Package 'redeemR'

November 2, 2022

Title What the Package Does (One Line, Title Case)

Version 0.0.0.9000

Description What the package does (one paragraph).

License `use_mit_license()`, `use_gpl3_license()` or friends to pick a license

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 7.1.2

R topics documented:

`AddDatatoplot_clustering` *AddDatatoplot_clustering This prepare the clonal clustering data to plot*

Description

AddDatatoplot_clustering This prepare the clonal clustering data to plot

Usage

```
AddDatatoplot_clustering(object, ...)
```

Arguments

`object`	mitoTracin class

`AddDatatoplot_clustering,mitoTracing-method` *AddDatatoplot_clustering This prepare the clonal clustering data to plot*

Description

AddDatatoplot_clustering This prepare the clonal clustering data to plot

Usage

```
## S4 method for signature 'mitoTracing'
AddDatatoplot_clustering(object)
```

Arguments

`object`	mitoTracin class

Value

mitoTracing class

`AddDist` *AddDist This add Jaccard, Dice, Jaccard3W distance and stored in DistObjects*

Description

AddDist This add Jaccard, Dice, Jaccard3W distance and stored in DistObjects

Usage

```
AddDist(object, ...)
```

Arguments

`object`	mitoTracin class

`AddDist,mitoTracing-method` *AddDist This add Jaccard, Dice, Jaccard3W distance and stored in DistObjects*

Description

AddDist This add Jaccard, Dice, Jaccard3W distance and stored in DistObjects

Usage

```
## S4 method for signature 'mitoTracing'
AddDist(
  object,
  jaccard = T,
  dice = T,
  jaccard3w = T,
  w_jaccard = T,
  w_cosine = T,
  weightDF = NULL,
  NN = 1,
  LSIdist = T,
  dim = 2:50
)
```

Arguments

`object`	mitoTracin class
`jaccard`	default=T
`dice`	default=T
`jaccard3w`	default=T
`w_jaccard`	default=T

w_cosine	default=T
NN	To replace NA, which means a variant shown in the object is not shown in the weight vector, with a number, default is 1 for jaccard system.
LSIdist	default=T
dim	the dimensions to use to calculate LSI distance default is 2:50
weight	A two column dataframe, "Variant"(The variant name should match cell-variant matrix column, e.g, Variants310TC), "weight" (numeric)

Value

mitoTracing class

AddHemSignature *Function to add hematopoietic signatures from Griffin_Signatures*

Description

This function allows you to input a seurat object, add the signatures and return an seurat object

Usage

```
AddHemSignature(object = Donor01_BMMC_Multiome_wrapper.filtered)
```

Arguments

object	a seurat object

Value

a seurat object

AddTree *Add_Tree Optional, if a phylogentic tree object phylo is already available, can be directly added to the mitoTracing*

Description

Add_Tree Optional, if a phylogentic tree object phylo is already available, can be directly added to the mitoTracing

Usage

```
AddTree(object, phylo, ...)
```

Arguments

object	mitoTracin class
phylo	phyogenetic tree object

`AddTree,mitoTracing-method` *Add_Tree Optional, if a phylogentic tree object phylo is already available, can be directly added to the mitoTracing class in slot TREE*

Description

Add_Tree Optional, if a phylogentic tree object phylo is already available, can be directly added to the mitoTracing class in slot TREE

Usage

```
## S4 method for signature 'mitoTracing'
AddTree(object, phylo, record = "")
```

Arguments

`object`	mitoTracin class
`phylo`	phyogenetic tree object

Value

mitoTracing class

`Add_AssignVariant` *Add_AssignVariant a function to assign variants to edges based on maximum likihood*

Description

Add_AssignVariant a function to assign variants to edges based on maximum likihood

Usage

```
Add_AssignVariant(mitoTracing, n.cores, ...)
```

Arguments

`object`	mitoTracin class
`QualifiedTotalCts`	a big source data, usually at XXX/mitoV/final

Add_AssignVariant,mitoTracing-method

a function to assign variants to edges based on maximum likihood

Description

a function to assign variants to edges based on maximum likihood

Usage

```
## S4 method for signature 'mitoTracing'
Add_AssignVariant(mitoTracing = DN1_HSC_mitoTracing.VerySensitive, n.cores = 4)
```

Arguments

mitoTracing	Need to have mitoTracing@Ctx.Mtx.depth (By Add_DepthMatrix), mitoTracing@Cts.Mtx mitoTracing@Cts.Mtx.bi, mitoTracing@TREE

Value

mitoTracing with @AssignedVarian list of two p is a probability matrix of variants vs edges (Rowsum is 1) and Variant.assign.report, a dataframe (Variant|Edge.Assign|prob)

Add_DepthMatrix

Add_DepthMatrix Optional, add a matrix with same dimension with the Cts.Mtx and Cts.Mtx.bi, which display the depths

Description

Add_DepthMatrix Optional, add a matrix with same dimension with the Cts.Mtx and Cts.Mtx.bi, which display the depths

Usage

```
Add_DepthMatrix(object, QualifiedTotalCts, ...)
```

Arguments

object	mitoTracin class
QualifiedTotalCts	a big source data, usually at XXX/mitoV/final

---

`Add_DepthMatrix,mitoTracing-method`
Add_DepthMatrix Optional, add a matrix with same dimension with the Cts.Mtx and Cts.Mtx.bi, which display the depths

---

Description

Add_DepthMatrix Optional, add a matrix with same dimension with the Cts.Mtx and Cts.Mtx.bi, which display the depths

Usage

```
## S4 method for signature 'mitoTracing'
Add_DepthMatrix(object, QualifiedTotalCts)
```

