

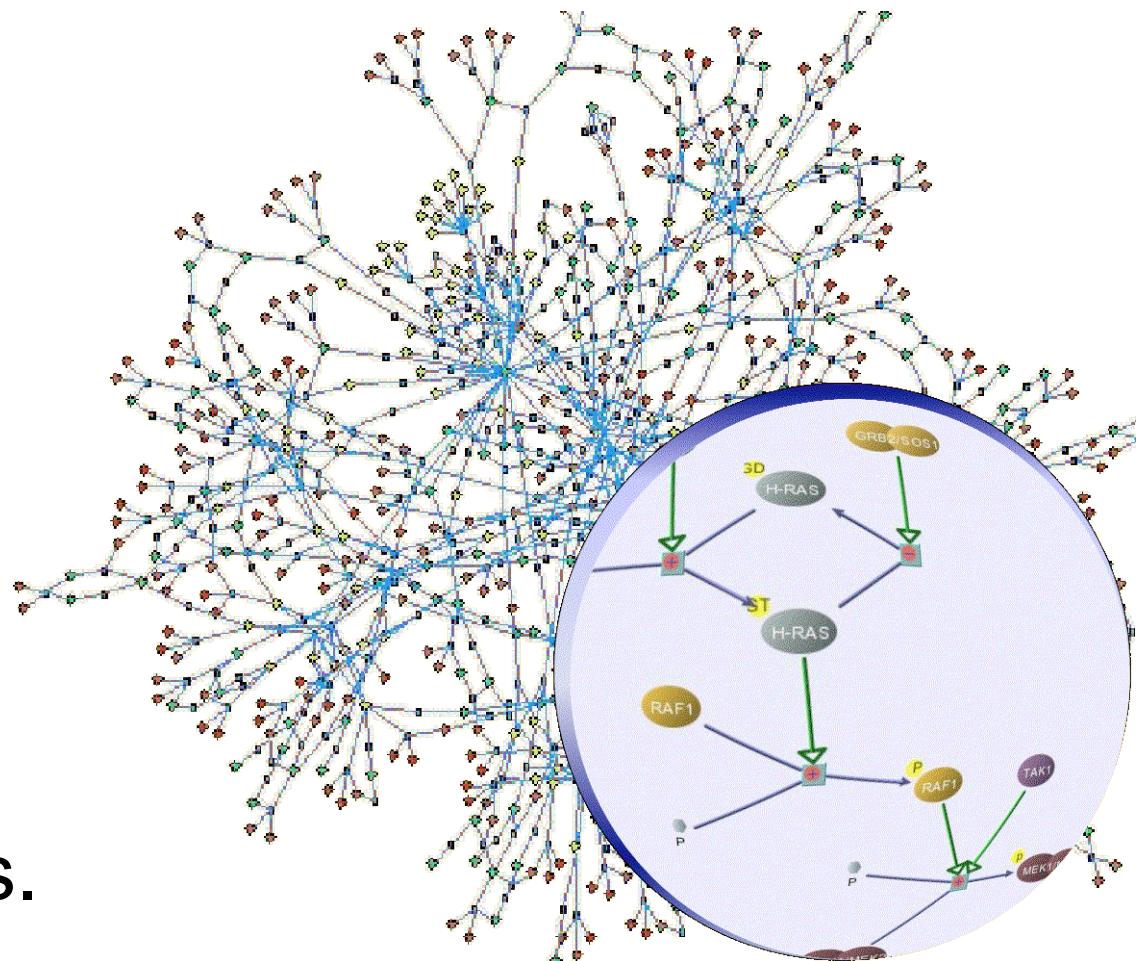
Pathway Ontology, Visualization Notation & Methods in PATIKA



By *PATIKA Team*,
Bilkent Center for Bioinformatics, Ankara, Turkey

PATIKA project *motivation*

- Essential being able to *store, integrate, access, and analyze* huge amounts of **pathway data** effectively with the help of tools.



PATIKA project goals & work-packages

- The project aims include development of valuable tools for
 - efficient data analysis through **effective visualization**,
- The project work-packages include
 - define an **ontology**,
 - develop **software tools** and construct a **database** using this ontology along with an effective environment for *pathway data_storage, integration, query, visualization and analysis*.

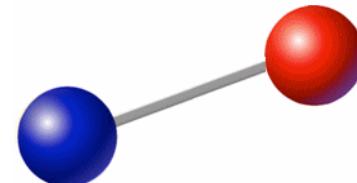
PATIKA ontology *basics*

[Demir et al, *Bioinformatics*, 2002 & 2004]

■ Graph-based representation at two levels:

□ Bio(logical)entity level:

- **genetic** (e.g. proteins), **chemical** (e.g. ions, ATP, lipids) or **physical** (e.g. light, heat, radiation) entities
- **interactions** such as PPI and transcriptional regulation



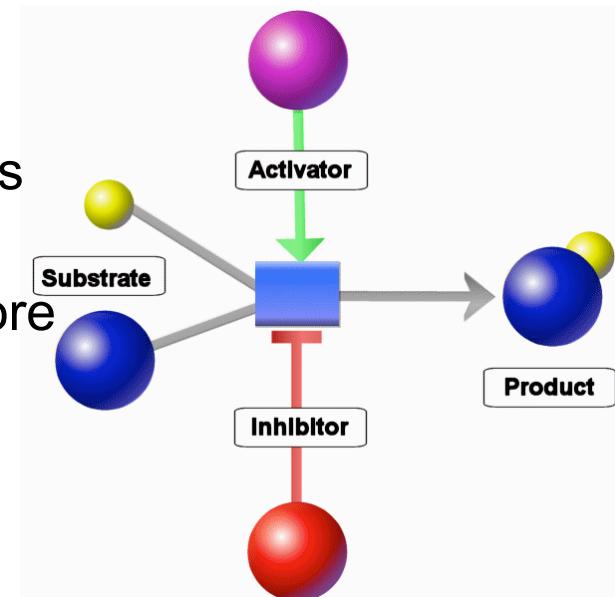
□ Mechanistic level

PATIKA ontology *basics*

[Demir et al, *Bioinformatics*, 2002 & 2004]

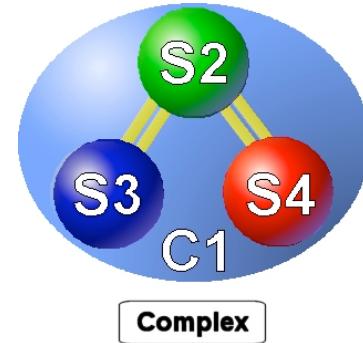
■ Graph-based representation at two levels:

- Bio(logical)entity level
- Mechanistic level:
 - bioentities have **states** (native, phosphorylated and MDM2-bound states of p53)
 - every state is associated with one or more **transitions**
 - a state is either *substrate*, *product*, or *effector* (activator or inhibitor) of a particular transition



PATIKA ontology *molecular complexes*

- Molecular complex (formed by a group of states with certain bind relations between them) [**use of compound graphs**]

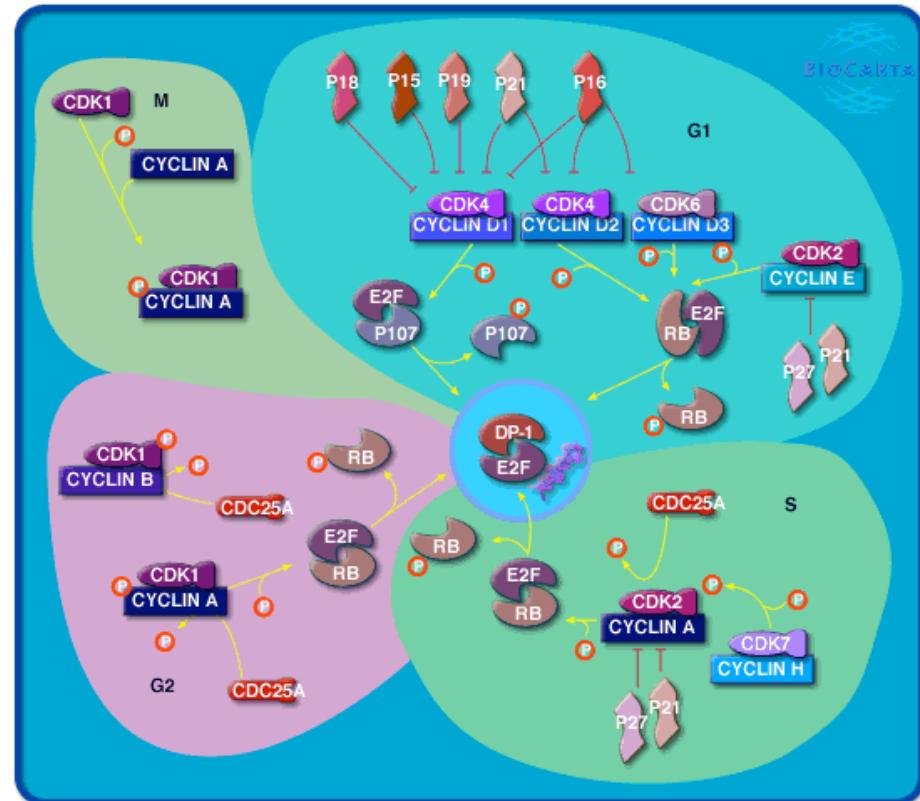


PATIKA ontology *abstractions*

- Possibly nested **groupings** of states and transitions [*use of compound graphs*]
- Help manage complexity and capture domain specific conventions
- Categorized according to biological meaning:
 - **Regular** abstractions
 - **Homology** abstractions
 - **Incomplete** abstractions

PATIKA ontology *regular abstractions*

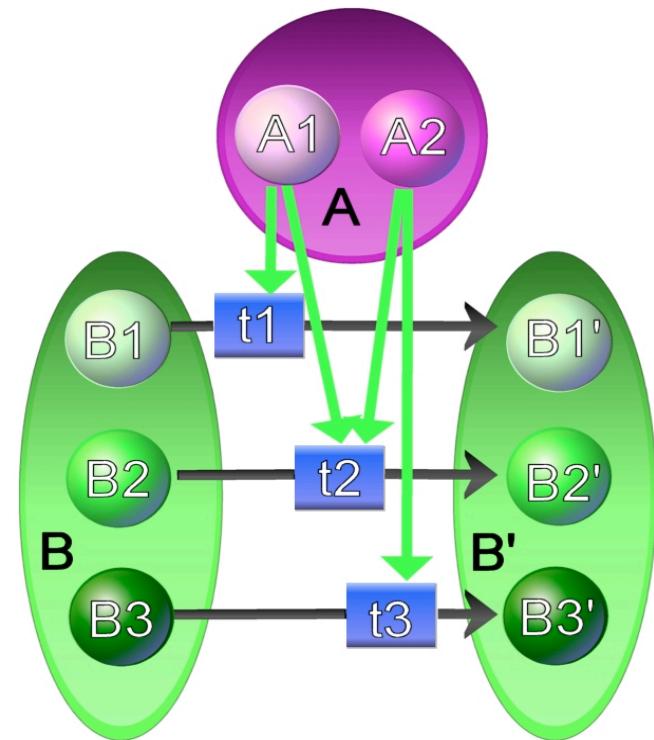
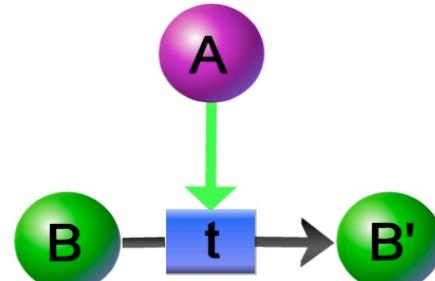
■ Example: Cell cycle, sub-phases: M, G1, S, G2.



