Parsing PID pathway data

Zuguang Gu

September 11, 2012

Pathway Interaction Database (PID) (http://pid.nci.nih.gov/) provides interaction data of pathways in several formats (XML and BioPAX). Here we parsed the PID XML format data (ftp://ftp1.nci.nih.gov/pub/PID/XML/).

Data stored in PID XML file can be divided into four levels: pathway level, interaction level, node level and gene/compound level. The relationship between four levels is visualized in figure 1. Generally speaking, a pathway catalogue contains a list of pathways. Each pathway is composed of a list of interactions. Each interaction is represented as relation between an input node and an output node. Each node has different characters depending on whether it is a single protein, a complex or other non-protein molecules.

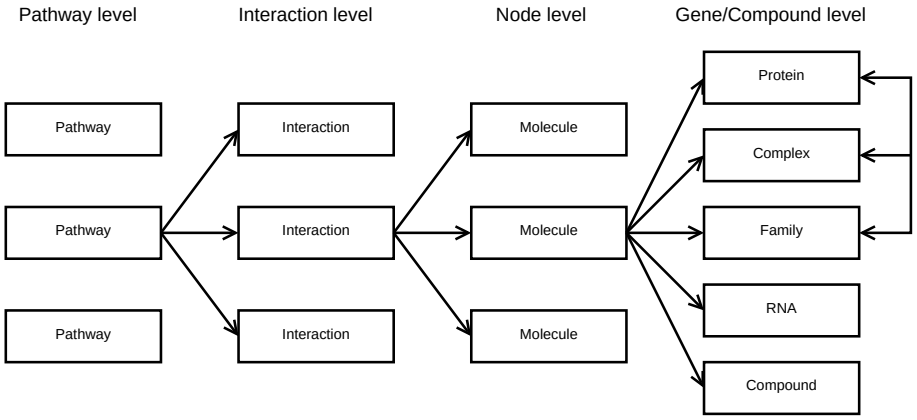

Figure 1: Representation of PID pathway data structure.

Pathway level data is embedded in `Pathway` block (figure 2). Basic information of pathways is provided here such as the pathway ID, full name and short name of the pathway. The `PathwayComponentList` block contains interactions that are involved in the pathway. Interactions are represented as interaction IDs (value of `interaction_idref` parameter) that can be linked to the specific interaction records. It should be noted that in CePa we take the short name of the pathway as the pathway unique ID rather than the value specified in `id` parameter, `Pathway` tag, because the value for `id` parameter is only for cross-reference in XML file and the short name is the unique ID in PID web site.

The pathway level data stored in R looks like as follows in which each pathway is an item of a list and contains a vector of interaction IDs.

```
<Pathway id="200094" subnet="false">
  <Organism>Hs</Organism>
  <LongName>Validated nuclear estrogen receptor beta network</LongName>
  <ShortName>erb_genomic_pathway</ShortName>
  <Source id="5">NCI-Nature Curated</Source>
  <CuratorList>
    <Curator>Kira Anthony</Curator>
  </CuratorList>
  <ReviewerList>
    <Reviewer>Julie M Hall</Reviewer>
    <Reviewer>Ann M Nardulli</Reviewer>
  </ReviewerList>
  <PathwayComponentList>
    <PathwayComponent interaction_idref="204635" />
    <PathwayComponent interaction_idref="204645" />
    <PathwayComponent interaction_idref="204637" />
    <PathwayComponent interaction_idref="204642" />
    <PathwayComponent interaction_idref="204639" />
    <PathwayComponent interaction_idref="204641" />
    <PathwayComponent interaction_idref="204643" />
    <PathwayComponent interaction_idref="204644" />
    <PathwayComponent interaction_idref="204638" />
    <PathwayComponent interaction_idref="204640" />
  </PathwayComponentList>
```

Figure 2: An example of pathway block in PID XML file.

```
> head(PID.db$NCI$pathList, n = 2)

$wnt_signaling_pathway
 [1] "203098" "203087" "203104" "203106" "203125" "203127"
 [7] "203092" "203142" "203097" "203111" "203099" "203118"
[13] "203103" "203137" "203143" "203091" "203088" "203141"
[19] "203128" "203089" "203101" "203117" "203126" "203140"
[25] "203095" "203108" "203119" "203129" "203113" "203120"
[31] "203094" "203102" "203122" "203136" "203145" "203105"
[37] "203123" "203144" "203130" "203132" "203124" "203107"
[43] "203110" "203093" "203133" "203090" "203096" "203109"
[49] "203100" "203121" "203116" "203112"

$cdc42_reg_pathway
 [1] "203416" "203396" "203420" "203393" "203405" "203418"
 [7] "203392" "203415" "203388" "203408" "203389" "203390"
[13] "203403" "203412" "203398" "203410" "203406" "203395"
[19] "203391" "203401" "203394" "203419" "203404" "203397"
[25] "203399" "203407" "203402" "203400" "203413" "203411"
[31] "203409" "203414"
```

