Combining OmniPath annotations and networks

Marton Olbei

09/12/2020

In this tutorial we show you how to query interactions from one of the many resources included in OmniPath, we map additional annotations to the data, and combine the two to build a small tissue specific network out of them

We'll start by importing libraries, first `OmnipathR`, and `dplyr` for data wrangling.

```
library(OmnipathR)
library(dplyr)
```

Importing interactions

With the `OmnipathR` package loaded, we can download a subset of the interaction data. By calling the `import_omnipath_annotations` function we assign interaction data from the **BioGrid** resource to the `interactions` variable restricting it to human interaction data, by selecting the taxonomy ID **9606**.

It is important to mention, that the `OmnipathR` library queries can be replicated in browser too, as they access specific URLs, depending on the parameters we give here.

```
interactions <- import_all_interactions(
  resources = c('SIGNOR'),
  organism = 9606
  )
```

```
## Downloaded 11619 interactions.
```

The R query above translates to the following URL for example. Note that only the required resources and taxIDs are selected, but the rest of the query remained intact. Feel free to give it a go in your browser:

URL: https://omnipathdb.org/interactions?genesymbols=yes&resources=SIGNOR&datasets=dorothea,kinaseextra,ligrecextra,lncrna_mrna,mirnatarget,omnipath,pathwayextra,tf_mirna,tf_target,tfregulons&dorothea_levels=A,B&fields=sources,references,curation_effort,dorothea_level&license=academic

The result should look something like below (in R).

```
interactions %>% tibble()
```

```
## # A tibble: 11,619 x 17
##    source target source_genesymb~ target_genesymb~ is_directed is_stimulation
##    <chr>  <chr>  <chr>            <chr>                  <int>          <int>
##  1 Q13976 Q13507 PRKG1            TRPC3                      1              0
##  2 P18031 Q9H1D0 PTPN1            TRPV6                      1              0
```

```
##  3 P63244 Q9BX84 RACK1            TRPM6                     1             0
##  4 Q96QT4 P04083 TRPM7            ANXA1                     1             1
##  5 P17612 Q9GZU1 PRKACA           MCOLN1                    1             0
##  6 Q16539 P49137 MAPK14           MAPKAPK2                  1             1
##  7 P49137 O95453 MAPKAPK2         PARN                      1             0
##  8 P49757 P46531 NUMB             NOTCH1                    1             1
##  9 O00548 P46531 DLL1             NOTCH1                    1             1
## 10 P31749 O15111 AKT1             CHUK                      1             1
## # ... with 11,609 more rows, and 11 more variables: is_inhibition <int>,
## #   consensus_direction <int>, consensus_stimulation <int>,
## #   consensus_inhibition <int>, dip_url <chr>, sources <chr>, references <chr>,
## #   curation_effort <int>, dorothea_level <chr>, n_references <int>,
## #   n_resources <int>
```

Importing annotations

To give these interactions a bit more depth, we can map annotation data to the interactions. To take a look at the available annotation resources in OmniPath, call the `get_annotation_resources()` function by running the line below, or calling it from the console.

```
get_annotation_resources()
```

```
##  [1] "Adhesome"             "Almen2009"            "Baccin2019"
##  [4] "CancerGeneCensus"     "CancerSEA"            "CellCellInteractions"
##  [7] "CellPhoneDB"          "CellPhoneDB_complex"  "ComPPI"
## [10] "CORUM_Funcat"         "CORUM_GO"             "CSPA"
## [13] "CSPA_celltype"        "DGIdb"                "DisGeNet"
## [16] "EMBRACE"              "Exocarta"             "GO_Intercell"
## [19] "GPCRdb"               "Guide2Pharma"         "HGNC"
## [22] "HPA_secretome"        "HPA_subcellular"      "HPA_tissue"
## [25] "HPMR"                 "HPMR_complex"         "ICELLNET"
## [28] "ICELLNET_complex"     "Integrins"            "IntOGen"
## [31] "iTALK"                "KEGG-PC"              "kinase.com"
## [34] "Kirouac2010"          "LOCATE"               "LRdb"
## [37] "Matrisome"            "MatrixDB"             "MCAM"
## [40] "Membranome"           "MSigDB"               "NetPath"
## [43] "OPM"                  "Phobius"              "Phosphatome"
## [46] "Ramilowski_location"  "Ramilowski2015"       "SignaLink_function"
## [49] "SignaLink_pathway"    "SIGNOR"               "Surfaceome"
## [52] "TCDB"                 "TFcensus"             "TopDB"
## [55] "UniProt_family"       "UniProt_keyword"      "UniProt_location"
## [58] "UniProt_tissue"       "UniProt_topology"     "Vesiclepedia"
## [61] "Zhong2015"
```

Let's import tissue enrichment data from the **Human Protein Atlas**. Calling the `import_omnipath_annotations` function first we select the proteins we'd like to gather information on, followed by the resources we are pulling the data from.

