ligrec_vignette

December 8, 2020

1 Receptor-ligand permutation test

This vignette demonstrates the usage of the permutation test as described in CellPhoneDB with the interations extracted from OmniPath database developed by Saezlab.

The downside is that in the original implementation, apart from being inefficient, the CellPhoneDB database has only been manually curated for human interactions. To overcome this issue, we make use of the OmniPath database (containing CellPhoneDB as one of its many sources) which also focuses on literature curated rodent signalling pathways.

1.1 Imports

[1]:
```
import numpy as np
import pandas as pd

import scanpy as sc
import squidpy as sp

from anndata import AnnData
```

1.2 Usage

[2]:
```
help(sp.gr.ligrec)
```

```
Help on function ligrec in module squidpy.gr._ligrec:

ligrec(adata: 'AnnData', cluster_key: 'str', interactions:
'Optional[InteractionType]' = None, complex_policy: 'str' = 'min', threshold:
'float' = 0.01, fdr_method: 'Optional[str]' = None, fdr_axis: 'str' =
'clusters', copy: 'bool' = False, key_added: 'Optional[str]' = None, **kwargs)
-> 'Optional[LigrecResult]'
    Perform the permutation test as described in [CellPhoneDB20]_.

    Parameters
    ----------
    adata
        Annotated data object. Must contain :attr:`anndata.AnnData.raw`
attribute.
    interactions
```

```
        Interaction to test. The type can be one of:

            - :class:`pandas.DataFrame` - must contain at least 2 columns named
`'source'` and `'target'`.
            - :class:`dict` - dictionary with at least 2 keys named `'source'`
and `'target'`.
            - :class:`typing.Sequence` - Either a sequence of :class:`str`, in
which case all combinations are
              produced, or a sequence of :class:`tuple` of 2 :class:`str` or a
:class:`tuple` of 2 sequences.

        If `None`, the interactions are extracted from :mod:`omnipath`. Protein
complexes can be specified by
        delimiting the components using `_`, such as `'alpha_beta_gamma'`.
    complex_policy
        Policy on how to handle complexes. Can be one of:

            - `'min'` - select gene with the minimum average expression.
              This is the same as in [CellPhoneDB20]_.
            - `'all'` - select all possible combinations between complexes
`'source'` and `'target'`.
    interactions_params
        Keyword arguments for
:func:`omnipath.interactions.import_intercell_network` defining the
interactions.
        These datasets from [OmniPath16]_ are used by default: `'omnipath'`,
`'pathwayextra'` `'kinaseextra'`,
        `'ligrecextra'`.
    transmitter_params
        Keyword arguments for
:func:`omnipath.interactions.import_intercell_network` defining the transmitter
        side of intercellular connections.
    receiver_params
        Keyword arguments for
:func:`omnipath.interactions.import_intercell_network` defining the receiver
        side of intercellular connections.
    cluster_key
        Key in :attr:`anndata.AnnData.obs` where clusters are stored.
    clusters
        Clusters from :attr:`anndata.AnnData.obs` ``[cluster_key]``. Can be
specified either as a sequence
        of :class:`tuple` or just a sequence of cluster names, in which case all
combinations are created.
    n_perms
        Number of permutations for the permutation test.
    threshold
        Do not perform permutation test if any of the interacting components is
being expressed
```

```
        in less than ``threshold`` percent of cells within a given cluster.
    seed
        Random seed for reproducibility.

    Returns
    -------
    :class:`collections.namedtuple` or None
        If ``copy = False``, updates ``adata.uns[{key_added}]`` with the
following triple:

            - `'means'` - :class:`pandas.DataFrame` containing the mean
expression.
            - `'pvalues'` - :class:`pandas.DataFrame` containing the possibly
corrected p-values.
            - `'metadata'` - :class:`pandas.DataFrame` containing interaction
metadata.

        Otherwise, just returns the result.

        `NaN` p-values mark combinations for which the mean expression of one of
the interacting components was `0`
        or it didn't pass the ``threshold`` percentage of cells being expressed
within a given cluster.
```

2 Load the mouse data

[3]:
```
adata = sc.datasets.paul15()
```

```
WARNING: In Scanpy 0.*, this returned logarithmized data. Now it returns non-
logarithmized data.
```

```
… storing 'paul15_clusters' as categorical
Trying to set attribute `.uns` of view, copying.
```

Normalize and create `.raw`.

