bp2: S4 object model of BioPAX level 2

Ossi Koivistoinen

June 10, 2010

1 Introduction

Bp2 package provides S4 object implementation of BioPAX level 2 [1] and some helper functions for analyzing the data. The object model is created from an RDF file read with Rredland library that is part of BioConductor [2].

2 BioPAX level 2 object model

2.1 Creating bp2 model

The bp2 package is meant for processing RDF data in a `data.frame` as returned by *Rredland* library. Any correctly formatted `data.frame` will do so `bp2` package does not have a direct dependency to `Rredland` package.

Reactome pathway 69278 that describes the human mitotic cell cycle (*Reactome_69278.owl*) is used as an example in this vignette, but it is not included with the package because of copyright issues and it's large size. The file can be loaded in BioPAX level 2 format from Reactome [3].

Creating RDF `data.frame` with `Rredland` library:

```
> require(Rredland)
```

```
A redland RDF world has been created in package:Rredland as ..GredlWorld.
```

```
> biopax.file.path <- system.file("data/Reactome_69278.owl", package = "bp2")
> biopax.owl.file.uri <- paste("file://", biopax.file.path, sep = "")
> biopax.rdf <- readRDF(biopax.owl.file.uri, storageType = "bdb")
> biopax.rdf.data.frame <- as(biopax.rdf, "data.frame")
```

```
Getting size by traversing stream.
```

```
> freeRedl(biopax.rdf)
```

Instance of `bp2.model` is created from the `data.frame` with `bp2.create.model` function:

```
> require(bp2)
> my.bp2 <- bp2.create.model(biopax.rdf.data.frame)
```

```
Allocating objects
Setting forward relations between objects
Setting inverse relations between objects
```

```
> my.bp2

Object of class "bp2.model" with slots: all.objects
symbol.names
predicate.names
all.slot.names
```

In the above example my.bp2 is a bp2.model S4 object. Printing the object to screen shows the slots of the object.

2.2 Saving and loading bp2.model objects

The bp2.model object can be saved and loaded, but there is one special command, bp2.resume(bp2.model), that has to be called after loading the object. This is because bp2 package maintains its internal environment that it uses on many operations and that environment must correspond the bp2.model object that is being processed.

Saving bp2.model objects:

```
> save(my.bp2, file = "my_bp2.RData")
```

Loading bp2.model object:

```
> load("my_bp2.RData")
> bp2.resume(my.bp2)
```

3 Model structure

bp2.model object contains a list in slot all.objects containing all BioPAX objects defined in the source OWL/RDF file. The positions in the list serve as numeric IDs of the objects. There can be (usually one) NULL objects in the list if RDF data contained objects that were not recognized by the model. The NULLs are important because the positions of the list are significant.

The bp2 package uses its own reference system implemented similar to R factors: All fields of all objects in bp2.model are vectors of integers. These integers are indexes of symbol.names list within the bp2.model object. The names of all.objects list are located in the beginning of symbol.names list. Thus, if there is integer value in any object that is less than length(my.bp2@all.objects) it is a reference to the object with that location in all.objects list and the name of that object is found with the same index in symbol.names list. If the integer is larger, then it is just a symbol name, not a reference to another object and the string value of the symbol is found in symbol.names list.

The show method for the objects automatically retrieves the names from the symbol name list so printing a objects gives human readable output.

Print an object with the symbolic names:

```
> my.bp2@all.objects[[2]]

Object of class bp2.smallMolecule
 http://www.reactome.org/biopax#ADP__ChEBI_16761_
     chemicalFormula =>
     structure =>
     molecularWeight =>
     physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_
      physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_1
      physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_2
      physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_3
      physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_4
      physicalEntity.of => http://www.reactome.org/biopax#ADP__cytosol_5
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_1
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_10
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_11
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_12
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_13
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_14
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_15
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_16
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_17
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_18
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_19
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_2
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_20
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_21
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_22
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_23
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_24
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_25
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_26
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_3
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_4
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_5
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_6
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_7
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_8
      physicalEntity.of => http://www.reactome.org/biopax#ADP__nucleoplasm_9
     availability =>
     name => "ADP ChEBI:16761^^<http://www.w3.org/2001/XMLSchema#string>"
     shortName => "ADP^^<http://www.w3.org/2001/XMLSchema#string>"
     bp2.comment =>
     dataSource => http://www.reactome.org/biopax#ReactomeDataSource
     xref => http://www.reactome.org/biopax#ChEBI_16761
     synonyms => "Adenosine 5'-diphosphate^^<http://www.w3.org/2001/XMLSchema#string>"
     rdf.id => http://www.reactome.org/biopax#ADP__ChEBI_16761_
```