Arguments

`object`	mitoTracin class
`QualifiedTotalCts`	a big source data, usually at XXX/mitoV/final, If needed, edit V1, the cell name, which may have additional postfix due to combine

Value

mitoTracing class

`add_derived_profile_info`
This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/treemut.R#L68

Description

This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/tree

Usage

```
add_derived_profile_info(
  profile_df,
  samples = sprintf("s%s", 0:(nchar(profile_df$profile[1]) - 1))
)
```

`Add_tree_cut` *Add_tree_cut a function to cut tree using assigned variant as branch-length on edge*

Description

Add_tree_cut a function to cut tree using assigned variant as branch-length on edge

Usage

```
Add_tree_cut(mitoTracing, MinCell, N, ...)
```

Arguments

`mitoTracing`	Need to have had the tree built
`MinCell`	The minimum number of cells in each clone, otherwise merge with sibling
`N`	branch length to cut the tree

`Add_tree_cut,mitoTracing-method` *a function to cut tree using assigned variant as branch-length on edge*

Description

a function to cut tree using assigned variant as branch-length on edge

Usage

```
## S4 method for signature 'mitoTracing'
Add_tree_cut(
  mitoTracing = DN4_stemcell_mitoTracing.seed.verysensitive,
  MinCell = 30,
  N = 1,
  prob.cut = 0.3,
  Dumpcut = 100
)
```

Arguments

`mitoTracing`	Need to have had the tree built
`MinCell`	The minimum number of cells in each clone, otherwise merge with sibling
`N`	branch length to cut the tree
`Dumpcut`	Number of can be tolerated to be removed to fulfill the right side. The small value-> Less unassignment, big clones

ATAC_Wrapper *Wrap Seurat ATAC clustering*

Description

This function allows you to perform standard sc-ATAC clustering

Usage

```
ATAC_Wrapper(MTX, res = 0.3, dim1 = 1, dim2 = 20)
```

Arguments

MTX	sparse Matrix of class "dgCMatrix", each row is a peak, each column is a cell,
res	clustering resolution, default=0.5

Value

this returns seurat object with ATAC clustering

Examples

```
bmmc.filtered.atac<-SeuratLSIClustering(PeakVSCell.filtered.Mtx) #each row is a peak, eac
```

BinaryDist *Compute distances for binary distances*

Description

Compute distances for binary distances

Usage

```
BinaryDist(M, method = "Jaccard")
```

Arguments

M	the binary matrix, Each row is a cell, each column is a variant, generated by Make_matrix
method	distance method, choose from Jaccard, Dice, 3WJaccard, Simpson, Kulczynski2, Ochiai, Hamming

Value

dist object

Examples

```
d.Jaccard<-BinaryDist(object@Cts.Mtx.bi,method="Jaccard")
```

`Clone_FinderMarker` *Define a function to perform Find marker for top vs bottom clones This function was developed based on DN4T2.basics.ipynb*

Description

Define a function to perform Find marker for top vs bottom clones This function was developed based on DN4T2.basics.ipynb

Usage

```
Clone_FinderMarker(
  topClones,
  bottomClones,
  HSC_Multiome_wrapper = Donor04_HSC_Multiome_wrapper,
  HSC_mitoTracing,
  assay = "SCT",
  test = "wilcox"
)
```

Arguments

`topClones`	a vector of clone ID eg. c("1","3","7"),this must be in HSC_mitoTracing@CellMeta$Clone_merge
`bottomClones`	a vector of clone ID eg. c("2","5"), this must be in HSC_mitoTracing@CellMeta$Clone_merge
`HSC_mitoTracing`	mitoTracing object for HSC
`test`	the statistic method to use for DE, a wrapper function from Seurat FindAllMarkers
`ob`	Seurat object (Multiomics), the postfix needs to be compatible with HSC_mitoTracing, the cells will be matched by cell names

`ComputeRejectRate` *Function to compute the reject rate(The filtering rate in concensus variant calling)*

Description

This function allows you to computae the filtering rate for each single cell

Usage

```
ComputeRejectRate(WD)
```

Arguments

`WD`	The path to the work space usually XXX/mitoV/final

Value

a dataframe that store the percentage of variant in a given threahold again total

Examples

```
DN9_BMMC_RejectRate<-ComputeRejectRate("/lab/solexa_weissman/cweng/Projects/MitoTracing_V
```

CountVperCell *Internal function in plot_variant*

Description

Internal function in plot_variant

Usage

```
CountVperCell(x, name, CellN)
```

Arguments

x	CellVar.Sum$VN
name	c
CellN	nrow(CellVar.Sum)

Examples

```
CountVperCell(CellVar.Sum$VN,c,CellN=nrow(CellVar.Sum)))
```

Create_mitoTracing *Create_mitoTracing*

Description

This function is to create mitoTracing with basic information

Usage

```
Create_mitoTracing(
  GTsummary_list,
  depth_list,
  feature.list_list,
  meta_list,
  labels,
  thr = "VerySensitive",
  qualifiedCellCut = 10,
  OnlyHetero = T,
  VAFcut = 1,
  Cellcut = 2,
  maxctscut = 2
)
```

Arguments

GTsummary_list	simply put GTSummary (Generated by CW_mgatk.read) into list, this allows mergeing multiple dataset this way.
depth_list	simply put depth(Generated by DepthSummary) into list, this allows mergeing multiple dataset this way.
feature.list_list	simply put feature.list(Generated by Vfilter_v3) into list, this allows mergeing multiple dataset this way.
labels	a vector of labels for the samples.
thr	One of the following "Total","VerySensitive","Sensitive","Specific"
qualifiedCellCut	The minimum median mitochondrial coverage for a qualified cell, default is 10
OnlyHetero	If only consider the heteroplasmy variants, default is T
VAFcut	only use variants with VAF smaller than VAFcut. Default is 1. We can use smaller value to constrain into only using rare variants
Cellcut	only use variants with at least cellcut cells carry
maxctscut	only use variants with at least in one cell with at leaset maxctscut variant fragments