[4]:
```
sc.pp.normalize_per_cell(adata)

adata.raw = adata.copy()
```

2.1 Run the CellPhoneDB's permutation test

2.1.1 Use only CellPhoneDB as a resource

For mouse data, CellPhoneDB uses the ortholog genes, downloaded from biomart. They convert the mouse genes into their human orthologs and use that as an input (latest source).

[5]:
```
res = sp.gr.ligrec(adata, "paul15_clusters",
                   fdr_method=None, copy=True,
                   interactions_params={"resources": "CellPhoneDB"},
                   threshold=0.1, seed=0, n_perms=10000, n_jobs=1)
df = res.pvalues
print("Number of CellPhoneDB interactions (mouse data):", len(df))

df.head()
```

```
/home/michal/.miniconda3/envs/spatial/lib/python3.8/site-
packages/requests/__init__.py:89: RequestsDependencyWarning: urllib3 (1.26.2) or
chardet (3.0.4) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({}) doesn't match a supported "
```

```
WARNING: Removed `1` duplicate gene(s)
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=10000.0),␣
 ↪HTML(value='')))
```

```
Number of CellPhoneDB interactions (mouse data): 9
```

[5]:

cluster_1		10GMP								\
cluster_2		10GMP	11DC	12Baso	13Baso	14Mo	15Mo	16Neu	17Neu	18Eos
source	target									
GRN	TNFRSF1A	0.2868	0.4736	0.531	NaN	NaN	NaN	NaN	0.204	0.7944
TGFB1	TGFBR3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	TGFBR2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0061
	ITGAV	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
TGFBR3	TGFB1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

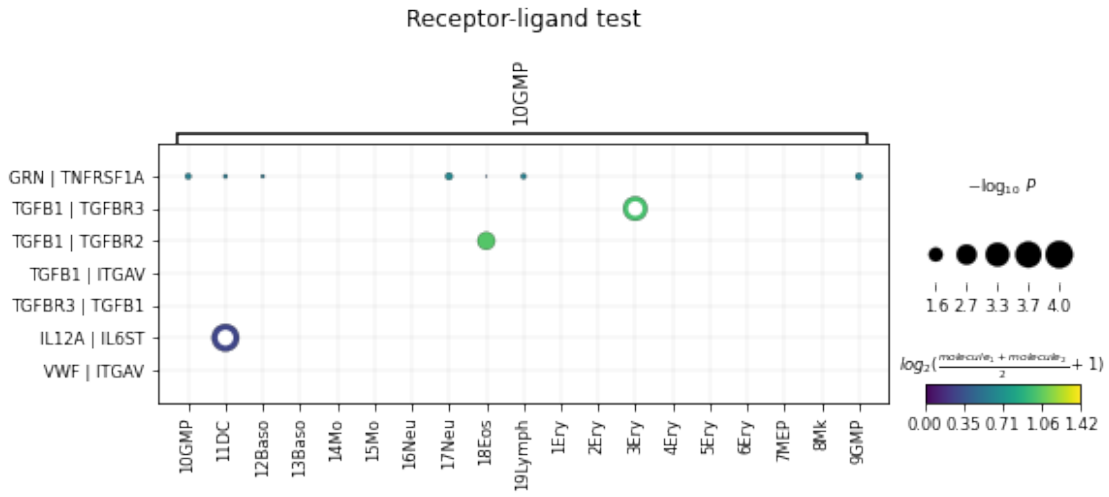
cluster_1		…	9GMP							\
cluster_2		19Lymph	…	19Lymph	1Ery	2Ery	3Ery	4Ery	5Ery	6Ery	7MEP
source	target		…								
GRN	TNFRSF1A	0.3359	…	0.1427	NaN	NaN	NaN	NaN	0.8665	NaN	NaN
TGFB1	TGFBR3	NaN	…	NaN	NaN	NaN	0.0648	NaN	NaN	NaN	NaN
	TGFBR2	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
	ITGAV	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
TGFBR3	TGFB1	NaN	…	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

cluster_1			
cluster_2		8Mk	9GMP
source	target		
GRN	TNFRSF1A	NaN	0.0908
TGFB1	TGFBR3	0.0165	NaN
	TGFBR2	NaN	NaN
	ITGAV	NaN	NaN
TGFBR3	TGFB1	NaN	NaN

```
[5 rows x 361 columns]
```

The tori mark significant p-values (`alpha=0.001` by default). `molecule1` belongs to the source cluster (top) whereas `moleule2` to the target clusters.

