**Internal functions:**

* my\_CV
* my\_FC\_row
* my\_FC
* apply\_sparseMatrix
* sweep\_sparseMatrix

**User functions:**

* initialize function
  + Creates a biostats single cell object which supports multi-sample sc/sn RNAseq dataset
  + @param count\_file: character string representing count file to be read (must be a file type supported by fread)
  + @param meta\_file: character string representing meta data file to be read (must be a file type supported by fread)
  + @param sampleId\_col: character string representing a column in meta data that contains sample id information
  + @param treatment\_col: character string representing a column in meta data that contains treatment information
  + @param cluster\_col: character string representing a column in meta data that contains information of the cell’s cell type or cluster
  + @param count\_data: a count data matrix in sparse matrix format
  + @param meta\_data: meta data in data frame format
  + @param rds\_file: character string representing rds file for input
  + @param mode: internal use
  + @return: R6 object containing counts, cells, genes, meta data, MT rows, normalized counts, and pseudo bulk counts
  + @examples:
  + Example 1: Initialize using count\_file, meta\_file, sampleId\_col, treatment\_col, and cluster\_col arguments.   
      
    This will read single cell data into memory from count\_file (rows are genes, columns are cells, gene names in column 1, and cell names in column names) and read meta data from meta\_file. Sample ids, treatments, and clusters are in columns sampleId\_col, treatment\_col, and cluster\_col of the meta data.
  + Example 2: Initialize using count table, meta data, sampleId\_col, treatment\_col, and cluster\_col.

Initialize with count table (a sparse matrix, rows are genes, columns are cells, gene names in row names, and cell names in column names) and meta\_data data frame.

Sample ids, treatments, and clusters are in columns sampleId\_col, treatment\_col, and cluster\_col of the meta data.

* + Example 3: Initialize from an rds file with sampleId\_col, treatment\_col, and cluster\_col arguments  
      
    Read single cell data into memory from a rds\_file assuming there is a meta.data slot that includes sample id, treatment, and cluster information. Sample ids, treatments, and clusters are in columns sampleId\_col, treatment\_col, and cluster\_col of the meta data.
* make\_QCplots
  + Creates a number of QC plots for the sc/snRNAseq data
  + @param out\_file: character vector representing the file name of an output pdf file of QC plots
  + @return: None
* apply\_filter():
  + Applies the 1st round of biostats filtering pipeline. Note that this filter is applied to all cells of the experiment.
  + @param MTfilter: if TRUE then remove mitochondrial genes. Default is TRUE.
  + @param lib\_size\_low and lib\_size\_high: only keep cells that have a library size between lib\_size\_low and lib\_size\_high [inclusive].   
    Defaults are 200 <= lib\_size <= 20,000,000
  + @param min.cells.per.gene: if perc\_filter is FALSE, then only keep genes have expression in at least min.cells.per.gene cells. Default is 50.
  + @param min.genes.per.cell: keep cells with expression in at least min.genes.per.cell genes. Default is 500.
  + @param min.perc.cells.per.gene: if perc\_filter is TRUE, then only keep genes have expression in at least min.perc.cells.per.gene \* 100 percent of cells. Default is 0.01.
  + @param perc\_filter: if TRUE, apply the cells.per.gene filter using percentages (expressed as a decimal) rather than an absolute threshold. Default is TRUE.
  + @returns None: this function will filter your R6 object in-place
* apply\_filter\_R6():
  + Applies the 1st round of biostats filtering pipeline. Note that this filter is applied to all cells of the experiment.
  + @param MTfilter: if TRUE then remove mitochondrial genes. Default is TRUE.
  + @param lib\_size\_low and lib\_size\_high: only keep cells that have a library size between lib\_size\_low and lib\_size\_high [inclusive].   
    Defaults are 200 <= lib\_size <= 20,000,000
  + @param min.cells.per.gene: if perc\_filter is FALSE, then only keep genes have expression in at least min.cells.per.gene cells. Default is 50.
  + @param min.genes.per.cell: keep cells with expression in at least min.genes.per.cell genes. Default is 500.
  + @param min.perc.cells.per.gene: if perc\_filter is TRUE, then only keep genes have expression in at least min.perc.cells.per.gene \* 100 percent of cells. Default is 0.01.
  + @param perc\_filter: if TRUE, apply the cells.per.gene filter using percentages (expressed as a decimal) rather than an absolute threshold. Default is TRUE.
  + @returns: a new R6 object representing the data after one round of filtering
* subset():
  + Subsets your R6 object based on a provided condition
  + @param condition: a logical condition which uses the columns of the meta data
  + @examples:

sce\_subset <- filterR1$subset(Disease.x %in% c('SPMS'))

* down\_sample()
  + Downsamples cells to a specified number
  + @param number: function will downsample the cells to the amount specified by number
  + @returns: R6 object with downsampled counts and meta data
  + @example: sce\_down <- sce\_qc$down\_sample(775)
* set\_group\_mode():
  + Sets DE analysis to “group” mode (ie comparing groups within a specified cell type)
  + @param cluster\_of\_interest: character string representing cell type of interest
  + @ref\_group: character string representing reference group or character vector of reference groups
  + @alt\_group: character string representing non-reference group or character vector of non-reference groups
  + @returns: None
* set\_cluster\_mode():
  + Sets DE analysis to “cluster” mode (ie comparing clusters/cell types within a specific group)
  + @param group\_of\_interest: your group of interest as a character string
  + @param ref\_cluster: your reference cluster as a character string or a character vector of reference clusters
  + @param alt\_cluster: your non-reference cluster as a character string or a character vector of non-reference clusters or you may specify the use of all clusters that are not in ref\_cluster by using the character string, “others”
  + @returns: None
* apply\_filter\_contrasts\_R6()
  + Apply the 2nd round of biostats filtering. For “group” mode, the filtering is applied to ref\_group and alt\_group for the given cell type of interest. For “cluster mode”, the filtering is applied to ref\_cluster and alt\_cluster for a given group of interest
  + @param min.cells.per.gene: minimum cells expressed per gene. This filter is applied if perc.cells.filter is FALSE and cells.per.gene.filter is TRUE. Default is 50.
  + @param min.perc.cells.per.gene: minimum % cells expressed per gene (use decimal form of percentage). This threshold is applied if perc.cells.filter is TRUE and cells.per.gene.filter is TRUE. Default is 0.10 but recent simulations suggest 0.05 is better.
  + @param perc.cells.filter: TRUE means apply cell.per.gene filtering by use of a percentage rather than absolute threshold. If the percentage results in a number less than min.cells.per.gene, the code will automatically switch to min.cells.per.gene absolute thresholding. Default is TRUE.
  + @param min.cells.per.gene.type: type of cell per gene filtering. If it has the value, “and”, then require expression in both groups. If it has the value “or”, then require expression in either group. Default is “and”.
  + @param cells.per.gene.filter: TRUE means apply cells per gene filtering. Default is TRUE.
  + @param perc.filter = T: if TRUE then apply the 75th percentile filtering. Default is TRUE.
  + @param perc.filter.type: type of percentile filtering. If it has the value, “and”, any gene that has a 75th percentile of zero in both groups will be filtered out. If it has the value, “or”, any gene that has a 75th percentile of zero in either group will be filtered out. Default is “and”
  + @param perc\_threshold: percentile threshold, which is 75th percentile by default. Express the percentile as a decimal.
  + @param min.ave.pseudo.bulk.cpm: cpm filtering threshold. Default is 1.
  + @param pseudo.bulk.cpm.filter: if TRUE then apply a cpm filter on the pseudo-bulk counts. Default is TRUE.
  + @returns a new R6 object with the filtered counts and filtered meta data
* t\_test\_pipeline()
  + Runs the pseudo-bulk DE t-test pipeline
  + @returns data.frame of DE results
* ancova\_pipeline()
  + Runs the pseudo-bulk ancova pipeline
  + @param covs: vector of character strings representing your covariates
  + @returns data.frame of DE results
* u\_test\_pipeline()
  + Runs the pseudo-bulk u-test DE pipeline
  + @returns data.frame of DE results
* edgeR\_pipeline()
  + Runs the pseudo-bulk edgeR DE pipeline
  + @param covs: vector of character strings representing your covariates
