Customize intercellular networks

Marton Olbei

10/12/2020

In this tutorial we show you how to query interactions from the intercellular interactions in OmniPath, and go over the various attributes accompanying them.

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

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

The intercellular interactions in OmniPath are collated from a number of sources. When putting together a query, you can select all of these, or just a preferred subset of them. The `get_intercell_resources` function returns the list of datasets included in the database.

```
get_intercell_resources()
```

```
##  [1] "Adhesome"             "Almen2009"            "Baccin2019"
##  [4] "CellCellInteractions" "CellPhoneDB"          "ComPPI"
##  [7] "CSPA"                 "DGIdb"                "EMBRACE"
## [10] "GO_Intercell"         "GPCRdb"               "Guide2Pharma"
## [13] "HGNC"                 "HPA_secretome"        "HPMR"
## [16] "ICELLNET"             "Integrins"            "iTALK"
## [19] "Kirouac2010"          "LOCATE"               "LRdb"
## [22] "Matrisome"            "MatrixDB"             "MCAM"
## [25] "Membranome"           "OmniPath"             "OPM"
## [28] "Phobius"              "Ramilowski_location"  "Ramilowski2015"
## [31] "SignaLink_function"   "Surfaceome"           "TopDB"
## [34] "UniProt_keyword"      "UniProt_location"     "UniProt_topology"
## [37] "Zhong2015"
```

These resources contain a large variety of actors we can use to build intercellular interactions. Take a peek at a generalized list of these categories by using the `get_intercell_generic_categories` function.

This list is also accessible from the browser, at https://omnipathdb.org/intercell_summary. Using the `get_intercell_categories` command returns the complete list.

```
get_intercell_generic_categories()
```

```
##  [1] "plasma_membrane"                  "transmembrane"
##  [3] "peripheral"                       "transmembrane_predicted"
##  [5] "receptor"                         "adhesion"
##  [7] "ligand"                           "cell_surface_ligand"
##  [9] "ecm"                              "secreted"
```

```
## [11] "ion_channel"                          "cell_surface"
## [13] "transporter"                          "ligand_regulator"
## [15] "receptor_regulator"                   "plasma_membrane_transmembrane"
## [17] "cell_adhesion"                        "extracellular"
## [19] "matrix_adhesion"                      "secreted_receptor"
## [21] "plasma_membrane_peripheral"           "plasma_membrane_regulator"
## [23] "secreted_enzyme"                      "cell_surface_enzyme"
## [25] "matrix_adhesion_regulator"            "secreted_peptidase"
## [27] "intracellular"                        "cell_surface_peptidase"
## [29] "extracellular_peptidase"              "secreted_peptidase_inhibitor"
## [31] "secreted_enyzme"                      "desmosome"
## [33] "ecm_regulator"                        "gap_junction"
## [35] "sparc_ecm_regulator"                  "intracellular_intercellular_related"
## [37] "adherens_junction"                    "ion_channel_regulator"
## [39] "tight_junction"
```

Now that we have seen the resources and categories, we have to go over a few definitions related to them to make sure everything is clear going forward.

