Package 'jrSiCKLSNMF'

July 6, 2023

Type Package

Title Multimodal Single-Cell Omics Dimensionality Reduction

Version 1.2.1

Description Methods to perform Joint graph Regularized Single-Cell Kullback-Leibler Sparse Non-negative Matrix Factorization ('jrSiCKLSNMF', pronounced ``junior sickles NMF'') on quality controlled single-cell multimodal omics count data. 'jrSiCKLSNMF' specifically deals with dual-assay scRNA-seq and scATAC-seq data. This package contains functions to extract meaningful latent factors that are shared across omics modalities. These factors enable accurate cell-type clustering and facilitate visualizations. Methods for preprocessing, clustering, and mini-batch updates and other adaptations for larger datasets are also included. For further details on the methods used in this package please see Ellis, Roy, and Datta (2023) <doi:10.3389/fgene.2023.1179439>.

License GPL-3

Encoding UTF-8

RoxygenNote 7.2.3

Imports Rcpp (>= 1.0.9), igraph, umap, kknn, ggplot2, methods, stats, rlang, Matrix, data.table,parallel,pbapply,cluster,MASS,clValid,factoextra, foreach,irlba,scran,Rdpack

Suggests knitr, rmarkdown

LinkingTo Rcpp, RcppArmadillo, RcppProgress

LazyData true

LazyDataCompression xz

VignetteBuilder knitr

RdMacros Rdpack

NeedsCompilation yes

Author Dorothy Ellis [aut, cre] (<https://orcid.org/0000-0002-8624-0042>), Susmita Datta [ths], Kenneth Perkins [ctb] (Util.h function author, http://programmingnotes.org/), Renaud Gaujoux [ctb] (Author of .nndsvd R adaptation)

Maintainer Dorothy Ellis <ddemoreellis@gmail.com>

Depends R (>= 3.5.0)

Repository CRAN

Date/Publication 2023-07-06 18:40:04 UTC

R topics documented:

`AddSickleJrMetadata` *Add metadata to an object of class SickleJr*

Description

Add any type of metadata to an object of class SickleJr. Metadata are stored in list format under the name specified in `metadataname` of each node in slot `metadata`.

Usage

```
AddSickleJrMetadata(SickleJr, metadata, metadataname)
```

Arguments

`SickleJr`	An object of class SickleJr holding at least one count matrix of omics data
`metadata`	Metadata to add to the SickleJr object; there are no restrictions on type
`metadataname`	A string input that indicates the desired name for the added metadata.

Value

An object of class SickleJr with added metadata

Examples

```
SimSickleJrSmall<-AddSickleJrMetadata(SimSickleJrSmall,
SimData$cell_type,"cell_types_full_data")
```

`BuildKNNGraphLaplacians`

Build KNN graphs and generate their graph Laplacians

Description

Generate graph Laplacians for graph regularization of jrSiCKLSNMF from the list of raw count matrices using a KNN graph. Note that this is only appropriate when the number of features is considerably greater than the number of cells in all modalities. If this is not the case, please use `BuildSNNGraphLaplacians` or any other method of graph construction that does not rely on the Euclidean distance and store the graph Laplacians for each modality as a list in the `graph.laplacian.list` slot.

Usage

```
BuildKNNGraphLaplacians(SickleJr, k = 20)
```

Arguments

`SickleJr`	An object of class SickleJr
`k`	Number of KNN neighbors to calculate; by default, is set to 20

Value

An object of class SickleJr with a list of graph Laplacians in sparse matrix format added to the `graph.laplacian.list` slot

References

Lun AT, McCarthy DJ, Marioni JC (2016). "A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor." *F1000Research*, **5**. ISSN 1759796X, doi:10.12688/F1000RESEARCH.9501.2/DOI, `https://pubmed.ncbi.nlm.nih.gov/27909575/`.

