RefNet

Paul Shannon

April 27, 2020

Contents

1 Introduction

RefNet allows you to query a large and growing collection of data sources to obtain annotated molecular interactions. Many of these sources are well-known, and many of them are from the PSCIQUIC collaboration. Other sources, *native* to this package, are culled from recent publications.

We emphasize that *RefNet* is a query tool, not a download tool. Molecular interactions are often transient, and frequently dependent upon cell-type and biological context. The rich diversity of the interactions returned by RefNet queries should always be examined closely for relevance to the actual biological topic being studied. To assist in this, RefNet interactions include annotations which describe

- detection method
- interaction type
- publication identifiers

RefNet's query interface (the `interactions` method) supports numerous filtering parameters. Combined with post-processing tools the package offers, *RefNet* provides a curatorial tool for constructing context-specific molecular networks.

2 Providers: interaction data sources

What are the currently available interaction data sources (hereafter called **providers**)?

```
> library(RefNet)
> refnet <- RefNet()

[1] initializing PSICQUIC...
[1] initializing RefNet from AnnotationHub...
[1] RefNet ready.

> providers(refnet)

$native
[1] "gerstein-2012"         "hypoxiaSignaling-2006" "stamlabTFs-2012"
[4] "recon202"              "gerstein-2012"         "hypoxiaSignaling-2006"
[7] "stamlabTFs-2012"       "recon202"

$PSICQUIC
 [1] "BioGrid"           "bhf-ucl"           "ChEMBL"
 [4] "DIP"               "DIP-IMEx"          "HPIDb"
 [7] "InnateDB"          "InnateDB-All"      "IntAct"
[10] "IMEx"              "mentha"            "MPIDB"
[13] "iRefIndex"         "MatrixDB"          "MINT"
[16] "Reactome"          "Reactome-FIs"      "EBI-GOA-miRNA"
[19] "I2D"               "UniProt"           "MBInfo"
[22] "BindingDB"         "VirHostNet"        "BAR"
[25] "EBI-GOA-nonIntAct"
```

The structure of this list reveals the two classes of providers currently offered: those which are directly contained in *RefNet* and those which are obtained via the *Bioconductor* package *PSICQUIC*. The former group ("native") will in general hold smaller, special purpose collections. New *PSICQUIC* providers, and new interactions from existing providers become available automatically. Other classes of providers in addition to these two may be added as well.

3 Quick Start: Interactions for **E2F3**

To introduce *RefNet's* principal function, `interactions`, we will query two providers for interactions with the *E2F3* transcription factor:

- **gerstein-2012**: transcription factors (TFs) and their targets, from Architecture of the human regulatory network derived from ENCODE data[1], obained by chromatin immunoprecipitation assay.
- **BioGrid**: "The Biological General Repository for Interaction Datasets (BioGRID) is a public database that archives and disseminates genetic and protein interaction data from model organisms and humans (thebiogrid.org). BioGRID currently holds over 720,000 interactions curated from both high-throughput datasets and individual focused studies, as derived from over 41,000 publications in the primary literature. Complete coverage of the entire literature is maintained for budding yeast (S. cerevisiae), fission yeast (S. pombe) and thale cress (A. thaliana), and efforts to expand curation across multiple metazoan species are underway. Current curation drives are focused on particular areas of biology to enable insights into conserved networks and pathways that are relevant to human health."

The `interactions` method has nine arguments, eight of which are optional. In practice, one or more (typically three: `id`, `species`, `provider`) are always used. For example, to obtain interactions for the transcription factor **E2F3**:

```
> if("Biogrid" %in% unlist(providers(refnet), use.names=FALSE)){
+    tbl.1 <- interactions(refnet, species="9606", id="E2F3", provider=c("gerstein-2012", "BioGrid"))
+    dim(tbl.1)
+    }
```

The full set of arguments, of which all but the first is optional:

- *object* a RefNet instance
- *id=NA* a list of one or more identifiers
- *species=NA* limit interactions to organisms, described with the NCBI taxonomy codes
- *speciesExclusive=TRUE* force all interactions to be within the specified species
- *type=NA* limit interactions to interaction types
- *provider=NA* limit interactions by providers
- *detectionMethod=NA* limit interactions by detection methods
- *publicationID=NA*
- *quiet=TRUE*

Thus *RefNet's* `interactions` method is designed for focused use in a curatorial mode, in which one limits a query by providing values to some or all of these arguments, iteratively creating a biologically relevant network of interactions, as we will demonstrate below. One *could* retrieve all interactions from all providers by calling `interactions` with all defaulted arguments. This would take a very long time, would be a disservice to the PSICQUIC providers, and would be of little benefit. It may be reasonable in some circumstances to retrieve full data sets from the *native* providers. This is demonstrated in their respective man pages.

4 Query Results: understanding the data.frame

A `data.frame` is returned by the `interactions` method. Because PSICQUIC provides many of the data sources used by *RefNet*, and because the PSI community provide a common results format which was the result of much deliberation, *RefNet* returns a `data.frame` with all of the standard PSICQUIC columns, with several (sometimes many) columns added.

Some of these additional columns are copied directly from the provider source data. *recon2* for instance, provides metabolic reaction interactions, and characterizes each reaction as reversible or not. If your query providers includes *recon2* then you will see this column added to your results. Other providers report other non-PSICQUIC data columns. *RefNet* constructs a data.frame which includes the union of all data columns reported by all of the providers, neccessarily including many missing values (currently represented in PSICQUIC style, with a '-").

Four "entity name" columns are also added to every results `data.frame`, in an attempt to solve – or at least, to ameliorate – the "identifier problem", in which different providers prefer different naming schemes for the interactions they report. PSICQUIC providers, most of whom are interested primarily in protein-protein interactions, tend to report interaction pairs as interacting proteins, using a variety of naming schemes (UniProt.kb, RefSeq, Ensembl, STRING).

Current bioinformatic practice, however, commonly describes protein interactions in terms of interactions between the genes which code for the interacting proteins. This practice is reflected in the PSICQUIC query convention: gene symbol names are used in PSICQUIC queries.

We support that practice in order to get good results from PSICQUIC providers. For *RefNet* native sources, we go further, and look for query matches against *any* identifier provided by the native source: a reaction name, a small molecule metabolite, a protein, gene symbol or an entrez geneID.

Furthermore, and crucially, every interaction in the results data.frame includes these four extra name columns:

- **A.common** a familiar, readable name, e.g. "E2F3", "acetyl-coa transport"
- **B.common**
- **A.canonical** a more formal identifier, e.g. "1871", "R_ACCOAgt"
- **B.canonical**

These columns are added to the RefNet native sources as they are parsed into the package. You must add them to interactions obtained from RefNet PSICQUIC sources by invoking the IDMapper class, from the PSICQUIC package.

```
> if("IntAct" %in% unlist(providers(refnet), use.names=FALSE)){
+    tbl.2 <- interactions(refnet, id="E2F3", provider="IntAct", species="9606")
+    dim(tbl.2)
+    idMapper <- IDMapper("9606")
+    tbl.3 <- addStandardNames(idMapper, tbl.2)
+    dim(tbl.3)
+    tbl.3[, c("A.name", "B.name", "A.id", "B.id", "type", "provider")]
+    }

   A.name B.name  A.id B.id                                type provider
1    SSR3   E2F3  6747 1871 psi-mi:MI:0915(physical association)   IntAct
2    BCL6   E2F3   604 1871          psi-mi:MI:0914(association)   IntAct
3   TFDP2   E2F3  7029 1871          psi-mi:MI:0914(association)   IntAct
4   TFDP3   E2F3 51270 1871          psi-mi:MI:0914(association)   IntAct
5     RB1   E2F3  5925 1871 psi-mi:MI:0915(physical association)   IntAct
6   TFDP1   E2F3  7027 1871          psi-mi:MI:0914(association)   IntAct
7    MSH2   E2F3  4436 1871 psi-mi:MI:0915(physical association)   IntAct
8    E2F3   RBL1  1871 5933          psi-mi:MI:0914(association)   IntAct
9    E2F3    DEK  1871 7913 psi-mi:MI:0915(physical association)   IntAct
10   E2F3    DEK  1871 7913 psi-mi:MI:0915(physical association)   IntAct
```