Value

mitoTracing class

CW_mgatk.read *Function to read in mitoV outputs*

Description

This function allows you to read raw data from XX/final folder, the output from mitoV

Usage

```
CW_mgatk.read(path, Processed = F)
```

Arguments

path	The XX/final folder, the output from mitoV
Processed	Boolean variable (Default F), if true directly readRDS("VariantsGTSummary.RDS") or, generate and saveout "VariantsGTSummary.RDS"

Value

this returns depth which is a list of 4 df (Total/VerySensitive/Sensitive/Specific), each is a genotype summary

Examples

```
WD<-"/lab/solexa_weissman/cweng/Projects/MitoTracing_Velocity/SecondaryAnalysis/Donor01_C
DN1CD34_1.VariantsGTSummary<-CW_mgatk.read(WD,Processed =T)
```

Datatoplots-class *An intermediate S4 class Datatoplots*

Description

An intermediate S4 class Datatoplots

Slots

`clustering` dataframe that store the data to plot

DE.gettripple *DE.gettripple*

Description

This function is to prepare the data format that is used to differentially expression calling. It include the raw matrix; data.info and size effect

Usage

```
DE.gettripple(datapair, cpcol, withscran = F)
```

Arguments

`datapair`	tyhe datapair generated from datapair.mk
`cpcol`	The column name for comparison.
`withscran`	if true, use deconvolution to calculate size effect.

Value

This will return .tri.dummy file that is the input for DE analysis

Examples

```
ROCKvsnorock.endo.tri.dummy<-DE.gettripple(ROCKvsnorock.endo.paired,cpcol="name")
```

DepthSummary *Function to summarize the depth (Total that passed Q30)*

Description

This function allows you to summarize the depth

Usage

```
DepthSummary(path, CellSubset = NA, cellSubSetName = NA)
```

Arguments

path	The XX/final folder, the output from mitoV
CellSubset	A vector of ATAC cell names for subsetting, default is NA
cellSubSetName	a string to name this Subset, should explain with the CellSubset
Processed	Boolean variable(Default T), if true directly readRDS("depth.RDS") or, generate and saveout "depth.RDS"

Value

this returns depth which is a list of 4 list(Total/VerySensitive/Sensitive/Specific), each contains 2 df, summarize mito coverage by Pos/Cell

Examples

```
WD<-"/lab/solexa_weissman/cweng/Projects/MitoTracing_Velocity/SecondaryAnalysis/Donor01_C
DN1CD34_1.depth<-DepthSummary(WD,Processed = T)
```

df2ProfileMtx *This is a convinience function, internal*

Description

This is a convinience function, internal

Usage

```
df2ProfileMtx(df)
```

DistObjects-class *An intermediate S4 class Datatoplots*

Description

An intermediate S4 class Datatoplots

Slots

`jaccard` distance object dist: Jaccard distance

`Dice` distance object dist: Dice distance

`jaccard3W` distance object dist: jaccard3W

DoDE *DoDE*

Description

This is the main function for calculating differentially expressed genes

Usage

```
DoDE(tri.dummy, cpcol, onlyoneSample = F, cpus = 16)
```

Arguments

`tri.dummy`	this is generated from DE.gettripple
`cpcol`	the column in tri.dummy$info, the contents of which are used for iteratively compare with one another
`onlyoneSample`	If true, regress out batch effect. Notice, there should be a "Sample" column in in tri.dummy$info that indicate sample or donor or batch
`cpus`	a number of cpus being used for calculation, default is 16

Value

return a list that includes all DE result iteratively

Examples

```
ROCKvsnorock.endo.de<-DoDE(ROCKvsnorock.endo.tri.dummy,"name",onlyoneSample=T,cpus=16)
```

`FromDist2Graph` *FromDist2Graph From disttance object or matrix to graph, default is to return igraph object This function was developed based on*

Description

FromDist2Graph From disttance object or matrix to graph, default is to return igraph object This function was developed based on

Usage

```
FromDist2Graph(d, k.param = 30, return_igraph = T)
```

Arguments

`d`	the distance matrix, this can be either dist or a matrix
`k.param`	K default is 30
`return_igraph`	Wheather return igraph, default is T which return igraph. Otherwise, return adjacent matrix

Value

igraph or adjacent matrix

`GEM_Wrapper` *Wrap Seurat RNA clustering*

Description

This function allows you to perform standard sc-RNA clustering

Usage

```
GEM_Wrapper(mtx = bmmc.data$`Gene Expression`, exp = "DN1_BMMC1", res = 0.5)
```

Arguments

`mtx`	sparse Matrix of class "dgCMatrix", each row is a gene, each column is a cell,
`exp`	The name of this sample/experiment
`res`	clustering resolution, default=0.5

Value

this returns seurat object with RNA clustering

Examples

```
bmmc.data=Read10X(data.dir = "XX/CellRanger/Donor01_BMMC_1/outs/filtered_feature_bc_matri
docluster_GEM(mtx=bmmc.data$`Gene Expression`,exp="DN1_BMMC1")
```

`get_ancestral_nodes` *This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/treemut.R#L68*

Description

This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/tree

Usage

```
get_ancestral_nodes(node, edge, exclude_root = TRUE)
```

`GTSummary` *Function to generate GTS summary*

Description

This function allows you to summarize the meta data for each genotyped variant

Usage

```
GTSummary(RawGenotypes, filterN = T)
```

Arguments

`RawGenotypes`	Well-named "RawGenotypes.Sensitive.StrandBalance" file in function CW_mgatk.read
`filterN`	Boolean variable, if true filter out the variant with "N"