Examples

```
SimSickleJrSmall<-BuildKNNGraphLaplacians(SimSickleJrSmall)
```

BuildSNNGraphLaplacians *Build SNN graphs and generate their graph Laplacians*

Description

Generate graph Laplacians for graph regularization of jrSiCKLSNMF from the list of raw count matrices using an SNN graph. SNN is more robust to situations where the number of cells outnumbers the number of features. Uses the `scran` package's `BuildSNNGraph` function (Lun et al. 2016)

Usage

```
BuildSNNGraphLaplacians(SickleJr, k = 20)
```

Arguments

`SickleJr`	An object of class SickleJr
`k`	Number of KNN neighbors to calculate SNN graph; defaults to 20

Value

An object of class SickleJr with list of graph Laplacians in sparse matrix format added to its `graph.laplacian.list` slot

References

Lun AT, McCarthy DJ, Marioni JC (2016). "A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor." *F1000Research*, **5**. ISSN 1759796X, doi:10.12688/F1000RESEARCH.9501.2/DOI, https://pubmed.ncbi.nlm.nih.gov/27909575/.

Examples

```
SimSickleJrSmall<-BuildSNNGraphLaplacians(SimSickleJrSmall)
```

CalculateUMAPSickleJr *Calculate the UMAP for an object of class SickleJr*

Description

Perform UMAP on the $\mathbf{H}$ matrix alone (default) or within a modality by using UMAP on the $W^v H$ corresponding to modality v.

Usage

```
CalculateUMAPSickleJr(
  SickleJr,
  umap.settings = umap::umap.defaults,
  modality = NULL
)
```

Arguments

SickleJr	An object of class SickleJr
umap.settings	Optional settings for the umap; defaults to umap.defaults
modality	A number corresponding to the desired modality; if set, will perform UMAP on $\mathbf{W}^{\text{modality}}$ on $\mathbf{H}$ alone; not recommended for datasets of more than 1000 cells

Value

An object of class SickleJr with UMAP output based on the $\mathbf{H}$ matrix alone or within a modality added to its umap slot

References

McInnes L, Healy J, Saul N, Großberger L (2018). "UMAP: Uniform Manifold Approximation and Projection." *Journal of Open Source Software*, **3**(29), 861. ISSN 2475-9066, doi:10.21105/JOSS.00861, https://joss.theoj.org/papers/10.21105/joss.00861.

Examples

```
#Since this example has only 10 observations,
#we need to modify the number of neighbors from the default of 15
umap.settings=umap::umap.defaults
umap.settings$n_neighbors=2
SimSickleJrSmall<-CalculateUMAPSickleJr(SimSickleJrSmall,
umap.settings=umap.settings)
SimSickleJrSmall<-CalculateUMAPSickleJr(SimSickleJrSmall,
umap.settings=umap.settings,modality=1)
SimSickleJrSmall<-CalculateUMAPSickleJr(SimSickleJrSmall,
umap.settings=umap.settings,modality=2)
```

ClusterSickleJr *Cluster the* $\mathbf{H}$ *matrix*

Description

Perform k-means, spectral clustering, clustering based off of the index of the maximum latent factor, or Louvain community detection on the $\mathbf{H}$ matrix. Defaults to k-means.

Usage

```
ClusterSickleJr(
  SickleJr,
  numclusts,
  method = "kmeans",
  neighbors = 20,
  louvainres = 0.3
)
```

Arguments

SickleJr	An object of class SickleJr
numclusts	Number of clusters; can be NULL when method is "max" or "louvain"
method	String holding the clustering method: can choose "kmeans" for k-means clustering, "spectral" for spectral clustering, "louvain" for Louvain community detection or "max" for clustering based on the maximum row value; note that "max" is only appropriate for jrSiCKLSNMF with L2 norm row regularization
neighbors	Number indicating the number of neighbors to use to generate the graphs for spectral clustering and Louvain community detection: both of these methods require the construction of a graph first (here we use KNN); defaults to 20 and unused when the clustering method equal to "kmeans" or "max"
louvainres	Numeric containing the resolution parameter for Louvain community detection; unused for all other methods

Value

SickleJr- an object of class SickleJr with added clustering information

References

Blondel VD, Guillaume JL, Lambiotte R, Lefebvre E (2008). "Fast unfolding of communities in large networks." *Journal of Statistical Mechanics: Theory and Experiment*, **2008**(10), P10008. ISSN 1742-5468, doi:10.1088/17425468/2008/10/P10008, 0803.0476, https://iopscience.iop.org/article/10.1088/1742-5468/2008/10/P10008.

