categoryCompare2 Scripts

Robert M Flight

2018-01-04 16:44:50

Scripts??

This version of `categoryCompare2` now includes a set of executable scripts that can be used to perform an analysis from the command line. This document attempts to document what these scripts do. The scripts are found in the folder `executables` from the package directory.

You can run the `install_executables` function to move the scripts somewhere more useful, or `list_executables` to provide the locations of the scripts so that you can make aliases and change their permissions yourself so that they are accessible from the command line directly.

Workflow

Feature Lists

`categoryCompare2` assumes you have one or more feature lists (normally genes) that you want to run enrichment on and compare the enrichments from the feature lists. For enrichment, one also needs the full list or **universe** of features that were measured in the experiment.

The easiest way to generate the input file in the expected format is to provide newline separated feature lists, each one in a separate file, where the file name indicates what condition the features came from.

In this example, we have the gene **symbols** for the differential genes at two different timepoints, **10** and **48** hours. These were calculated using the `estrogen` dataset from Bioconductor.

```
head $test_loc/10_symbol.txt
```

```
## TK1
## ELOVL2
## AGR2
## PCNA
## CDC6
## MCM6
## TFF1
## OLFM1
## MCM7
## CCNE2
```

```
head $test_loc/48_symbol.txt
```

```
## TK1
## TFF1
## MYBL2
## TOP2A
## HMGB2
## PCLAF
## UBE2C
## XBP1
## CDC20
## CCNA2
```

There is also a file with the full set of features measured on the Affymetrix chip for this data set.

To generate the feature file, we need to combine these individual files into a single JSON file. This uses `feature_files_2_json.R`

```
$exec_loc/feature_files_2_json.R --help
```

```
## Usage:
##   feature_files_2_json.R [--json=<jsonfile>] [--file1=<file1>] [--file2=<file2>] [--file3=<file3>] [
##   feature_files_2_json.R (-h | --help)
##
## Description: Concatenates multiple feature list files into a single JSON file. Each
## feature list will be named according to the file name that it came from. So, for example
## if you did:
##
## feature_files_2_json.R --file1=treatment1.txt --file2=treatment2.txt --universe=universe.txt
##
## Then there would be *treatment1*, *treatment2*, and *universe* in the JSON file
##
## Note that if *universe* is not supplied, then it will be the combination of all
## of the other feature lists supplied.
##
## Options:
##   --json=<jsonfile>     The JSON file to save to [default: features.json]
##   --file1=<file1>       The first list of features (optional)
##   --file2=<file2>       The second list of features (optional)
##   --file3=<file3>       The third list of features (optional)
##   --file4=<file4>       The fourth list of features (optional)
##   --universe=<universe> All the features measured (optional)
```

Let's run it!

```
$exec_loc/feature_files_2_json.R --json="$results_loc/features.json" \
  --file1="$test_loc/10_symbol.txt" \
  --file2="$test_loc/48_symbol.txt" \
  --universe="$test_loc/universe_symbol.txt"
```

Now we can see that the `features.json` file has the three sets of gene symbols in it.

```
head $results_loc/features.json
```

```
## {
##   "10_symbol": ["TK1", "ELOVL2", "AGR2", "PCNA", "CDC6", "MCM6", "TFF1", "OLFM1", "MCM7", "CCNE2", "
##   "48_symbol": ["TK1", "TFF1", "MYBL2", "TOP2A", "HMGB2", "PCLAF", "UBE2C", "XBP1", "CDC20", "CCNA2"
##   "universe": ["TK1", "ELOVL2", "AGR2", "PCNA", "CDC6", "MCM6", "TFF1", "OLFM1", "MCM7", "CCNE2", "F
## }
```

Annotations

```
$exec_loc/create_annotations.R --help
```

```
## Usage:
##   create_annotations.R [--input=<input-file>] [--orgdb=<organism_db>] [--feature-type=<feature-type>]
##   create_annotations.R (-h | --help)
##
## Description: Either take a feature to annotation JSON file, and reverse them
```

```
##   to create the needed annotation object for categoryCompare, or generate the
##   annotations using a Bioconductor organism or chip database.
##
## Options:
##   --input=<input-file>                  An input JSON file of features to annotations (optional)
##   --orgdb=<organism_db>                 A Bioconductor organism or chip database to use (optional)
##   --feature-type=<feature-type>         What type of features to use from the database [default: ENTI
##   --annotation-type=<annotation-type>   The annotation type [default: GO]
##   --json=<jsonfile>                     The output file [default: annotations.json]
```

Organism DB

Often times you will want to generate annotations from an included organism database that generated and maintained by Bioconductor. These are *normally* the `org.*.eg.db`, also known as organism entrez gene databases. There are other databases, including the various measurement platform databases.

See `?get_db_annotation` for all the various feature types and annotation types available.