Value

Genotypes.summary a dataframe that summarize several metrics for each genotype

Examples

```
Usually used inside of function CW_mgatk.read
```

LineageBiasPlot *plot_npSummary to plot the lineage composition*

Description

plot_npSummary to plot the lineage composition

Usage

```
LineageBiasPlot(npresult, pre)
```

Arguments

npresult	from ProgenyMapping_np
pre	Any short description for this plot to print with the plot

MakeAllNodes *Define a function make the Allnodes(Node|Parent|Freq|CladeSize), where Freq is the number of variants assigned to the node(as ending point) from mitotracing object,*

Description

Define a function make the Allnodes(Node|Parent|Freq|CladeSize), where Freq is the number of variants assigned to the node(as ending point) from mitotracing object,

Usage

```
MakeAllNodes(
  mitotracing = DN4_stemcell_mitoTracing.seed.verysensitive,
  prob.cut = 0.3
)
```

Arguments

mitotracing	a mitotracing object already have the tree built
prob.cut	The probability cutoff to include confidently assigned variant

`MakeDF4Regress` *MakeDF4Regress Define a function to make two dataframe for regression analysis This function was developed based on HSC_multiome_Het_2.ipynb*

Description

MakeDF4Regress Define a function to make two dataframe for regression analysis This function was developed based on HSC_multiome_Het_2.ipynb

Usage

```
MakeDF4Regress(
  multiome_wrapper = Donor04_HSC_Multiome_wrapper,
  mitoTracing = DN4_stemcell_mitoTracing.seed.sensitive,
  progeny_np = DN4_HSC_LSI_progeny_np,
  assay = "SCT",
  useNPimputation = T,
  maxcloneUMI = 10
)
```

Arguments

`multiome_wrapper`	This outject should includes all and more than HSCs cells in mitoTracing
`mitoTracing`	scMitoTracing object for HSC
`progeny_np`	run via ProgenyMapping_np
`assay`	SCT for expression, ATAC for ATAC
`useNPimputation`	default is T, use all cells called by network propagation, inaddition to the top cells in mitoTracing
`maxcloneUMI`	default is 10, Only include genes, in the max clone the expression greater than 10

Value

list(mtx.clone=mtx.clone,mtx.clone.norm.scale=mtx.clone.norm.scale)

`MakeNN` *Define a function to make nn list, which can be further used to make adjacency matrix This scan row by row, looking for k.param nearest neighbours*

Description

Define a function to make nn list, which can be further used to make adjacency matrix This scan row by row, looking for k.param nearest neighbours

Usage

```
MakeNN(d, k.param = 15)
```

Arguments

d	Distance matrix, can be a dist object or matrix
k.param	Default is 15

Value

return an nn list, which has two components: nn$idx and nn$dist

---

Make_AnnTable *Make_AnnTable, Make a big dataframe, each row is a cell, each column includes info such as clonal UMAP, Clonal ID, ATAC/RNA/WNN UMAP, PCA, gene expression of chosen gene, etc. Require a MitoTracing object and a multiome wrapper that better matches the cells in the MitoTracing*

Description

Make_AnnTable, Make a big dataframe, each row is a cell, each column includes info such as clonal UMAP, Clonal ID, ATAC/RNA/WNN UMAP, PCA, gene expression of chosen gene, etc. Require a MitoTracing object and a multiome wrapper that better matches the cells in the MitoTracing

Usage

```
Make_AnnTable(
  Mitotracing = DN4_HSC_mitoTracing.Sensitive,
  Multiome = Donor04_HSC_Multiome_wrapper,
  clonal_features = c("nCount_mitoV", "seurat_clusters"),
  clonal_features_rename = c("nCount_mitoV", "clone_clusters"),
  CellMeta_features = c("meanCov", "nCount_RNA", "nFeature_RNA", "nCount_ATAC",
    "nFeature_ATAC", "CellType"),
  CellMeta_features_rename = c("Mito_meanCov", "nCount_RNA", "nFeature_RNA",
    "nCount_ATAC", "nFeature_ATAC", "CellType"),
  multiome_features = c("seurat_clusters"),
  multiome_features_rename = c("NewSeurat_cluster"),
  RNAUMAP = T,
  ATACUMAP = T,
  WNNUMAP = T,
  PCA = F,
  LSI = F,
  Variants = "",
  genes = "",
  peaks = "",
  PostTrans_from = c(2, 3),
  PostTrans_to = c(2, 1)
)
```

Arguments

Mitotracing	eg. DN4_HSC_mitoTracing.Sensitive
Multiome	eg. Donor04_HSC_Multiome_wrapper, Multiome_wrapper object that matches with the MitoTracing, a reclustering using Multi_Wrapper() is recommended
clonal_features	eg. c("nCount_mitoV","seurat_clusters"), The column names take from Mitotracing@Seurat@meta.data, importantly the clonal clusterings
clonal_features_rename	eg. c("nCount_mitoV","clone_clusters") Rename the clonal_features
CellMeta_features	eg. c("meanCov","nCount_RNA","nFeature_RNA","nCount_ATAC","nFeature_ATAC","CellType") The column names take from Mitotracing@CellMeta, may useful cell features
CellMeta_features_rename	eg. c("Mito_meanCov","nCount_RNA","nFeature_RNA","nCount_ATAC","nFeature_ATAC","Cell Rename the CellMeta
multiome_features	eg. c("seurat_clusters") The column names take from Multiome@meta.data
multiome_features_rename	eg. c("NewSeurat_cluster") Rename the column names for multiome_features
RNAUMAP	default T
ATACUMAP	Default T
WNNUMAP	Default T
PCA	Default T
LSI	Default T
Variants	Default "" can be a vector of variant names format is eg "Variants10020TC"
genes	Default "" can be a vector of gene names, for example c("HLF","CD34")
peaks	Default "" can be a vector of peaks names
PostTrans_from	Default c(2,3) # This is a tricky part eh nmerging files are involved, find the postfix from cellranger agg for different sample
PostTrans_to	Default c(2,1)