[<http://www.biocarta.org>]

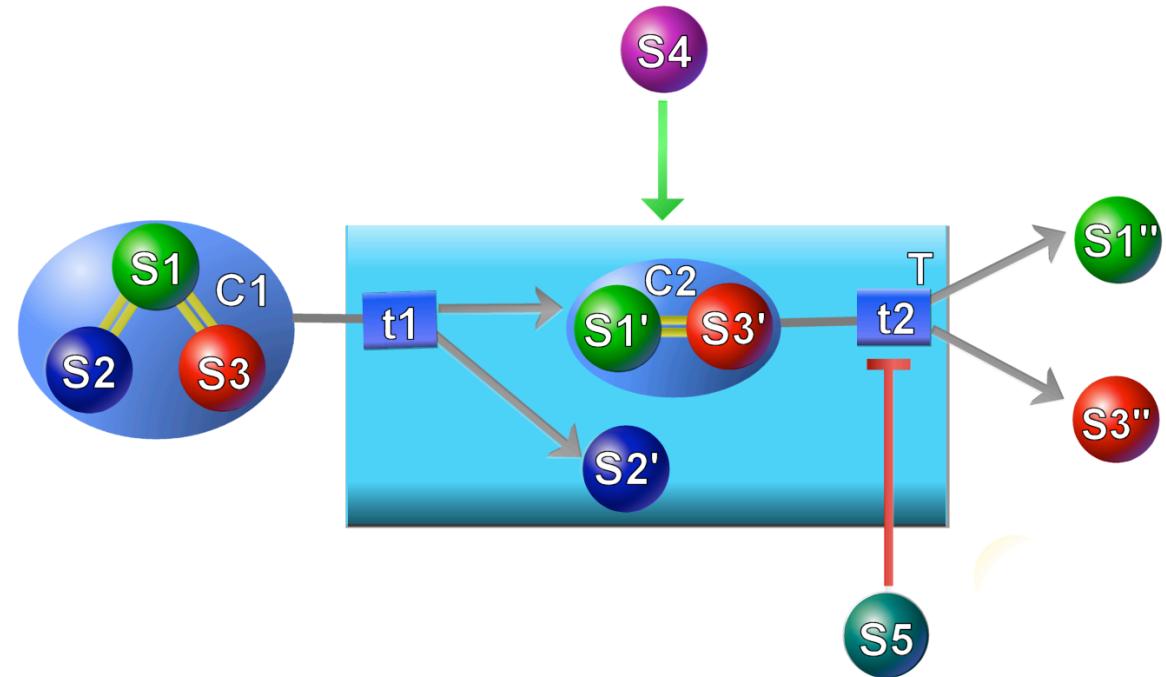
PATIKA ontology *homology abstractions*

- Evolutionary relations (e.g., wnt gene in human)



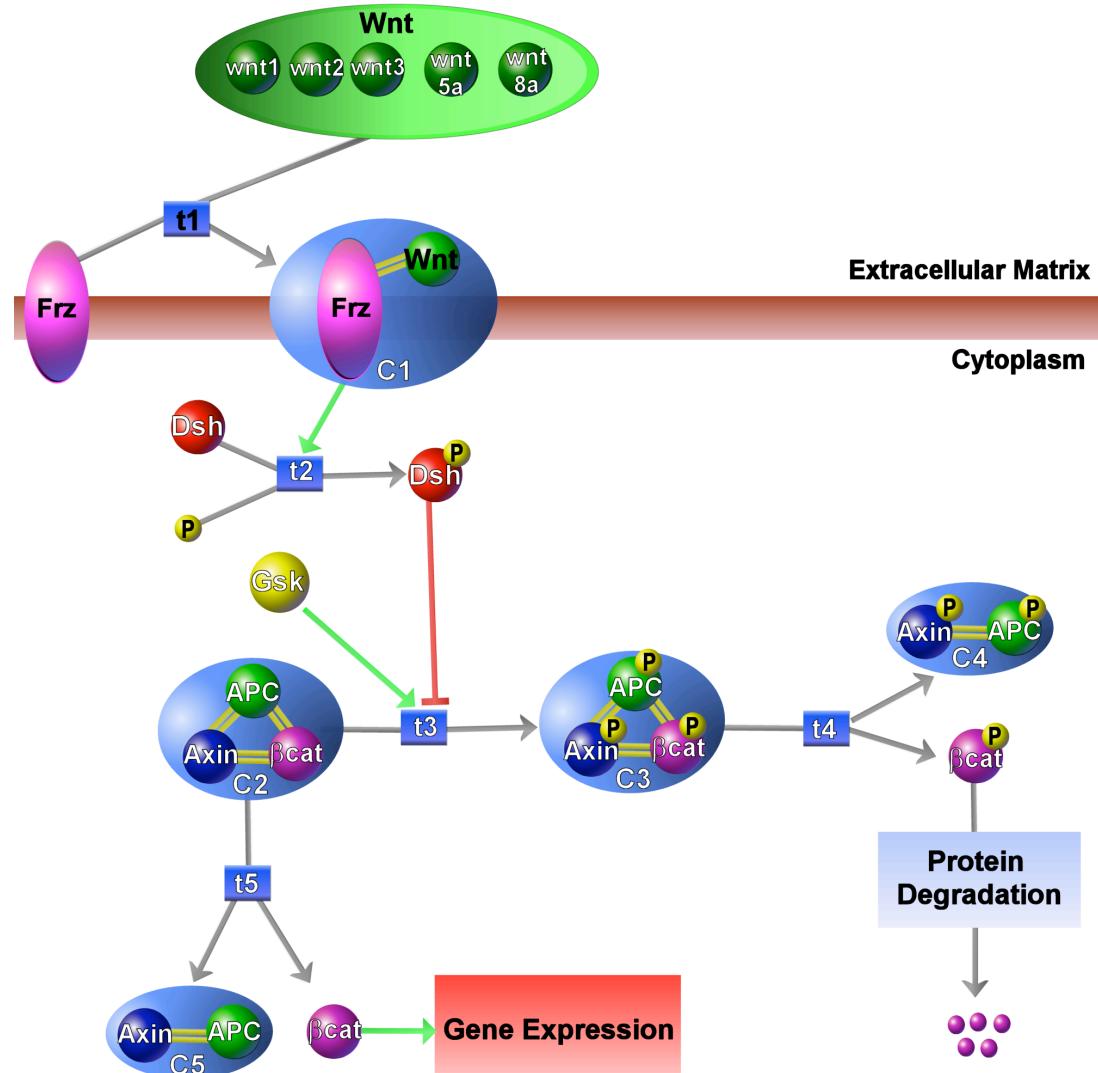
PATIKA ontology *incomplete abstractions*