[6]:
```
sp.pl.ligrec(res, source_groups="10GMP")
```



2.1.2 Use all available resources from OmniPath

[7]:
```
res = sp.gr.ligrec(adata, "paul15_clusters",
                   fdr_method=None, copy=True,
                   threshold=0.1, seed=0, n_perms=10000, n_jobs=1)
df = res.pvalues
print("Number of OmniPath interactions (mouse data):", len(df))

df.head()
```

```
WARNING: Removed `1` duplicate gene(s)
```

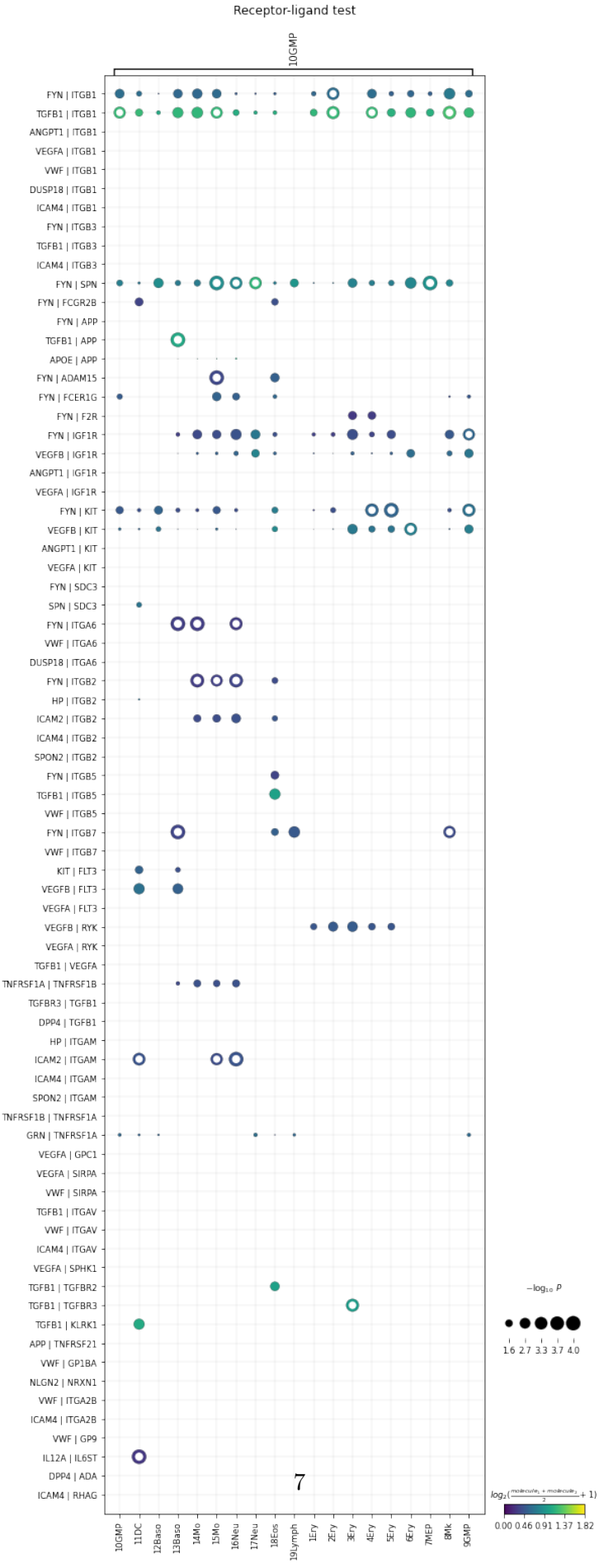
```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=10000.0),␣
↪HTML(value='')))
```

```
Number of OmniPath interactions (mouse data): 107
```

