**XPathway**

XPathway, a set of  tools that compares pathway activity analyzing mapping of contigs assembled from RNA-Seq reads to KEGG pathways. The XPathway analysis of pathway activity is based on expectation maximization and topological properties of pathway graphs.

The different tools that constitute XPathway are:

1.  **KGMLPathway2Graph**: Extraction tool of metabolic network

KGMLPathway2Graph aims at extracting metabolic pathways from KGML flatfile database. Readme, examples and software for KGMLPathway2Graph can be downloaded here.

2. **Link Gopher 1.3.3**:  Mozilla Firefox add-ons

3. **java code**:

4. **Python code**:

5. **shell script:**

**Pathway activity level pipeline**

**Pathway significance pipeline**

**Script 2 : Pathway extraction**

1. Run Kegg experiment on sample from **Script1**
2. Use Mozilla Firefox add-ons  “Link Gopher 1.3.3”

* open kegg result page “pathway map”
* use this filter “<http://www.kegg.jp/kegg-bin/show_pathway?@>” with link gopher to copy all green nodes per pathway. These are part of the pathway urls. Save the output in a file called All\_KGML\_URL.txt.

1. java code: extract all green node (KoNames.java)

- Read “green\_pathway” containing url of ko pathway for green nodes ( green nodes are organisms specific genes)

* input file: All\_KGML\_URL.txt
* output files: koNames.txt and setKonames

- eg:

input: http://www.kegg.jp/kegg-bin/show\_pathway?@ko00073/reference%3dwhite/default%3d%23bfffbf/K13356/K13357

Output: 2 text files containing: ko00073: #green and the other, ko00073: K13356, K13357 enumerate all green nodes (Count and save all K# after green color - #bfffbf) -

Result. ko00073: 2 and ko00073: K13356, K13357

These names are used as input to the next step. eg  ko00073 becomes ko00073.xml

1. Downloaded KGML file (eg. ko00073.xml) from Kegg using wget. This is a one time operation since ko xml file do not change.

wget "http://www.kegg.jp/kegg-bin/download?entry=ko00073&format=kgml" -O ko00073.xml

**NB: Some pathway do not have an associated KGML files. eg:** [**http://www.kegg.jp/kegg-bin/download?entry=ko01210&format=kgml**](http://www.kegg.jp/kegg-bin/download?entry=ko01210&format=kgml)

**to see what the pathway looks like, browse this url**

[**http://rest.kegg.jp/get/ko01212/kgml**](http://rest.kegg.jp/get/hsa01212/kgml)

**From this pathway: http://www.kegg.jp/kegg-bin/show\_pathway?ko01210**

**About KGML files**

**KGML is an exchange format of KEGG pathway maps. It is meant for outside users and is not used in any service or database update procedure within KEGG. KGML files, which are computationally generated from the manually defined KGML+ file, contain information about entries (KEGG objects) and two types of relationships.**

* **relations - relationships between boxes**
* **reactions - relationships between circles**

**For KGML containing both relations (between ko - orthologs) and reaction (between cpd - compound), consider only ko numbers along with relations.**

**If only reactions are present in the pathway, ignore the pathway for now.**

1. Run Qiong’s tool: KGMLPathway2Graph to convert pathway (KGML file) into graph of node and vertices. The ouput is a directory “Group” containing all graph

eg. The downloaded file ko00073.xml will be an input to Qiong's KGMLPathway2Graph tool to download all vertices and edges in the graph.

1. Step 4 and 5 are computed through a script - /data1/blanche/Bugula/Ko/koDownload.sh - and the output (graph) is saved in a directory.

* Input is a text file containing url of all pathway: koNames.txt from step 3
* output: All downloaded kgml files in a directory (koXML) and their corresponding graph files in another directory (Group or KoGrp)

1. Significance of pathway: This is computed through python script which takes as input the directory group from step 6 and a set of green nodes - setKonames - per pathway from step 3 and output a p\_value describing the significance of the pathway.

**Script 2 : Pathway extraction**

1. Run Kegg experiment on sample from **Script1**
2. Use Mozilla Firefox add-ons  “Link Gopher 1.3.3”

* open kegg result page “pathway map”
* use this filter “<http://www.kegg.jp/kegg-bin/show_pathway?@>” with link gopher to copy all green nodes per pathway. These are part of the pathway urls. Save the output in a file called All\_KGML\_URL.txt.

1. java code: extract all green node (KoNames.java)

- Read “green\_pathway” containing url of ko pathway for green nodes ( green nodes are organisms specific genes)

* input file: All\_KGML\_URL.txt
* output files: koNames.txt and setKonames

- eg:

input: http://www.kegg.jp/kegg-bin/show\_pathway?@ko00073/reference%3dwhite/default%3d%23bfffbf/K13356/K13357

Output: 2 text files containing: ko00073: #green and the other, ko00073: K13356, K13357 enumerate all green nodes (Count and save all K# after green color - #bfffbf) -

Result. ko00073: 2 and ko00073: K13356, K13357

These names are used as input to the next step. eg  ko00073 becomes ko00073.xml

1. Downloaded KGML file (eg. ko00073.xml) from Kegg using wget. This is a one time operation since ko xml file do not change.

wget "http://www.kegg.jp/kegg-bin/download?entry=ko00073&format=kgml" -O ko00073.xml

**NB: Some pathway do not have an associated KGML files. eg:** [**http://www.kegg.jp/kegg-bin/download?entry=ko01210&format=kgml**](http://www.kegg.jp/kegg-bin/download?entry=ko01210&format=kgml)

**to see what the pathway looks like, browse this url**

[**http://rest.kegg.jp/get/ko01212/kgml**](http://rest.kegg.jp/get/hsa01212/kgml)

**From this pathway: http://www.kegg.jp/kegg-bin/show\_pathway?ko01210**

**About KGML files**

**KGML is an exchange format of KEGG pathway maps. It is meant for outside users and is not used in any service or database update procedure within KEGG. KGML files, which are computationally generated from the manually defined KGML+ file, contain information about entries (KEGG objects) and two types of relationships.**

* **relations - relationships between boxes**
* **reactions - relationships between circles**

**For KGML containing both relations (between ko - orthologs) and reaction (between cpd - compound), consider only ko numbers along with relations.**

**If only reactions are present in the pathway, ignore the pathway for now.**

1. Run Qiong’s tool: KGMLPathway2Graph to convert pathway (KGML file) into graph of node and vertices. The ouput is a directory “Group” containing all graph

eg. The downloaded file ko00073.xml will be an input to Qiong's KGMLPathway2Graph tool to download all vertices and edges in the graph.

1. Step 4 and 5 are computed through a script - /data1/blanche/Bugula/Ko/koDownload.sh - and the output (graph) is saved in a directory.

* Input is a text file containing url of all pathway: koNames.txt from step 3
* output: All downloaded kgml files in a directory (koXML) and their corresponding graph files in another directory (Group or KoGrp)

1. Significance of pathway: This is computed through python script which takes as input the directory group from step 6 and a set of green nodes - setKonames - per pathway from step 3 and output a p\_value describing the significance of the pathway.