Value

AnnTable

Make_Cells4Nodes *Define a function to make a list, each contains the cell names for a node*

Description

Define a function to make a list, each contains the cell names for a node

Usage

```
Make_Cells4Nodes(
  tr = DN4_SLCT_HSC_w_jaccard.njtree@phylo,
  min.node.size = 10,
  max.node.fra = 0.33
)
```

Arguments

tr	phylo object (ape)
min.node.size	default is 10, only the nodes with more than 10 tips are included (# Minimum # tips in the node to be included)
max.node.fra	default is 0.33, only consider the nodes with less than max.node.fra*total cell number (# The up limit of the node size(Fraction of all tips) to be considered)

Value

return a list each contains the cell names for a node that meets the criteria

---

Make_matrix *Make_matrix This will make the matixies of Cell VS mitochondrial variants and return mitoTracing Results stored in Cts.Mtx and Cts.Mtx.bi*

Description

Make_matrix This will make the matixies of Cell VS mitochondrial variants and return mitoTracing Results stored in Cts.Mtx and Cts.Mtx.bi

Usage

```
Make_matrix(object)
```

Arguments

object	mitoTracin class

Make_matrix,mitoTracing-method *Make_matrix This will make the matixies of Cell VS mitochondrial variants and return mitoTracing Results stored in Cts.Mtx and Cts.Mtx.bi*

Description

Make_matrix This will make the matixies of Cell VS mitochondrial variants and return mitoTracing Results stored in Cts.Mtx and Cts.Mtx.bi

Usage

```
## S4 method for signature 'mitoTracing'
Make_matrix(object)
```

Arguments

object	mitoTracin class

Value

mitoTracin class

Make_tree *Make_tree This will generate a basic phylogenetic tree*

Description

Make_tree This will generate a basic phylogenetic tree

Usage

```
Make_tree(object, d = "jaccard", algorithm = "upgma", onlyreturntree = F, ...)
```

Arguments

object	mitoTracin class
d	"jaccard" or "Dice" or "jaccard3W"
algorithm	the algorithm used to build the tree, choose from "nj" and "upgma"

Make_tree,mitoTracing-method *Make_tree This will generate a basic phylogenetic tree*

Description

Make_tree This will generate a basic phylogenetic tree

Usage

```
## S4 method for signature 'mitoTracing'
Make_tree(object, d, algorithm, onlyreturntree = F)
```

Arguments

object	mitoTracin class
d	"jaccard" or "Dice" or "jaccard3W" or "w_jaccard" "w_cosine" "LSIdist"
algorithm	the algorithm used to build the tree, choose from "nj" and "upgma"

Value

mitoTracin class

MergeMtx *Function to Merge sparse Matrix*

Description

This function allows you to input a list of sparse matrix and merge by rownames, return a new sparse matrix

Usage

```
MergeMtx(mtx.list, postfix)
```

Arguments

mtx.list	A list of sparse matrix to be merged
postfix	a vector of postfix (Usually are numbers that added at the end of cell names). Better be consistent with a merged MitoTracing object orders

Value

new sparse matrix

Examples

```
Donor4_HSC_HPC_BMMC.Mtx<-MergeMtx(list(Donor04_BMMC_Multiome_wrapper$seurat@assays$RNA@cc
Donor4_HSC_HPC_BMMC.RNA.seurat<-GEM_Wrapper(Donor4_HSC_HPC_BMMC.Mtx)
```

mitoTracing-class *Major mitoTracing class that store clonal-resolved multi-omics*

Description

Major mitoTracing class that store clonal-resolved multi-omics

Slots

GTsummary.filtered The Mitochondrial genotype data frame

CellMeta Store meta data for each cell type

V.fitered.list a list of data frame of variant metrics, VAF, cellN, etc (each for different stringency),

UniqueV A character showing the number of usable variant

Cts.Mtx A sparse matrix cell-mitoVariants, store the variant count

Cts.Mtx.bi A sparse matrix cell-mitoVariants, The variant count has been binarized into 0 and 1

Ctx.Mtx.depth A sparse matrix cell-mitoVariants(total counts for each position), store the variant count

para A character showing the parameter of this object

`Seurat` Seurat object storing the clonal clustering results

`DataToplotList` The customized class of Datatoplots: A list of dataframe for further plotting

`DistObjects` The customized class that stores the cell-cell distances

`TREE` The customized class that wraps phylogenetic tree

`Motifenrich.binom` *Motifenrich.binom In house function to compute enrichment from Fimo This function was developed based on HSC_multiome_Het.ipynb and HSC_multiome_Het_2.ipynb*

Description

Motifenrich.binom In house function to compute enrichment from Fimo This function was developed based on HSC_multiome_Het.ipynb and HSC_multiome_Het_2.ipynb

Usage

```
Motifenrich.binom(queryP.motif, controlP.motif, alt = "greater")
```

Arguments

`queryP.motif`	can be a subset of all.motif.sig
`controlP.motif`	can be all.motif.sig
`alt`	default is greater

