Alteration of intercellular signaling in ulcerative colitis

December 17, 2020

Analysising single cell RNA-seq data for inter and intracellular interactions. The first step is to import the singel cell RNA-seq data file. The import file is the cell cluster averaged gene expression files. OmniPath is accesed by the python client.

[1]:
```
import pandas as pd
import igraph as ig
import os
from itertools import permutations
import numpy as np
```

[2]:
```
import omnipath as op
```

```
C:\Users\modos\.conda\envs\Single_cell_RNA_seq\lib\site-
packages\requests\__init__.py:91: RequestsDependencyWarning: urllib3 (1.26.2) or
chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
```

First lets check the OmniPAth version and data.

[3]:
```
print(op.__server_version__)
print(op.options)
```

```
0.11.34
Options(url='https://omnipathdb.org', license=<License.ACADEMIC>,
cache=<FileCache[size=3, path='C:\\Users\\modos\\.cache\\omnipathdb']>,
autoload=True, convert_dtypes=True, num_retries=3, timeout=5.0, chunk_size=8196)
```

Now we can donwload OmniPath intercellular network and see what kind of columns are in the intercellular dataframe.

[4]:
```
intercell_network = op.interactions.import_intercell_network()
```

[5]:
```
intercell_network.head()
```

[5]:

	index	source	target	is_stimulation	is_inhibition	consensus_direction	\
0	0	P14416	P48995	True	False	True	
1	2	P14416	P48995	True	False	True	
2	5	P14416	P48995	True	False	True	
3	7	Q13255	P48995	True	False	True	
4	9	Q13255	P48995	True	False	True	

```
   consensus_stimulation  consensus_inhibition dip_url  curation_effort  …  \
0                   True                 False     nan                1  …
1                   True                 False     nan                1  …
2                   True                 False     nan                1  …
3                   True                 False     nan                1  …
4                   True                 False     nan                1  …

  category_source_intercell_target uniprot_intercell_target  \
0                resource_specific                   P48995
1                resource_specific                   P48995
2                        composite                   P48995
3                resource_specific                   P48995
4                resource_specific                   P48995

  genesymbol_intercell_target entity_type_intercell_target  \
0                       TRPC1                      protein
1                       TRPC1                      protein
2                       TRPC1                      protein
3                       TRPC1                      protein
4                       TRPC1                      protein

   consensus_score_intercell_target  transmitter_intercell_target  \
0                                 1                         False
1                                 3                         False
2                                 3                         False
3                                 1                         False
4                                 3                         False

  receiver_intercell_target secreted_intercell_target  \
0                      True                     False
1                      True                     False
2                      True                     False
3                      True                     False
4                      True                     False

  plasma_membrane_transmembrane_intercell_target  \
0                                          False
1                                          False
2                                          False
3                                          False
4                                          False

  plasma_membrane_peripheral_intercell_target
0                                       False
1                                       False
2                                       False
```

```
3                                        False
4                                        False

[5 rows x 46 columns]
```

[6]:
```
intercell_network.columns
```

[6]:
```
Index(['index', 'source', 'target', 'is_stimulation', 'is_inhibition',
       'consensus_direction', 'consensus_stimulation', 'consensus_inhibition',
       'dip_url', 'curation_effort', 'references', 'sources',
       'references_stripped', 'n_references', 'n_sources', 'n_primary_sources',
       'category_intercell_source', 'parent_intercell_source',
       'database_intercell_source', 'scope_intercell_source',
       'aspect_intercell_source', 'category_source_intercell_source',
       'uniprot_intercell_source', 'genesymbol_intercell_source',
       'entity_type_intercell_source', 'consensus_score_intercell_source',
       'transmitter_intercell_source', 'receiver_intercell_source',
       'secreted_intercell_source',
       'plasma_membrane_transmembrane_intercell_source',
       'plasma_membrane_peripheral_intercell_source',
       'category_intercell_target', 'parent_intercell_target',
       'database_intercell_target', 'scope_intercell_target',
       'aspect_intercell_target', 'category_source_intercell_target',
       'uniprot_intercell_target', 'genesymbol_intercell_target',
       'entity_type_intercell_target', 'consensus_score_intercell_target',
       'transmitter_intercell_target', 'receiver_intercell_target',
       'secreted_intercell_target',
       'plasma_membrane_transmembrane_intercell_target',
       'plasma_membrane_peripheral_intercell_target'],
      dtype='object')
```

[7]:
```
intercell_network.shape
```

[7]:
```
(30265, 46)
```

Here we can check what are the categories what we can use for intercellular interactions. We can use the intercellular interctions directly, hovewer it contain porteins which are invovled in the adhesion process and binds to the intracelluar domains of receptor proteins. That is why we suggest the fitlering as in the paper.

[8]:
```
print (set(intercell_network["category_intercell_source"]))
```

```
{'desmosome', 'cell_surface_ligand', 'cell_surface_enzyme', 'secreted_receptor',
'ecm', 'receptor_regulator', 'adhesion', 'tight_junction', 'cell_adhesion',
'matrix_adhesion_regulator', 'cell_surface_peptidase', 'ecm_regulator',
'secreted_enzyme', 'ligand_regulator', 'gap_junction', 'ligand'}
```

[9]:
```
print (set(intercell_network["category_intercell_target"]))
```

```
{'adherens_junction', 'receptor', 'ion_channel_regulator', 'desmosome',
'adhesion', 'tight_junction', 'cell_adhesion', 'matrix_adhesion', 'transporter',
'ion_channel', 'gap_junction'}
```

[10]:
```
filtered_source_types =␣
 ↪["cell_surface_enzyme","cell_surface_ligand","ligand","secreted_enzyme","secreted_receptor"
          ␣
 ↪"adhesion","cell_adhesion","tight_junction","cell_surface_peptidase","gap_junction","desmos
filtered_target_types =␣
 ↪["adhesion","receptor","cell_adhesion","tight_junction","gap_junction","ion_channel","trans
               "adherens_junction"]
```

[11]:
```
filtered_intercell_network = intercell_network[intercell_network.
 ↪category_intercell_source.isin(filtered_source_types)]
filtered_intercell_network =␣
 ↪filtered_intercell_network[filtered_intercell_network.
                                                          ␣
 ↪category_intercell_target.isin(filtered_target_types)]
filtered_intercell_network.shape
```

[11]:
```
(20784, 46)
```

The next step is to preprocess the data files. The imput file is the cell type specfic expression. The data are from Smillie et al 2019 (https://pubmed.ncbi.nlm.nih.gov/31348891/). Each column is the average per cell type in transcript per million. From that we will build a simple cell-cell interaction count. It is based on expression tresholding. Our assumptation was that the interactions are important if they are exisit in any way, so we have chosen a realtievely low expression treshold. We invite the user for testing various tresholds in their own work. You will need for this the expression data which are in this example's folder.