Lun AT, McCarthy DJ, Marioni JC (2016). "A step-by-step workflow for low-level analysis of single-cell RNA-seq data with Bioconductor." *F1000Research*, **5**. ISSN 1759796X, doi:10.12688/F1000RESEARCH.9501.2/DOI, https://pubmed.ncbi.nlm.nih.gov/27909575/.

Ng AY, Jordan MI, Weiss Y (2001). "On spectral clustering: analysis and an algorithm." In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, 849–856.

Schliep K, Hechenbichler K (2016). "kknn: Weighted k-Nearest Neighbors." https://cran.r-project.org/package=kknn.

Xu W, Liu X, Gong Y (2003). "Document clustering based on non-negative matrix factorization." *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval*, 267–273. doi:10.1145/860435.860485, https://dl.acm.org/doi/10.1145/860435.860485.

Examples

```
SimSickleJrSmall<-ClusterSickleJr(SimSickleJrSmall,3)
SimSickleJrSmall<-ClusterSickleJr(SimSickleJrSmall,method="louvain",neighbors=5)
SimSickleJrSmall<-ClusterSickleJr(SimSickleJrSmall,method="spectral",neighbors=5,numclusts=3)
#DO NOT DO THIS FOR REAL DATA; this is just to illustrate max clustering
SimSickleJrSmall<-SetLambdasandRowReg(SimSickleJrSmall,rowReg="L2Norm")
SimSickleJrSmall<-ClusterSickleJr(SimSickleJrSmall,method="max")
```

CreateSickleJr *Create an object of class SickleJr*

Description

Using a list of sparse count matrices, create an object of class SickleJr and specify the names of these count matrices.

Usage

```
CreateSickleJr(count.matrices, names = NULL)
```

Arguments

count.matrices	A list of quality-controlled count matrices with pre-filtered features where each modality corresponds to each matrix in the list
names	Optional parameter with names for the count matrices in vector format

Value

An object of class SickleJr with sparse count matrices added to the `count.matrices` slot

Examples

```
ExampleSickleJr<-CreateSickleJr(SimData$Xmatrices)
```

DetermineClusters *Perform clustering diagnostics*

Description

A wrapper for the `clValid` and `fviz_nbclust` functions to perform clustering diagnostics

Usage

```
DetermineClusters(
  SickleJr,
  numclusts = 2:20,
  clusteringmethod = "kmeans",
  diagnosticmethods = c("wss", "silhouette", "gap_stat"),
  clValidvalidation = "internal",
  createDiagnosticplots = TRUE,
  runclValidDiagnostics = TRUE,
  printPlots = TRUE,
  printclValid = TRUE,
  subset = FALSE,
  subsetsize = 1000,
  seed = NULL
)
```

Arguments

`SickleJr`	An object of class SickleJr
`numclusts`	A vector of integers indicating the number of clusters to test
`clusteringmethod`	String holding the clustering method: defaults to k-means; since the other methods are not implemented in jrSiCKLSNMF, it is recommended to use k-means.
`diagnosticmethods`	Vector of strings indicating which methods to plot. Defaults to all three of the available: wss, silhouette, and gap_stat
`clValidvalidation`	String containing validation method to use for `clValid`. Defaults to internal.
`createDiagnosticplots`	Boolean indicating whether to create diagnostic plots for cluster size
`runclValidDiagnostics`	Boolean indicating whether to calculate the diagnostics from `clValid`
`printPlots`	Boolean indicating whether to print the diagnostic plots
`printclValid`	Boolean indicating whether to print the diagnostic results from `clValid`
`subset`	Boolean indicating whether to calculate the diagnostics on a subset of the data rather than on the whole dataset.
`subsetsize`	Numeric value indicating size of the subset
`seed`	Numeric value holding the random seed

Value

An object of class SickleJr with cluster diagnostics added to its `clusterdiagnostics` slot

References

Brock G, Pihur V, Datta S, Datta S (2008). "clValid: An R Package for Cluster Validation." *Journal of Statistical Software*, **25**(4), 1–22. https://www.jstatsoft.org/v25/i04/.