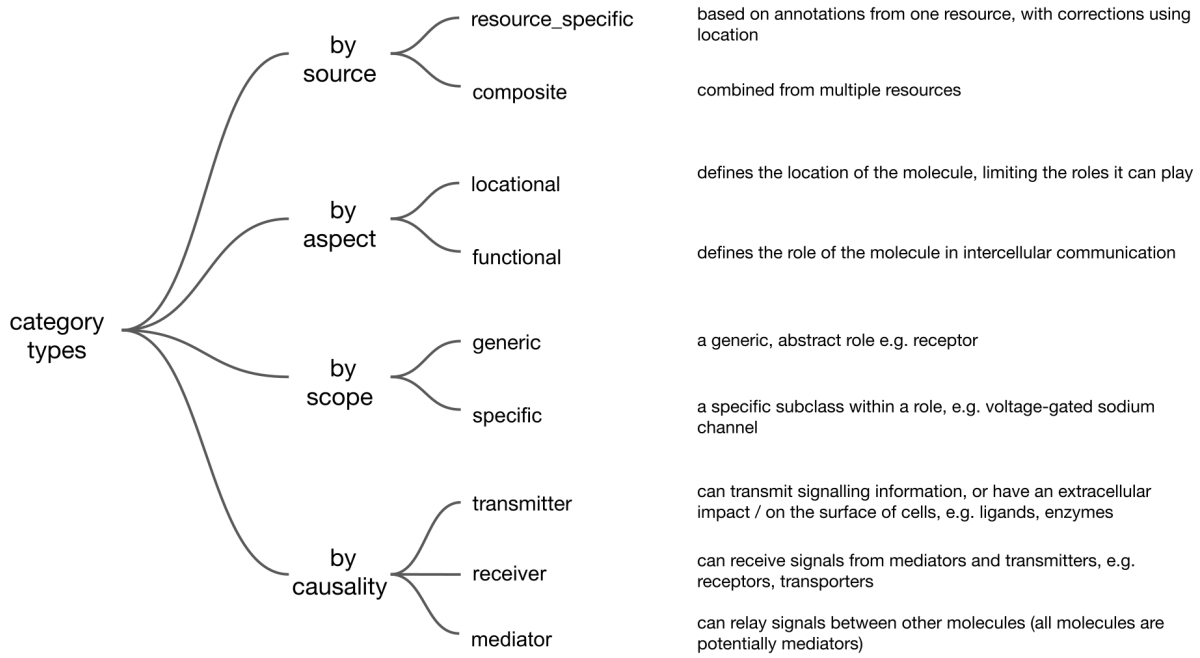

Figure 1: definitions

To import an intercellular network we call the `import_intercell_network` function. The function has three main steps:

- first, we have to define the datasets to import from under `interactions_param`
- following that, we can specify the qualities of the transmitting proteins, such as which categories they should belong to, what should be their scope, source, aspect and so on under `transmitter_param`
- once we are done with that, we should specify the same for the receiving molecules under `receiver_param`

In this example below we generate a large intercellular network, where we are attempting to connect ligands to receptors.

These steps can be individually traced back through URLs:

1. interaction parameters: https://omnipathdb.org/interactions?genesymbols=yes&datasets=omnipath,pathwayextra,ligrecextra&organisms=9606&fields=sources,references,curation_effort&license=academic
2. transmitter parameters: https://omnipathdb.org/intercell?scope=generic&categories=ligand&causality=trans&license=academic
3. receiver parameters: https://omnipathdb.org/intercell?scope=generic&categories=receptor&causality=rec&license=academic

```
interactions <- import_intercell_network(
    interactions_param = list(
        datasets = c('omnipath', 'pathwayextra', 'ligrecextra')
    ),
    transmitter_param = list(
        categories = c('ligand')
    ),
    receiver_param = list(
        categories =c('receptor')
    )
)
```

```
interactions
```

```
## # A tibble: 7,604 x 44
##    category_interc~ parent_intercel~ source target category_interc~
##    <chr>            <chr>            <chr>  <chr>  <chr>
##  1 ligand           ligand           A6NMZ7 O75056 receptor
##  2 ligand           ligand           A6NMZ7 O75578 receptor
##  3 ligand           ligand           A6NMZ7 P05106 receptor
##  4 ligand           ligand           A6NMZ7 P05556 receptor
##  5 ligand           ligand           A6NMZ7 P06756 receptor
##  6 ligand           ligand           A6NMZ7 P08514 receptor
##  7 ligand           ligand           A6NMZ7 P08648 receptor
##  8 ligand           ligand           A6NMZ7 P13612 receptor
##  9 ligand           ligand           A6NMZ7 P16070 receptor
## 10 ligand           ligand           A6NMZ7 P16144 receptor
## # ... with 7,594 more rows, and 39 more variables:
## #   parent_intercell_target <chr>, target_genesymbol <chr>,
## #   source_genesymbol <chr>, is_directed <int>, is_stimulation <int>,
## #   is_inhibition <int>, consensus_direction <int>,
## #   consensus_stimulation <int>, consensus_inhibition <int>, dip_url <chr>,
## #   sources <chr>, references <chr>, curation_effort <int>, n_references <int>,
## #   n_resources <int>, database_intercell_source <chr>,
## #   scope_intercell_source <chr>, aspect_intercell_source <chr>,
## #   category_source_intercell_source <chr>, genesymbol_intercell_source <chr>,
## #   entity_type_intercell_source <chr>, consensus_score_intercell_source <int>,
## #   transmitter_intercell_source <lgl>, receiver_intercell_source <lgl>,
## #   secreted_intercell_source <lgl>,
## #   plasma_membrane_transmembrane_intercell_source <lgl>,
## #   plasma_membrane_peripheral_intercell_source <lgl>,
## #   database_intercell_target <chr>, scope_intercell_target <chr>,
## #   aspect_intercell_target <chr>, category_source_intercell_target <chr>,
```

```
## #   genesymbol_intercell_target <chr>, entity_type_intercell_target <chr>,
## #   consensus_score_intercell_target <int>, transmitter_intercell_target <lgl>,
## #   receiver_intercell_target <lgl>, secreted_intercell_target <lgl>,
## #   plasma_membrane_transmembrane_intercell_target <lgl>,
## #   plasma_membrane_peripheral_intercell_target <lgl>
```