Mixed queries produce many columns.

```
> if("Biogrid" %in% unlist(providers(refnet), use.names=FALSE)){
+    tbl.4 <- interactions(refnet, id="ACO2", provider=c("gerstein-2012", "BioGrid"))
+    tbl.5 <- addStandardNames(idMapper, tbl.4)
+    sort(colnames(tbl.5))
+    }
```

5 Curation

Let us now examine more closely the interactions returned from the E2F3 query above, to demonstrate the curation process *RefNet* is designed to support.

5.1 detectionMethod and interaction type

Of the 54 interactions returned by that query, 10 come from *gerstein-2012* and 44 from *BioGrid*.

```
> if(exists("tbl.5")){
+     dim(tbl.5)
+     table(tbl.5$provider)
+     }
```

With what methods were these interactions detected? What interaction types were reported? Note that twelve of the BioGrid interactions were identified in a high-throughput "two hybrid" experiment, and may deserve less weight than interactions from small-scale experiments such as "western blotting" and "enzymatic study".

```
> if(exists("tbl.5")){
+    options(width=180)
+    tbl.info <- with(tbl.5, as.data.frame(table(detectionMethod, type, provider)))
+    tbl.info <- tbl.info[tbl.info$Freq>0,]
+    tbl.info
+    options(width=80)
+    }
```

5.2 Detect and Examine Duplicates

A query will often return duplicate interactions, either redundant reports of the same interaction from the same experiment and published paper, or essentially identical interactions between two entities discovered and reported more than once. You will often want to eliminate these duplicates as you build out a network. And, in general, you will want to keep the interactions which are most reliably reported, which are most specifically observed, and which come from from well-regarded experiments. You may wish to select only those interactions which come from small-scale experiments involving a cell-type identical with, or similar to, the one you are modeling.

To help with this, *RefNet* offers two related functions: `detectDuplicateInteractions` and `pickBestFromDupGroup`.

```
> if("Biogrid" %in% unlist(providers(refnet), use.names=FALSE)){
+    tbl.6 <- interactions(refnet, species="9606", id="E2F3", provider=c("gerstein-2012", "BioGrid"))
+    tbl.7 <- addStandardNames(idMapper, tbl.6)
+    tbl.withDups <- detectDuplicateInteractions(tbl.7)
+    }
```

The last function call adds a "dupGroup" column, identifying ten groups, each of which has the same two interacting molecules. The "0" group has special status: it contains unique interactions, of which 28 were found. dupGroup number 1 has three interactions:

```
> if(exists("tbl.withDups")){
+     options(width=180)
+     table(tbl.withDups$dupGroup)
+     subset(tbl.withDups, dupGroup==1)[, c("A.name", "B.name", "type", "detectionMethod", "publicationID")]
+     options(width=80)
+     }
```