Kassambara A, Mundt F (2020). *factoextra: Extract and Visualize the Results of Multivariate Data Analyses*. R package version 1.0.7, https://CRAN.R-project.org/package=factoextra.

Examples

```
#Since these data are too small, the clValid diagnostics do not run
#properly. See the vignette for an example with the clValid diagnostics
SimSickleJrSmall<-DetermineClusters(SimSickleJrSmall,numclusts=2:5,runclValidDiagnostics=FALSE)
```

`DetermineDFromIRLBA` *Create elbow plots of the singular values derived from IRLBA to determine D for large datasets*

Description

This generates v+1 plots, where v is the number of data modalities, of the approximate singular values generated by IRLBA.There is one plot for each modality and then a final plot that concatenates all of the modalities together. Choose the largest elbow value among the three plots.

Usage

```
DetermineDFromIRLBA(SickleJr, d = 50)
```

Arguments

`SickleJr`	An object of class SickleJr
`d`	Number of desired factors; it is important to select a number that allows you to see a clear elbow: defaults to 50.

Value

An object of class SickleJr with plots for IRLBA diagnostics added to its `plots` slot

References

Baglama J, Reichel L (2005). "Augmented Implicitly Restarted Lanczos Bidiagonalization Methods." *SIAM Journal on Scientific Computing*, **27**(1), 19–42. ISSN 10648275, doi:10.1137/04060593X.

Examples

```
SimSickleJrSmall<-DetermineDFromIRLBA(SimSickleJrSmall,d=5)
```

GenerateWmatricesandHmatrix *Initialize the* $\mathbf{W}$ *matrices in each modality and the shared* $\mathbf{H}$ *matrix*

Description

Create the $\mathbf{W}^v$ matrices and $\mathbf{H}$ matrix via non-negative double singular value decomposition (NNDSVD) (Boutsidis and Gallopoulus 2008; Gaujoux and Seoighe 2010) or randomization. For randomization, the algorithm runs for 10 rounds for the desired number of random initializations and picks the $\mathbf{W}^v$ matrices and $\mathbf{H}$ matrix with the lowest achieved loss.

Usage

```
GenerateWmatricesandHmatrix(
  SickleJr,
  d = 10,
  random = FALSE,
  numberReps = 100,
  seed = 5,
  minibatch = FALSE,
  batchsize = -1,
  random_W_updates = FALSE,
  subsample = 1:dim(SickleJr@count.matrices[[1]])[2],
  usesvd = FALSE
)
```

Arguments

SickleJr	An object of class SickleJr
d	Number of latent factors to use: defaults to 10
random	Boolean indicating whether to use random initialization (TRUE) or NNDSVD (FALSE): default is NNDSVD
numberReps	Number of random initializations to use: default is 5
seed	Random seed for reproducibility of random initializations
minibatch	Indicates whether or not to use the mini-batch algorithm
batchsize	Size of batches for mini-batch NMF
random_W_updates	Indicates whether to only update each $\mathbf{W}^v$ once per round of $\mathbf{H}$ updates; only appropriate for mini-batch algorithms
subsample	A vector of values to use for subsampling; only appropriate when determining proper values for d.
usesvd	Indicates whether to use R's singular value decomposition function svd (TRUE) or irlba (FALSE), default is FALSE; use irlba for larger datasets to increase performance

Value

SickleJr An object of class SickleJr with the $\mathbf{W}^v$ matrices and $\mathbf{H}$ matrix added.

References

Boutsidis C, Gallopoulos E (2008). "SVD based initialization: A head start for nonnegative matrix factorization." *Pattern Recognition*, **41**(4), 1350–1362. ISSN 00313203, doi:10.1016/J.PATCOG.2007.09.010.

Gaujoux R, Seoighe C (2010). "A flexible R package for nonnegative matrix factorization." *BMC Bioinformatics*, **11**(1), 1–9. ISSN 14712105, doi:10.1186/1471210511367/FIGURES/5, https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-367.

