**Functions**

**compute\_binomial\_deviance\_\_genuine\_and\_randomized\_counts**

* purpose
  + compute binomial deviance [1] for genuine and randomized data
  + provide information to make a filtering decision – to specify a number of genes for analysis
* parameters
  + required
    - pandas dataframe: UMI counts
    - index (row labels): gene IDs
    - column labels: cell IDs
  + optional
    - number of randomizations to be performed: default = 11
    - percentiles of binomial deviance to be listed in output “summary statistics” data frame – python list. default: = [.25,.5,.75,.9,.95, .96, .97, .98, .99,.995,.999]
* output value: list
  + pandas dataframe: binomial deviance for genuine data
  + pandas dataframe: binomial deviance for each set of randomized counts: one column per randomized data set
  + pandas dataframe: summary statistics
    - median percentiles of binomial deviance for randomized data
    - number of genes with binomial deviance larger than the median percentiles of the randomized data

**filter\_UMI\_counts**

* purpose
  + from a set of UMI counts, extract data for genes with highest binomial deviance
* parameters
  + required
    - pandas dataframe: UMI counts
    - index (row labels): gene IDs
    - column labels: cell IDs
    - pandas dataframe: binomial deviance for genuine data – for example, returned by the function

**compute\_binomial\_deviance\_\_genuine\_and\_randomized\_counts**

* + - number of genes for which counts are to be extracted
* output value: list
  + pandas dataframe: extracted count data: has same columns as input UMI dataframe

**MLE\_cell\_cluster\_models**

* purpose
  + given a clustering of UMI count data, calculate the maximum likelihood estimates of the relative abundance of genes in clusters
* parameters
  + required
    - pandas dataframe: UMI counts
    - index (row labels): gene IDs
    - column labels: cell IDs
    - pandas dataframe: one or more clusterings of the UMI count data
    - index (row labels): cell IDs
    - column labels: clustering IDs (integers) – example: 3 means that the clustering has 3 clusters; the values in the column range from 0 to 2
* output value: list
  + python dict containing results for each input clustering, plus the null model (1 cluster) and the saturated model (number of clusters equals the number of cells).
    - keys: clustering IDs, e.g. 3 for the clustering with 3 clusters
    - values: dicts, each with 3 elements
      * key: 'cell\_cluster\_gene\_totals'

value: pandas dataframe with one row per gene, one column per cluster

each entry equals the total count of all cells in the cluster for the specific gene

* + - * key: 'pi\_hat’

value: pandas dataframe with one row per gene, one column per cluster

each entry equals the MLE

* + - * key: ‘NLL’

value: negative log likelihood

* + pandas dataframe containing negative log likelihood and likelihood ratio statistics

**Caution if clusterings are calculated with Mclust in R**

* Mclust returns clusters numbered from 1 through the number of clusters.
* Python numberings start with zero.
* The python programs in this suite follow this policy, for example, in a clustering with 8 clusters, the clusters are numbered 0 through 7
* When using Mclust output with these programs, **clusters must first be renumbered**.

**Reference**

1. Townes F W, Hicks S C, Aryee M J et al.: Feature selection and dimension reduction for single-cell RNA-Seq based on a multinomial model. *Genome Biol* 20, 295 (2019).