`Multi_Wrapper` *Wrap Seurat Multiomics clustering*

Description

This function allows you to perform standard sc-multiome clustering

Usage

```
Multi_Wrapper(

    path = "/lab/solexa_weissman/cweng/Projects/MitoTracing_Velocity/SecondaryAn
  atacmin = 1000,
  umimin = 1000,
  CellID = NULL
)
```

Arguments

`path`	this should be the path to the cell-ranger results XX/outs
`atacmin`	minimum atac fragment for each cell, default is 1000
`umimin`	minimum rna umi for each cell, default is 1000
`cellID`	to be used for input(useful for re-clustering), default is NULL which will use the info from path/per_barcode_metrics.csv

Value

this returns seurat object with both RNA and ATAC

Examples

```
Multi_Wrapper(path="XX/CellRanger/Donor01_BMMC_1/outs/")
```

MutationProfile.bulk

Function to plot bulk level mutation signatures

Description

This function allows you to plot the mito mutation signatures

Usage

```
MutationProfile.bulk(cell_variants)
```

Arguments

cell_variants	a vector of variants formated as c('93_A_G"103_G_A"146_T_C'

Value

p from ggplot2

Examples

```
MutationProfile.bulk(DN1CD34_1.Variants.feature.lst[[name]]$Variants
```

NN2M

Define a function convert nn list to adjacency matrix that can be further used for igraph

---

Description

Define a function convert nn list to adjacency matrix that can be further used for igraph

Usage

```
NN2M(nn)
```

Arguments

nn	nn list, which has two components: nn$idx and nn$dist

Value

return an nn.matrix. This is adjacency matrix can be input to igraph graph<-graph_from_adjacency_matrix(nn.matrix,diag = F,mode = "undirected")

plot_depth *Function to plot the mito depth summary*

Description

This function allows you to plot both position-wise and cell-wise mito depth summary

Usage

```
plot_depth(depth = DN1CD34_1.depth, name = "", w = 10, h = 3)
```

Arguments

depth	The .depth file by function DepthSummary
name	The plot name shown on top
w	the Width of the plot, default=10
h	the height of the plot default=3

Value

directly out put the plot

Examples

```
plot_depth(DN1CD34_1.depth$Total,"Total")
```

plot_npSummary *plot_npSummary to assess the outputlevel*

Description

plot_npSummary to assess the outputlevel

Usage

```
plot_npSummary(npresult, orderby = "Total.norm", pre)
```

Arguments

npresult	from ProgenyMapping_np
orderby	Normalize by, so far can work with "Total.norm" and "Total.norm_NPadj"
pre	Any short description for this plot to print with the plot

plot_variant *Function to plot variant metrics*

Description

This function allows you to plot the mito mutation metrics For each category(stringency), p1: Variant allele frequency(VAF); p2: Heteroplasmy histogram p3: CellN(Number of caells that carry the variants) VS maxcts(The number of variant counts in the highest cell) p4: Histogram to show the distribution of the number of variant per cell

Usage

```
plot_variant(
  GTSummary,
  feature.list,
  depth,
  cat = c("Total", "VerySensitive", "Sensitive", "Specific"),
  p4xlim = 50,
  QualifyCellCut = 10
)
```

Arguments

GTSummary	The GTSummary file read in by function CW_mgatk.read
feature.list	The variant feature list generated by Vfilter_v3
depth	The .depth file by function DepthSummary
cat	The category(or the striengency to be ploted), default is c("Total","VerySensitive","Sensitive","Specifi
p4xlim	the p4 xlim(number of variant per cell), default is 50
QualifyCellCut	median coverage for qualified cells, default is 10

Value

no returns, directly plot

Examples

```
plot_variant(DN1CD34_1.VariantsGTSummary,DN1CD34_1.Variants.feature.lst,depth=DN1CD34_1.d
```

ProgenyMapping *Define a function to perform single-cell based hard porogeny assignment This function was developed based on DN4T2.basics.ipynb*

Description

Define a function to perform single-cell based hard porogeny assignment This function was developed based on DN4T2.basics.ipynb

Usage

```
ProgenyMapping(
  HSC_mitoTracing = DN4_PhenoHSC_mitoTracing.verysensitive,
  Full_mitoTracing = DN4_BMMC_HSPC_HSC_mitoTracing.verysensitive,
  distCut = 0.95,
  d = "w_jaccard"
)
```

Arguments

HSC_mitoTracing	The HSC_mitoTracing is the mitoTracing object for defined HSC
Full_mitoTracing	The FULL_mitoTracing is the mitoTracing object for the full BMMC_HSPC_HSC
distCut	Default is 0.95, the distance, below which I define as the related progeny

---

ProgenyMapping_np *ProgenyMapping_np Define a function to compute network propagation based probability FromDist2Graph is needed to convert fistance matrix into MNN graph*

Description

ProgenyMapping_np Define a function to compute network propagation based probability FromDist2Graph is needed to convert fistance matrix into MNN graph

Usage

```
ProgenyMapping_np(
  HSC_mitoTracing = DN4_stemcell_mitoTracing.seed.verysensitive,
  Full_mitoTracing = DN4_BMMC_HSPC_HSC_mitoTracing.verysensitive,
  CloneCol = "Clone_merge",
  k = 30,
  gm = 0.5,
  useLSI = F,
  useSCAVENGE_LSI = F,
  subsample = F,
  ProbCut = 0.7,
  Celltype = "Rig.CellType"
)
```