A toy example below:

We assign the annotations of the proteins TP53 and LMNA from the tissue section of the HPA into the HPA_small variable. The "wide" setting pivots the data from a long format to wide, which gives us a nice table from the queried data.

```
HPA_small <- import_omnipath_annotations(
  proteins = c('TP53', 'LMNA'),
  resources = c('HPA_tissue'),
  wide = TRUE
  )
```

```
## Downloaded 1549 annotation records.
```

These queries are also URL accessible. This toy query translates to the following: URL: https://omnipathdb.org/annotations?resources=HPA_tissue&proteins=TP53,LMNA&license=academic

Let's get the unique proteins from SIGNOR into a vector (list)

```
protein_list <- c(interactions$source, interactions$target) %>% unique()
```

We can pass this into `import_omnipath_annotations` just like above. To ensure sensible runtimes for this tutorial here we restrict the queried proteins to the first 500 in the list with the [1:500] slice.

```
HPA_signor <- import_omnipath_annotations(
  proteins = protein_list[1:500],
  resources = c('HPA_tissue'),
  wide = TRUE
  )
```

```
## Downloaded 325162 annotation records.
```

Now that we have the data, let's filter it down to a specific case, like thyroid cancer. We filter out rows where the levels of the proteins are not "not detected", i.e. we only move forward with the ones that are detected.

```
HPA_thyroid_cancer <- HPA_signor %>%
  filter(
    tissue == "thyroid cancer",
    level != "Not detected"
    )
```

Combining datasets into a tissue specific network

We can combine the two datasets (`interactions` and `HPA_thyroid_cancer`) with an `inner join`. This operation returns records that have matching values in both tables. We first map the annotation to the `source` column (i.e. the initial member of each interaction) and we "pipe" (forward) these results with the `%>%` symbol into a second join, where we map the results of the first operation to the `target` column proteins.

```
thyroid_cancer_network <- inner_join(
  interactions, HPA_thyroid_cancer, # here we select the two dataframes
  by=c("source"="uniprot")) %>% # the names of the columns to combine them on
  inner_join(
    HPA_thyroid_cancer, # the first dataframe is the result of the first operation here, second
    by=c("target"="uniprot"), # the names of the columns to combine them on
    suffix=c(".source", ".target")) # suffixes, so that we know which column corresponds to

thyroid_cancer_network %>% tibble()
```

```
## # A tibble: 966 x 45
##    source target source_genesymb~ target_genesymb~ is_directed is_stimulation
##    <chr>  <chr>  <chr>            <chr>                  <int>          <int>
##  1 Q16539 P49137 MAPK14           MAPKAPK2                   1              1
##  2 P49757 P46531 NUMB             NOTCH1                     1              1
##  3 P31749 O15111 AKT1             CHUK                       1              1
##  4 O15111 P19838 CHUK             NFKB1                      1              1
##  5 P38936 P06493 CDKN1A           CDK1                       1              0
##  6 Q93009 Q00987 USP7             MDM2                       1              1
##  7 P60484 Q70Z35 PTEN             PREX2                      1              0
##  8 Q70Z35 P60484 PREX2            PTEN                       1              0
##  9 P30086 P04049 PEBP1            RAF1                       1              1
## 10 P45985 P45983 MAP2K4           MAPK8                      1              1
## # ... with 956 more rows, and 39 more variables: is_inhibition <int>,
## #   consensus_direction <int>, consensus_stimulation <int>,
## #   consensus_inhibition <int>, dip_url <chr>, sources <chr>, references <chr>,
## #   curation_effort <int>, dorothea_level <chr>, n_references <int>,
## #   n_resources <int>, genesymbol.source <chr>, entity_type.source <chr>,
## #   organ.source <chr>, tissue.source <chr>, level.source <chr>,
## #   status.source <chr>, prognostic.source <chr>, favourable.source <chr>,
## #   pathology.source <chr>, n_not_detected.source <chr>, n_low.source <chr>,
## #   n_medium.source <chr>, n_high.source <chr>, score.source <chr>,
## #   genesymbol.target <chr>, entity_type.target <chr>, organ.target <chr>,
## #   tissue.target <chr>, level.target <chr>, status.target <chr>,
## #   prognostic.target <chr>, favourable.target <chr>, pathology.target <chr>,
## #   n_not_detected.target <chr>, n_low.target <chr>, n_medium.target <chr>,
## #   n_high.target <chr>, score.target <chr>
```

The result is a thyroid cancer specific interaction network, roughly 2% the size of the original **SIGNOR** network.

In this tutorial we learned:

- How to query interaction data from one of the many resoruces in OmniPath
- How to query annotation data from OmniPath, and combine it with the interaction sources
- How to generate a tissue specific network with the usage of the two datatypes