Interaction level data is embedded in `Interaction` block (figure **??**). The most important part in it is `InteractionComponentList` block. In the block, interaction is represented by a list of nodes (in this block, nodes are also called

molecules) and their relations. There are basically three types of molecules in an interaction: input molecules, output molecules and agents identified by `role_type` parameter, `InteractionComponent` tag. A detailed characters and relations between molecules are provided such as the location of the molecule and type of the interaction. The molecules involved in the interaction are recorded with molecule IDs that can be linked to the detailed information of them. CePa only extracts the input molecule ID, output molecule ID and the agent ID to construct interactions.

The interactions in PID database some kind look like chemical reaction equations in which agents are similar to enzymes. In order to establish a pathway network, some transformations should be applied. For example in figure 3, the input molecule and the output molecule are same with just different locations, so it is not proper if we use the interaction in which input molecule directs to output molecule while leaving the agent alone. Thus we use the following rules:

1. All agents direct to input molecules.
2. If there is no input molecule, agents direct to output molecules.
3. All input molecules direct to output molecules.
4. Self-loop is not allowed.

```
<Interaction interaction_type="modification" id="206272">
  <Source id="5">NCI-Nature Curated</Source>
  <EvidenceList>
    <Evidence value="NIL">NIL</Evidence>
  </EvidenceList>
  <ReferenceList>
    <Reference pmid="11972023">11972023</Reference>
  </ReferenceList>
  <InteractionComponentList>
    <InteractionComponent role_type="input" molecule_idref="200592">
      <Label label_type="location" value="extracellular region" />
    </InteractionComponent>
    <InteractionComponent role_type="agent" molecule_idref="208363">
      <Label label_type="activity-state" value="active" />
    </InteractionComponent>
    <InteractionComponent role_type="output" molecule_idref="200592">
      <Label label_type="location" value="cytoplasm" />
    </InteractionComponent>
  </InteractionComponentList>
</Interaction>
```

Figure 3: An example of interaction block in PID XML file.

The interaction data stored in R looks like as follows in which the fist column are the interaction IDs the second and the third columns are the input node IDs and the output node IDs involved in.

```
> head(PID.db$NCI$interactionList)
```

```
  interaction.id  input output
1         503376 507485 506711
2         503376 507487 507485
3         204164 202538 208490
4         204164 208487 208490
5         100688 101169 101176
6         100688 101177 101176
```

Node level and gene level data are embedded in `Molecule` block (figure 4). Each molecule record is unified with an ID. The `molecule_type` parameter identifies the type of the molecule. If the type is protein, the link to UniProt or GenBank database is given. If the molecule is a complex or protein family, the record only provides the component IDs that can be queried from other `Molecule` blocks. Thus it is convenient to construct a complex molecule that can be composed of proteins, complex and families, repeatedly. So, with `Molecule` data, the mapping from molecule IDs to protein/gene IDs can be generated.

```
<Molecule molecule_type="protein" id="203391">
  <Name name_type="UP" long_name_type="UniProt" value="Q8NI08" />
  <Name name_type="PF" long_name_type="preferred symbol" value="ERAP140" />
</Molecule>
<Molecule molecule_type="complex" id="212736">
  <Name name_type="PF" long_name_type="preferred symbol" value="IHH N/PTCH1" />
  <ComplexComponentList>
    <ComplexComponent molecule_idref="203467">
    </ComplexComponent>
    <ComplexComponent molecule_idref="203465">
    </ComplexComponent>
  </ComplexComponentList>
</Molecule>
```

Figure 4: An example of molecule block in PID XML file.

The mapping data stored in R looks like as follows in which the first column is the node IDs and the second column is the gene IDs (here using gene symbols).

```
> head(PID.db$NCI$mapping)

  node.id  symbol
1  202230 ARHGAP6
2  201405    XIAP
3  503376  SLC7A2
4  203548   SATB1
5  201647    CRY2
6  508774     CRH
```

The parsing procedure has been implemented as Perl scripts, the scripts can directly generate an RData file that can be loaded into R session.

URL for parsing script: http://mcube.nju.edu.cn/jwang/lab/soft/cepa/