This results in 7604 interactions. Let's narrow it down by restricting it with some of the categorical data outlined above.

```
interactions_small <- import_intercell_network(
    interactions_param = list(
        datasets = c('omnipath', 'pathwayextra', 'ligrecextra')
    ),
    transmitter_param = list(
        categories = c('ligand'),
        scope = c('specific') # let's restrict the scope to be specific
    ),
    receiver_param = list(
        categories =c('receptor')
    )
)
```

```
interactions_small
```

```
## # A tibble: 183 x 44
##    category_interc~ parent_intercel~ source target category_interc~
##    <chr>            <chr>            <chr>  <chr>  <chr>
##  1 ligand           ligand           O00548 P46531 receptor
##  2 ligand           ligand           O00548 Q04721 receptor
##  3 ligand           ligand           O00548 Q99466 receptor
##  4 ligand           ligand           O00548 Q9UM47 receptor
##  5 ligand           ligand           P01889 O43908 receptor
##  6 ligand           ligand           P01889 P01732 receptor
##  7 ligand           ligand           P01889 P04234 receptor
##  8 ligand           ligand           P01889 P09693 receptor
##  9 ligand           ligand           P01889 P10966 receptor
## 10 ligand           ligand           P01889 P26715 receptor
## # ... with 173 more rows, and 39 more variables: parent_intercell_target <chr>,
## #   target_genesymbol <chr>, source_genesymbol <chr>, is_directed <int>,
## #   is_stimulation <int>, is_inhibition <int>, consensus_direction <int>,
## #   consensus_stimulation <int>, consensus_inhibition <int>, dip_url <chr>,
## #   sources <chr>, references <chr>, curation_effort <int>, n_references <int>,
## #   n_resources <int>, database_intercell_source <chr>,
## #   scope_intercell_source <chr>, aspect_intercell_source <chr>,
## #   category_source_intercell_source <chr>, genesymbol_intercell_source <chr>,
## #   entity_type_intercell_source <chr>, consensus_score_intercell_source <int>,
## #   transmitter_intercell_source <lgl>, receiver_intercell_source <lgl>,
## #   secreted_intercell_source <lgl>,
## #   plasma_membrane_transmembrane_intercell_source <lgl>,
## #   plasma_membrane_peripheral_intercell_source <lgl>,
## #   database_intercell_target <chr>, scope_intercell_target <chr>,
## #   aspect_intercell_target <chr>, category_source_intercell_target <chr>,
## #   genesymbol_intercell_target <chr>, entity_type_intercell_target <chr>,
```

```
## #   consensus_score_intercell_target <int>, transmitter_intercell_target <lgl>,
## #   receiver_intercell_target <lgl>, secreted_intercell_target <lgl>,
## #   plasma_membrane_transmembrane_intercell_target <lgl>,
## #   plasma_membrane_peripheral_intercell_target <lgl>
```

In this tutorial we learned:

- the data sources used to generate intercellular interactions
- the qualities of intercellular interactors
- the functions to generate and specify intercellular interactions