[12]:
```
df_cell_imm = pd.read_csv("imm_all_expression_condition.tsv", sep="\t",␣
 ↪index_col=0, header=0)
df_cell_fib = pd.read_csv("fib_all_expression_condition.tsv",sep="\t",␣
 ↪index_col=0, header=0)
df_cell_epi =  pd.read_csv("epi_all_expression_condition.tsv",sep="\t",␣
 ↪index_col=0, header=0)
df_cell_exp = df_cell_imm.join(df_cell_fib, how="outer") #Keeping all genes
df_cell_exp = df_cell_exp.join(df_cell_epi, how="outer")
```

[13]:
```
df_cell_exp.head()
```

[13]:

	RNA.CD8..IELs_Healthy	RNA.CD8..LP_Healthy	RNA.CD4..Memory_Healthy	\
7SK	0.030182	0.042158	0.016792	
A1BG	0.062424	0.025848	0.049498	
A1BG-AS1	0.276995	0.198592	0.138406	
A1CF	0.002978	0.005669	0.000000	
A2M	0.023545	0.214442	0.202609	

```
          RNA.MT.hi_Healthy  RNA.Cycling.T_Healthy  RNA.NKs_Healthy  \
7SK                0.000000               0.000000         0.228991
A1BG               0.208431               0.321939         0.000000
A1BG-AS1           0.295042               0.000000         0.498747
A1CF               0.000000               0.025840         0.000000
A2M                0.167921               0.097767         0.545677

          RNA.CD4..Activated.Fos.lo_Healthy  \
7SK                                0.000000
A1BG                               0.045158
A1BG-AS1                           0.130186
A1CF                               0.000000
A2M                                0.144894

          RNA.CD4..Activated.Fos.hi_Healthy  RNA.CD8..IELs_Uninflamed  \
7SK                                0.015092                  0.030573
A1BG                               0.018531                  0.053037
A1BG-AS1                           0.256156                  0.305482
A1CF                               0.000000                  0.079150
A2M                                0.166956                  0.010304

          RNA.MT.hi_Uninflamed  …  RNA.Immature.Goblet_Inflamed  \
7SK                   0.000000  …                       0.00000
A1BG                  0.000000  …                       0.00000
A1BG-AS1              0.000000  …                           NaN
A1CF                  0.000000  …                       0.66123
A2M                   0.243075  …                       0.00000

          RNA.Stem_Uninflamed  RNA.Immature.Enterocytes.2_Inflamed  \
7SK                  0.001679                             0.000000
A1BG                 0.013224                             0.002473
A1BG-AS1                  NaN                                  NaN
A1CF                 0.211192                             0.848589
A2M                  0.000000                             0.000000

          RNA.Goblet_Inflamed  RNA.Tuft_Inflamed  RNA.Enterocytes_Inflamed  \
7SK                  0.000000           7.005192                  0.002465
A1BG                 0.089265           0.000000                  0.000000
A1BG-AS1                  NaN                NaN                       NaN
A1CF                 0.187466           0.000000                  0.486224
A2M                  0.000000           0.000000                  0.000000

          RNA.Best4..Enterocytes_Inflamed  RNA.Enteroendocrine_Inflamed  \
7SK                              0.007913                      0.000000
A1BG                             0.000000                      0.000000
A1BG-AS1                              NaN                           NaN
A1CF                             0.778587                      2.653654
```

```
A2M                                0.000000                       0.000000

          RNA.M.cells_Inflamed  RNA.M.cells_Uninflamed
7SK                   0.000000                0.000000
A1BG                  0.000000                0.000000
A1BG-AS1                   NaN                     NaN
A1CF                  0.457293                0.737544
A2M                   0.000000                0.000000

[5 rows x 153 columns]
```

We will choose the mean -2SD of the whole data set. First we do a log2 based transformation.

[14]:
```
df_cell_exp[df_cell_exp.values == 0] = "NaN"
```

[15]:
```
df_cell_exp.head()
```

[15]:
```
         RNA.CD8..IELs_Healthy RNA.CD8..LP_Healthy RNA.CD4..Memory_Healthy  \
7SK                  0.0301819           0.0421578                0.0167924
A1BG                  0.062424           0.0258481                0.0494975
A1BG-AS1              0.276995            0.198592                 0.138406
A1CF                0.00297762          0.00566872                      NaN
A2M                   0.0235452            0.214442                 0.202609

         RNA.MT.hi_Healthy RNA.Cycling.T_Healthy RNA.NKs_Healthy  \
7SK                    NaN                   NaN        0.228991
A1BG              0.208431              0.321939             NaN
A1BG-AS1          0.295042                   NaN        0.498747
A1CF                   NaN             0.0258403             NaN
A2M               0.167921             0.0977667        0.545677

         RNA.CD4..Activated.Fos.lo_Healthy RNA.CD4..Activated.Fos.hi_Healthy  \
7SK                                    NaN                          0.0150918
A1BG                             0.0451578                          0.0185314
A1BG-AS1                          0.130186                           0.256156
A1CF                                   NaN                                NaN
A2M                               0.144894                          0.166956

         RNA.CD8..IELs_Uninflamed RNA.MT.hi_Uninflamed  …  \
7SK                     0.0305725                  NaN  …
A1BG                    0.0530365                  NaN  …
A1BG-AS1                 0.305482                  NaN  …
A1CF                    0.0791505                  NaN  …
A2M                     0.0103037             0.243075  …

         RNA.Immature.Goblet_Inflamed RNA.Stem_Uninflamed  \
7SK                               NaN          0.00167886
A1BG                              NaN           0.0132239
```

```
A1BG-AS1                          NaN                NaN
A1CF                          0.66123           0.211192
A2M                               NaN                NaN

         RNA.Immature.Enterocytes.2_Inflamed RNA.Goblet_Inflamed  \
7SK                                      NaN                NaN
A1BG                               0.0024732           0.089265
A1BG-AS1                                 NaN                NaN
A1CF                                0.848589           0.187466
A2M                                      NaN                NaN

         RNA.Tuft_Inflamed RNA.Enterocytes_Inflamed  \
7SK                7.00519               0.00246488
A1BG                   NaN                      NaN
A1BG-AS1               NaN                      NaN
A1CF                   NaN                 0.486224
A2M                    NaN                      NaN

         RNA.Best4..Enterocytes_Inflamed RNA.Enteroendocrine_Inflamed  \
7SK                           0.00791332                          NaN
A1BG                                 NaN                          NaN
A1BG-AS1                             NaN                          NaN
A1CF                            0.778587                      2.65365
A2M                                  NaN                          NaN

         RNA.M.cells_Inflamed RNA.M.cells_Uninflamed
7SK                       NaN                   NaN
A1BG                      NaN                   NaN
A1BG-AS1                  NaN                   NaN
A1CF                 0.457293              0.737544
A2M                       NaN                   NaN

[5 rows x 153 columns]
```