[7]:
```
cluster_1       10GMP                                                   \
cluster_2       10GMP    11DC  12Baso  13Baso    14Mo    15Mo   16Neu   17Neu
source target
FYN     THY1      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
        ITGB1  0.0060  0.0843  0.7125  0.0068  0.0036  0.0067  0.4110  0.5669
TGFB1   ITGB1  0.0009  0.0206  0.1831  0.0022  0.0014  0.0010  0.0505  0.2222
ANGPT1  ITGB1     NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN
```

```
VEGFA  ITGB1      NaN     NaN     NaN     NaN     NaN     NaN     NaN     NaN

cluster_1                        …    9GMP                                \
cluster_2       18Eos 19Lymph    … 19Lymph    1Ery    2Ery    3Ery    4Ery
source target                    …
FYN    THY1       NaN     NaN    …     NaN     NaN     NaN     NaN     NaN
       ITGB1   0.3759     NaN    …     NaN  0.0953  0.0039  0.0001  0.0083
TGFB1  ITGB1   0.1867     NaN    …     NaN  0.1474  0.0585  0.0093  0.0590
ANGPT1 ITGB1      NaN     NaN    …     NaN  0.0744  0.0001     NaN  0.0018
VEGFA  ITGB1      NaN     NaN    …     NaN     NaN     NaN     NaN     NaN

cluster_1
cluster_2        5Ery    6Ery    7MEP     8Mk    9GMP
source target
FYN    THY1       NaN     NaN     NaN     NaN     NaN
       ITGB1   0.0944  0.0291  0.1237  0.0018  0.0292
TGFB1  ITGB1   0.2032  0.1306  0.2321  0.0170  0.0691
ANGPT1 ITGB1   0.0430  0.0104  0.0638  0.0005  0.0130
VEGFA  ITGB1      NaN     NaN     NaN     NaN     NaN

[5 rows x 361 columns]
```

[8]:
```
sp.pl.ligrec(res, source_groups="10GMP")
```


Receptor-ligand test
10GMP
FYN | ITGB1
TGFB1 | ITGB1
ANGPT1 | ITGB1
VEGFA | ITGB1
VWF | ITGB1
DUSP18 | ITGB1
ICAM4 | ITGB1
FYN | ITGB3
TGFB1 | ITGB3
ICAM4 | ITGB3
FYN | SPN
FYN | FCGR2B
FYN | APP
TGFB1 | APP
APOE | APP
FYN | ADAM15
FYN | FCER1G
FYN | F2R
FYN | IGF1R
VEGFB | IGF1R
ANGPT1 | IGF1R
VEGFA | IGF1R
FYN | KIT
VEGFB | KIT
ANGPT1 | KIT
VEGFA | KIT
FYN | SDC3
SPN | SDC3
FYN | ITGA6
VWF | ITGA6
DUSP18 | ITGA6
FYN | ITGB2
HP | ITGB2
ICAM2 | ITGB2
ICAM4 | ITGB2
SPON2 | ITGB2
FYN | ITGB5
TGFB1 | ITGB5
VWF | ITGB5
FYN | ITGB7
VWF | ITGB7
KIT | FLT3
VEGFB | FLT3
VEGFA | FLT3
VEGFB | RYK
VEGFA | RYK
TGFB1 | VEGFA
TNFRSF1A | TNFRSF1B
TGFBR3 | TGFB1
DPP4 | TGFB1
HP | ITGAM
ICAM2 | ITGAM
ICAM4 | ITGAM
SPON2 | ITGAM
TNFRSF1B | TNFRSF1A
GRN | TNFRSF1A
VEGFA | GPC1
VEGFA | SIRPA
VWF | SIRPA
TGFB1 | ITGAV
VWF | ITGAV
ICAM4 | ITGAV
VEGFA | SPHK1
TGFB1 | TGFBR2
TGFB1 | TGFBR3
TGFB1 | KLRK1
APP | TNFRSF21
VWF | GP1BA
NLGN2 | NRXN1
VWF | ITGA2B
ICAM4 | ITGA2B
VWF | GP9
IL12A | IL6ST
DPP4 | ADA
ICAM4 | RHAG
10GMP
11DC
12Baso
13Baso
14Mo
15Mo
16Neu
17Neu
18Eos
19Lymph
1Ery
2Ery
3Ery
4Ery
5Ery
6Ery
7MEP
8Mk
9GMP
$-\log_{10} p$
1.6 2.7 3.3 3.7 4.0
$log_2(\frac{molecule_1 + molecule_2}{2} + 1)$
0.00 0.46 0.91 1.37 1.82

