

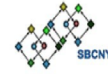
Paxtools, a Java API for BioPAX

Paxtools: Basics

- A complete and consistent implementation of BioPAX specification:
 - POJO Beans
 - Getters/Setters
- Handles non-OO OWL constructs
 - Subproperties, reverse properties, anonymous classes

When to use Paxtools

- Develop BioPAX using software tools
- Extract simplified information
- Write Pathway specific algorithms
- Run complex searches:
 - Path, Pattern and Graph queries.
- Convert
 - BioPAX → SBGN
 - PSI-MI → BioPAX
 - BioPAX L1,L2 → L3
- Access PC2 web service



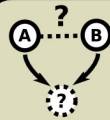
Pathway and interaction data (BioPAX/PSI-MI)
from different data sources

Files

Web Service



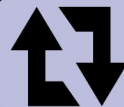
Paxtools



Graph
Queries



Algorithms



Export



Database
Storage

Embed

Use

Develop
Software



ChiBE



Cytoscape



Network Analysis



Gene Set Enrichment Analysis



Validation Reports



Visualization

Getting started

- Hosted within SourceForge BioPAX project
 - “Fat” jars
 - Maven
 - Sources
- Documentation
 - Project documentation
 - API docs
 - Getting Started guide

Developing Paxtools

- If you get stuck or have questions, look for support in the Paxtools support mailing list:
<https://lists.sourceforge.net/lists/listinfo/biopax-paxtools>.
- For general questions related to BioPAX, use BioPAX discuss mailing list:
<http://groups.google.com/group/biopax-discuss>.

Module Structure

- Core
- Jena IO
- Query
- PathwayCommons client
- PSI-MI converter
- Sif, GSEA, SBGN converter
- BioPAX converter

My first model

```
BioPAXFactory factory =  
BioPAXLevel.L3.getDefaultFactory();  
  
Model model = factory.createModel();  
  
Protein protein1 =  
model.addNew(Protein.class,  
"http://biopax.org/tutorial/protein1");
```


Adding fields

```
protein1.setStandardName("standard test");
```

```
protein1.setDisplayName("test");
```

```
BiochemicalReaction rxn1 =  
model.addNew(BiochemicalReaction.class,  
"http://biopax.org/tutorial/rxn1" );
```

```
rxn1.addLeft(protein1);
```

Core Tools

- Editors
 - Flexible, reflection based property editors
- Accessors
 - Xpath like access/search
- Algorithms
 - Traverse, Normalize, Compare, Merge, Extract, Clone
- Persistence
 - Hibernate and Hibernate Search mappings

SIF and other reductions

- Converts BioPAX to simpler formats
 - SIF to convert processes to binary relationships based on templates
 - SIFNX – customizable SIF exporter
 - GSEA to get gene sets
 - Activity Network Format : PD → AF conversion for BioPAX (Work in Progress)

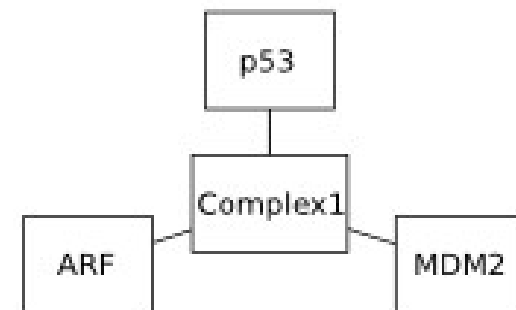
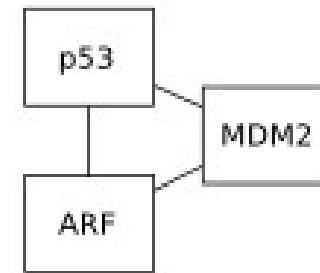
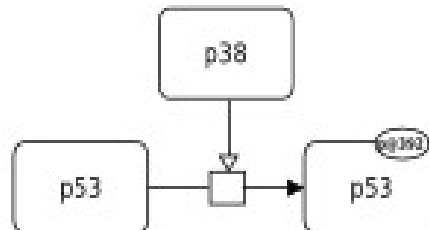
COMPONENT_OF: The first entity is a component of the second entity, which is a complex. This interaction is transient in the sense that A component_of B and B component_of C implies A component_of C. This interaction is directed.