[16]:
```
data_matrix = df_cell_exp.to_numpy()
data_matrix_log2 = np.log2(data_matrix.astype(float)) #We need to change the␣
 ↪data file to floats
mean_cell = np.nanmean(data_matrix_log2)
std = np.nanstd(data_matrix_log2)
```

[17]:
```
mean_cell, std
```

[17]:
```
(-2.160708171397226, 3.00937762482671)
```

[18]:
```
import seaborn as sns
```

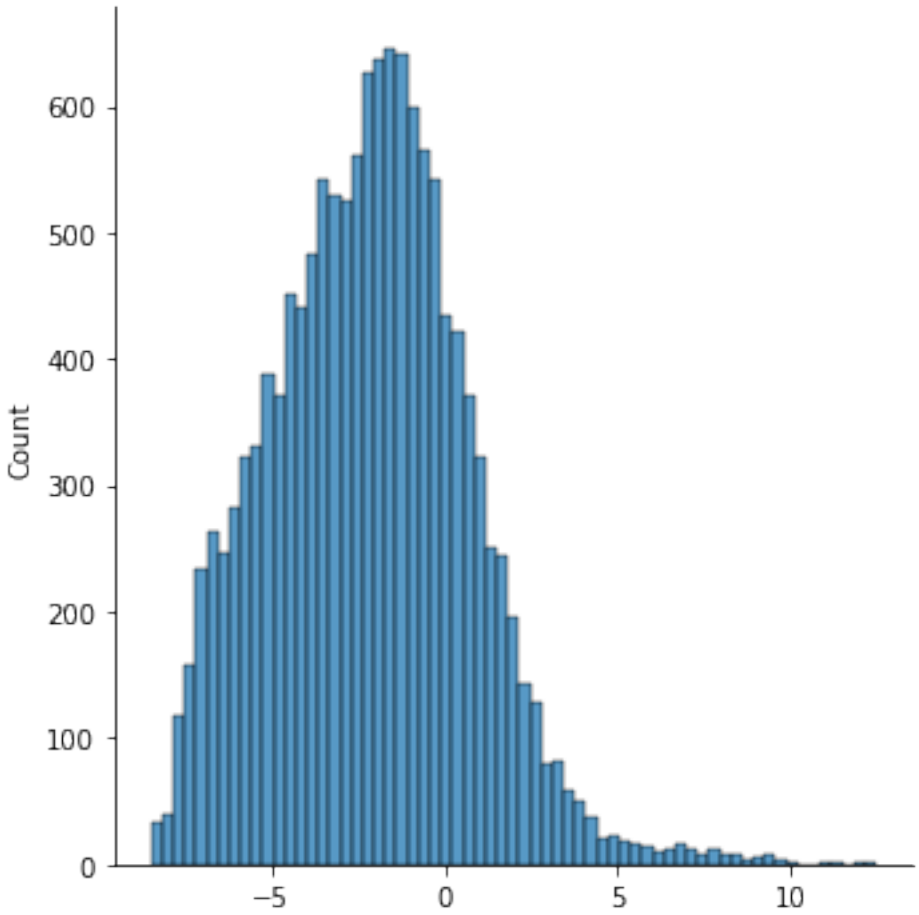
We can check the distribuiton of the cell types. Here only for one.

[19]:
```
data_matrix_log2
```

[19]:
```
array([[-5.0501723 , -4.56805614, -5.89604351, …,         nan,
                nan,         nan],
       [-4.00175632, -5.27379813, -4.33649916, …,         nan,
                nan,         nan],
       [-1.85206773, -2.33211901, -2.8530207 , …,         nan,
                nan,         nan],
       …,
       [-3.47290816, -4.19215007, -4.40580567, …, -2.06536459,
         0.35303725,  0.44592408],
       [-4.38586399, -2.27399124, -3.85313444, …, -4.58601467,
        -8.31527547,         nan],
       [-2.5735855 , -1.76640068, -1.1066736 , …, -4.14328163,
        -7.335609  ,         nan]])
```

[20]:
```
sns.displot(data_matrix_log2[:,0])
```

[20]:
```
<seaborn.axisgrid.FacetGrid at 0x19ee52d6c88>
```



After checking the histogram we can say that we will use the mean minus 2 standard deviation of the expressed genes.

[21]:
```
data_matrix_log2 > (mean_cell-2*std)
```

```
C:\Users\modos\.conda\envs\Single_cell_RNA_seq\lib\site-
packages\ipykernel_launcher.py:1: RuntimeWarning: invalid value encountered in
greater
  """Entry point for launching an IPython kernel.
```

[21]:
```
array([[ True,  True,  True, …, False, False, False],
       [ True,  True,  True, …, False, False, False],
       [ True,  True,  True, …, False, False, False],
       …,
       [ True,  True,  True, …,  True,  True,  True],
       [ True,  True,  True, …,  True, False, False],
       [ True,  True,  True, …,  True,  True, False]])
```

For simplicity we call "NaN" each number which is below the treshold.