3 Load the human data

[9]:
```
adata = sc.datasets.pbmc3k_processed()
adata
```

[9]:
```
AnnData object with n_obs × n_vars = 2638 × 1838
    obs: 'n_genes', 'percent_mito', 'n_counts', 'louvain'
    var: 'n_cells'
    uns: 'draw_graph', 'louvain', 'louvain_colors', 'neighbors', 'pca',
'rank_genes_groups'
    obsm: 'X_pca', 'X_tsne', 'X_umap', 'X_draw_graph_fr'
    varm: 'PCs'
    obsp: 'distances', 'connectivities'
```

3.1 Run the CellPhoneDB's permutation test

3.1.1 Use only CellPhoneDB as a resource

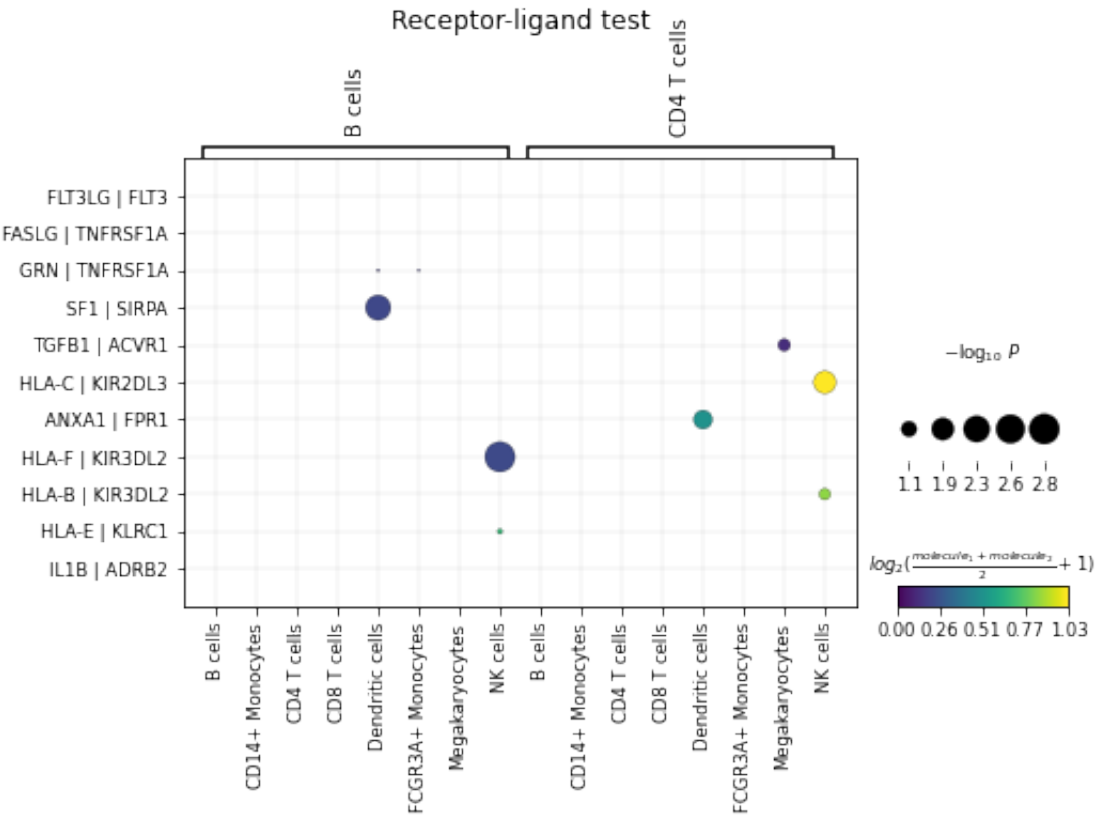
[10]:
```
res = sp.gr.ligrec(adata, "louvain",
                   fdr_method=None, copy=True,
                   interactions_params={"resources": "CellPhoneDB"},
                   threshold=0.1, seed=0, n_perms=10000, n_jobs=1)
df = res.pvalues
print("Number of CellPhoneDB interactions (human data):", len(df))

df.head()
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=10000.0),␣
↪HTML(value='')))


Number of CellPhoneDB interactions (human data): 76
```

