keith Laderoute

Patrick Lincoln

Merrill Knapp

Ian Mason

Andy Poggio

Carolyn Talcott

Plan

- 0 Context
- 0 Overview
- 0 Pathway Logic Models
 - Maude specification
- 0 Curating 101
- 0 Computing with a PL model: PLA Demo

Biological Systems

- 0 Biological processes are complex
- 0 Dynamics that range over huge timescales
 - Microseconds to years
- 0 Nonlinear systems
- 0 Spatial scales over 12 orders of magnitude
 - Single protein to cell, cell to whole organism
- 0 Oceans of experimental biological data generated
- 0 Biologists build mental models of important biological processes

Symbolic Systems Biology

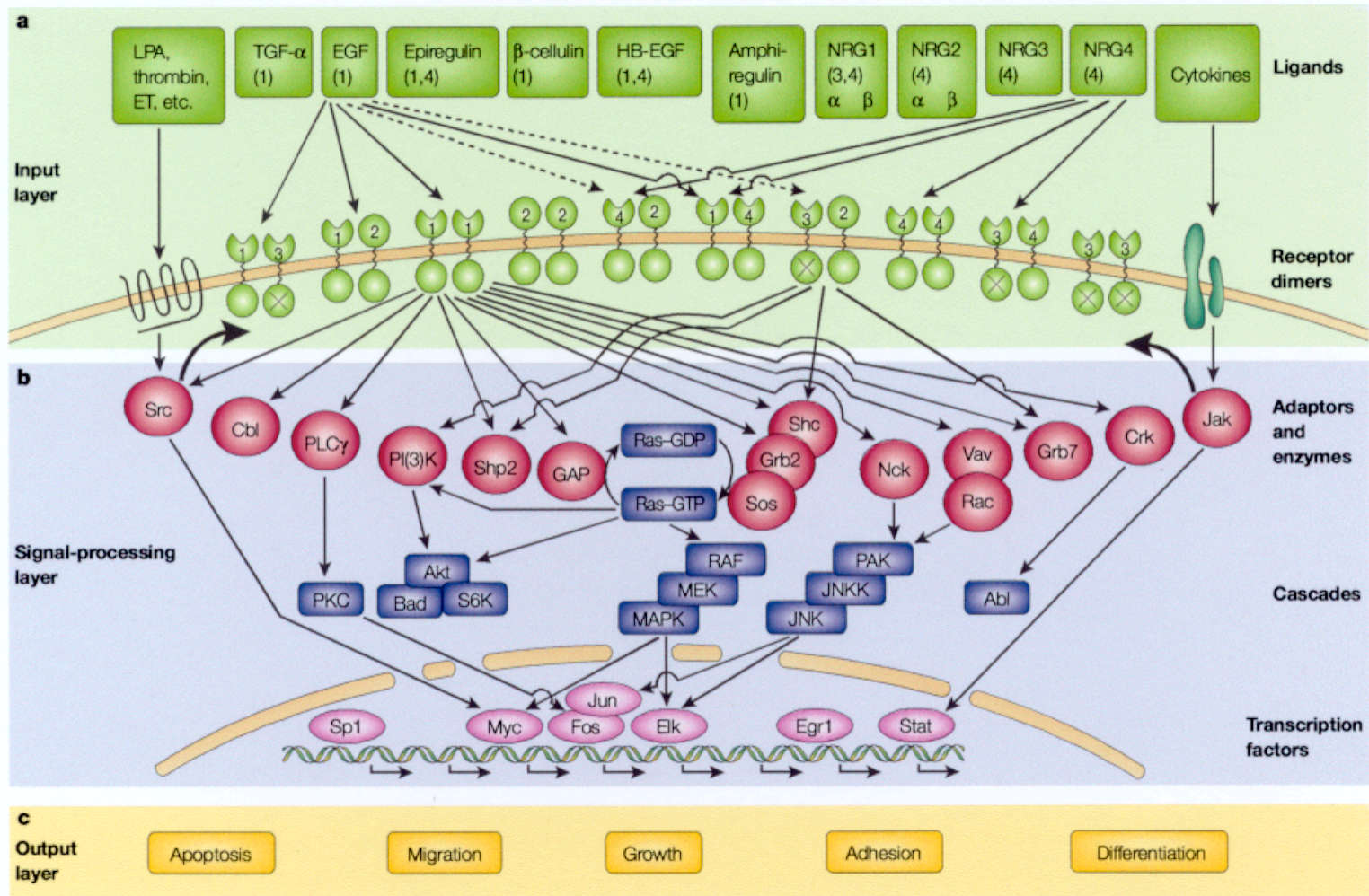
Proposed Definition:

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

Our goals

- Develop formal models that are as close as possible to domain experts' mental models
- Develop formal tools able to reason abstractly about these models
- Compose different views/models for systems level understanding (e.g., circadian clock subnetworks are integrated with metabolic, survival, and growth subnetworks)
- Produce power tools for biologists to help them refine their understanding, plan experiments, develop new theories ...

