Overview

by Alexander Pico

2018-03-24

WikiPathways is a well-known repository for biological pathways that provides unique tools to the research community for content creation, editing and utilization. **R** is a powerful programming language and environment for statistical and exploratory data analysis. *rwikipathways* leverages the WikiPathways API to communicate between **R** and WikiPathways, allowing any pathway to be queried, interrogated and downloaded in both data and image formats. Queries are typically performed based on "Xrefs", standardized identifiers for genes, proteins and metabolites. Once you can identified a pathway, you can use the WPID (WikiPathways identifier) to make additional queries.

Prerequisites

All you need is this **rwikipathways** package! If you're viewing this vignette, then you've probably already installed and loaded **rwikipathways**, e.g., by means of:

source("https://bioconductor.org/biocLite.R") biocLite("rwikipathways")

or for developers:

install.packages("devtools") library(devtools) install_github('wikipathways/rwikipathways', build_vignettes=TRUE) library(rwikipathways)

Getting started

Lets first get oriented with what WikiPathways contains. For example, here's how you check to see which species are currently supported by WikiPathways:

```
    listOrganisms()
```

You should see 30 or more species listed. This list is useful for subsequent queries that take an *organism* argument, to avoid misspelling.

Next, let's see how many pathways are available for Human:

```
    hs.pathways <- listPathways('Homo sapiens')
    hs.pathways
```

Yikes! That is a lot of information. Let's break that down a bit:

```
    ?listPathways
    length(hs.pathways)
```

Ok. The help docs tell us that for each Human pathway we are getting a lot of information. A simple *length* function might be all you really want to know. Or if you're interested in just one particular piece of information, check out these functions:

```
?listPathwayIds
?listPathwayNames
?listPathwayUrls
```

These return simple lists containing just a particular piece of information for each pathway result.

Finally, there's another way to find pathways of interest: by Xref. An Xref is simply a standardized identifier form an official source. WikiPathways relies on BridgeDb to provide dozens of Xref sources for genes, proteins and metabolites. See the full list at https://github.com/bridgedb/BridgeDb/blob/master/org.bridgedb.bio/resources/org/bridgedb/bio/datasources.txt

With **rwikipathways**, the approach is simple. Take a supported identifier for a molecule of interest, e.g., an official gene symbol from HGNC, "TNF" and check the *system code* for the datasource, e.g., HGNC = H (this comes from the second column in the datasources.txt table linked to above), and then form your query:

```
tnf.pathways <- findPathwaysByXref('TNF','H')
tnf.pathways
```

Ack! That's a lot of information. We provide not only the pathway information, but also the search result strength in case you want to rank results, etc. Again, if all you're interested in is WPIDs, names or URLs, then there are these handy alternatives that will just return simple lists:

```
?findPathwayIdsByXref
?findPathwayNamesByXref
?findPathwayUrlsByXref
```

Be aware: a simple *length* function may be misleading here since a given pathway will be listed multiple times if the Xref is present mutiple times.

My favorite pathways

At this point, we should have one or more pathways identified from the queries above. Let's assume we identified 'WP554', the Ace Inhibitor Pathway (https://www.wikipathways.org/index.php/Pathway:WP554). We will use its WPID (WP554) in subsequent queries.

First off, we can get information about the pathway (if we didn't already collect it above):

```
getPathwayInfo('WP554')
```

Next, we can get all the Xrefs contained in the pathway, mapped to a datasource of our choice. How convenient! We use the same *system codes* as described above. So, for example, if we want all the genes listed as Entrez Genes from this pathway:

```
getXrefList('WP554','L')
```

Alternatively, if we want them listed as Ensembl IDs instead, then…

```
getXrefList('WP554', 'En')
```

And, if we want the metabolites, drugs and other small molecules associated with the pathways, then we'd simply provide the system code of a chemical database, e.g., Ch (HMBD), Ce (ChEBI) or Cs (Chemspider):

```
getXrefList('WP554', 'Ch')
getXrefList('WP554', 'Ce')
getXrefList('WP554', 'Cs')
```

It's that easy!

Give me more

We also provide methods for retrieving pathways as data files and as images. The native file format for WikiPathways is GPML, a custom XML specification. You can retrieve this format by…

```
gpml <- getPathway('WP554')
```

Maybe you just want to get the image of the pathway to embed in your own website, for example:

```
svg <- getColoredPathway('WP554')
```

You can save the svg to file with the function:

writeLines(svg, 'rwikipathways_example.svg')

And, as you may have suspected from the function name, you can also add color to the pathway to highlight specific elements. You will need to know the *graphIds* of the elements, however, which can be a bit tricky. Consider this an advanced features. Here's an example:

```
svg = getColoredPathway(pathway="WP554", graphId="ef1f3", color="DD88FF");
```

This will produce an svg image of the pathway with the ACE gene highlighted by color.

Wrapping up

That's an overview of the **rwikipathways** package. Check out other vignettes and the complete list of package functions:

```
browseVignettes('rWikiPathways')
help(package='rWikiPathways')
```