- Regular methodology to represent incomplete information



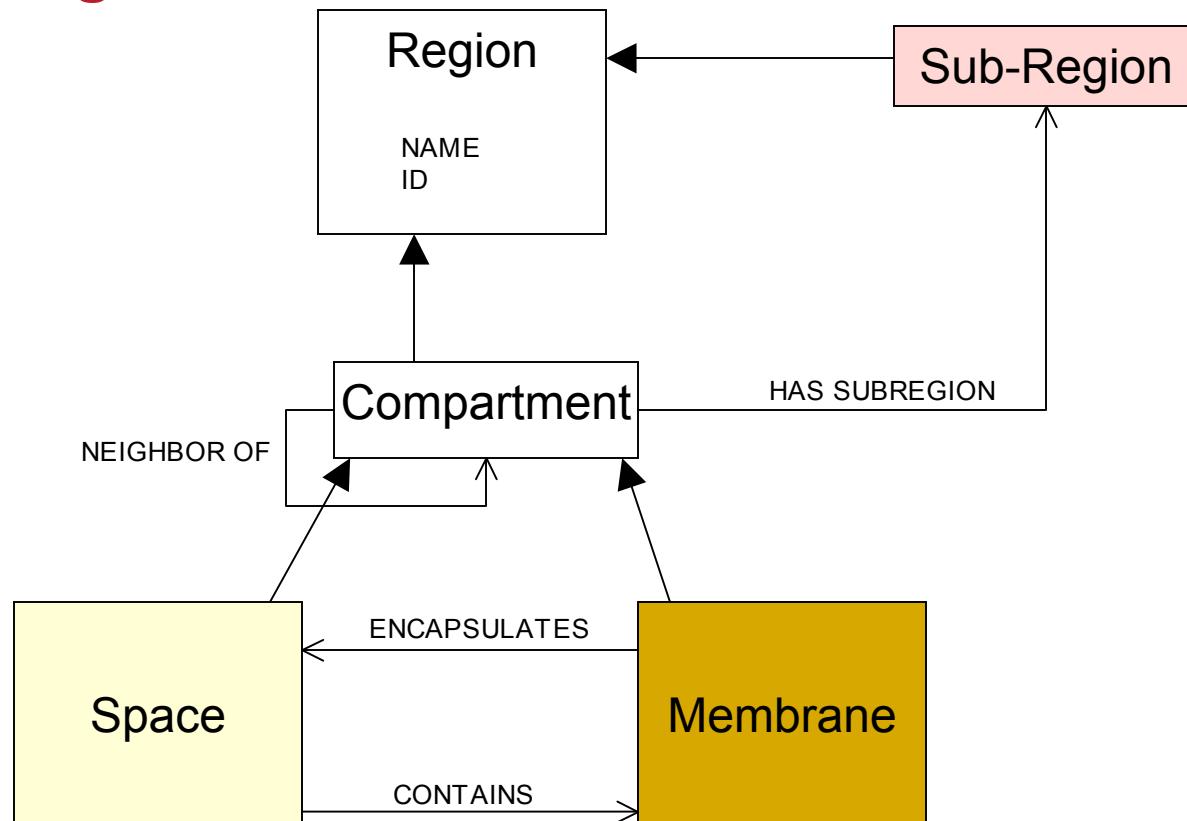
PATIKA ontology example

Conanical *wnt* pathway



PATIKA ontology *cell model*

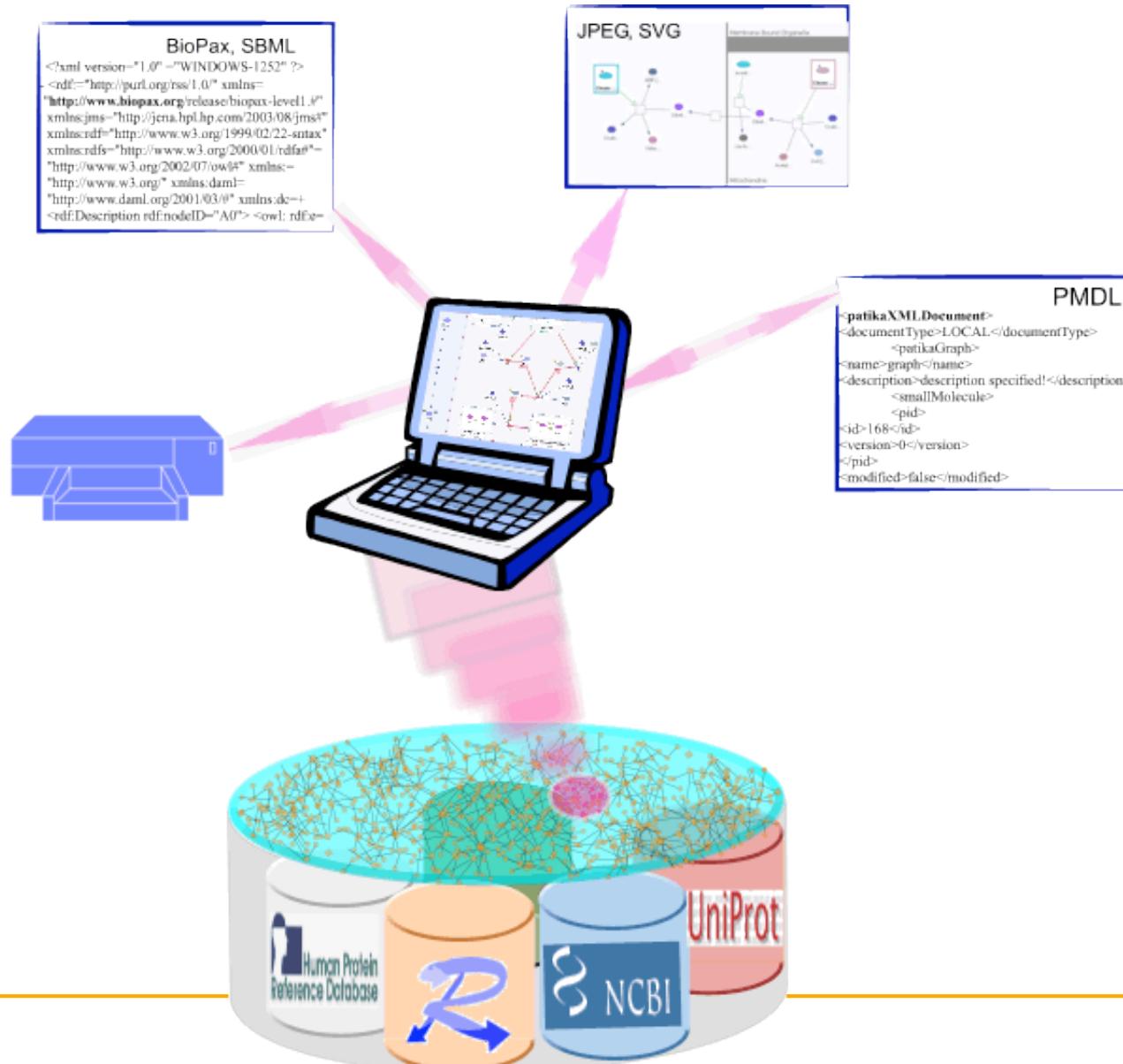
- Containment, neighborhood and sub-region properties are used for modeling cell environment of **particular organism**.



PATIKA tools

- **PATIKA/*lite* 1.0** [*Demir et al, Bioinformatics, 2002*]: **Basics** implemented in first attempt.
- **PATIKA/*web* 1.0** [*Dogrusoz et al, Bioinformatics, 2006*]: **Full ontology support; Web-based** thin-client version; not intended for submissions to database.
- **PATIKA/*pro*** [*prototype available*]: **Full ontology support; author tool** for submissions and integration to database.

PATIKA tools *data flow*



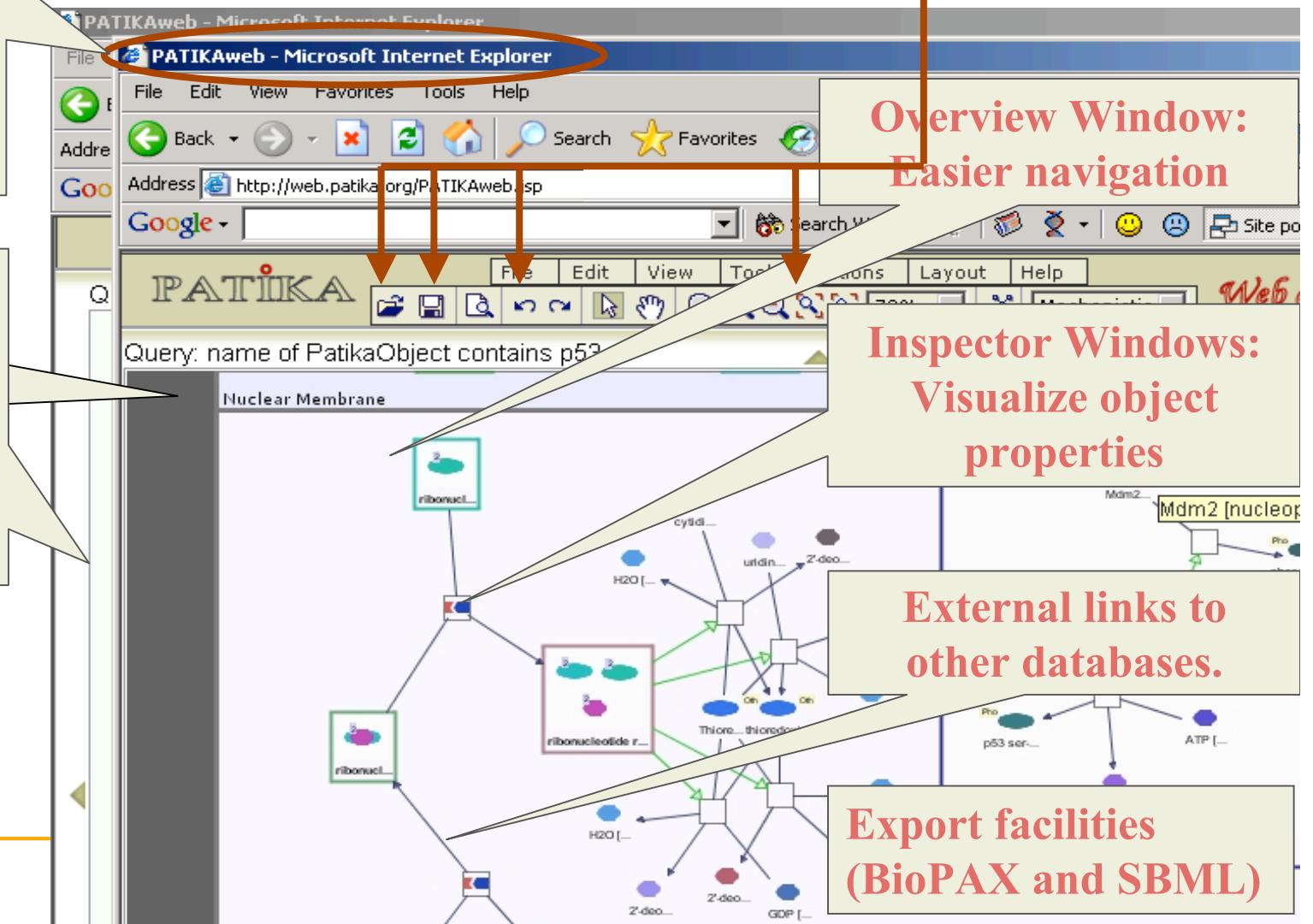
PATIKAweb (web.patika.org) key features

Easy access to
PATIKA DB
through a Web
browser:
web.patika.org

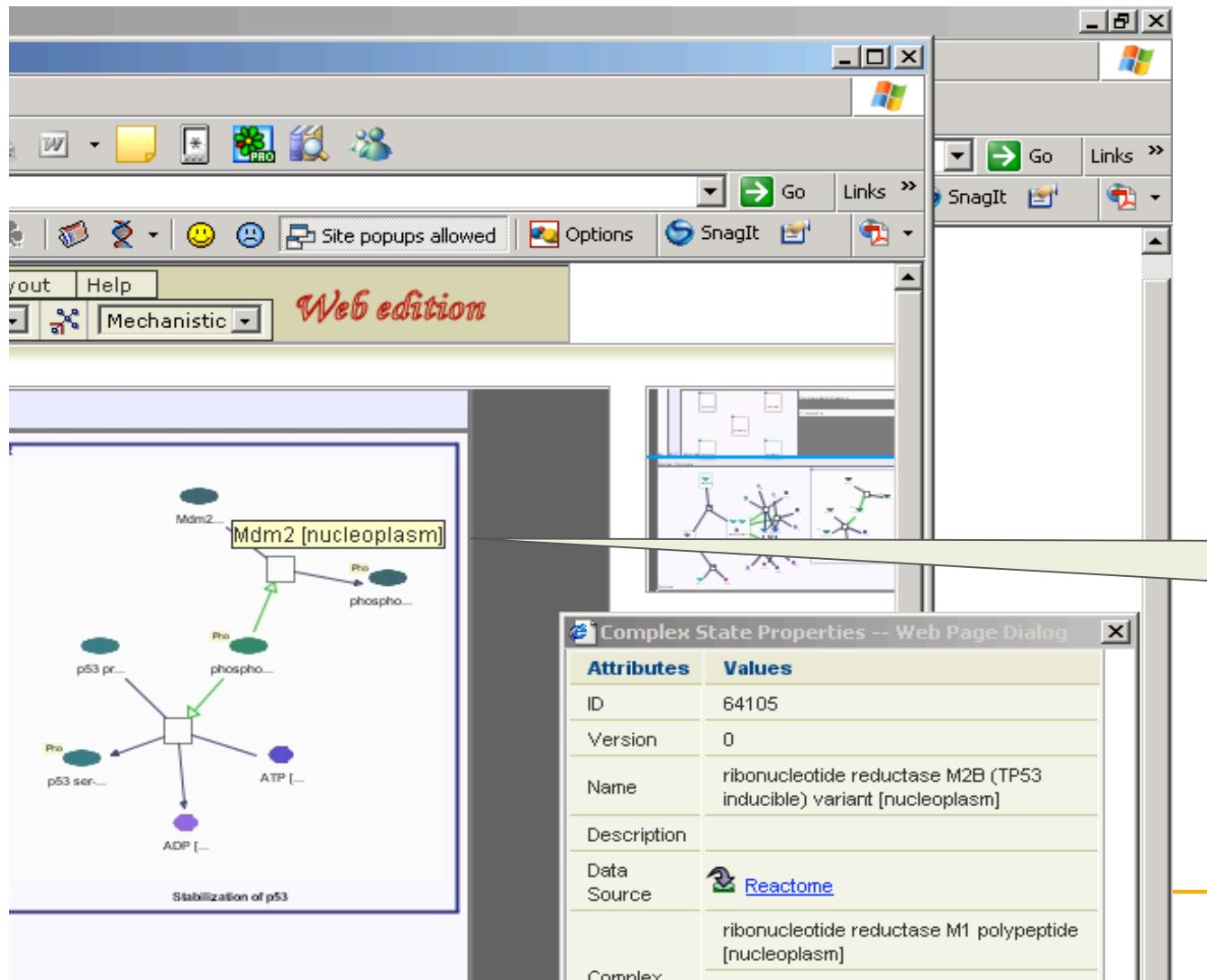
Multiple
View facility
for
Mechanistic
and Bioentity
levels of a
pathway model

www.patika.org

Pathway Editor functions: Save-load, undo, zoom, pan...