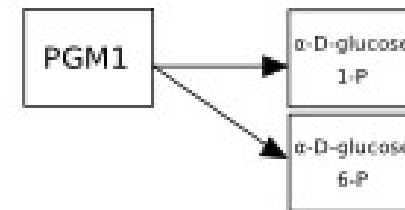
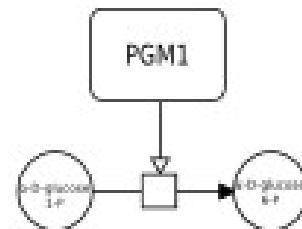
IN_SAME_COMPONENT: Two entities belong to the same molecular complex. This does not necessarily mean they interact directly. In a complex with n molecules, this rule will create a clique composed of $n(n-1)/2$ interactions. This interaction is undirected.



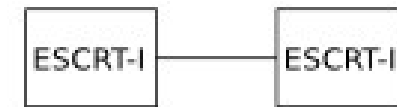
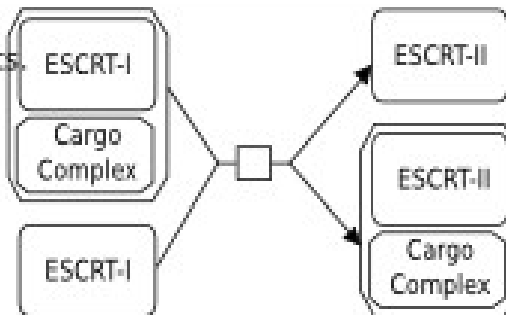
STATE_CHANGE: The first entity controls a reaction that changes the state of the second entity, e.g. by phosphorylation or other posttranslational modification, or by a change in subcellular location. This interaction is directed.



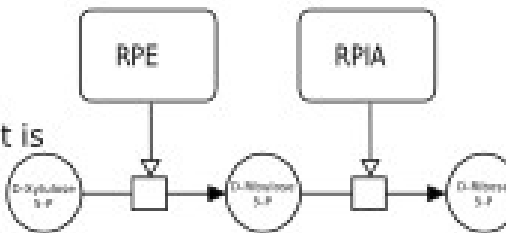
METABOLIC_CATALYSIS: The first entity catalyzes a reaction that either consumes or produces the second entity. This interaction is directed.



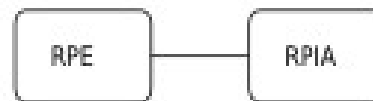
REACTS_WITH: The entities participate in a conversion as substrates or products. Controllers are not included. This interaction is undirected.



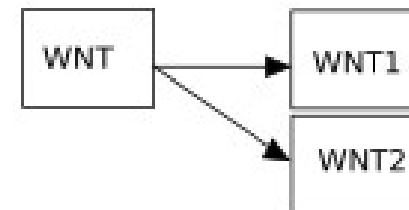
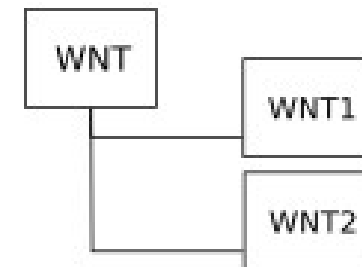
SEQUENTIAL_CATALYSIS: The entities catalyze two conversions that are connected via a common molecule, e.g. the first entity produces a substrate that is consumed by the second entity. This interaction is directed.



INTERACTS_WITH: The entities participate in an interaction. Controllers are not included. This interaction is undirected.



GENERIC_OF: The first entity is a generic form of the second entity. This interaction is directed.