[10]:
```
cluster_1     B cells                                                    \
cluster_2     B cells CD14+ Monocytes CD4 T cells CD8 T cells Dendritic cells
source target
DLL1   NOTCH1     NaN             NaN         NaN         NaN             NaN
TNF    NOTCH1     NaN             NaN         NaN         NaN             NaN
DLL3   NOTCH1     NaN             NaN         NaN         NaN             NaN
DLL1   NOTCH2     NaN             NaN         NaN         NaN             NaN
DLL3   NOTCH2     NaN             NaN         NaN         NaN             NaN

cluster_1                                              CD14+ Monocytes  \
cluster_2     FCGR3A+ Monocytes Megakaryocytes NK cells         B cells
source target
```

```
DLL1   NOTCH1               NaN            NaN      NaN             NaN
TNF    NOTCH1               NaN            NaN      NaN             NaN
DLL3   NOTCH1               NaN            NaN      NaN             NaN
DLL1   NOTCH2               NaN            NaN      NaN             NaN
DLL3   NOTCH2               NaN            NaN      NaN             NaN

cluster_1                      … Megakaryocytes          NK cells  \
cluster_2     CD14+ Monocytes  … Megakaryocytes NK cells  B cells
source target                  …
DLL1   NOTCH1             NaN  …             NaN      NaN      NaN
TNF    NOTCH1             NaN  …             NaN      NaN      NaN
DLL3   NOTCH1             NaN  …             NaN      NaN      NaN
DLL1   NOTCH2             NaN  …             NaN      NaN      NaN
DLL3   NOTCH2             NaN  …             NaN      NaN      NaN

cluster_1                                                          \
cluster_2     CD14+ Monocytes CD4 T cells CD8 T cells Dendritic cells
source target
DLL1   NOTCH1             NaN         NaN         NaN             NaN
TNF    NOTCH1             NaN         NaN         NaN             NaN
DLL3   NOTCH1             NaN         NaN         NaN             NaN
DLL1   NOTCH2             NaN         NaN         NaN             NaN
DLL3   NOTCH2             NaN         NaN         NaN             NaN

cluster_1
cluster_2     FCGR3A+ Monocytes Megakaryocytes NK cells
source target
DLL1   NOTCH1               NaN            NaN      NaN
TNF    NOTCH1               NaN            NaN      NaN
DLL3   NOTCH1               NaN            NaN      NaN
DLL1   NOTCH2               NaN            NaN      NaN
DLL3   NOTCH2               NaN            NaN      NaN

[5 rows x 64 columns]
```