PATIKAw^eb (web.patika.org) key features



Graph-based
querying and
modeling

On-the-fly pathway
drawing generation
and automated layout

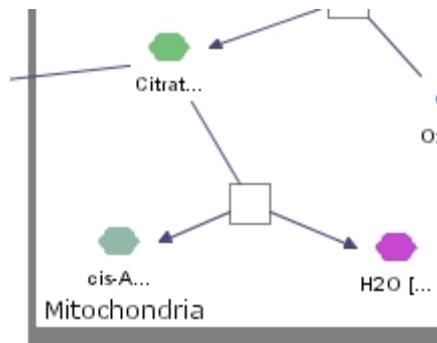
Pathway
integration like
“pieces of a puzzle”

Graphical notation *design criteria*

- Easily distinguishable pathway elements
 - Distinct UIs for pathway elements of distinct types



- Compartments easy to recognize



Graphical notation design criteria

■ Effective analysis of pathway elements

- Info boxes, labels



Protein

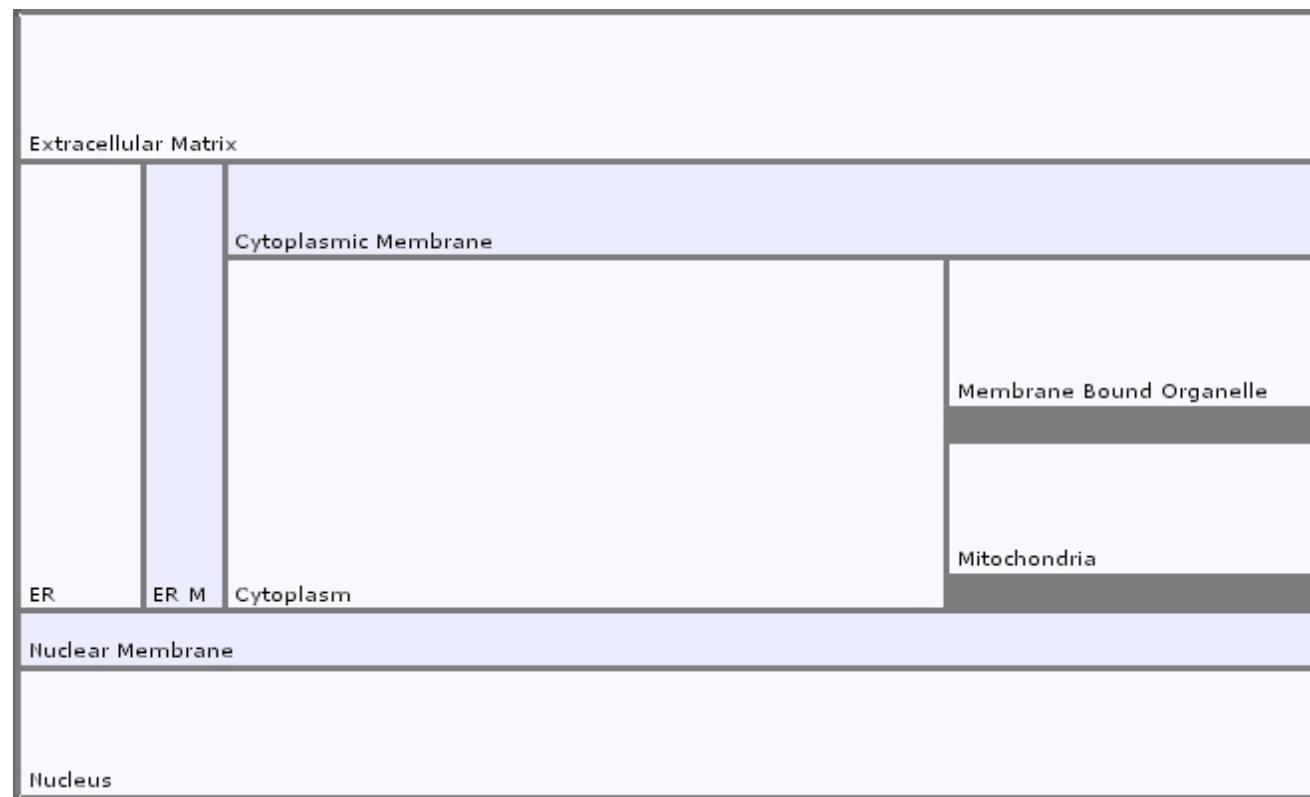
- Inspectors

The screenshot shows a graphical user interface for pathway analysis. On the left, a network diagram displays various nodes, including a protein complex (purple oval with 'Sub 2'), a branched-chain amino acid (blue hexagon), a proton (purple hexagon labeled 'H+ [m...]', with a green hexagon below it), and a methyl group (purple hexagon labeled 'methyl...'). A context menu is open over the protein complex node, listing options: 'Navigate', 'Select Neighbors in View', 'Find Neighbors', and 'Properties'. The 'Properties' option is highlighted. To the right of the diagram is a detailed 'Complex State Properties' window. This window has a title bar 'Complex State Properties -- Web Page D...' and a close button 'X'. It contains a table with two columns: 'Attributes' and 'Values'. The table rows are:

Attributes	Values
ID	59568
Version	0
Name	branched-chain amino acid aminotransferase, cytosolic, holoenzyme [cytosol]
Description	
Data Source	Reactome Entrez PubMed
Complex Members	Pyridoxal phosphate [cytosol] branched-chain amino acid aminotransferase, cytosolic, apoenzyme [cytosol]

Graphical notation *visual cell model*

- Custom (organism-specific) cell model using **resizable** rectangular compartments

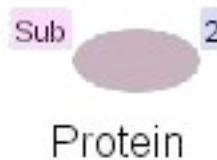


Graphical notation *bioentity view*

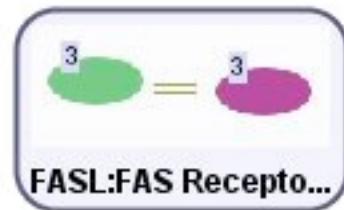


Graphical notation *mechanistic view*

■ Simple States:



■ Complexes:



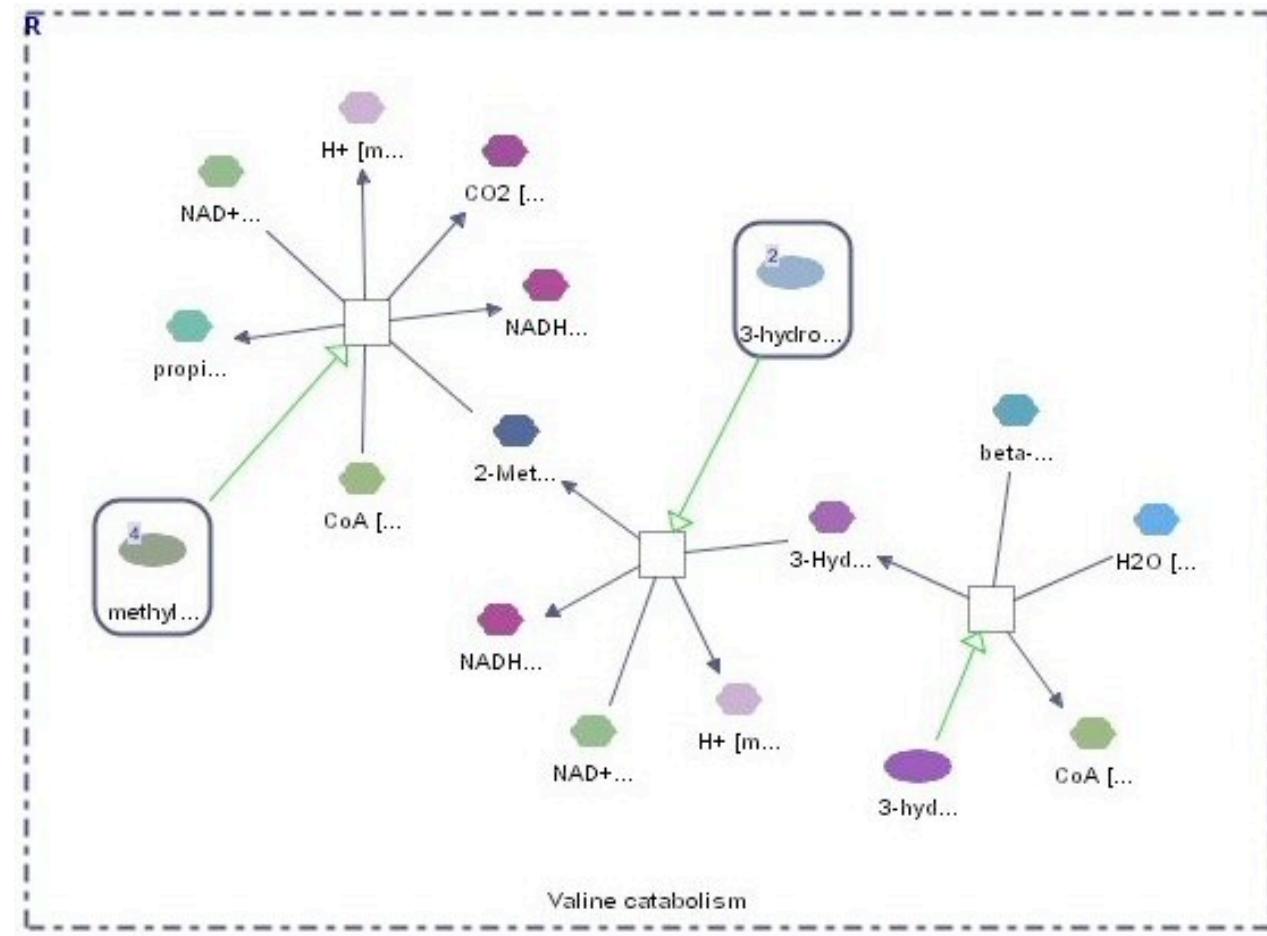
Graphical notation *mechanistic view*

■ Abstractions:

R: *regular*

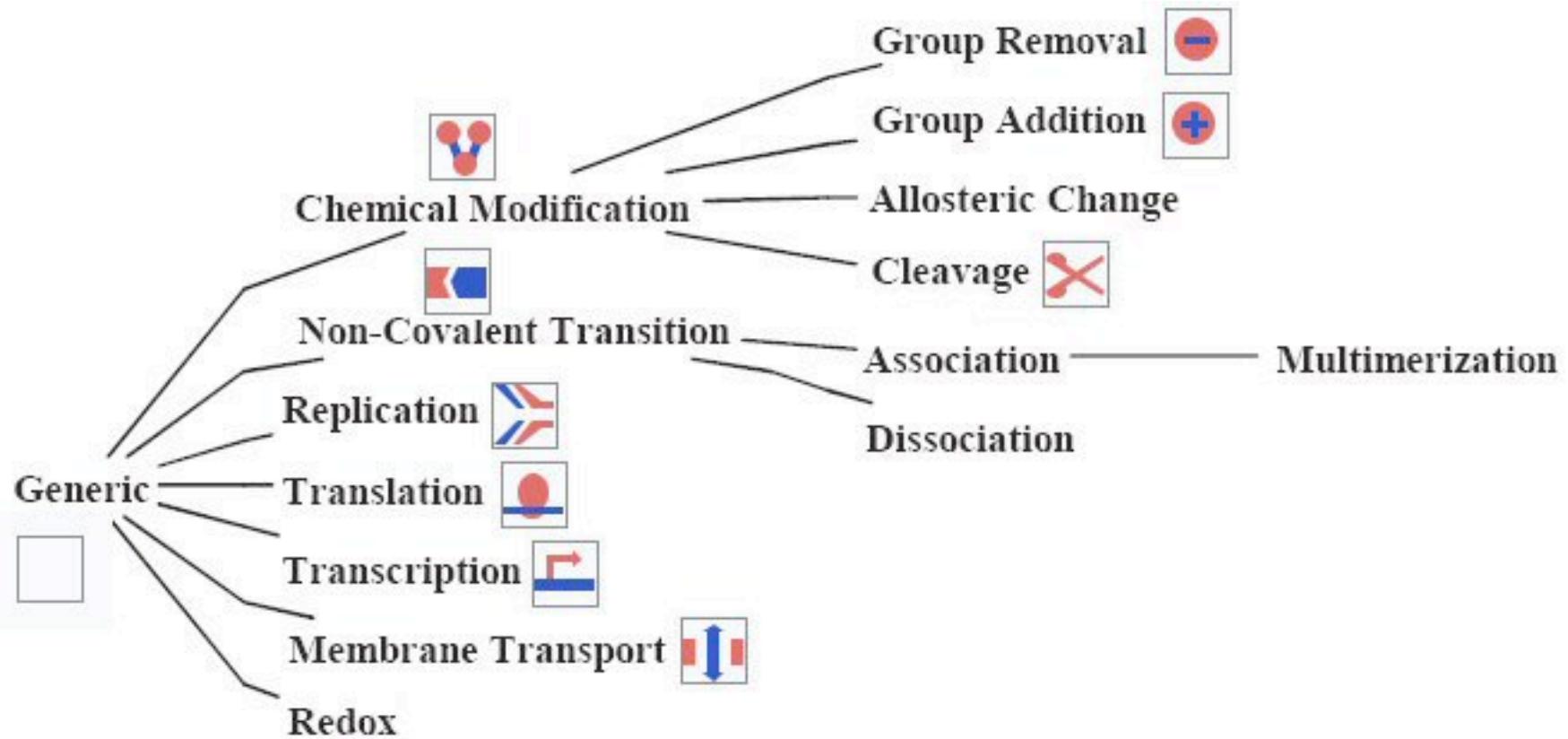
H: *homology*

I: *incomplete*



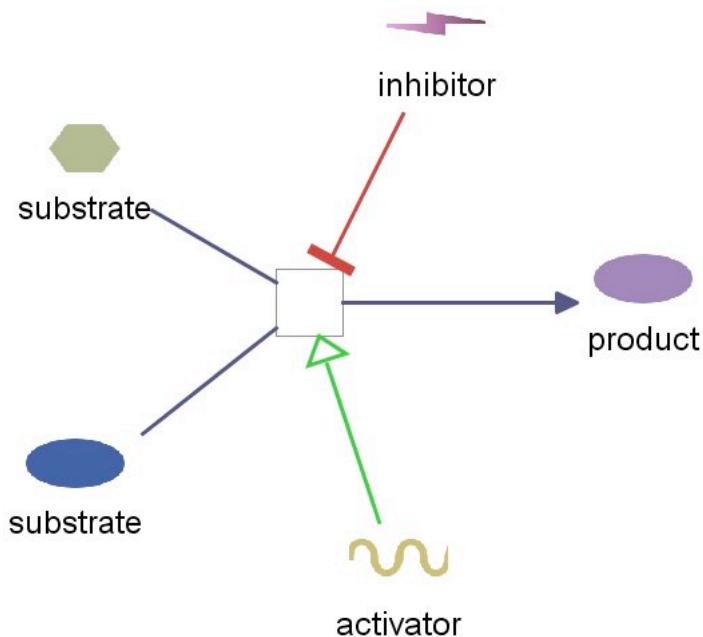
Graphical notation *mechanistic view*

■ Transitions:



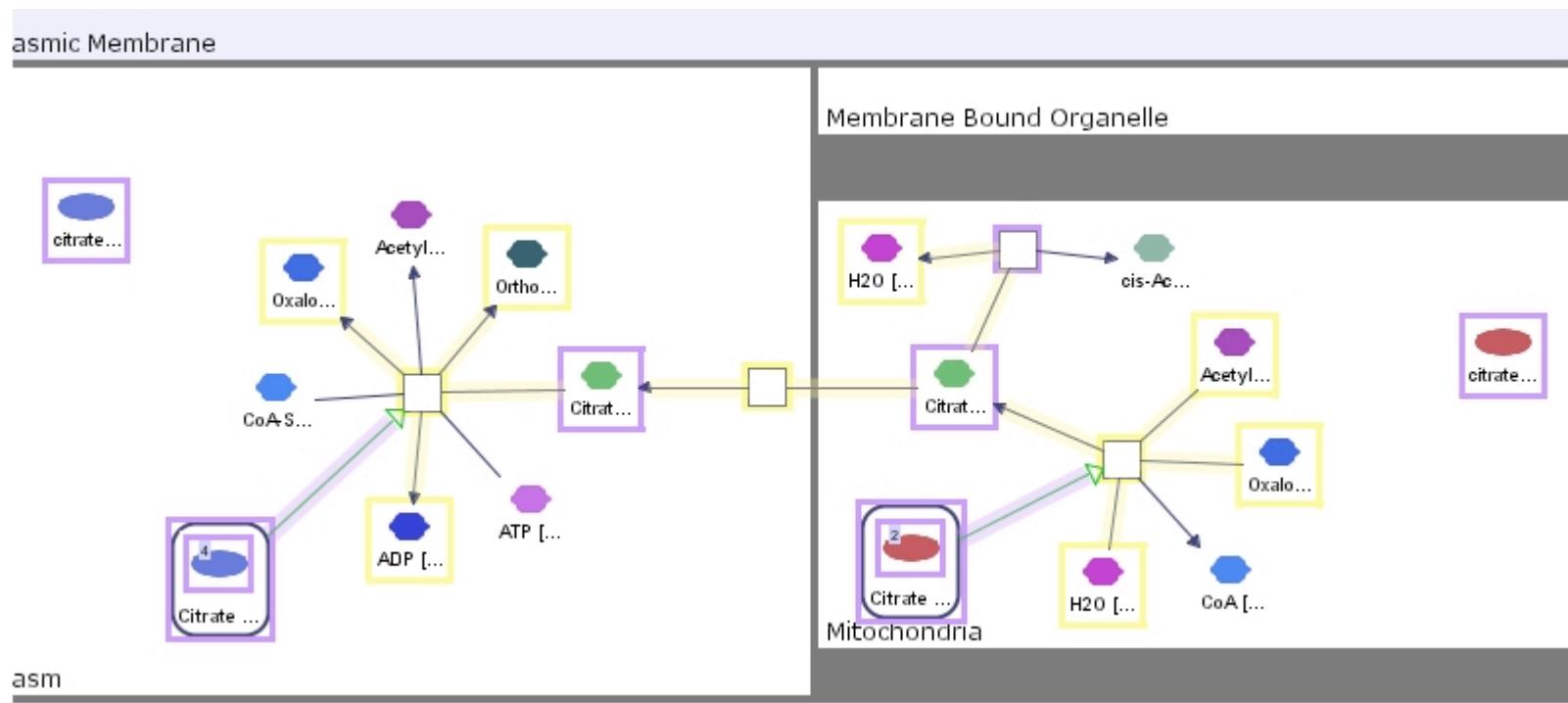
Graphical notation *mechanistic view*

■ Interactions:

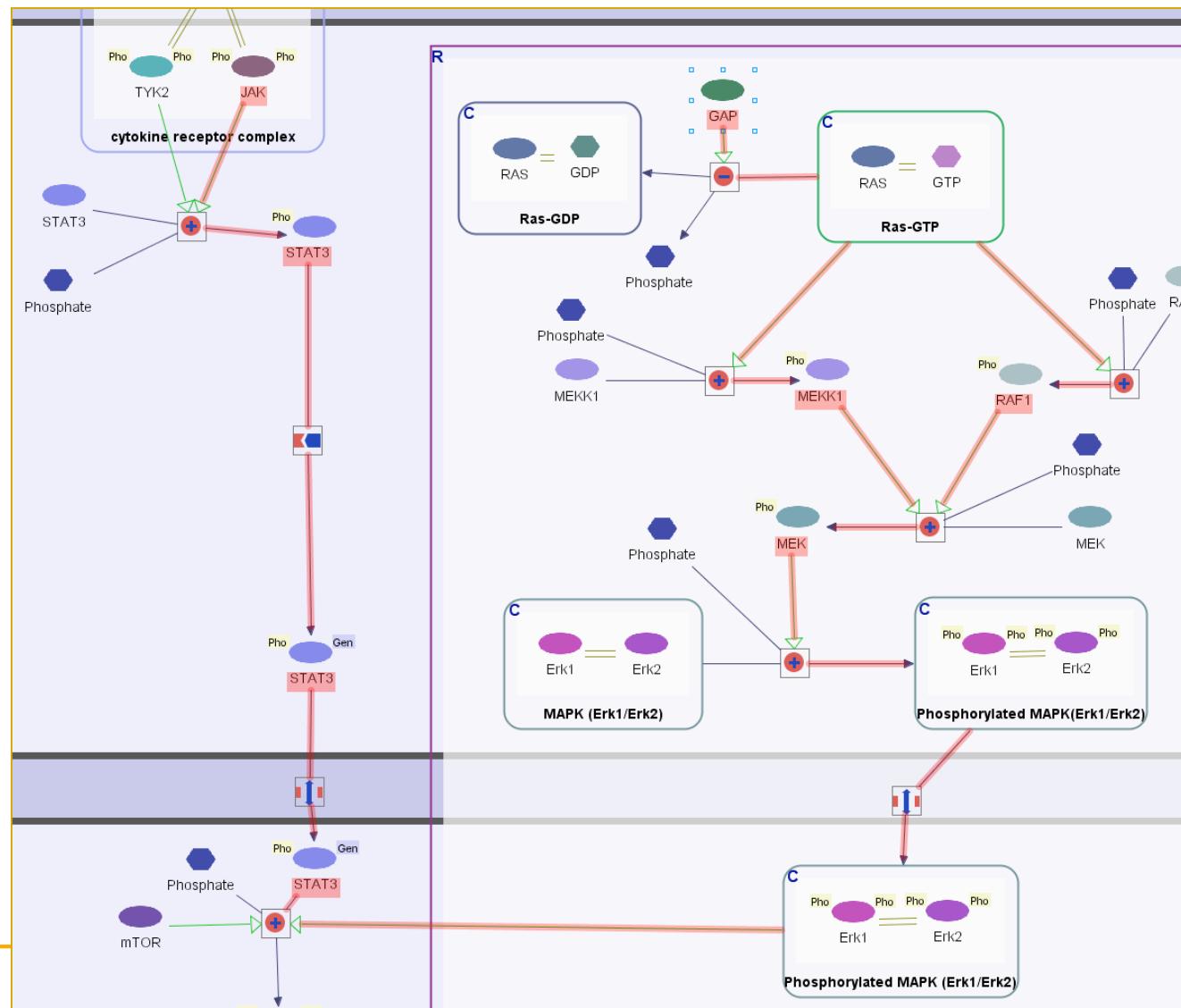


Visualization methods *design criteria*

- Clear, comprehensible presentation
 - Automated layout (static & incremental)
 - Highlighting, color coding, holos