[22]:
```
df_cell_log2 = pd.DataFrame(data=data_matrix_log2, index=df_cell_exp.index␣
↪,columns=df_cell_exp.columns)
```

[23]:
```
df_cell_log2.head()
```

[23]:

	RNA.CD8..IELs_Healthy	RNA.CD8..LP_Healthy	RNA.CD4..Memory_Healthy	\
7SK	-5.050172	-4.568056	-5.896044	
A1BG	-4.001756	-5.273798	-4.336499	
A1BG-AS1	-1.852068	-2.332119	-2.853021	
A1CF	-8.391626	-7.462761	NaN	
A2M	-5.408426	-2.221338	-2.303227	

	RNA.MT.hi_Healthy	RNA.Cycling.T_Healthy	RNA.NKs_Healthy	\
7SK	NaN	NaN	-2.126639	
A1BG	-2.262358	-1.635142	NaN	
A1BG-AS1	-1.761008	NaN	-1.003621	
A1CF	NaN	-5.274231	NaN	
A2M	-2.574148	-3.354513	-0.873881	

	RNA.CD4..Activated.Fos.lo_Healthy	\
7SK	NaN	
A1BG	-4.468881	
A1BG-AS1	-2.941349	
A1CF	NaN	
A2M	-2.786930	

	RNA.CD4..Activated.Fos.hi_Healthy	RNA.CD8..IELs_Uninflamed	\

```
7SK                              -6.050093                 -5.031619
A1BG                             -5.753884                 -4.236870
A1BG-AS1                         -1.964903                 -1.710843
A1CF                                   NaN                 -3.659258
A2M                              -2.582460                 -6.600697

          RNA.MT.hi_Uninflamed  …  RNA.Immature.Goblet_Inflamed  \
7SK                        NaN  …                           NaN
A1BG                       NaN  …                           NaN
A1BG-AS1                   NaN  …                           NaN
A1CF                       NaN  …                     -0.596775
A2M                 -2.040528   …                           NaN

          RNA.Stem_Uninflamed  RNA.Immature.Enterocytes.2_Inflamed  \
7SK                 -9.218304                                 NaN
A1BG                -6.240706                           -8.659406
A1BG-AS1                  NaN                                 NaN
A1CF                -2.243375                           -0.236862
A2M                       NaN                                 NaN

          RNA.Goblet_Inflamed  RNA.Tuft_Inflamed  RNA.Enterocytes_Inflamed  \
7SK                       NaN           2.808425                 -8.664267
A1BG                -3.485761                NaN                       NaN
A1BG-AS1                  NaN                NaN                       NaN
A1CF                -2.415296                NaN                 -1.040307
A2M                       NaN                NaN                       NaN

          RNA.Best4..Enterocytes_Inflamed  RNA.Enteroendocrine_Inflamed  \
7SK                             -6.981502                           NaN
A1BG                                  NaN                           NaN
A1BG-AS1                              NaN                           NaN
A1CF                            -0.361069                       1.40798
A2M                                   NaN                           NaN

          RNA.M.cells_Inflamed  RNA.M.cells_Uninflamed
7SK                        NaN                     NaN
A1BG                       NaN                     NaN
A1BG-AS1                   NaN                     NaN
A1CF                -1.12881                  -0.4392
A2M                        NaN                     NaN

[5 rows x 153 columns]
```

We will call every genee which is expressed below the treshold with “NaN”.

[24]:
```
df_cell_log2[df_cell_log2.values < (mean_cell-2*std)] = "NaN"
```

```
C:\Users\modos\.conda\envs\Single_cell_RNA_seq\lib\site-
```

```
packages\ipykernel_launcher.py:1: RuntimeWarning: invalid value encountered in
less
  """Entry point for launching an IPython kernel.
```

Let's see what we hjave done and the avaialbe cell types.

[25]:
```
df_cell_log2.head()
```

[25]:

	RNA.CD8..IELs_Healthy	RNA.CD8..LP_Healthy	RNA.CD4..Memory_Healthy	\
7SK	-5.05017	-4.56806	-5.896044	
A1BG	-4.00176	-5.2738	-4.336499	
A1BG-AS1	-1.85207	-2.33212	-2.853021	
A1CF	NaN	-7.46276	NaN	
A2M	-5.40843	-2.22134	-2.303227	

	RNA.MT.hi_Healthy	RNA.Cycling.T_Healthy	RNA.NKs_Healthy	\
7SK	NaN	NaN	-2.126639	
A1BG	-2.262358	-1.635142	NaN	
A1BG-AS1	-1.761008	NaN	-1.003621	
A1CF	NaN	-5.274231	NaN	
A2M	-2.574148	-3.354513	-0.873881	

	RNA.CD4..Activated.Fos.lo_Healthy	RNA.CD4..Activated.Fos.hi_Healthy	\
7SK	NaN	-6.05009	
A1BG	-4.46888	-5.75388	
A1BG-AS1	-2.94135	-1.9649	
A1CF	NaN	NaN	
A2M	-2.78693	-2.58246	

	RNA.CD8..IELs_Uninflamed	RNA.MT.hi_Uninflamed	…	\
7SK	-5.031619	NaN	…	
A1BG	-4.236870	NaN	…	
A1BG-AS1	-1.710843	NaN	…	
A1CF	-3.659258	NaN	…	
A2M	-6.600697	-2.040528	…	

	RNA.Immature.Goblet_Inflamed	RNA.Stem_Uninflamed	\
7SK	NaN	NaN	
A1BG	NaN	-6.24071	
A1BG-AS1	NaN	NaN	
A1CF	-0.596775	-2.24338	
A2M	NaN	NaN	