We see three interactions in this dupGroup. Because the pubmed ID is the same, and the interacting proteins are the same, albeit ordered differently, we surmise that this may just be one interaction between **FZR1** and **E2F3**. We prefer the "enzymatic study", "direct interaction" version, for the extra specificity they imply, but an examination of the abstract of the source publication is often helpful:

```
> noquote(pubmedAbstract("22580460", split=TRUE))

 [1]
 [2] 1. Cell Cycle. 2012 May 15;11(10):1999-2005. doi: 10.4161/cc.20402. Epub 2012 May
 [3] 15.
 [4]
 [5] APC/C (Cdh1) controls the proteasome-mediated degradation of E2F3 during cell
 [6] cycle exit.
```

```
 [7]
 [8] Ping Z(1), Lim R, Bashir T, Pagano M, Guardavaccaro D.
 [9]
[10] Author information:
[11] (1)Hubrecht Institute-KNAW and University Medical Center Utrecht, Utrecht, The
[12] Netherlands.
[13]
[14] E2F transcription factors regulate gene expression in concert with the
[15] retinoblastoma tumor suppressor family. These transcriptional complexes are
[16] master regulators of cell cycle progression and, in addition, control the
[17] expression of genes involved in DNA repair, G 2/M checkpoint and differentiation.
[18] E2F3 has recently attracted particular attention, because it is amplified in
[19] various human tumors. Here we show that E2F3 becomes unstable as cells exit the
[20] cell cycle. E2F3 degradation is mediated by the anaphase-promoting
[21] complex/cyclosome and its activator Cdh1 (APC/C (Cdh1) ). E2F3 interacts with
[22] Cdh1 but not Cdc20, the other APC/C activator. Enforced expression of Cdh1
[23] results in proteasome-dependent degradation of E2F3, whereas the overexpression
[24] of Cdc20 has no effect on E2F3 turnover. Finally, silencing of Cdh1 by RNA
[25] interference stabilizes E2F3 in differentiating neuroblastoma cells. These
[26] findings indicate that the APC/C (Cdh1) ubiquitin ligase targets E2F3 for
[27] proteasome-dependent degradation during cell cycle exit and neuronal
[28] differentiation.
[29]
[30] DOI: 10.4161/cc.20402
[31] PMCID: PMC3359123
[32] PMID: 22580460  [Indexed for MEDLINE]
[33]
```

FZR1 is not mentioned in the abstract. Perhaps an alternate name for this gene has been used? To explore that possibiity, first obtain the entrez geneID, then see what aliases are known for it.

```
> library(org.Hs.eg.db)
> if(exists("tbl.withDups")){
+    geneID <- unique(subset(tbl.withDups, A.common=="FZR1")$A.canonical)
+    suppressWarnings(select(org.Hs.eg.db, keys=geneID, columns="ALIAS", keytype="ENTREZID"))
+    }
```

Interpreting **Cdh1** as **FZR1**, and based on the text of the abstract, we can with high confidence claim that **FZR1** interacts directly with **E2F3** resulting in its proteasome-dependent degradation: a specific, attested molecular interaction likely to be of strong interest. In the next section we demonstrate some *RefNet* function calls which speed up this process of curation.

5.3 Programmatically Eliminate Duplicates

A common *RefNet* scenario is to query all providers for interactions with a gene or protein of interest, and then – the total reported interactions being quite large – programmatically elminate all but the most interesting non-redundant interactions.

```
> providers <- intersect(unlist(providers(refnet), use.names=FALSE),
+                        c("BIND", "BioGrid", "IntAct", "MINT",
+                          "gerstein-2012"))
> tbl.8 <- interactions(refnet, species="9606", id="E2F3",
+                       provider=providers)
> tbl.9 <- addStandardNames(idMapper, tbl.8)
> dim(tbl.9)

[1] 322  32
```

Duplicate interactions can only be detected if both participating entities have canonical names assigned to them. Some PSICQUIC providers return identifiers which `addStandardNames` cannot (at the present time) map to standard identifiers. We eliminate interactions involving those few identifiers. A case-by-case "manual" study of these interactions will sometimes be warranted.

```
> removers <- with(tbl.9, unique(c(grep("^-$", A.id),
+                                  grep("^-$", B.id))))
> if(length(removers) > 0)
+     tbl.10 <- tbl.9[-removers,]
> dim(tbl.10)

[1] 311  32
```

In order to distinguish better interactions from worse an ordered list of interaction types must be provided. (For now, this is the only ranking criteria we support; detectionMethod and provider ranking will be added in the future). To begin, we must first find out the interaction types present in the current set:

```
> options(width=120)
> table(tbl.10$type)
```

```
      psi-mi:MI:0403(colocalization)   psi-mi:MI:0407(direct interaction)          psi-mi:MI:0914(association)
                                   3                                     19                                  251
psi-mi:MI:0915(physical association)
                                  38
```