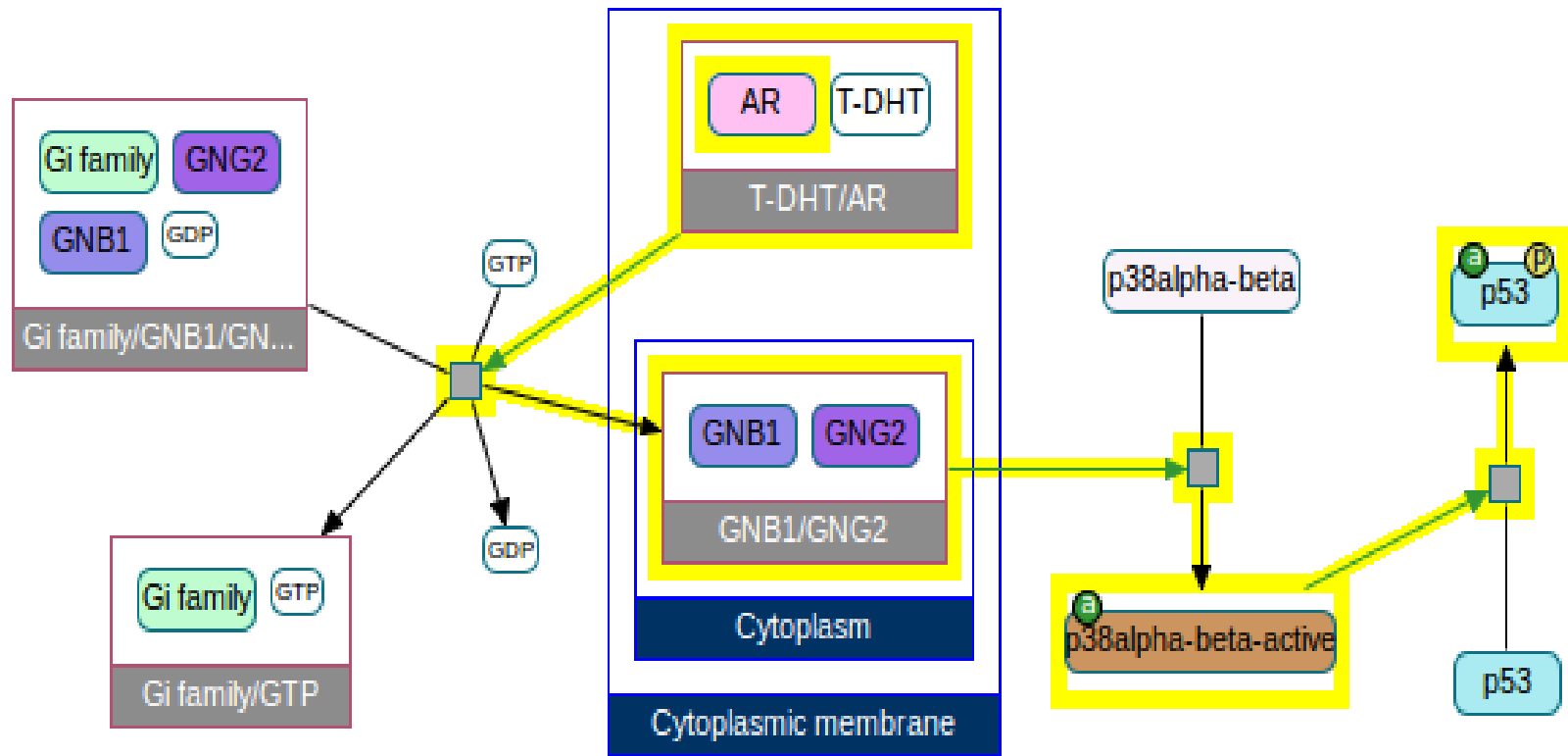


Graph Searches

- Neighborhood
- Graph-of-interest
- Common upstream/downstream
- Shortest Path
- New powerful Pattern search system
 - Regular expressions for pathways

Accessing Pathway Commons

- PC client distributed as a separate Maven project.
- Seamless access to full PC2 web service
- Filters, simple and graph searches, Xpath like accessors.



Paxtools from the command line

- merge file1 file2 output
- toSif file1 output
- toSifnx file1 outEdges outNodes prop1,prop2,...
- validate path out [xml|html|biopax] [auto-fix] [normalize] [only-errors] [maxerrors=n]
- integrate file1 file2 output
- toLevel3 file1 output
- fromPsimi level file1 output
- toGSEA file1 output database crossSpeciesCheck
- fetch file1 id1,id2,.. output
- getNeighbors file1 id1,id2,.. output

Get Involved

- Use Paxtools and report issues and feature requests. See : <http://biopax.sourceforge.net/paxtools/issue-tracking.html>
- Respond to questions by other users at [biopax-paxtools](#)
- Try your hand at fixing bugs and implementing requests
- Improve this documentation.
- Spread the word. Let other people know about BioPAX and Paxtools.

Acknowledgements

@New York

- Ozgun Babur
- Arman Aksoy
- Onur Sumer
- Chris Sander

@Toronto

- Igor Rodchenkov
- Gary Bader

@Tokyo

- Ken I. Fukuda

BioPAX Community