	RNA.Immature.Enterocytes.2_Inflamed	RNA.Goblet_Inflamed	\
7SK	NaN	NaN	
A1BG	NaN	-3.48576	
A1BG-AS1	NaN	NaN	
A1CF	-0.236862	-2.4153	

```
A2M                                      NaN                  NaN

          RNA.Tuft_Inflamed  RNA.Enterocytes_Inflamed  \
7SK                2.808425                       NaN
A1BG                    NaN                       NaN
A1BG-AS1                NaN                       NaN
A1CF                    NaN                  -1.04031
A2M                     NaN                       NaN

          RNA.Best4..Enterocytes_Inflamed  RNA.Enteroendocrine_Inflamed  \
7SK                               -6.9815                           NaN
A1BG                                  NaN                           NaN
A1BG-AS1                              NaN                           NaN
A1CF                            -0.361069                       1.40798
A2M                                   NaN                           NaN

          RNA.M.cells_Inflamed  RNA.M.cells_Uninflamed
7SK                        NaN                    NaN
A1BG                       NaN                    NaN
A1BG-AS1                   NaN                    NaN
A1CF                 -1.12881                 -0.4392
A2M                        NaN                    NaN

[5 rows x 153 columns]
```

[26]:
```
print(set(df_cell_log2.columns))
```

```
{'RNA.Macrophages_Uninflamed', 'RNA.CD4..PD1._Inflamed', 'RNA.ILCs_Inflamed',
'RNA.Cycling.TA_Healthy', 'RNA.TA.1_Inflamed', 'RNA.WNT2B..Fos.lo.1_Uninflamed',
'RNA.Tregs_Healthy', 'RNA.Best4..Enterocytes_Inflamed',
'RNA.Enterocyte.Progenitors_Inflamed', 'RNA.Microvascular_Inflamed',
'RNA.Goblet_Inflamed', 'RNA.CD4..Activated.Fos.hi_Healthy',
'RNA.CD8..IELs_Inflamed', 'RNA.WNT5B..2_Uninflamed',
'RNA.Inflammatory.Fibroblasts_Inflamed', 'RNA.CD8..IELs_Healthy',
'RNA.DC2_Inflamed', 'RNA.Tregs_Uninflamed',
'RNA.Immature.Enterocytes.1_Inflamed', 'RNA.WNT5B..2_Healthy',
'RNA.Follicular_Uninflamed', 'RNA.Inflammatory.Monocytes_Uninflamed',
'RNA.Microvascular_Uninflamed', 'RNA.Plasma_Uninflamed',
'RNA.CD8..IL17._Inflamed', 'RNA.CD8..IELs_Uninflamed', 'RNA.DC1_Inflamed',
'RNA.M.cells_Uninflamed', 'RNA.WNT5B..1_Inflamed',
'RNA.WNT2B..Fos.lo.2_Uninflamed', 'RNA.CD4..Activated.Fos.lo_Uninflamed',
'RNA.Cycling.Monocytes_Inflamed', 'RNA.NKs_Healthy',
'RNA.Secretory.TA_Inflamed', 'RNA.GC_Healthy', 'RNA.Cycling.B_Inflamed',
'RNA.Inflammatory.Monocytes_Healthy', 'RNA.MT.hi_Inflamed',
'RNA.Post.capillary.Venules_Inflamed', 'RNA.Cycling.T_Healthy',
'RNA.WNT2B..Fos.hi_Inflamed', 'RNA.RSPO3._Inflamed',
'RNA.Cycling.TA_Uninflamed', 'RNA.Enterocyte.Progenitors_Healthy',
'RNA.CD69..Mast_Inflamed', 'RNA.TA.2_Inflamed', 'RNA.TA.2_Healthy',
```

```
'RNA.DC1_Uninflamed', 'RNA.Endothelial_Inflamed', 'RNA.WNT5B..2_Inflamed',
'RNA.Pericytes_Healthy', 'RNA.Inflammatory.Monocytes_Inflamed',
'RNA.Inflammatory.Fibroblasts_Uninflamed', 'RNA.Microvascular_Healthy',
'RNA.M.cells_Inflamed', 'RNA.Glia_Healthy', 'RNA.MT.hi_Uninflamed',
'RNA.DC2_Healthy', 'RNA.Macrophages_Healthy', 'RNA.Macrophages_Inflamed',
'RNA.CD4..PD1._Healthy', 'RNA.Cycling.T_Uninflamed', 'RNA.CD69..Mast_Healthy',
'RNA.Enteroendocrine_Inflamed', 'RNA.Tregs_Inflamed',
'RNA.Secretory.TA_Healthy', 'RNA.CD4..Activated.Fos.hi_Uninflamed',
'RNA.Follicular_Inflamed', 'RNA.Plasma_Healthy',
'RNA.Immature.Enterocytes.1_Uninflamed', 'RNA.Glia_Inflamed',
'RNA.Best4..Enterocytes_Healthy', 'RNA.Best4..Enterocytes_Uninflamed',
'RNA.Stem_Inflamed', 'RNA.Tuft_Inflamed', 'RNA.Enterocytes_Uninflamed',
'RNA.Goblet_Uninflamed', 'RNA.CD4..Memory_Healthy', 'RNA.RSPO3._Uninflamed',
'RNA.Immature.Goblet_Inflamed', 'RNA.Enterocytes_Healthy',
'RNA.CD4..Activated.Fos.hi_Inflamed', 'RNA.MT.hi_Healthy',
'RNA.CD8..IL17._Healthy', 'RNA.WNT2B..Fos.lo.2_Inflamed', 'RNA.Goblet_Healthy',
'RNA.Cycling.B_Uninflamed', 'RNA.Cycling.Monocytes_Healthy',
'RNA.Pericytes_Inflamed', 'RNA.Immature.Enterocytes.2_Uninflamed',
'RNA.ILCs_Uninflamed', 'RNA.WNT2B..Fos.lo.1_Inflamed',
'RNA.Myofibroblasts_Healthy', 'RNA.CD4..Activated.Fos.lo_Healthy',
'RNA.Immature.Enterocytes.2_Healthy', 'RNA.WNT2B..Fos.lo.2_Healthy',
'RNA.M.cells_Healthy', 'RNA.CD8..IL17._Uninflamed',
'RNA.WNT2B..Fos.hi_Uninflamed', 'RNA.DC2_Uninflamed', 'RNA.TA.1_Uninflamed',
'RNA.Cycling.TA_Inflamed', 'RNA.Immature.Goblet_Healthy',
'RNA.Secretory.TA_Uninflamed', 'RNA.CD4..Memory_Uninflamed',
'RNA.Enteroendocrine_Uninflamed', 'RNA.Tuft_Uninflamed', 'RNA.Plasma_Inflamed',
'RNA.CD8..LP_Healthy', 'RNA.Stem_Uninflamed', 'RNA.Glia_Uninflamed',
'RNA.Follicular_Healthy', 'RNA.NKs_Uninflamed', 'RNA.DC1_Healthy',
'RNA.WNT5B..1_Uninflamed', 'RNA.CD4..Activated.Fos.lo_Inflamed',
'RNA.CD69..Mast_Healthy.1', 'RNA.NKs_Inflamed', 'RNA.Stem_Healthy',
'RNA.Inflammatory.Fibroblasts_Healthy', 'RNA.CD8..LP_Inflamed',
'RNA.CD69..Mast_Uninflamed.1', 'RNA.GC_Inflamed',
'RNA.Enterocyte.Progenitors_Uninflamed', 'RNA.GC_Uninflamed',
'RNA.Enterocytes_Inflamed', 'RNA.CD69..Mast_Inflamed.1',
'RNA.WNT2B..Fos.lo.1_Healthy', 'RNA.ILCs_Healthy',
'RNA.Post.capillary.Venules_Healthy', 'RNA.CD4..Memory_Inflamed',
'RNA.Myofibroblasts_Inflamed', 'RNA.WNT2B..Fos.hi_Healthy',
'RNA.WNT5B..1_Healthy', 'RNA.RSPO3._Healthy', 'RNA.Cycling.B_Healthy',
'RNA.CD8..LP_Uninflamed', 'RNA.Enteroendocrine_Healthy', 'RNA.TA.1_Healthy',
'RNA.Tuft_Healthy', 'RNA.Post.capillary.Venules_Uninflamed',
'RNA.Myofibroblasts_Uninflamed', 'RNA.Immature.Enterocytes.2_Inflamed',
'RNA.TA.2_Uninflamed', 'RNA.Immature.Goblet_Uninflamed',
'RNA.Endothelial_Uninflamed', 'RNA.Cycling.Monocytes_Uninflamed',
'RNA.Pericytes_Uninflamed', 'RNA.Immature.Enterocytes.1_Healthy',
'RNA.Cycling.T_Inflamed', 'RNA.CD69..Mast_Uninflamed',
'RNA.CD4..PD1._Uninflamed', 'RNA.Endothelial_Healthy'}
```