Examples

```
SimSickleJrSmall<-SetLambdasandRowReg(SimSickleJrSmall,
lambdaWlist=list(10,50),lambdaH=500,rowReg="None")
SimSickleJrSmall<-GenerateWmatricesandHmatrix(SimSickleJrSmall,d=5,usesvd=TRUE)
```

`jrSiCKLSNMF` *Run jrSiCKLSNMF outside of a SickleJr object*

Description

Perform joint non-negative matrix factorization (NMF) across multiple modalities of single-cell data. To measure the discrepancy between two distributions, one can use the Poisson Kullback-Leibler divergence (`diffFunc` ="`klp`") or the Frobenius norm (`diffFunc` ="`fr`"). It is also possible to set graph regularization constraints on $\mathbf{W}^v$ and either a sparsity constraint on $\mathbf{H}$ or an L2 norm constraint on the rows of $\mathbf{H}$. This function passes by reference and updates the variables `WL` and `H` and does not require data to be in an object of type SickleJr. `RunjrSiCKLSNMF` calls this function. If your data are in an object of class SickleJr, please use the `RunjrSiCKLSNMF` function instead.

Usage

```
jrSiCKLSNMF(
  datamatL,
  WL,
  H,
  AdjL,
  DL,
  lambdaWL,
  lambdaH,
  initsamp,
  suppress_warnings,
  diffFunc = "klp",
  Hconstraint = "None",
  differr = 1e-06,
```

```
  rounds = 1000L,
  display_progress = TRUE,
  minibatch = TRUE,
  batchsize = 100L,
  random_W_updates = TRUE,
  minrounds = 100L
)
```

Arguments

`datamatL`	An R list where each entry contains a normalized, sparse $\mathbf{X}^v$ matrix corresponding to single-cell modality v
`WL`	An R list containing initialized values of the $\mathbf{W}^v$ within each modality v
`H`	A matrix containing initialized values for the shared $\mathbf{H}$
`AdjL`	An R list containing all of the adjacency matrices for the feature-feature similarity graphs in sparse format; note that `D` $-$ `Adj` is the graph Laplacian
`DL`	An R list containing all of the degree matrices of the feature-feature similarity graphs; note that `D` $-$ `Adj` is the graph Laplacian
`lambdaWL`	A list of the $\lambda_{\mathbf{W}^v}$ corresponding to modality v
`lambdaH`	A double containing the desired value for $\lambda_{\mathbf{H}}$
`initsamp`	A vector of randomly selected rows of `H` on which to run the objective function
`suppress_warnings`	A Boolean that indicates whether warnings should be suppressed
`diffFunc`	A string indicating what type of divergence to use; set to the Poisson Kullback-Leibler divergence ("`klp`") by default, but the Frobenius norm ("`fr`") is also available
`Hconstraint`	A string that indicates whether you want to set an L2 norm constraint on the rows of $\mathbf{H}$. Enter 'None' for no constraints or 'L2Norm' to set the L2 norm of each row of $\mathbf{H}$ to 1
`differr`	A double containing the tolerance
`rounds`	A double containing the number of rounds
`display_progress`	A Boolean indicating whether to display the progress bar
`minibatch`	A Boolean indicating whether to use the mini-batch version of the algorithm
`batchsize`	Number of batches for mini-batch updates
`random_W_updates`	A Boolean indicating whether to update $\mathbf{W}^v$ once per epoch (TRUE) or after every update of the subset of $\mathbf{H}$ (FALSE) for the mini-batch algorithm.
`minrounds`	A minimum number of rounds for the algorithm to run: most useful for the mini-batch algorithm

Value

An R list containing values for the objective function.

References

Cai D, He X, Wu X, Han J (2008). "Non-negative matrix factorization on manifold." *Proceedings - IEEE International Conference on Data Mining, ICDM*, 63–72. ISSN 15504786, doi:10.1109/ICDM.2008.57.

Greene D, Cunningham P (2009). "A matrix factorization approach for integrating multiple data views." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **5781 LNAI**(PART 1), 423–438. ISSN 03029743, doi:10.1007/9783642041808_45/COVER, https://link.springer.com/chapter/10.1007/978-3-642-04180-8_45.