Arguments

HSC_mitoTracing	The HSC_mitoTracing is the mitoTracing object for defined HSC, have already gone through Add_DepthMatrix–Add_AssignVariant–Add_tree_cut, otherwise, need othereise, need a column in CellMeta that indicates the clone ID
Full_mitoTracing	The FULL_mitoTracing is the mitoTracing object for the full BMMC_HSPC_HSC
CloneCol	"Clone_merge"

k	the k.param used for MNN graph
gm	gamma default is 0.05 which mean 95% information is passing out
ProbCut	The cutoff of the maximum probability for a given progeny cell(If the maximum probability is lower than ProbCut, it will be filtered)
Celltype	The column to be used in aggregate into lineages

Value

a list of two ALLmeta.npClone (A meta data with last column npClone), np_mat (the network propagation matrix))

---

quick_w_cosine *Compute weighted cosine distance*

Description

Compute weighted cosine distance

Usage

```
quick_w_cosine(M, w)
```

Arguments

M	the binary matrix, Each row is a cell, each column is a variant, generated by Make_matrix
w	weight for each variant, a vector

Value

dist object

---

quick_w_jaccard *Compute weighted jaccard distance*

Description

Compute weighted jaccard distance

Usage

```
quick_w_jaccard(M, w)
```

Arguments

M	the binary matrix, Each row is a cell, each column is a variant, generated by Make_matrix
w	weight for each variant, a vector

Value

dist object

Reclustering *Function to reclustering a seurat object*

Description

This function allows you to input a seurat object(multiome), redo clustering. Usually this is after subset

Usage

```
Reclustering(ob)
```

Arguments

ob	a seurat object

Value

a seurat object

Reclustering_hm *Function to reclustering_hm a seurat object with Harmony*

Description

This function allows you to input a seurat object(multiome), redo clustering harmony by a certain column in meta data. Usually this is after subset

Usage

```
Reclustering_hm(
  ob = DN4_RigHSC_T1T2_Multiome_wrapper_filtered.anno,
  HarmonyBy = "TimePoint"
)
```

Arguments

ob	a seurat object
HarmonyBy	The columne name in meta that will be used for Harmony

Value

a seurat object

`reconstruct_genotype_summary` *This is a function borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/treemut.R#L68 Input phylo object, return a "profile matrix"–Edge(or denoted as the ending node) vs cell. a 0, 1 character string that indicate what cells in a given node*

Description

This is a function borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/treemut.R#L68 Input phylo object, return a "profile matrix"–Edge(or denoted as the ending node) vs cell. a 0, 1 character string that indicate what cells in a given node

Usage

```
reconstruct_genotype_summary(phylo)
```

Arguments

`phylo`	phylo an ape object

Value

df includes df$df which is a big data frame, and df$sample that is the cell names

`Runplot_scale_2` *plot_npSummary to assess the outputlevel vs lineage bias, normalize by assigned*

Description

plot_npSummary to assess the outputlevel vs lineage bias, normalize by assigned

Usage

```
Runplot_scale_2(
  datatoplot = DN4_HSC_LSI_progeny$LineageSummary$output_lineage.summary.pct.sca
  pre
)
```

Arguments

`datatoplot`	A slot from the result of ProgenyMapping_np : datatoplot.scale
`pre`	Any short description for this plot to print with the plot

Runplot_scale_3 *plot_npSummary to assess the outputlevel vs lineage bias, normalize by HSC original clone size*

Description

plot_npSummary to assess the outputlevel vs lineage bias, normalize by HSC original clone size

Usage

```
Runplot_scale_3(
  datatoplot = DN4_HSC_LSI_progeny$LineageSummary$output_lineage.summary.pct.sca
  pre
)
```

Arguments

datatoplot	A slot from the result of ProgenyMapping_np : datatoplot.scale
pre	Any short description for this plot to print with the plot

Run_Lin_regression *Run_Lin_regression*

Description

Firstly used in HSC_multiome_Het_2.ipynb

Usage

```
Run_Lin_regression(
  LinOut,
  regress_factor = c("OutLevel.scale", "OutLevel_NPadj.scale", "Lym", "Mye", "MK
    "ME"),
  n.cores = 8
)
```

Arguments

LinOut	produced by MakeDF4Regress
n.cores	=8

`Run_Lin_regression_poi` *Run_Lin_regression_poi Firstly used in HSC_multiome_Het_2.ipynb This function was developed based on*

Description

Run_Lin_regression_poi Firstly used in HSC_multiome_Het_2.ipynb This function was developed based on

Usage

```
Run_Lin_regression_poi(
  LinOut,
  regress_factor = c("OutLevel.scale", "OutLevel_NPadj.scale", "Lym", "Mye", "MK
    "ME"),
  n.cores = 8
)
```

Arguments

`LinOut`	produced by MakeDF4Regress
`regress_factor`	default is c("OutLevel.scale","OutLevel_NPadj.scale","Lym","Mye","MK","ME")
`n.cores`	=8

`SeuratLSIClustering` *SeuratLSIClustering This will use the mito variants for Seurat clustering (LSI based)*

Description

SeuratLSIClustering This will use the mito variants for Seurat clustering (LSI based)

Usage

```
SeuratLSIClustering(object, ...)
```

Arguments

`object`	mitoTracin class

`SeuratLSIClustering,mitoTracing-method`
SeuratLSIClustering This will use the mito variants for Seurat clustering (LSI based)

Description

SeuratLSIClustering This will use the mito variants for Seurat clustering (LSI based)

Usage

```
## S4 method for signature 'mitoTracing'
SeuratLSIClustering(
  object,
  binary = T,
  res = 0.6,
  lsidim = 2:50,
  rmvariants = c("Variants310TC", "Variants3109TC", "Variants5764CT")
)
```

Arguments

`binary`	Default is tree, to make use of the binary matrix
`res`	Default os 0.3, the resolution of the clustering
`mitoTracing`	class