Visualization methods *design criteria*



Visualization methods *design criteria*

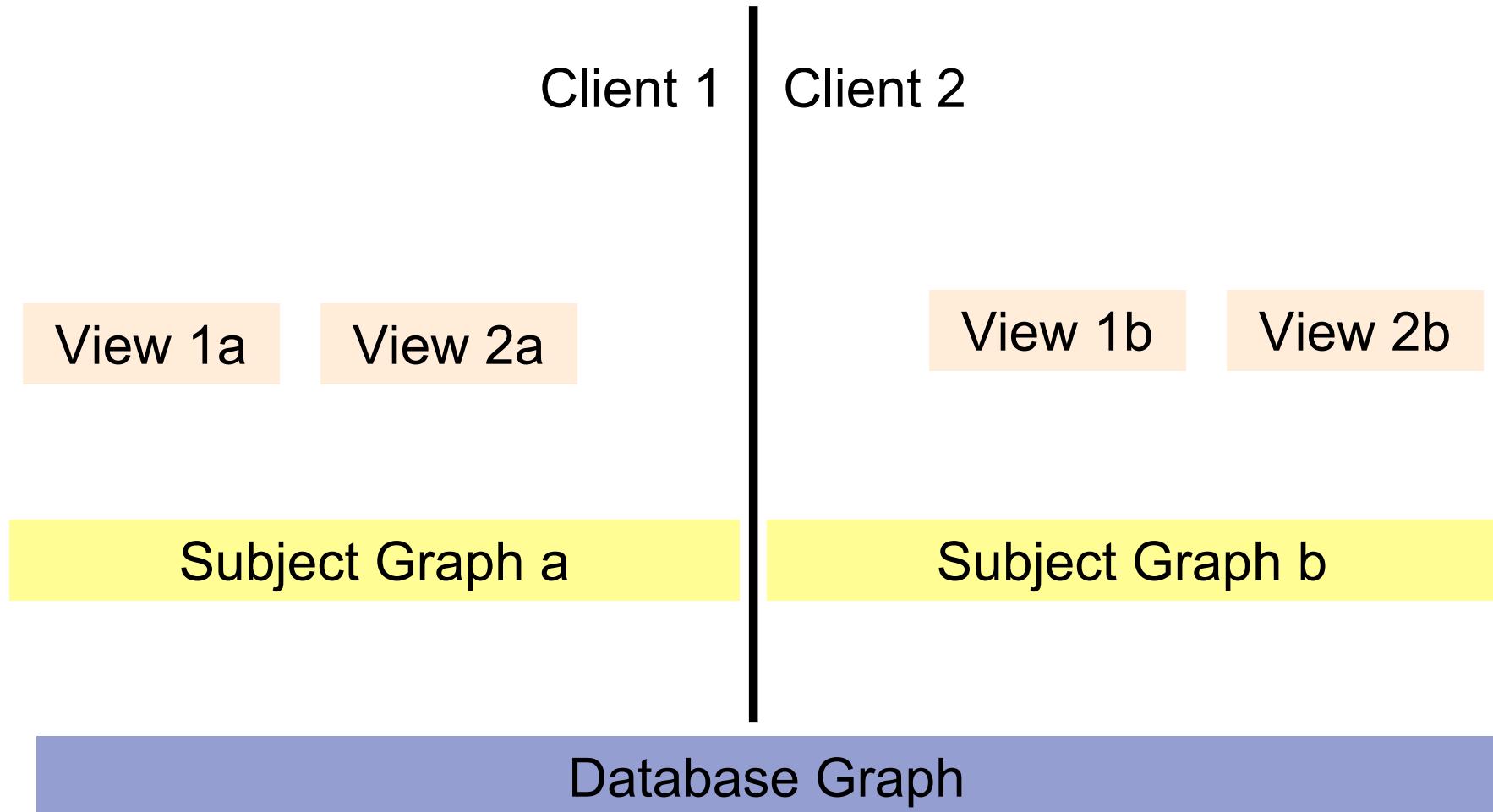
■ Complexity management

- Expand/collapse, overview window, multiple views

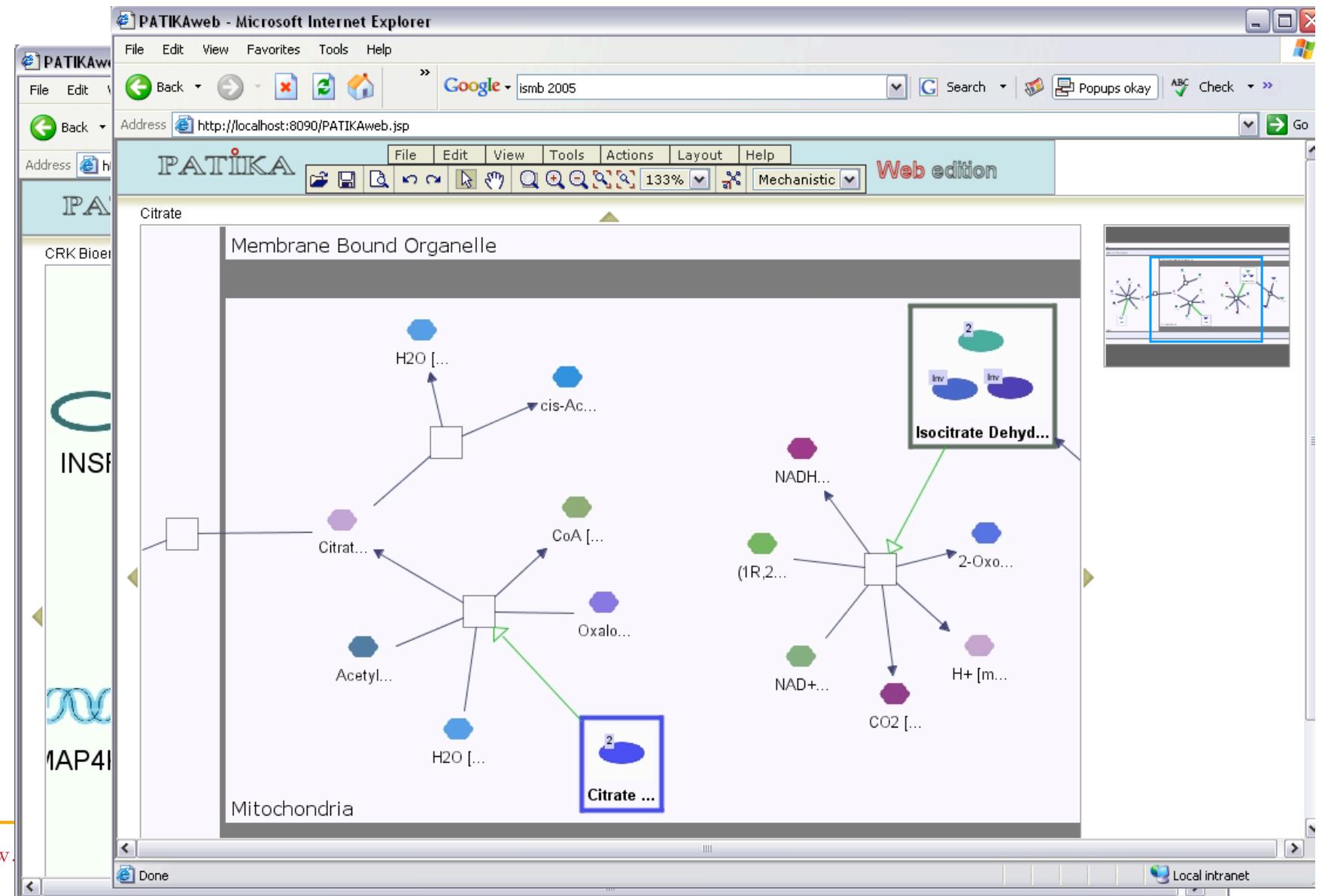
■ Effective editing

- Pathway element addition/removal
- Geometry changes
- Resizable compartments

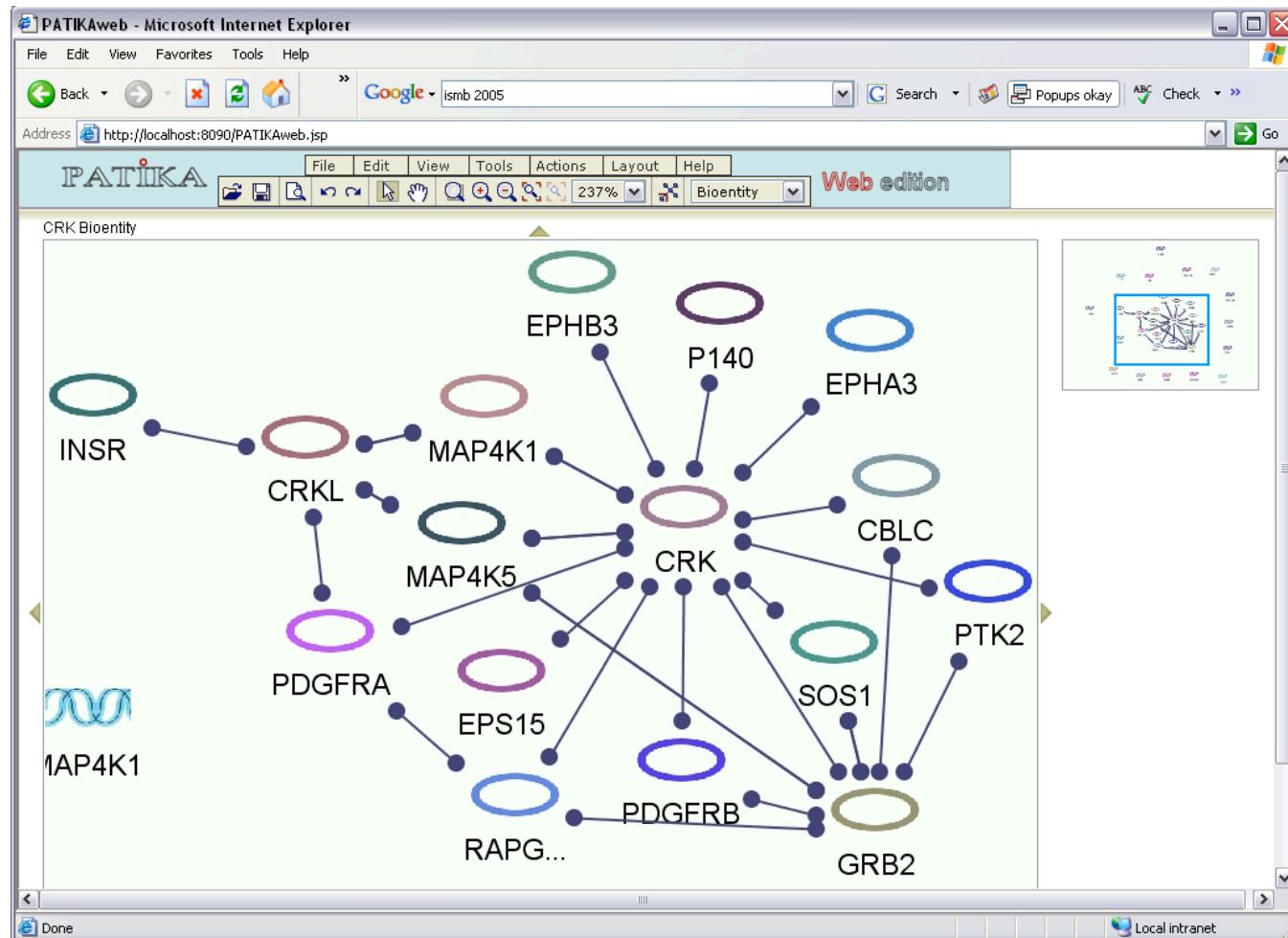
Complexity management *multiple views*



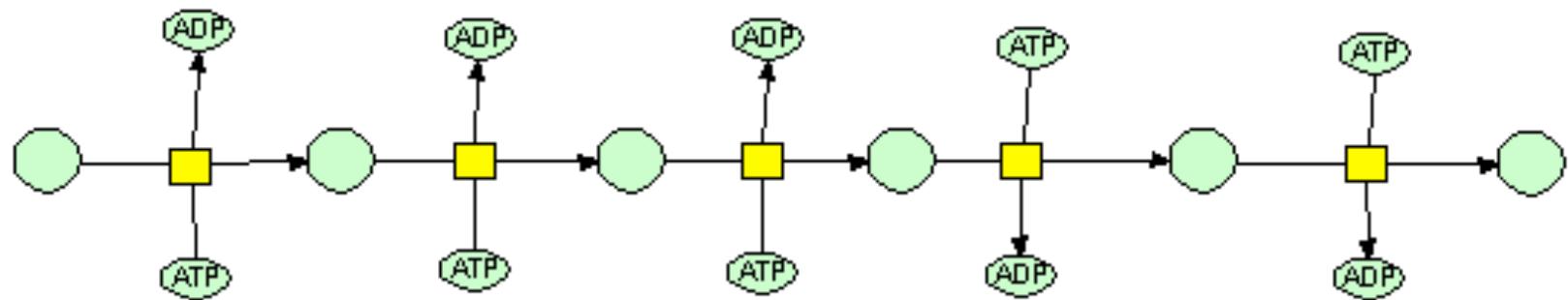
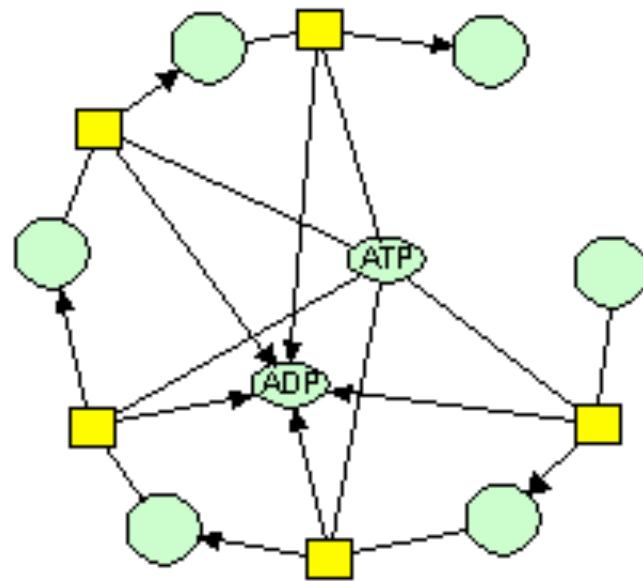
Complexity management *multiple views*