Eddelbuettel D, François R (2011). "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel D, Sanderson C (2014). "RcppArmadillo: Accelerating R with high-performance C++ linear algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. http://dx.doi.org/10.1016/j.csda.2013.02.005.

Elyanow R, Dumitrascu B, Engelhardt BE, Raphael BJ (2020). "NetNMF-SC: Leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis." *Genome Research*, **30**(2), 195–204. ISSN 15495469, doi:10.1101/gr.251603.119, https://pubmed.ncbi.nlm.nih.gov/31992614/.

Le Roux J, Weniger F, Hershey JR (2015). "Sparse NMF: half-baked or well done?" Mitsubishi Electric Research Laboratories (MERL), Cambridge.

Lee DD, Seung HS (2000). "Algorithms for Non-negative Matrix Factorization." In Leen T, Dietterich T, Tresp V (eds.), *Advances in Neural Information Processing Systems*, volume 13. https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

Liu J, Wang C, Gao J, Han J (2013). "Multi-view clustering via joint nonnegative matrix factorization." *Proceedings of the 2013 SIAM International Conference on Data Mining*, 252–260. doi:10.1137/1.9781611972832.28.

`MinibatchDiagnosticPlot` *Plot a diagnostic plot for the mini-batch algorithm*

Description

To ensure sufficient convergence of the loss for jrSiCKLSNMF with mini-batch updates, we plot the loss vs the number of iterations for the mini-batch algorithm. After a certain number of iterations, the loss should appear to oscillate around a value. Before continuing with downstream analyses, please ensure that the loss exhibits this sort of behavior. For the mini-batch algorithm, it is not possible to use the convergence criteria used for the batch version of the algorithm.

Usage

```
MinibatchDiagnosticPlot(SickleJr)
```

Arguments

SickleJr	An object of class SickleJr

Value

An object of class SickleJr with mini-batch diagnostic plots added to the plots slot.

Examples

```
SimSickleJrSmall<-MinibatchDiagnosticPlot(SimSickleJrSmall)
```

NormalizeCountMatrices

Normalize the count matrices and set whether to use the Poisson KL divergence or the Frobenius norm

---

Description

Normalize the count data within each modality. The default normalization, which should be used when using the KL divergence, is median library size normalization (Zheng et al. 2017; Elyanow et al. 2020). To perform median library size normalization, each count within a cell is divided by its library size (i.e. the counts within a column are divided by the column sum). Then, all values are multiplied by the median library size (i.e. the median column sum). To use the Frobenius norm, set frob=TRUE to $\log(x+1)$ normalize your count data and use a desired scaleFactor. You may also use a different form of normalization and store these results in the normalized.count.matrices slot.

Usage

```
NormalizeCountMatrices(SickleJr, diffFunc = "klp", scaleFactor = NULL)
```

Arguments

SickleJr	An object of class SickleJr
diffFunc	A string set to "klp" when using the Poisson KL divergence or to "fr" when using the Frobenius norm: default is KL divergence; this also determines the type of normalization
scaleFactor	A single numeric value (if using the same scale factor for each modality) or a list of numeric values to use (if using different scale factors in different modalities) as scale factors for the $\log(x+1)$ normalization when diffFunc="fr"

Value

An object of class SickleJr with a list of sparse, normalized data matrices added to its normalized.count.matrices slot

References

Elyanow R, Dumitrascu B, Engelhardt BE, Raphael BJ (2020). "NetNMF-SC: Leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis." *Genome Research*, **30**(2), 195–204. ISSN 15495469, doi:10.1101/gr.251603.119, https://pubmed.ncbi.nlm.nih.gov/31992614/.

Zheng GX, Terry JM, Belgrader P, Ryvkin P, Bent ZW, Wilson R, Ziraldo SB, Wheeler TD, McDermott GP, Zhu J, Gregory MT, Shuga J, Montesclaros L, Underwood JG, Masquelier DA, Nishimura SY, Schnall-Levin M, Wyatt PW, Hindson CM, Bharadwaj R, Wong A, Ness KD, Beppu LW, Deeg HJ, McFarland C, Loeb KR, Valente WJ, Ericson NG, Stevens EA, Radich JP, Mikkelsen TS, Hindson BJ, Bielas JH (2017). "Massively parallel digital transcriptional profiling of single cells." *Nature Communications*, **8**. ISSN 20411723, doi:10.1038/NCOMMS14049, https://pubmed.ncbi.nlm.nih.gov/28091601/.