Value

mitoTracing class

`show,mitoTracing-method`
show This will show the basics of mitoTracin class

Description

show This will show the basics of mitoTracin class

Usage

```
## S4 method for signature 'mitoTracing'
show(object)
```

Arguments

`object`	mitoTracin class

Value

print out basics

`split_profile` *This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/treemut.R#L68*

Description

This is a convinience function, internal borrowed from https://github.com/NickWilliamsSanger/treemut/blob/main/R/tree

Usage

```
split_profile(profile)
```

`str2vector` *This is a convinience function, internal*

Description

This is a convinience function, internal

Usage

```
str2vector(x)
```

`Subset_MitoTracing` *Subset_MitoTracing Subset a mitotracing object by selecting a subset of cells, return a new MitoTracing object with only 4 slots: para; CellMeta; Cts.Mtx.bi; UniqueV, can be used for downstreme compute distance, clonal clustering, make tree, etc*

Description

Subset_MitoTracing Subset a mitotracing object by selecting a subset of cells, return a new MitoTracing object with only 4 slots: para; CellMeta; Cts.Mtx.bi; UniqueV, can be used for downstreme compute distance, clonal clustering, make tree, etc

Usage

```
Subset_MitoTracing(MitoTracing, Cells, ExtraInfo = "Subset from ... ")
```

Arguments

`Cells`	Important, give a vector of Cell names(ATAC cell names)
`ExtraInfo`	Extra information, usually "Subset from ..."
`Mitotracing`	The Parent MitoTracing object eg. DN4_HSC_mitoTracing.Sensitive

Value

MitoTracing Object

Tomerge_v2 *Tomerge_v2*

Description

This function is to quickly merge two dataframe by rownames, but can choose to leave A or B all information

Usage

```
Tomerge_v2(A, B, leavex = T, leavey = F)
```

Arguments

A	dataframe A
B	dataframe B

Value

return a data frame with merged information

Examples

```
Tomerge_v2(A,B)
```

Translate_RNA2ATAC *Function to translate the RNA barcode into ATAC barcode and add a column*

Description

This function allows you to input the metadata with row name as cell barcode

Usage

```
Translate_RNA2ATAC(
  meta = bmmc.filtered@meta.data,
  PostFix = T,
  bclength = 16,
  from = c(1, 2, 3),
  to = c(1, 2, 3)
)
```

Arguments

meta	a dataframe with the row names as the RNA cell barcode usually with the post -1
bclength	The cell barcode length, default is 16
from	A vector of the postfix, usually is c(1,2,3,...), it depends on how many samples are aggregated in Cellranger RNA part
to	A vector of the postfix, those cooresponds to the postfix added in scMitoTracing, in general, if it matches, then simply c(1,2,3,...), but in case not match, here provides a way to transform into scMitoTracing order

Value

meta a dataframe

Examples

```
Translate_RNA2ATAC(meta)
```

Translate_simple_ATAC2RNA

Translate_simple_ATAC2RNA

Description

This function allows you to input the ATAC name to translate to RNA name

Usage

```
Translate_simple_ATAC2RNA(
  name,
  PostFix = T,
  bclength = 16,
  from = c(1, 2, 3),
  to = c(1, 2, 3)
)
```

Arguments

name	RNA name, as the RNA cell barcode usually with the post -1
bclength	The cell barcode length, default is 16
from	A vector of the postfix, usually is c(1,2,3,...), it depends on how many samples are aggregated in Cellranger RNA part
to	A vector of the postfix, those cooresponds to the postfix added in scMitoTracing, in general, if it matches, then simply c(1,2,3,...), but in case not match, here provides a way to transform into scMitoTracing order

Value

RNA name Translate_RNA2ATAC(a vector of RNA names)

Translate_simple_RNA2ATAC *Translate_simple_RNA2ATAC*

Description

This function allows you to input the RNA name to translate to ATAC name

Usage

```
Translate_simple_RNA2ATAC(
  name,
  PostFix = T,
  bclength = 16,
  from = c(1, 2, 3),
  to = c(1, 2, 3)
)
```

Arguments

name	RNA name, as the RNA cell barcode usually with the post -1
bclength	The cell barcode length, default is 16
from	A vector of the postfix, usually is c(1,2,3,...), it depends on how many samples are aggregated in Cellranger RNA part
to	A vector of the postfix, those cooresponds to the postfix added in scMitoTracing, in general, if it matches, then simply c(1,2,3,...), but in case not match, here provides a way to transform into scMitoTracing order

Value

ATAC name Translate_RNA2ATAC(a vector of RNA names)

TREE-class *An intermediate S4 class Tree that store tree info*

Description

An intermediate S4 class Tree that store tree info

Slots

`phylo` the phylo tree class from ape package

`treedata` treedata class from tidytree

`records` character to store annotations

`Vfilter_v3` *Function to filter variants*

Description

This function allows you to filter variants

Usage

```
Vfilter_v3(
  InputSummary,
  depth,
  Rmvhomo = F,
  Min_Cells = 2,
  Max_Count_perCell = 2,
  QualifyCellCut = 10
)
```

Arguments

`InputSummary`	The GTSummary file read in by function CW_mgatk.read
`depth`	The .depth file by function DepthSummary
`Rmvhomo`	Boolean (Default F) If true, remove the homozygous variants
`Min_Cells`	Default 2, A qualified variant needs the minimum number of cells that have this variant
`Max_Count_perCell`	Default 2, A qualified variant needs to show at least 2 counts in one cell
`QualifyCellCut`	Default 10, Minimum depth for a qualified cell

Value

this returns feature.list

Examples

```
DN1CD34_1.Variants.feature.lst<-Vfilter_v3(InputSummary=DN1CD34_1.VariantsGTSummary,depth
```

Index