We will create an annotation object using the `cellular component` sub-ontology of the Gene Ontology mapped to gene *symbols*.

```
$exec_loc/create_annotations.R --orgdb="org.Hs.eg.db" \
  --feature-type="SYMBOL" \
  --annotation-type="CC" \
  --json="$results_loc/annotations.json"
```

```
##
## Attaching package: 'S4Vectors'
##
## The following object is masked from 'package:base':
##
##     expand.grid
```

User Provided Annotations

If you have annotations you want to use, they should be in a JSON file where each JSON object is the feature, with the annotations that it has.

An example would be:

```
head $test_loc/create_annotations_example_input.json
```

```
## {
##   "AARSD1": ["GO:0005575", "GO:0005575", "GO:0005622", "GO:0005622", "GO:0005623", "GO:0005623", "GO
##   "ADIG": ["GO:0005575", "GO:0005575", "GO:0005622", "GO:0005623", "GO:0005634", "GO:0005737", "GO:0
##   "ADNP": ["GO:0005575", "GO:0005576", "GO:0005615", "GO:0005622", "GO:0005623", "GO:0005634", "GO:0
##   "AFMID": ["GO:0005575", "GO:0005575", "GO:0005622", "GO:0005622", "GO:0005623", "GO:0005623", "GO:
##   "AIF1": ["GO:0001726", "GO:0001891", "GO:0005575", "GO:0005575", "GO:0005575", "GO:0005622", "GO:0
##   "ALOX15B": ["GO:0005575", "GO:0005575", "GO:0005575", "GO:0005576", "GO:0005622", "GO:0005622", "G
##   "AMPD2": ["GO:0005575", "GO:0005575", "GO:0005622", "GO:0005623", "GO:0005737", "GO:0005829", "GO:
##   "ANKRD37": ["GO:0005575", "GO:0005622", "GO:0005623", "GO:0005634", "GO:0005737", "GO:0005739", "G
##   "AP3M2": ["GO:0005575", "GO:0005575", "GO:0005622", "GO:0005622", "GO:0005623", "GO:0005623", "GO:
```

Using this file, you would run:

```
$exec_loc/create_annotations.R --input="create_annotations_example_input.json" \
  --feature-type="SYMBOL" \
```

```
  --annotation-type="CC" \
  --json="$results_loc/annotations.json"
```

In this case the `--feature-type` and `--annotation-type` should reflect what the original sources that were used to generate these annotations.

Comparisons!

Finally, with the features and annotations in hand, we can do the enrichments and comparisons between them using `categoryCompare2.R`.

```
$exec_loc/categoryCompare2.R --help
```

```
## Usage:
##   categoryCompare2.R [--features=<feature-file>] [--output-directory=<save-location>] [--annotations=
##   categoryCompare2.R [--config=<config-file>]
##   categoryCompare2.R [--default-config]
##   categoryCompare2.R (-h | --help)
##   categoryCompare2.R (-v | --version)
##
## Description: Runs categoryCompare2 on one or more feature lists. Note that there are
## a lot of parameters, with several defaults. See the accompanying vignette for a
## description of what each one does. The default values can be changed at the command
## line, or using a yaml config file.
## Options:
##   --config=<config-file>                A YAML configuration file [default: NULL]
##   --default-config                      Display a default configuration file
##   --features=<feature-file>                The JSON file containing the features (genes) [default: fe
##   --output-directory=<save-location>    Where to save the results [default: cc2_results]
##   --annotations=<annotation-source>     The annotations to use, as a file [default: annotations.json]
##   --enrichment-test=<enrichment-test>   What type of test to do [default: hypergeometric]
##   --enrichment-direction=<direction>    Do you want over- or under-enrichment [default: over]
##   --p-adjustment=<p-value adjustment>   What kind of p-value correction to perform [default: BH]
##   --p-cutoff=<p-value cutoff>           What cutoff is required to denote significance? [default: 0.0
##   --count-cutoff=<min-genes>            How many genes need to be annotated to keep the annotation?
```

So, with our generated files, we would now do:

```
$exec_loc/categoryCompare2.R --features="$results_loc/features.json" \
  --annotations="$results_loc/annotations.json" \
  --p-cutoff=0.001 \
  --output-directory="$results_loc"
```

Which provides the output in /tmp/RtmpaAngU3/full_table.txt. Here is a preview of the results:

name	description	X10_symbol.sig	X48_symbol.sig	X10_symbol.mea
GO:0000779	condensed chromosome, centromeric region	FALSE	TRUE	TRUE
GO:0000780	condensed nuclear chromosome, centromeric region	FALSE	TRUE	TRUE
GO:0000781	chromosome, telomeric region	FALSE	TRUE	TRUE
GO:0005634	nucleus	FALSE	TRUE	TRUE
GO:0005663	DNA replication factor C complex	TRUE	FALSE	TRUE
GO:0015630	microtubule cytoskeleton	FALSE	TRUE	TRUE