[11]:
```
sp.pl.ligrec(res, source_groups=["CD4 T cells", "B cells"])
```



3.1.2 Use all available resources from OmniPath

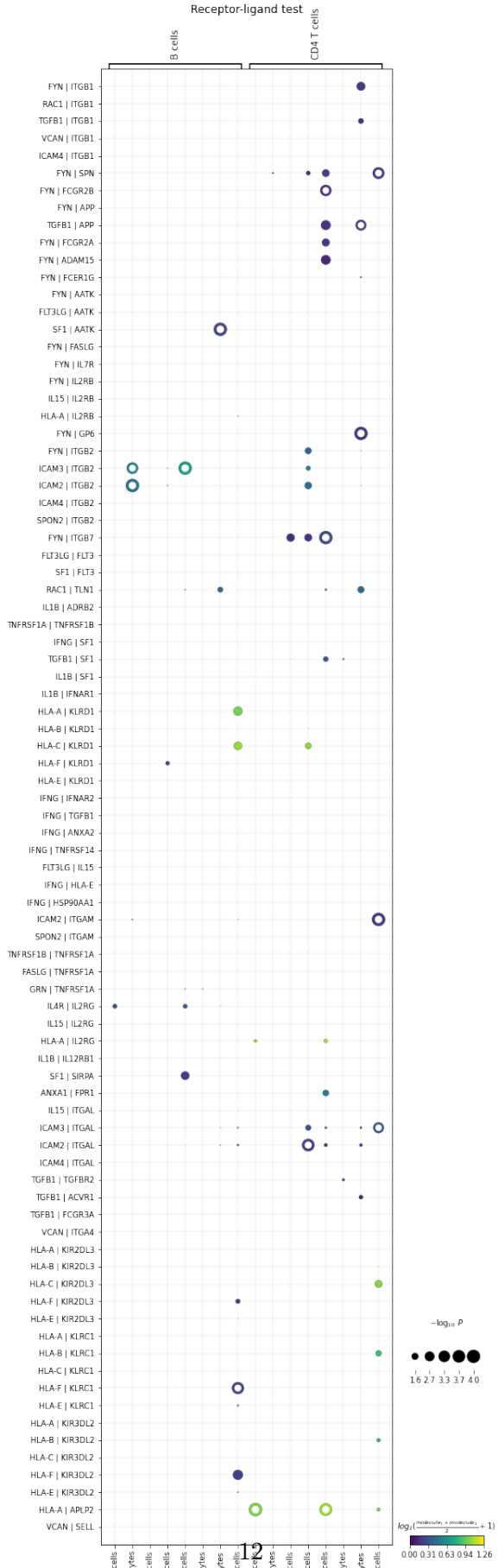
[12]:
```
res = sp.gr.ligrec(adata, "louvain",
                   fdr_method=None, copy=True,
                   threshold=0.1, seed=0, n_perms=10000, n_jobs=1)
df = res.pvalues
print("Number of OmniPath interactions (human data):", len(df))

df.head()
```

```
HBox(children=(HTML(value=''), FloatProgress(value=0.0, max=10000.0),␣
 ↪HTML(value='')))


Number of OmniPath interactions (human data): 510
```

[12]:
```
cluster_1     B cells                                                    \
cluster_2     B cells CD14+ Monocytes CD4 T cells CD8 T cells Dendritic cells
source target
FYN    ITGB1      NaN             NaN         NaN         NaN             NaN
RAC1   ITGB1      NaN             NaN         NaN         NaN             NaN
HGF    ITGB1      NaN             NaN         NaN         NaN             NaN
```

```
TGFB1  ITGB1      NaN             NaN         NaN         NaN             NaN
TGFB3  ITGB1      NaN             NaN         NaN         NaN             NaN

cluster_1                                              CD14+ Monocytes  \
cluster_2     FCGR3A+ Monocytes Megakaryocytes NK cells         B cells
source target
FYN    ITGB1                NaN            NaN      NaN             NaN
RAC1   ITGB1                NaN         0.9954      NaN             NaN
HGF    ITGB1                NaN            NaN      NaN             NaN
TGFB1  ITGB1                NaN            NaN      NaN             NaN
TGFB3  ITGB1                NaN            NaN      NaN             NaN

cluster_1                      … Megakaryocytes          NK cells  \
cluster_2     CD14+ Monocytes  … Megakaryocytes NK cells  B cells
source target                  …
FYN    ITGB1              NaN  …            NaN      NaN      NaN
RAC1   ITGB1              NaN  …         0.1756      NaN      NaN
HGF    ITGB1              NaN  …            NaN      NaN      NaN
TGFB1  ITGB1              NaN  …            NaN      NaN      NaN
TGFB3  ITGB1              NaN  …            NaN      NaN      NaN

cluster_1                                                          \
cluster_2     CD14+ Monocytes CD4 T cells CD8 T cells Dendritic cells
source target
FYN    ITGB1              NaN         NaN         NaN             NaN
RAC1   ITGB1              NaN         NaN         NaN             NaN
HGF    ITGB1              NaN         NaN         NaN             NaN
TGFB1  ITGB1              NaN         NaN         NaN             NaN
TGFB3  ITGB1              NaN         NaN         NaN             NaN

cluster_1
cluster_2     FCGR3A+ Monocytes Megakaryocytes NK cells
source target
FYN    ITGB1                NaN         0.0018      NaN
RAC1   ITGB1                NaN         0.0912      NaN
HGF    ITGB1                NaN            NaN      NaN
TGFB1  ITGB1                NaN         0.0007      NaN
TGFB3  ITGB1                NaN            NaN      NaN

[5 rows x 64 columns]
```