To see the integer values:

```
> unclass(my.bp2@all.objects[[2]])
```

```
<S4 Type Object>
attr(,"chemicalFormula")
integer(0)
```

```
attr(,"structure")
integer(0)
attr(,"molecularWeight")
integer(0)
attr(,"physicalEntity.of")
 [1]  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27
[26] 28 29 30 31 32 33 34 35
attr(,"availability")
integer(0)
attr(,"name")
[1] 5405
attr(,"shortName")
[1] 5406
attr(,"bp2.comment")
integer(0)
attr(,"dataSource")
[1] 2942
attr(,"xref")
[1] 505
attr(,"synonyms")
[1] 5479
attr(,"rdf.id")
[1] 2
```

Count the numbers of objects in the model and extract all pathways:

```
> table(sapply(my.bp2@all.objects, class))
```

```
                         NULL                 bp2.bioSource
                            1                             1
      bp2.biochemicalReaction                 bp2.catalysis
                          169                            76
                  bp2.complex                   bp2.control
                          172                            21
               bp2.dataSource  bp2.openControlledVocabulary
                            1                             7
                  bp2.pathway               bp2.pathwayStep
                           78                           246
           bp2.physicalEntity bp2.physicalEntityParticipant
                           41                           590
                  bp2.protein           bp2.publicationXref
                          312                           316
         bp2.relationshipXref           bp2.sequenceFeature
                           39                           111
         bp2.sequenceInterval       bp2.sequenceParticipant
                           31                           844
             bp2.sequenceSite             bp2.smallMolecule
                          119                            14
          bp2.unificationXref
                         1236
```

```
> pathways <- bp2.get.instances.of(my.bp2@all.objects, "bp2.pathway")
```

There are functions to convert object or list of objects to numeric ids and the ids back to objects:

```
> my.id <- bp2.objects2ids(pathways[[1]])
> my.ids <- bp2.objects2ids(pathways)
```

4 Processing BioPAX objects

Function `bp2.find.followers(...)` is provided for extracting data from the objects. The mandatory argument is a vector of object ids to process. The function returns a list that contains one entry for each object id provided as an argument. Each list entry is a vector of integers that are IDs of other objects that are connected to the object.

Create list of all connections to other objects for all objects:

```
> not.null.objects <- my.bp2@all.objects[!(sapply(my.bp2@all.objects,
+     is.null))]
> follower.list <- bp2.find.followers(not.null.objects)

Thu Jun 10 23:27:17 2010 : 1000 / 4424
Thu Jun 10 23:27:18 2010 : 2000 / 4424
Thu Jun 10 23:27:18 2010 : 3000 / 4424
Thu Jun 10 23:27:19 2010 : 4000 / 4424
```

Note that this is a follower list presentation of graph that can be processed further with different graph packages, like `igraph` of `graphNEL`. This graph is not that useful per se, because all objects are closely connected via objects like *bp2.dataSource*. For this reason the slots that are used to build follower list can be configured. For example get only pathway steps of all pathways:

```
> pwSteps <- bp2.find.followers(pathways, include.slots = "pathwayComponents")
> pwSteps[1:5]

$`41`
[1]  169 3276  844

$`43`
[1]  181  846 3282

$`45`
[1] 44 89 42

$`47`
[1]  848 3284  183

$`49`
[1]   91  186   48 2959  532

> bp2.ids2objects(pwSteps[[1]])
```

```
$`http://www.reactome.org/biopax#Association_of_Cyclin_B_Cdc2_with_Cdc20_APC_C_complexStep`
Object of class bp2.pathwayStep
 http://www.reactome.org/biopax#Association_of_Cyclin_B_Cdc2_with_Cdc20_APC_C_complexStep
     stepInteractions => http://www.reactome.org/biopax#Association_of_Cyclin_B_Cdc2_with_Cdc2
     nextStep => http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho_APC_C_C
     nextStep.of => http://www.reactome.org/biopax#Activation_of_APC_C_Cdc20_by_dissociation_
      nextStep.of => http://www.reactome.org/biopax#Dephosphorylation_of_nuclear_Cyclin_B1_ph
     pathwayComponents.of => http://www.reactome.org/biopax#APC_C_Cdc20_mediated_degradation_
     bp2.comment =>
     rdf.id => http://www.reactome.org/biopax#Association_of_Cyclin_B_Cdc2_with_Cdc20_APC_C_c

$`http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho_APC_C_Cdc20_complexSte
Object of class bp2.pathwayStep
 http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho_APC_C_Cdc20_complexStep
     stepInteractions => http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho
      stepInteractions => http://www.reactome.org/biopax#ubiquitin_protein_ligase_activity_of
     nextStep => http://www.reactome.org/biopax#Degradation_of_multiubiquitinated_Cyclin_BSte
     nextStep.of => http://www.reactome.org/biopax#Association_of_Cyclin_B_Cdc2_with_Cdc20_AP
     pathwayComponents.of => http://www.reactome.org/biopax#APC_C_Cdc20_mediated_degradation_
     bp2.comment =>
     rdf.id => http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho_APC_C_Cdc2

$`http://www.reactome.org/biopax#Degradation_of_multiubiquitinated_Cyclin_BStep`
Object of class bp2.pathwayStep
 http://www.reactome.org/biopax#Degradation_of_multiubiquitinated_Cyclin_BStep
     stepInteractions => http://www.reactome.org/biopax#Degradation_of_multiubiquitinated_Cycl
     nextStep => http://www.reactome.org/biopax#Dissociation_of_Cdc20_from_APC_C_complexStep
     nextStep.of => http://www.reactome.org/biopax#Ubiquitination_of_Cyclin_B_by_phospho_APC_
     pathwayComponents.of => http://www.reactome.org/biopax#APC_C_Cdc20_mediated_degradation_
     bp2.comment =>
     rdf.id => http://www.reactome.org/biopax#Degradation_of_multiubiquitinated_Cyclin_BStep
```