Next we select healthy/uninflamed average expression data in the 5 selected cell types.

[27]:
```
df_macrophage_healthy = df_cell_log2['RNA.Macrophages_Healthy']
df_macrophage_uninflamed = df_cell_log2['RNA.Macrophages_Uninflamed']

df_DC1_healthy = df_cell_log2['RNA.DC1_Healthy']
df_DC1_uninflamed = df_cell_log2['RNA.DC1_Uninflamed']

df_Treg_healthy = df_cell_log2['RNA.Tregs_Healthy']
df_Treg_uninflamed = df_cell_log2['RNA.Tregs_Uninflamed']

df_myofibroblast_healthy = df_cell_log2['RNA.Myofibroblasts_Healthy']
df_myofibroblast_uninflamed = df_cell_log2['RNA.Myofibroblasts_Uninflamed']

df_goblet_healthy = df_cell_log2['RNA.Goblet_Healthy']
df_goblet_uninflamed = df_cell_log2['RNA.Goblet_Uninflamed']
```

For calcualting the intercellular interactions we can store theese dataframes in a dictionarry.

[28]:
```
healthy_cell_types = {"DC1": df_DC1_healthy,"Macrophage":␣
↪df_macrophage_healthy, "Goblet": df_goblet_healthy,
                      "Myofibroblast": df_myofibroblast_healthy, 'Treg':␣
↪df_Treg_healthy}

uninflamed_cell_types = {"DC1": df_DC1_uninflamed,"Macrophage":␣
↪df_macrophage_uninflamed, "Goblet": df_goblet_uninflamed,
                         "Myofibroblast": df_myofibroblast_uninflamed, 'Treg':␣
↪df_Treg_uninflamed}
```

Next we can use the intercellular interactions.

[29]:
```
# create dictionaries for source-target interactions and their annotations from␣
↪OmniPath
interactions = {}
interaction_annotation = {}

for i, interaction in filtered_intercell_network.iterrows():
    if interaction["genesymbol_intercell_source"] not in interactions:
        interactions[interaction["genesymbol_intercell_source"]] = []
    if interaction["genesymbol_intercell_target"] not in ␣
↪interactions[interaction["genesymbol_intercell_source"]]:
        interactions[interaction["genesymbol_intercell_source"]].
↪append(interaction["genesymbol_intercell_target"])
    source_tupple =␣
↪(interaction["genesymbol_intercell_source"],interaction["category_intercell_source"])
    target_tupple =␣
↪(interaction["genesymbol_intercell_target"],interaction["category_intercell_target"])
    if source_tupple not in interaction_annotation:
        interaction_annotation[source_tupple] = []
```

```
    if target_tupple not in interaction_annotation[source_tupple]:
        interaction_annotation[source_tupple].append(target_tupple)
```

Create all of the possible interactions of cells (independently from the condition) Important: directionality (A-B and B-A are different)

[30]:
```
def create_interactions(cells, healthy_cell_types,interactions):
    healthy_interactions = set()
    for source in healthy_cell_types[cells[0]].keys():
            # selecting those ones which play role in intercellular␣
 ↪communication as a transmitter
            if source in interactions.keys():
                # iterating through target cell type expressed genes
                for target in healthy_cell_types[cells[1]].keys():

                    # selecting those ones which play role in intercellular␣
 ↪communication as a receiver
                    if target in interactions[source]:
                        if str(healthy_cell_types[cells[1]][target]) != 'nan'␣
 ↪and str(healthy_cell_types[cells[0]][source]) != 'nan':
                            healthy_interactions.add((source, target))
    return healthy_interactions
```

[31]:
```
#Creating dictinaries dataframe for healthy and UC interactions
healthy_intercell_tupplellist = []
uc_intercell_tupplelist = []
```