The ErbB Signaling Network



Yardon and Sliwkowski, Nat. Rev. Mol. Cell Biol. 2: 127-137, 2001
 Nat. Rev. Mol. Cell Biol. 2: 127-137, 2001

About Pathway Logic

Pathway Logic (PL) is an approach to formal modeling biological processes based on **rewriting logic**.

Processes considered include signal transduction, metabolism, inter-cellular signalling, neuron systems.

The most mature PL models are of signal transduction processes. These processes are modeled at different levels of abstraction

- the overall state of proteins, or
- protein functional domains and their interactions

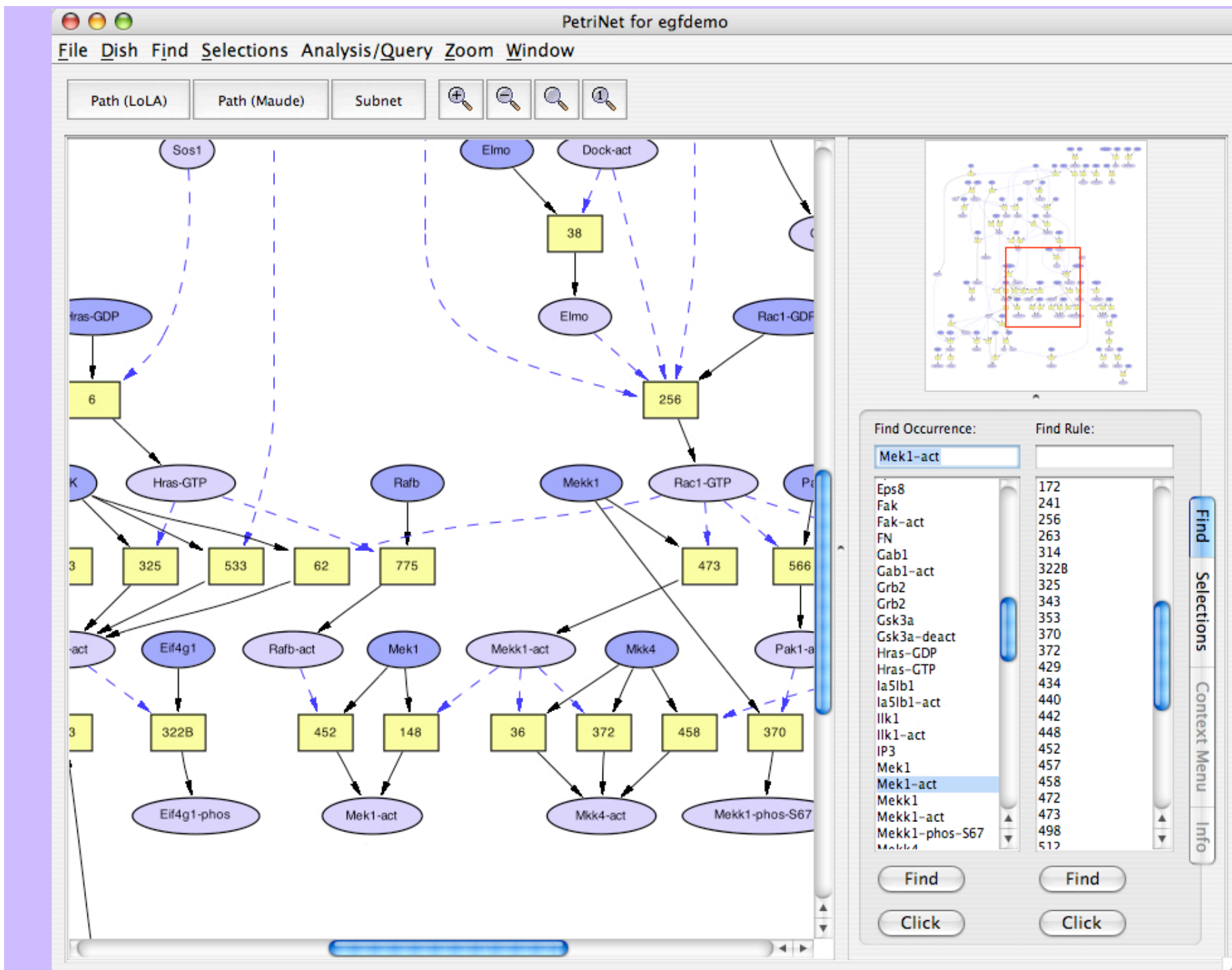
and different levels of detail, depending on available data and the questions to be asked.