5 Finding protein interations

Function `bp2.find.nodes` is used to find most relevant relations between pathways and protein complexes. The function recursively traverses a set of pathways and complexes following the relations presented in figure 1. The result of the function is a `bp2.nodes` object that contains various lists describing the relations between pathways. These lists can be used to construct different kinds of large graphs that describe biological processes [4].

```
> complexes <- bp2.get.instances.of(my.bp2@all.objects, "bp2.complex")
> nodes <- bp2.find.nodes(pathways, complexes)
```

```
Complex 100 / 172
Proteins not in nodes:
 character(0)
Solving inverse proteins
Solving common proteins between nodes
Almost ready
```

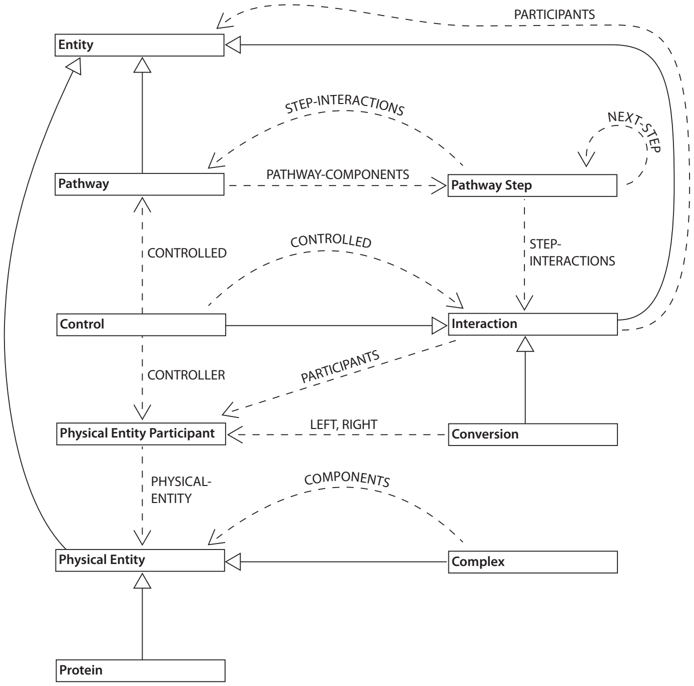

Figure 1: UML diagram of relationships relevant in finding *Protein* instances that are related to *Pathway* instances. Solid lines show inheritance and dashed lines references. *Control* objects have CONTROL-TYPE property that specifies the type of CONTROLLED relation (activation or inhibition).

```
> nodes

Object of class "bp2.nodes" with slots: ids
proteins
inverse.proteins
sub.nodes
nodes.with.common.proteins
nodes.with.activated.proteins
nodes.with.inhibited.proteins
activated.nodes
activated.proteins
inhibited.nodes
inhibited.proteins
all.followers
all.inhibited

> hist(sapply(nodes@proteins, length), plot = FALSE)[c("breaks",
+     "counts")]

$breaks
 [1]  0  5 10 15 20 25 30 35 40 45 50 55 60 65
```

```
$counts
 [1] 216  21   4   0   1   1   0   1   2   0   0   0   4
```

References

[1] Biopax: Biological pathways exchange. `http://www.biopax.org/`, April 2010.

[2] Bioconductor project. `http://www.bioconductor.org/`, April 2010.

[3] G. Joshi-Tope, M. Gillespie, I. Vastrik, P. D'Eustachio, E. Schmidt, B. de Bono, B. Jassal, G.R. Gopinath, G.R. Wu, L. Matthews, S. Lewis, E. Birney, and L. Stein. Reactome: a knowledgebase of biological pathways. *Nucl. Acids Res.*, 33(suppl.1):D428–432, 2005.

[4] Ossi Koivistoinen. Modeling cancer-associated transcriptional responses in cell-biological networks. Master's thesis, Aalto University School of Science and Technology, Espoo, Finland, 2010.