[32]:
```
for i in healthy_cell_types.keys():
    for j in healthy_cell_types.keys():
        if i!=j:
            print(i,j)
            # storing cell-type specific connections in sets
            healthy_interactions = create_interactions([i,j],␣
 ↪healthy_cell_types, interactions)
            UC_interactions = create_interactions([i,j],␣
 ↪uninflamed_cell_types,interactions)
            # writing out the condition specific interactions
            healthy_only = healthy_interactions.difference(UC_interactions)
            UC_only = UC_interactions.difference(healthy_interactions)
            #Adding the intercellular interactions
            healthy_intercell_tupplellist.append((i,j,len(healthy_only)))
            uc_intercell_tupplelist.append((i,j,len(UC_only)))

            with open(i + "_" + j + "_healthy_only.txt", 'w') as output_file_1:
                #print(UC_only)
                for inter in healthy_only:
```

```
                    for source_annotation in interaction_annotation:
                        if inter[0] == source_annotation[0]:
                            for target_annotation in␣
↪interaction_annotation[source_annotation]:
                                if inter[1] == target_annotation[0]:
                                    output_file_1.write(source_annotation[0] +␣
↪"," + source_annotation[1] + ","
                                                                + target_annotation[0]␣
↪+ "," + target_annotation[1] + "\n")
            with open(i + "_" + j + "_UC_only.txt", 'w') as output_file_2:
                for inter in UC_only:
                        for source_annotation in interaction_annotation:
                            if inter[0] == source_annotation[0]:
                                for target_annotation in␣
↪interaction_annotation[source_annotation]:
                                    if inter[1] == target_annotation[0]:
                                        output_file_2.
↪write(source_annotation[0] + "," + source_annotation[1] + ","
                                                                    +␣
↪target_annotation[0] + "," + target_annotation[1] + "\n")
```

```
DC1 Macrophage
DC1 Goblet
DC1 Myofibroblast
DC1 Treg
Macrophage DC1
Macrophage Goblet
Macrophage Myofibroblast
Macrophage Treg
Goblet DC1
Goblet Macrophage
Goblet Myofibroblast
Goblet Treg
Myofibroblast DC1
Myofibroblast Macrophage
Myofibroblast Goblet
Myofibroblast Treg
Treg DC1
Treg Macrophage
Treg Goblet
Treg Myofibroblast
```

Next we make the graphs of the interactions between the cells.

[33]:
```
g_healthy = ig.Graph.TupleList(healthy_intercell_tupplellist, weights=True,␣
↪directed=True)
g_uc = ig.Graph.TupleList(uc_intercell_tupplelist, weights=True, directed=True)
```

[34]:
```
healthy_intercell_tupplellist
```

[34]:
```
[('DC1', 'Macrophage', 912),
 ('DC1', 'Goblet', 781),
 ('DC1', 'Myofibroblast', 731),
 ('DC1', 'Treg', 557),
 ('Macrophage', 'DC1', 568),
 ('Macrophage', 'Goblet', 689),
 ('Macrophage', 'Myofibroblast', 617),
 ('Macrophage', 'Treg', 501),
 ('Goblet', 'DC1', 457),
 ('Goblet', 'Macrophage', 652),
 ('Goblet', 'Myofibroblast', 483),
 ('Goblet', 'Treg', 438),
 ('Myofibroblast', 'DC1', 437),
 ('Myofibroblast', 'Macrophage', 558),
 ('Myofibroblast', 'Goblet', 483),
 ('Myofibroblast', 'Treg', 387),
 ('Treg', 'DC1', 491),
 ('Treg', 'Macrophage', 800),
 ('Treg', 'Goblet', 682),
 ('Treg', 'Myofibroblast', 616)]
```

Now we can visualise the interactions. Igraph contain the visulasitation parameters as a dictionarry. You can also wirte out the edgelist to visualise it in cytoscape.

[35]:
```
g_healthy.es
```

[35]:
```
<igraph.EdgeSeq at 0x19ee53e1948>
```

[36]:
```
visual_style = {}
```

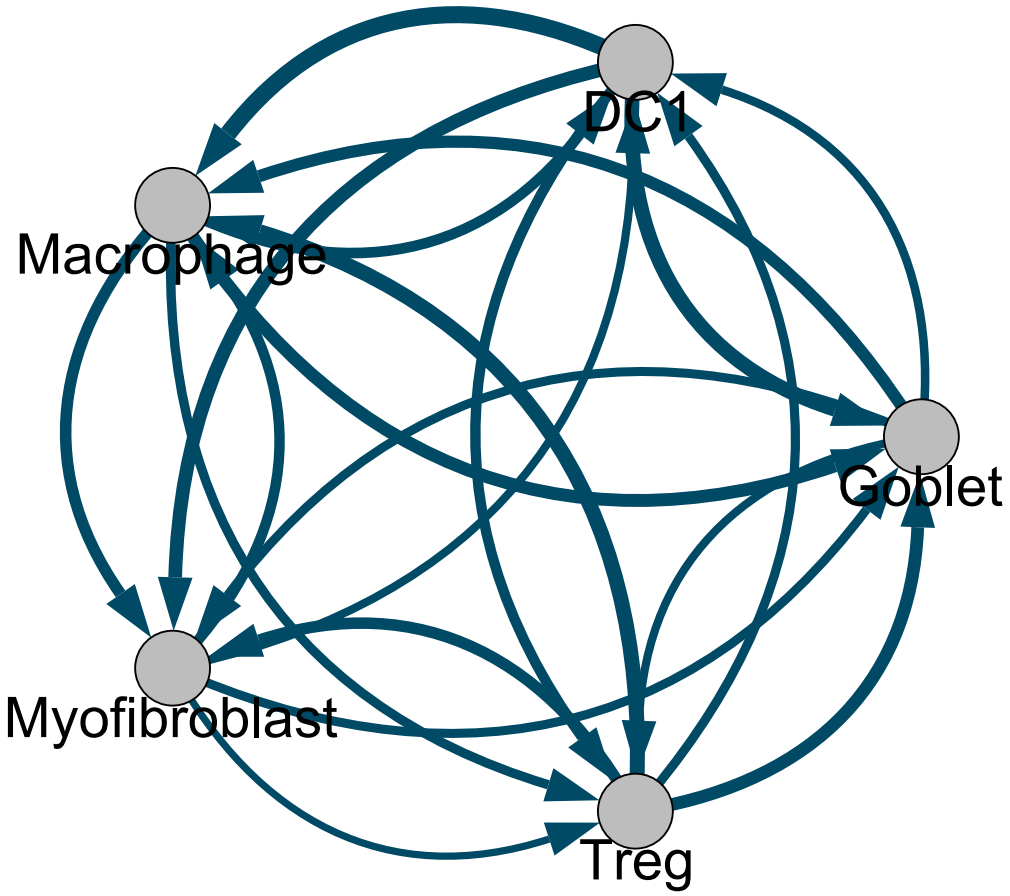
Below are the iGraph visual parameters which you can play with for your intercellular network.

