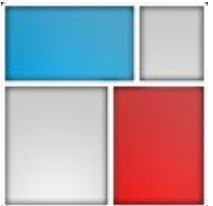


# reactome.db

Willem Ligtenberg  
Open Analytics  
[willem.ligtenberg@openanalytics.eu](mailto:willem.ligtenberg@openanalytics.eu)



# Why reactome.db

- I had quite some experience with Reactome
- Written a Cytoscape plugin (which will hopefully be published soon)
- We were using db packages like KEGG and GO, why was there no Reactome?
- I thought Bioconductor deserved a package like that
- I learned how to make it at a previous meeting in Heidelberg
- And apparently none too soon.



# KEGG.db

KEGG.db contains mappings based on older data because the original resource was removed from the the public domain before the most recent update was produced. This package should now be considered **deprecated** and future versions of Bioconductor may not have it available. One possible alternative to consider is to look at the **reactome.db** package



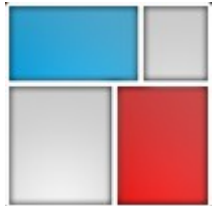
# KEGG.db

- Was (still is??) widely used
- Cannot be updated because required information is no longer publicly available
- reactome.db started off providing a similar interface, but using a different database
- Drop in replacement



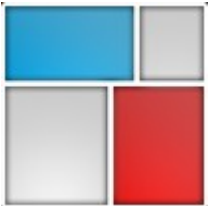
# KEGG.db

- Was (still is??) widely used
- Cannot be updated because required information is no longer publicly available
- reactome.db started off providing a similar interface, but using a different database
- Drop in replacement



# reactome.db take 2

- Marc Carslon: Why not all of Reactome?
- Current version contains a full dump of the mysql tables in sqlite.
- Still has the similar interface as the KEGG.db package
- Allows to use the full Reactome database structure



# Example KEGG like interface

- Let's get all the pathways for human  

```
pathways <- toTable(reactomePATHNAME2ID)  
pathwaysSelectedSpecies <- pathways[grep("Homo sapiens: "  
, iconv(pathways$path_name)), ]
```
- And now the genes in one of those pathways  

```
reactomeIds <- ls(reactomePATHID2EXTID)  
reactomePATHID2EXTID[[reactomeIds[1]]]
```



# Example full schema query

- Let's get the next reaction for a given reaction!

```
dbGetQuery(reactome_dbconn(), "  
SELECT do2._displayName, do._displayName  
FROM DatabaseObject AS do  
JOIN Event_2_precedingEvent AS e2p ON do.DB_ID = e2p.DB_ID  
JOIN DatabaseObject AS do2 ON e2p.precedingEvent = do2.DB_ID  
WHERE do2._displayName = 'alpha-D-Glucose + ATP => alpha-D-glucose 6-phosphate +  
ADP';")
```

- Get the input for a given reaction

```
dbGetQuery(reactome_dbconn(), "  
SELECT do._displayName, do2._displayName  
FROM DatabaseObject AS do  
JOIN ReactionlikeEvent_2_input AS r2i ON do.DB_ID = r2i.DB_ID  
JOIN DatabaseObject AS do2 ON r2i.input = do2.DB_ID  
WHERE do._displayName = 'Orc3 associates with Orc2 constitutively bound at origins of  
replication';")
```

- Warning: results are all species combined!
- Extra warning knowledge of the DB schema is required, and Reactome is quite normalised!





# RopenCL

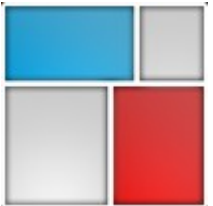
- OpenCL (Open Computing Language):  
is a framework for writing programs that execute across **heterogeneous** platforms consisting of central processing units (**CPUs**), graphics processing units (**GPUs**), DSPs and other processors.
- Package to help ease development of R packages which use OpenCL code
- Like Rcpp, but then for OpenCL

<http://repos.openanalytics.eu/html/ROpenCL.html>



# Acknowledgements

- Marc Carslon
- The previous BioConductor meeting in Heidelberg



We are hiring!  
[careers@openanalytics.eu](mailto:careers@openanalytics.eu)