The Pathway Logic Package

- o The Pathway Logic Assistant
 - builds on the IOP framework
 - Maude (meta-level) code for answering queries
 - JLambda / java code for interactive visualization
- o A demonstration model (Maude specification)
 - theops.maude --- specifies the basic data types
 - components.maude --- protein sorts and constants
 - rules.maude --- transduction rules
 - qq.maude --- initial states of interest
- o Additional models can be downloaded or developed by others and shared (for example the quiz model)

Pathway Logic Model

- 0 Cell components and state -- terms
 - Proteins/modifications -- Mekk2, Mekk2 - act
 - Cell structure -- membrane, nucleus, endosomes..
- 0 Network of reactions: set of rewrite rules
 - phosphorylation, activation, complex formation, translocation
- 0 Initial state: what proteins are present where
- 0 Pathways: assembled from initial state using rules
 - (by execution, search, model-checking)
- 0 Reactions relevant to a given state are exported as a PetriNet to take advantage of special purpose algorithms for model checking



Using PLA

- 0 Start PLA and open a dish (to display PetriNet)
- 0 Browse
 - surf using thumb nail -- drag square or click
 - click on an occurrence node, rule node, or edge
 - find an occurrence or rule
- 0 Analyze
 - set goals/avoids
 - subnet / incontext
 - path(LoLA)
 - path(Maude)
- 0 Print picture
- 0 Use menus, tools, key shortcuts, tabs ...

PL Models in Maude

Theops

Specifies data types used to represent cells:

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

Theops: Proteins

```
fmod PROTEIN is pr NAT .  
  sorts AminoAcid Protein .  
  subsort AminoAcid < Protein .  
  ops T Y S K P N : -> AminoAcid .  
  ....  
endfm
```

Some example proteins (see components for
declarations)

- 0 Egf (sort ErbB1L) EgfR (sort ErbB)
- 0 Raf1 Rkip Mek Erk Cbl

Theops: Things

```
fmod THING is pr PROTEIN .  
  sorts Thing Complex Chemical ...  
  subsorts Protein Complex Chemical ... < Thing .  
  op (_:_) : Thing Thing -> Complex [comm] .  
endfm
```

Example chemicals: Ca^{++} PIP2 PIP3

Example complex : (Raf1 : Rkip)

Theops: Soup

```
fmod SOUP is pr THING .  
  sort Soup .  
  subsort Thing < Soup .  
  op empty : -> Soup .  
  op ___ : Soup Soup -> Soup [assoc comm id: empty ] .  
  op _has_ : Soup Thing -> Bool .  
  .....  
endfm
```

Example soup: Erk Mek (Raf1 : Rkip)

Checking if a soup `has' a given thing

- 0 Erk Mek (Raf1 : Rkip) has Mek = true
- 0 Erk Mek (Raf1 : Rkip) has Raf1 = false
- 0 Erk Mek (Raf1 : Rkip) has Egf = false

Theops: Posttranslational 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

- 0 (bound phos) contains phos = true
- 0 (bound phos) contains act = false

Theops: 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

- 0 {CLm | Egfr }
- 0 {CLo | [Egf - bound] }
- 0 {CLc | Erk Mek (Raf1 : Rkip)

Theops: Cells

```
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 .
```

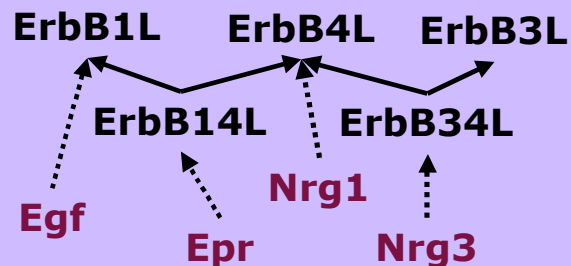
Example dish: PD(Egf [Cell | {Clm | Egfr } ...])