[37]:
```
# Curve the edges
visual_style["edge_curved"] = True
# Set the layout
my_layout = g_healthy.
 ↪layout_circle(order=["Goblet","Treg","Myofibroblast","Macrophage","DC1"])
visual_style["layout"] = my_layout
#Add annoation
visual_style["vertex_label"] = g_healthy.vs["name"]
#Calcualte the edge wheight relative the number of edges
visual_style["edge_width"] = 0.01 * np.array(g_healthy.es["weight"])

#Setting the vertex visualistation parameters -colours and label position
visual_style["vertex_label_size"] = 30
visual_style["vertex_color"] = "grey"
```

```
visual_style["vertex_label_dist"] = 0.75
visual_style["vertex_size"] = 40

#Setting the edge visualisation parameters
visual_style["edge_color"] = "#004C66" #hexa code for blue
visual_style["edge_arrow_size"] = 2

#Setting the visualistation place. igraph need margin yo work
visual_style["bbox"] = (600, 600)
visual_style["margin"] = 100
```

[38]:
```
ig.plot(g_healthy, "healthy.png", **visual_style)
```

[38]:

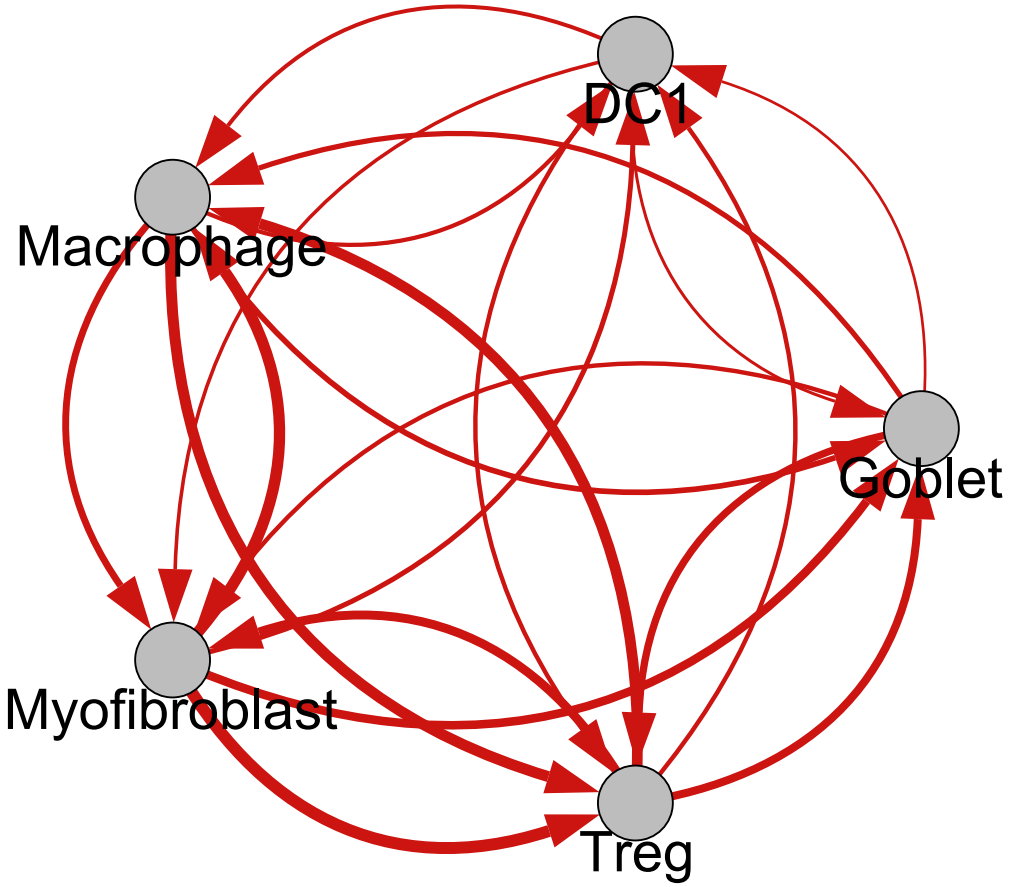


Now we can do the same for the uninflmaed UC cells.

[39]:
```
# Curve the edges
visual_style["edge_curved"] = True
# Set the layout - we keep the same layout for comapriosn
visual_style["layout"] = my_layout
#Add annoation
visual_style["vertex_label"] = g_uc.vs["name"]
#Calcualte the edge wheight relative the number of edges
visual_style["edge_width"] = 0.01 * np.array(g_uc.es["weight"])

#Setting the vertex visualistation parameters -colours and label position
visual_style["vertex_label_size"] = 30
visual_style["vertex_color"] = "grey"
visual_style["vertex_label_dist"] = 0.75
visual_style["vertex_size"] = 40

#Setting the edge visualisation parameters
visual_style["edge_color"] = "#CE1612" #hexa code for red
visual_style["edge_arrow_size"] = 2

#Setting the visualistation place. igraph need margin yo work
visual_style["bbox"] = (600, 600)
visual_style["margin"] = 100
```

[40]:
```
ig.plot(g_uc, "UC_uninflamed.png", **visual_style)
```

[40]:



Below you can see what kind of data we have used.

[41]:
```
import types
def imports():
    for name, val in globals().items():
        if isinstance(val, types.ModuleType):
            yield val.__name__
list(imports())
```

[41]:
```
['builtins',
 'builtins',
 'pandas',
 'igraph',
```

```
 'os',
 'numpy',
 'omnipath',
 'seaborn',
 'types']
```

[42]:
```
print('\n'.join(f'{m.__name__} {m.__version__}' for m in globals().values() if␣
↪getattr(m, '__version__', None)))
```

```
pandas 1.1.4
igraph 0.8.3
numpy 1.18.5
omnipath 1.0.0
seaborn 0.11.0
```