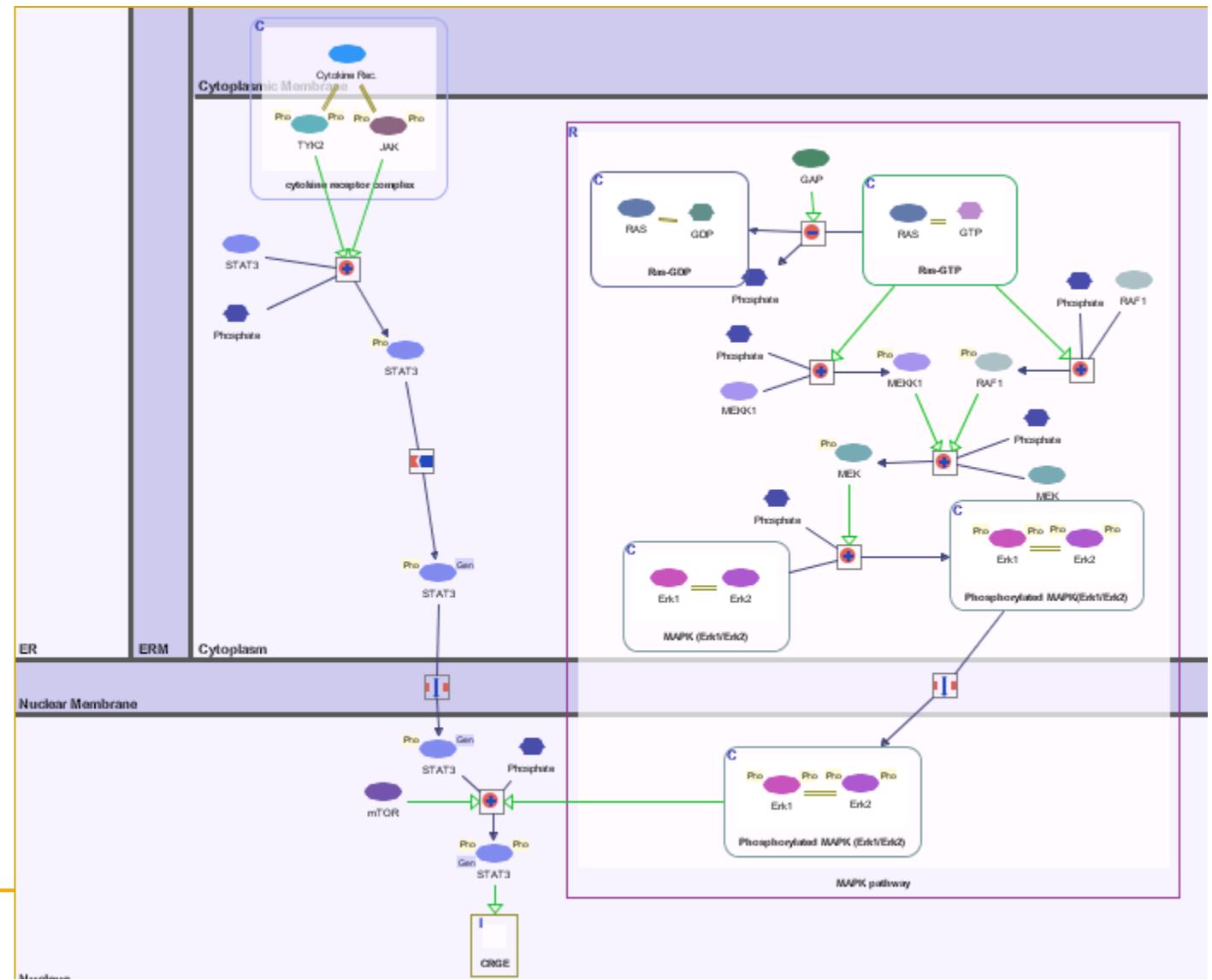
Complexity management overview window



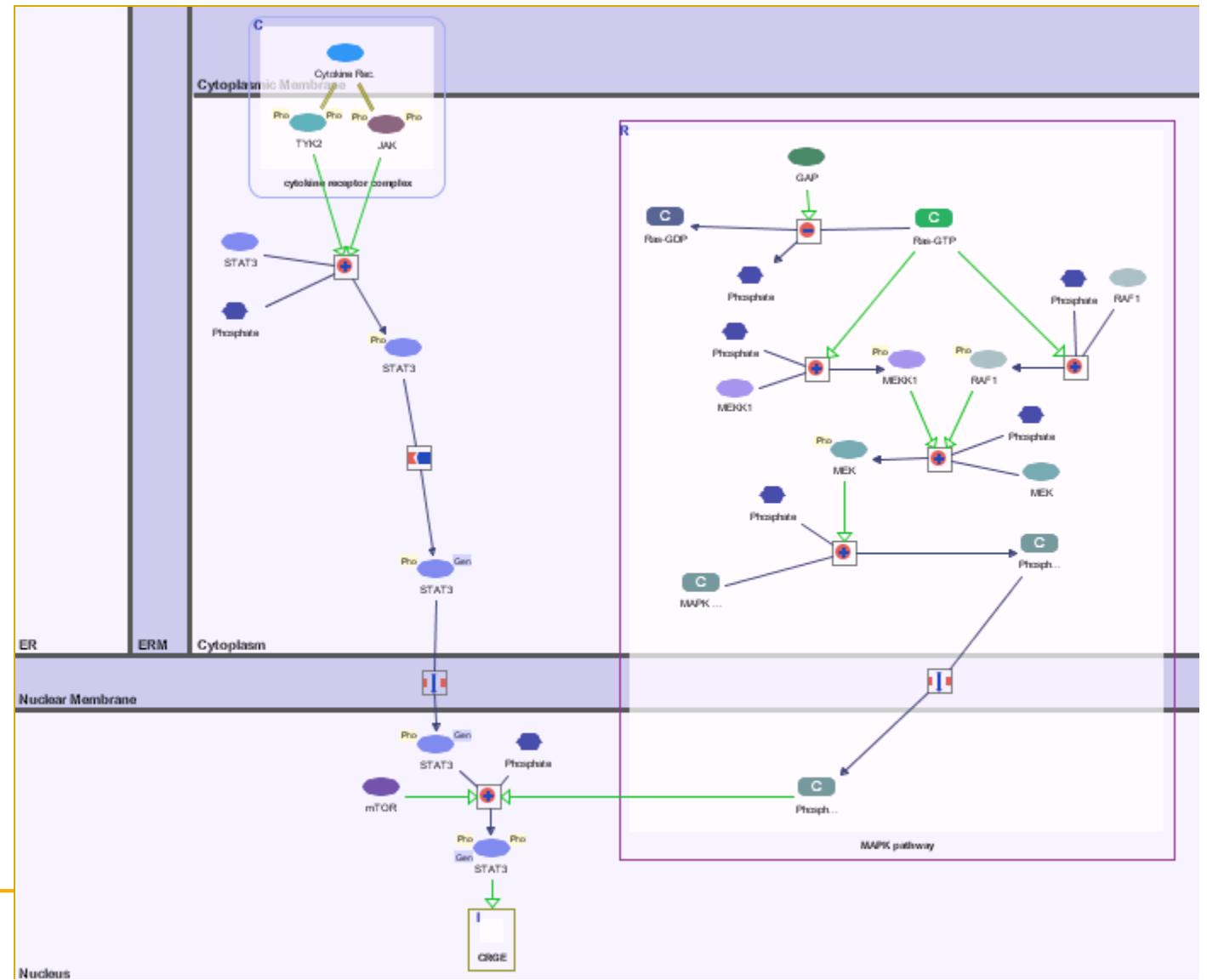
Complexity management *ubiquitous mol.*



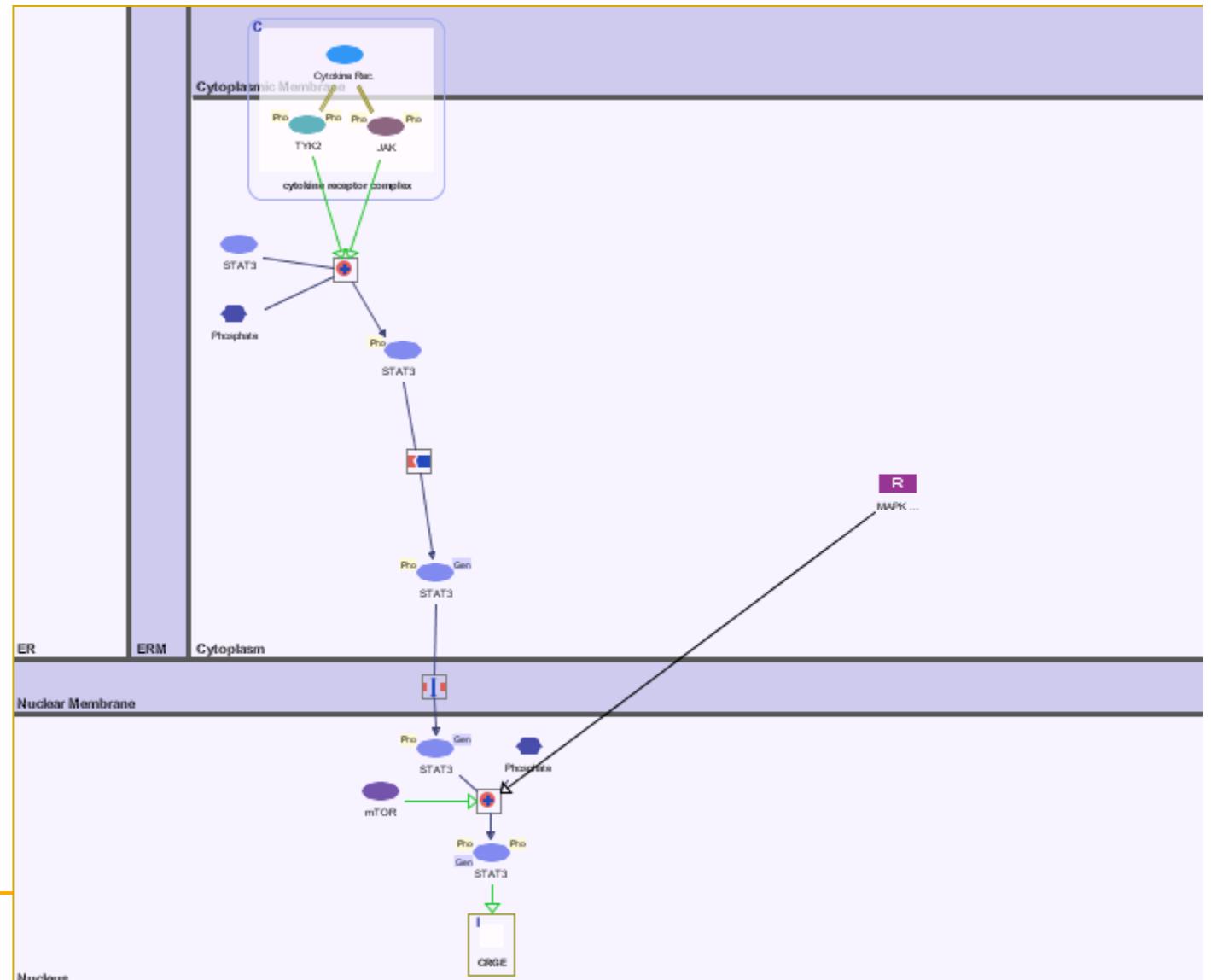
Complexity management *nesting*



Complexity management *collapse*



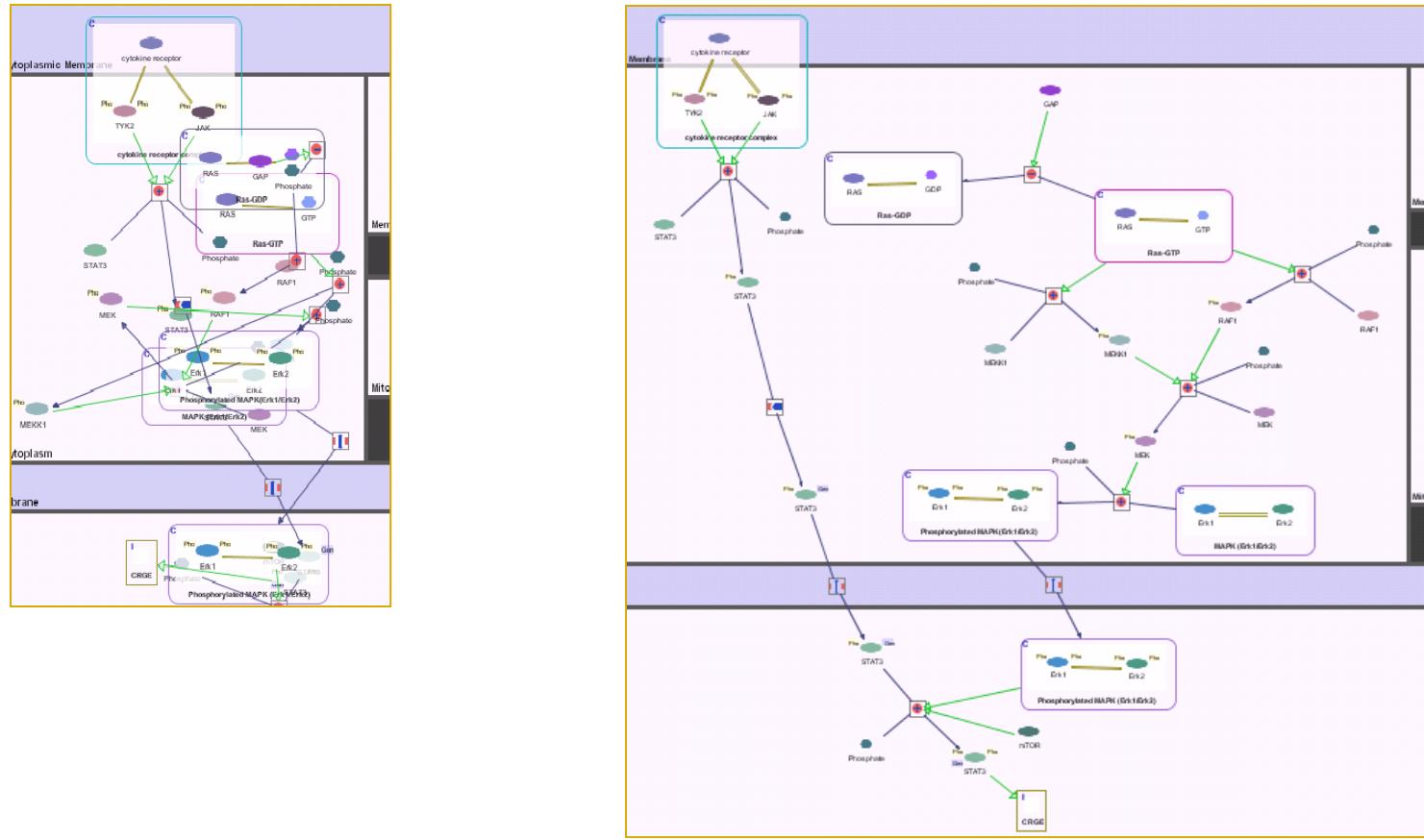
Complexity management *collapse*

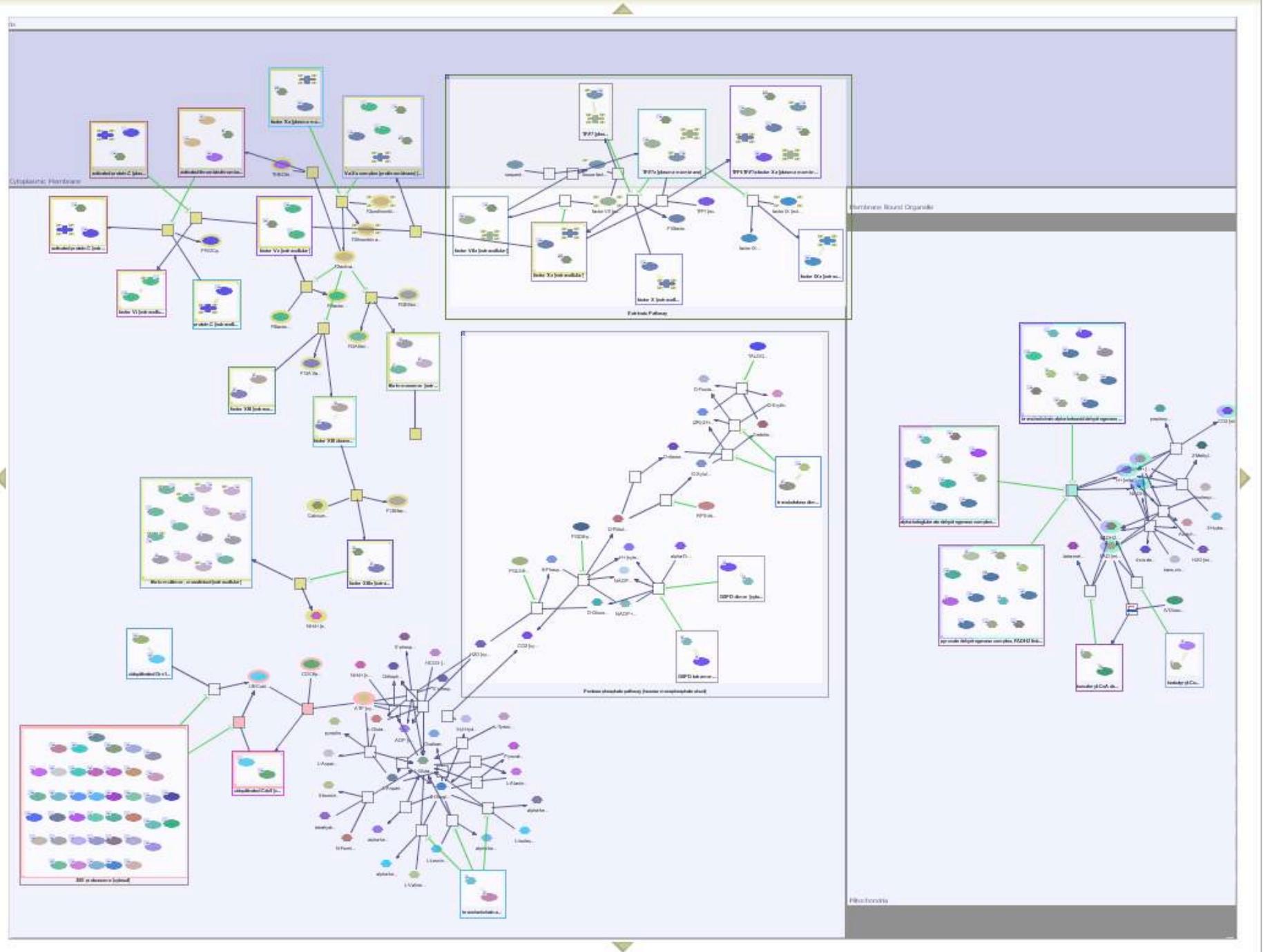


Automated layout

[Dogrusoz et al, INS, 2006; Dogrusoz et al, *Graph Drawing* 2004]

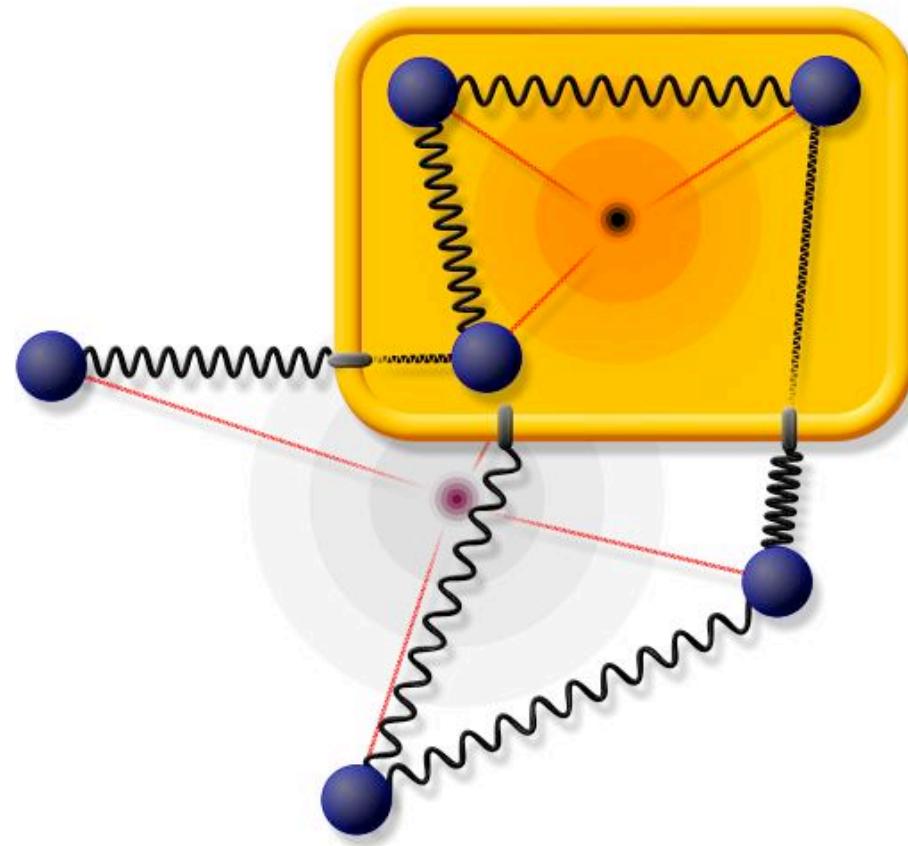
For easy-to-comprehend pathway drawings





Automated layout

[Dogrusoz et al, *INS*, 2006; Dogrusoz et al, *Graph Drawing 2004*]



Conclusion

- Ontology & graphical notation go hand-in-hand!
- Our ontology & graphical notation was highly influenced by **interactive visualization** and **querying** needs.
- Graphical notation should be designed in accordance with **visualization methods**.

Questions, comments?

Ugur Dogrusoz

Director, *Bilkent Center for Bioinformatics*
(<http://www.bilkent.edu.tr/~bcbi>)

Assistant Professor, *Computer Eng. Dept, Bilkent Univ., Ankara, Turkey*