[13]:
```
sp.pl.ligrec(res, source_groups=["CD4 T cells", "B cells"])
```


Receptor-ligand test
B cells
CD4 T cells
FYN | ITGB1
RAC1 | ITGB1
TGFB1 | ITGB1
VCAN | ITGB1
ICAM4 | ITGB1
FYN | SPN
FYN | FCGR2B
FYN | APP
TGFB1 | APP
FYN | FCGR2A
FYN | ADAM15
FYN | FCER1G
FYN | AATK
FLT3LG | AATK
SF1 | AATK
FYN | FASLG
FYN | IL7R
FYN | IL2RB
IL15 | IL2RB
HLA-A | IL2RB
FYN | GP6
FYN | ITGB2
ICAM3 | ITGB2
ICAM2 | ITGB2
ICAM4 | ITGB2
SPON2 | ITGB2
FYN | ITGB7
FLT3LG | FLT3
SF1 | FLT3
RAC1 | TLN1
IL1B | ADRB2
TNFRSF1A | TNFRSF1B
IFNG | SF1
TGFB1 | SF1
IL1B | SF1
IL1B | IFNAR1
HLA-A | KLRD1
HLA-B | KLRD1
HLA-C | KLRD1
HLA-F | KLRD1
HLA-E | KLRD1
IFNG | IFNAR2
IFNG | TGFB1
IFNG | ANXA2
IFNG | TNFRSF14
FLT3LG | IL15
IFNG | HLA-E
IFNG | HSP90AA1
ICAM2 | ITGAM
SPON2 | ITGAM
TNFRSF1B | TNFRSF1A
FASLG | TNFRSF1A
GRN | TNFRSF1A
IL4R | IL2RG
IL15 | IL2RG
HLA-A | IL2RG
IL1B | IL12RB1
SF1 | SIRPA
ANXA1 | FPR1
IL15 | ITGAL
ICAM3 | ITGAL
ICAM2 | ITGAL
ICAM4 | ITGAL
TGFB1 | TGFBR2
TGFB1 | ACVR1
TGFB1 | FCGR3A
VCAN | ITGA4
HLA-A | KIR2DL3
HLA-B | KIR2DL3
HLA-C | KIR2DL3
HLA-F | KIR2DL3
HLA-E | KIR2DL3
HLA-A | KLRC1
HLA-B | KLRC1
HLA-C | KLRC1
HLA-F | KLRC1
HLA-E | KLRC1
HLA-A | KIR3DL2
HLA-B | KIR3DL2
HLA-C | KIR3DL2
HLA-F | KIR3DL2
HLA-E | KIR3DL2
HLA-A | APLP2
VCAN | SELL
B cells
CD14+ Monocytes
CD4 T cells
CD8 T cells
Dendritic cells
FCGR3A+ Monocytes
Megakaryocytes
NK cells
B cells
CD14+ Monocytes
CD4 T cells
CD8 T cells
Dendritic cells
FCGR3A+ Monocytes
Megakaryocytes
NK cells
$-\log_{10} P$
1.6 2.7 3.3 3.7 4.0
$log_2(\frac{molecule_1 + molecule_2}{2} + 1)$
0.00 0.31 0.63 0.94 1.26

4 Conluding remarks

Using OmniPath as an interation source yields approx. ~7x more interactions than from CellPhoneDB for the selected human data and ~11x interactions for the seleted mouse data (internally in `squidpy.gr.ligrec`, we map the mouse gene symbols to human simply by uppercasing).

In the context of spatial tools, the goal is to use the permutation test from CellPhoneDB to analyze receptor-ligand interaction pairs in clusters that are spatially close.