# Pathway analysis with scBonita

## To perform pathway analysis, scBONITA uses the rules generated in Step 2.

In addition, scBONITA requires:

* a **metadata file** specifiying the treatments/experimental variables for each cell and
* a **contrast file** specifying the pairs of treatments to be compared.

## The pathway analysis script has the following arguments:

* **dataFile**
* Specify the name of the file containing processed scRNA-seq data
* **conditions**
* Specify the name of the file containing cell-wise condition labels, ie, metadata for each cell in the training dataset. The columns are condition variables and the rows are cells. The first column must be cell names that correspond to the columns in the training data file. The column names must contain the variables specified in the contrast file (see contrast –help for more information).
* **contrast**
* A text file where each line contains the two conditions (corresponding to column labels in the conditions file) are to be compared during pathway analysis.
* **conditions\_separator**
* Separator for the conditions file. Must be one of ‘comma’, ‘space’ or ‘tab’ (spell out words, escape characters will not work).”,
* **contrast\_separator**
* Separator for the contrast file. Must be one of ‘comma’, ‘space’ or ‘tab’ (spell out words, escape characters will not work).”,
* **pathwayList**
* Paths to GRAPHML files that should be used for scBONITA analysis. Usually networks from non-KEGG sources, saved in GRAPHML format. The default value is to use sll the networks initially used for rule inference.

## Example usage with the provided example files in the data folder:

python3.6 pathwayAnalysis.py --dataFile "data/trainingData.csv" --conditions "data/conditions.txt" --contrast "data/contrast.txt" --conditions\_separator "comma" --contrast\_separator "comma" --pathwayList "hsa00010"

## Output files from scBONITA Pathway Analysis

1. A comma-separated (CSV) file named as

**pvalues + contrast[0] + *vs* + contrast[1] + .csv**

For example, if the conditions to be compared (and specified in the contrasts file) are ‘control’ and ‘treatment’, the output file of scBONITA pathway analysis will be:

**pvalues\_control\_vs\_treatment.csv**

1. For each network, a file ending in \*\*\_importanceScores.csv\*\*

This file contains a table with the following columns:

* **Node**: Gene name
* **Importance Score**: Importance score for the node in the network, calculated using the provided training dataset
* **ObsERS**: Observed size of the equivalent rule set or ERS. This is the number of possible equally valid Boolean rules for this node.
* **MaxERS**: Maximum possible size of the ERS for that node. This is (2^n) - 1, where n is the number of incoming edges for the node in the network.
* Two columns named as the two conditions to be compared/specified in the contrast file: the expression of the cells in those two conditions
* **Upregulated\_in\_condition** A boolean column stating whethere the node is up regulated in ‘condition’

1. For each network, a GRAPHML file ending in “\_IS” with node attributes similar to the ones in the importance scores CSV

## View and analyze the output of scBONITA

#### Load required packages

import pandas as pd  
import matplotlib.pyplot as plt  
import networkx as nx  
from singleCell import \*

#### Example pvalues file

pvalues = pd.read\_csv("data/pvalues\_Control\_vs\_Treatment.csv")  
pvalues.head()

#### Make bubbleplots using generated pvalues

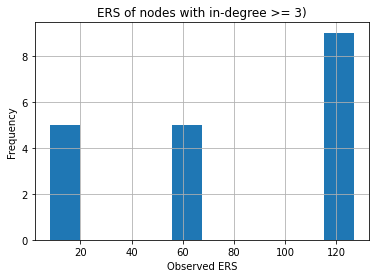
from singleCell import \*  
  
singleCell.makeBubblePlots(  
 pd.read\_csv("data/pvalues\_Control\_vs\_Treatment.csv"),  
 adjPValueThreshold=1,  
 wrap=25,  
 height=8,  
 width=10,  
 palette="colorblind",  
 saveAsPDF=True,  
 outputFile="example\_PA\_bubbleplot.pdf",  
)

#### Example importance scores file

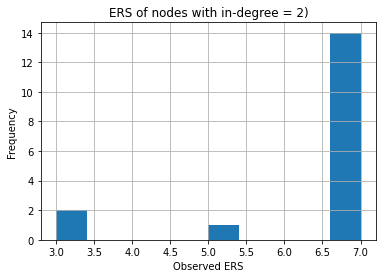
importanceScores = pd.read\_csv(  
 "data/hsa00010.graphml\_processed.graphml\_importanceScores.csv"  
)  
importanceScores.head()

#### Plot the equivalent rule set sizes for this network

importanceScores.loc[importanceScores["MaxERS"] == 127].hist(column="ObsERS")  
plt.xlabel("Observed ERS")  
plt.ylabel("Frequency")  
plt.title("ERS of nodes with in-degree >= 3)")  
plt.show()  
plt.clf()



importanceScores.loc[importanceScores["MaxERS"] == 7].hist(column="ObsERS")  
plt.xlabel("Observed ERS")  
plt.ylabel("Frequency")  
plt.title("ERS of nodes with in-degree = 2)")  
plt.show()  
plt.clf()



#### Visualize the output network in external software such as CytoScape or Gephi

graph = nx.read\_graphml("data/hsa00010\_IS\_Control\_vs\_Treatment.graphml")

print(pd.DataFrame.from\_dict(dict(graph.nodes(data=True)), orient="index"))