  + @returns data.frame of DE results
* limma\_pipeline():
  + Runs the pseudo-bulk limma DE pipeline
  + @param covs: vector of character strings representing your covariates
  + @returns data.frame of DE results
* limma\_cell\_level\_pipeline()
  + Runs the cell-level limma DE pipeline
  + @param covs: vector of character strings representing subject level covariates
  + @param cell\_level\_covs: vector of character strings representing cell level covariates
  + @returns: data frame of DE results
* DESeq2\_pipeline()
  + Runs the pseudo-bulk DESeq2 pipeline
  + @param covs: vector of character strings representing subject level covariates
  + @param shrink: if TRUE, apply shrinkage. Default is TRUE.
  + @param parallel: if TRUE, then use parallel computation. User is required to register cores with BiocParallel. Default is FALSE.
  + @returns: data frame of DE results
* glmmTMB\_pipeline()
  + Runs the glmmTMB pipeline
  + @param covs: vector of character strings representing covariates
  + @param family: model to use (nbinom2, nbinom1, poisson, nbinom2zi, or nbinom1zi). Default is nbinom2.
  + @param cores: number of cores to use, code will register the cores automatically. Cores = 4 by default.
  + @returns data.frame of DE results
* MAST\_pipeline()
  + Runs the MAST pipeline
  + @param covs: vector of character strings representing covariates
  + @param method: method to use for MAST pipeline. Options are glm, glmer, or bayesglm. Default is “glmer”.
  + @param ebayes: if TRUE, mast will use empirical bayes. Default is FALSE.
  + @param detection\_rate: if TRUE, use detection\_rate as a covariate. Default is TRUE.
  + @returns data.frame of DE results
* violinPlot()
  + Creates a violin plot
  + @param gene.name: gene of interest
  + @param de.method: which DE results table to use in annotating the plot. Default is “glmmTMB”
  + @param cell\_info: character string for column of meta data representing cell barcodes
  + @returns a ggplot
* volcanoPlot()
  + Creates a volcano plot
  + @param FDR\_threshold: your desired fdr threshold. Default is 0.05.
  + @param FC\_threshold: your desired FC threshold. Default is 1.
  + @param de.method: which DE results table to use
  + @param title: title of your plot. Default is “Volcano Plot”
  + @returns ggplot
* assayData()
  + Returns the count matrix slot of the R6 object
  + @return sparse matrix representation of the counts
* pseudoBulkData()
  + Returns the pseudobulk data slot of the R6 object
  + @return a R matrix of the pseudobulk data slot
* pData()
  + Returns the meta data slot of the R6 object
  + @return a R data.frame of the meta data
* get\_cells()
  + Return the cells slot of R6 object
  + @return vector of character strings of cell barcodes
* get\_genes()
  + Return the gene slot of the R6 object
  + @return vector of character strings of gene names
* get\_MT()
  + Return the MT genes of the R6 object
  + @return vector of row indexes for MT genes
* get\_lib\_sizes()
  + Return the library sizes of the R6 object
  + @return numeric vector of library sizes
* get\_norm\_counts()
  + Return the normalized counts of the R6 object
  + @return normalized counts as a sparse matrix
* get\_DE\_results()
  + Return DE pipeline results
  + @return list of results for each DE pipeline that has been executed. Each element of the list is a data frame
* get\_filter\_info()
  + Returns a record of each gene that has been filtered out and why.
  + @return list. MT\_gene element records a list of a genes removed due to MT filtering. less\_cell\_num element records genes which were filtered out due to cells.per.gene filter of round 1 filtering. less\_cell\_num\_group records which genes during 2nd round of filtering were filtered out due to cell.per.gene filtering. 75%\_percentile\_zero\_group entry records which genes are filtered out due to 75th percentile filter. low\_cpm entry records which genes are filtered out due to low cpm.