Components: Sorts

ErbBs and their ligands

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

sort ErbB1L ErbB4L ErbB14L ErbB34L .  *** ErbB Ligands
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

- o literature citations
 - pubmed id (type: review, data) brief description
- o 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.

```
r1[1.EgfR.on]: ?ErbB1L:ErbB1L
```

```
  [CellType:CellType | ct
```

```
    {CLO | clo
```

```
    {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 in words

If a cell contains:

any activated Ras ([?Ras:Ras - GTP]) in the CLi

any activated Pak ([?Pak:Pak - act]) in the CLi

Src in the CLi

unactivated Raf1 (Raf1) in the cytoplasm (CLc)

any 1433 (?1433:1433) in the CLc

any PP2A in the CLc

then:

Raf1 and 1433 will translocate to the CLi

Raf1 will become activated ([Raf1 - act])

Rule 280: Activation of Raf1 in Maude

```
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
....
```

Rule 280: Comments

Use of variables ie "?Ras:Ras"

The rule says that any Ras will work in this rule. This is an approximation. It should be interpreted as some Ras works, and we don't know which one.

The accuracy of the model depends on the specificity of the tools used in the experiments.

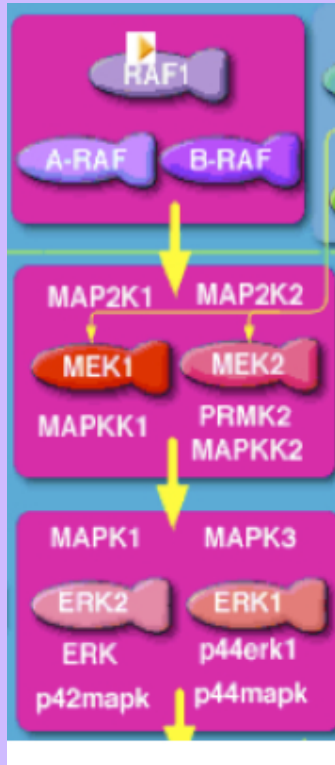
Why use variables for Ras and Pak but not PP2A?

PP2A is a complex made up of different combinations of three subunits - all of which have more than one isoform. The number of possible combinations is large and until there is information to distinguish their behavior it would only add clutter.

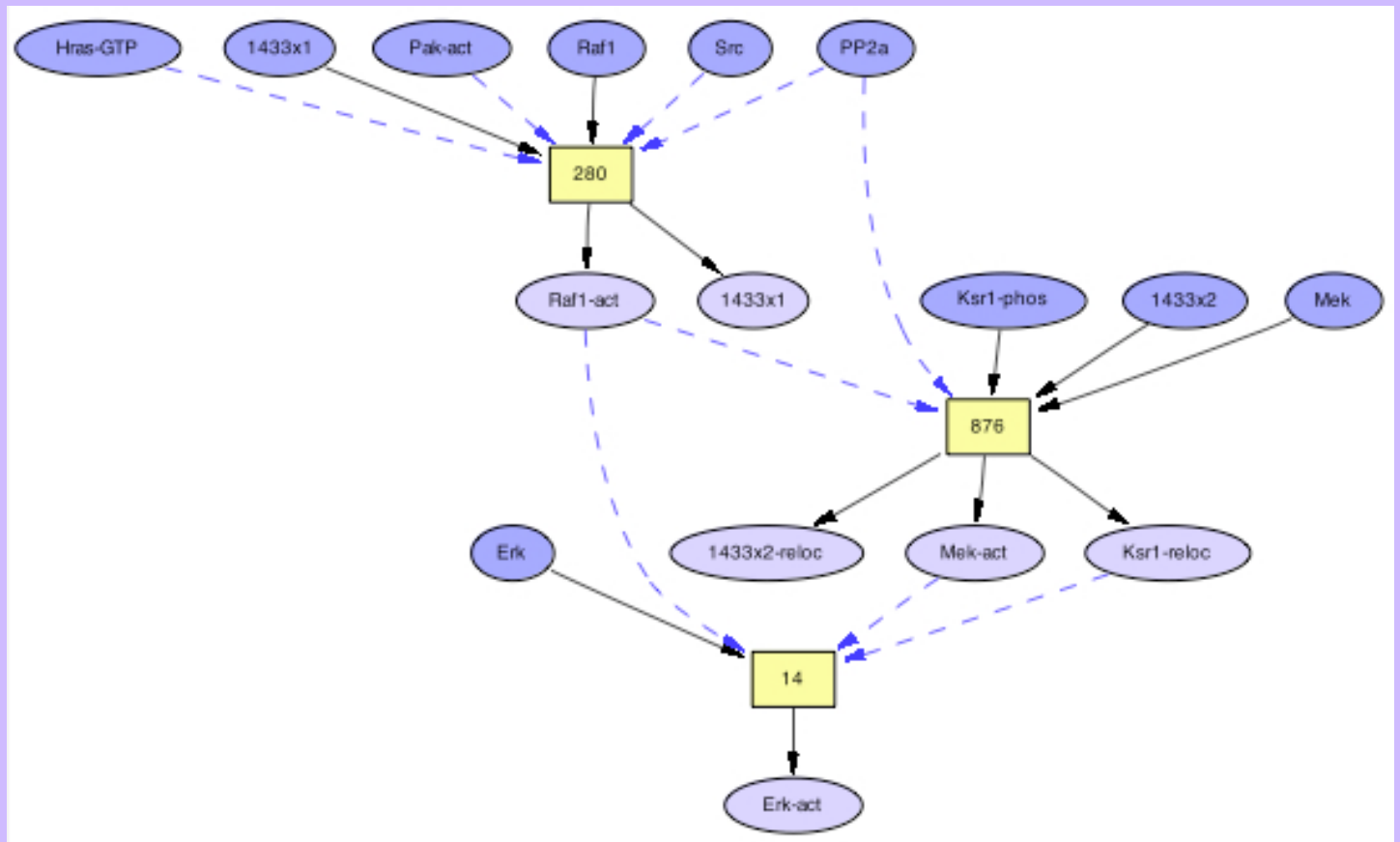
See annotated-rules.txt for more.

Chaining rules into pathways

Raf Mek Erk Pathway (RRME)



Biocarta
version



Pathway Logic version

Mek activation

If a cell contains:

activated Raf1 ([Raf1 - act]) in the CLi

unactivated Mek1 or Mek2 (?Mek:Mek) in the CLc

Ksr1 phosphorylated at S392 in the CLc ([Ksr1 - phos(S 392)])

any 1433 (?1433:1433) in the CLc

then:

Ksr1 will be dephosphorylated by PP2A

Mek, Ksr1, and 1433 will translocate to the CLi

Mek will be activated

rl[876.Mek.on]:

```
{CLi | cli [Raf1 - act] }
```

```
{CLc | clc PP2A ?Mek:Mek [Ksr1 - phos] ?1433:1433 }
```

=>

```
{CLi | cli [Raf1 - act] [?Mek:Mek - act] Ksr1 ?1433:1433 }
```

```
{CLc | clc PP2A } .
```

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.
```


Curation

Exercise: Rule curation

- 0 Task: create a rule from
 - Tyrosine Phosphorylation of Cbl upon Epidermal Growth Factor (EGF) Stimulation and Its Association with EGF Receptor and Downstream Signaling Proteins
 - Toru Fukazawa, Sachiko Miyake, Vimla Band
 - J Biol Chem 271(42):14554--14559, 1996 (PMID 8662998)
- 0 Materials (in folder quiz):
 - Reference as .pdf file (86-62-998.pdf)
 - Maude files:
 - theops (all of it)
 - components (cut to quiz-only stuff)
 - rules.maude (cut to quiz-only stuff)
 - qq.maude (cut to quiz-only stuff)
- 0 Hint: the rule concerns phosphorylation of the adaptor protein Cbl on a tyrosine

Curation rules: modification

Which modification to use: act, Yphos, reloc,

- o "act" is used for a kinase which is capable of phosphorylating its substrate i.e. (activated) and we don't know or don't care just how it is activated.
- o "Yphos" stands for "phosphorylated on tyrosine". It is used on for adapter proteins, such as Cbl, that require phosphorylation to bind to other proteins.
- o "reloc" stands for "relocated". It is used for adapter proteins that are recruited to another compartment but are not phosphorylated.

Curation rules: category

How do I tell whether a protein is a kinase or an adapter protein?

- o Look it up.
 - A good place to go is the Human Protein Resource Database. A kinase will have a kinase domain.
 - Another place is Gene Ontology - but be careful - GO is not always reliable and doesn't often give source references to double check.
 - Ask google! (also with care)
- o Infer it from the reference (activity assays).
- o Infer it from the other rules. Cbl is used in rule 2 as [Cbl - Yphos].

Curation rules: location

How do I know what compartment to use?

- 0 A general default for adapter proteins is that they start undmodified in the CLc and are recruited somewhere else.
- 0 In the quiz case, the data says that Cbl co-precipitates with EgfR which lives in the CLm. A protein that is located in a membrane tends to stay in the membrane. So Cbl probably is recruited to the CLi.
- 0 Look at other rules to see where it appears. Rule 2 has Cbl phosphorylated in CLi.
- 0 Look for similar rules. Rules with the same pattern: 4,5,10,11,13, and 14 all show things that are recruited to the CLi after EgfR is activated.

PLA Demo