We are, for now, not interested in interctions of unassigned type ("-"). We shall ingore "colocalization" as well.

```
> tbl.11 <- detectDuplicateInteractions(tbl.10)
> dupGroups <- sort(unique(tbl.11$dupGroup))
> preferred.types <- c("direct interaction",
+                      "physical association",
+                      "transcription factor binding")
> bestOfDups <- unlist(lapply(dupGroups, function(dupGroup)
+                         pickBestFromDupGroup(dupGroup, tbl.11, preferred.types)))
> deleters <- which(is.na(bestOfDups))
> if(length(deleters) > 0)
+     bestOfDups <- bestOfDups[-deleters]
> length(bestOfDups)

[1] 1

> tbl.12 <- tbl.11[bestOfDups,]
> tbl.12[, c("A.name", "B.name", "type", "provider", "publicationID")]

  A.name B.name                              type provider   publicationID
2   E2F3  ATAD2 psi-mi:MI:0407(direct interaction)  BioGrid pubmed:20855524
```

We thus obtain a high-confidence annotated list of E2F3 interactions.

6 Session info

Here is the output of `sessionInfo` on the system on which this document was compiled:

```
> toLatex(sessionInfo())
```

- R version 4.0.0 (2020-04-24), `x86_64-pc-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=C`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=en_US.UTF-8`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Running under: `Ubuntu 18.04.4 LTS`
- Matrix products: default
- BLAS: `/home/biocbuild/bbs-3.11-bioc/R/lib/libRblas.so`
- LAPACK: `/home/biocbuild/bbs-3.11-bioc/R/lib/libRlapack.so`
- Base packages: base, datasets, grDevices, graphics, methods, parallel, stats, stats4, utils
- Other packages: AnnotationDbi 1.50.0, AnnotationHub 2.20.0, Biobase 2.48.0, BiocFileCache 1.12.0, BiocGenerics 0.34.0, IRanges 2.22.0, PSICQUIC 1.26.0, RCurl 1.98-1.2, RefNet 1.24.0, S4Vectors 0.26.0, biomaRt 2.44.0, dbplyr 1.4.3, httr 1.4.1, org.Hs.eg.db 3.10.0, plyr 1.8.6, shiny 1.4.0.2
- Loaded via a namespace (and not attached): BiocManager 1.30.10, BiocStyle 2.16.0, BiocVersion 3.11.1, DBI 1.1.0, R6 2.4.1, RSQLite 2.2.0, Rcpp 1.0.4.6, XML 3.99-0.3, askpass 1.1, assertthat 0.2.1, bit 1.1-15.2, bit64 0.9-7, bitops 1.0-6, blob 1.2.1, compiler 4.0.0, crayon 1.3.4, curl 4.3, digest 0.6.25, dplyr 0.8.5, ellipsis 0.3.0, evaluate 0.14, fastmap 1.0.1, glue 1.4.0, hms 0.5.3, htmltools 0.4.0, httpuv 1.5.2, interactiveDisplayBase 1.26.0, knitr 1.28, later 1.0.0, lifecycle 0.2.0, magrittr 1.5, memoise 1.1.0, mime 0.9, openssl 1.4.1, pillar 1.4.3, pkgconfig 2.0.3, prettyunits 1.1.1, progress 1.2.2, promises 1.1.0, purrr 0.3.4, rappdirs 0.3.1, rlang 0.4.5, rmarkdown 2.1, stringi 1.4.6, stringr 1.4.0, tibble 3.0.1, tidyselect 1.0.0, tools 4.0.0, vctrs 0.2.4, xfun 0.13, xtable 1.8-4, yaml 2.2.1

References

[1] Mark B Gerstein, Anshul Kundaje, Manoj Hariharan, Stephen G Landt, Koon-Kiu Yan, Chao Cheng, Xinmeng Jasmine Mu, Ekta Khurana, Joel Rozowsky, Roger Alexander, et al. Architecture of the human regulatory network derived from encode data. *Nature*, 489(7414):91–100, 2012.