Examples

```
SimSickleJrSmall<-NormalizeCountMatrices(SimSickleJrSmall)
SimSickleJrSmall<-NormalizeCountMatrices(SimSickleJrSmall, diffFunc="fr",scaleFactor=1e6)
```

`PlotLossvsLatentFactors` *Create plots to help determine the number of latent factors*

Description

Generate plots of the lowest achieved loss after a pre-specified number of iterations (default 100) for each latent factor (defaults to 2:20). This operates similarly to a scree plot, so please select a number of latent factors that corresponds to the elbow of the plot. This method is not appropriate for larger sets of data (more than 1000 cells)

Usage

```
PlotLossvsLatentFactors(
  SickleJr,
  rounds = 100,
  differr = 1e-04,
  d_vector = c(2:20),
  parallel = FALSE,
  nCores = detectCores() - 1,
  subsampsize = NULL,
  minibatch = FALSE,
  random = FALSE,
  random_W_updates = FALSE,
  seed = NULL,
  batchsize = -1,
  lossonsubset = FALSE,
  losssubsetsize = dim(SickleJr@count.matrices[[1]])[2]
)
```

Arguments

SickleJr	An object of class SickleJr
rounds	Number of rounds to use: defaults to 100; this process is time consuming, so a high number of rounds is not recommended
differr	Tolerance for the percentage update in the likelihood: for these plots, this defaults to $1e-4$
d_vector	Vector of d values to test: default is 2 to 20
parallel	Boolean indicating whether to use parallel computation
nCores	Number of desired cores; defaults to the number of cores of the current machine minus 1 for convenience
subsampsize	Size of the random subsample (defaults to NULL, which means all cells will be used); using a random subsample decreases computation time but sacrifices accuracy
minibatch	Boolean indicating whether to use the mini-batch algorithm: default is FALSE
random	Boolean indicating whether to use random initialization to generate the $\mathbf{W}^v$ matrices and $\mathbf{H}$ matrix: defaults to FALSE
random_W_updates	Boolean parameter for mini-batch algorithm; if TRUE, only updates $\mathbf{W}^v$ once per epoch on the penultimate subset of $\mathbf{H}$; otherwise updates $\mathbf{W}^v$ after every update of the subset of $\mathbf{H}$
seed	Number representing the random seed
batchsize	Desired batch size; do not use if using a subsample
lossonsubset	Boolean indicating whether to calculate the loss on a subset rather than the full dataset; speeds up computation for larger datasets
losssubsetsize	Number of cells to use for the loss subset; default is total number of cells

Value

An object of class SickleJr with a list of initialized $\mathbf{W}^v$ matrices and an $\mathbf{H}$ matrix for each latent factor $d \in \{1, ..., D\}$ added to the WHinitials slot, a data frame holding relevant values for plotting the elbow plot added to the latent.factor.elbow.values slot, diagnostic plots of the loss vs. the number of latent factors added to the plots slot, and the cell indices used to calculate the loss on the subsample added to the lossCalcSubSample slot

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*, 2 edition. Springer International Publishing, Cham, Switzerland. ISBN 978-3-319-24277-4, doi:10.1007/9783319242774, https://ggplot2.tidyverse.org/.

Examples

```
SimSickleJrSmall@latent.factor.elbow.values<-data.frame(NULL,NULL)
SimSickleJrSmall<-PlotLossvsLatentFactors(SimSickleJrSmall,d_vector=c(2:5),
rounds=5,parallel=FALSE)
```

```
#Next, we commute 2 of these in parallel.
## Not run:
SimSickleJrSmall<-PlotLossvsLatentFactors(SimSickleJrSmall,
d_vector=c(6:7),rounds=5,parallel=TRUE,nCores=2)
## End(Not run)
```

`PlotSickleJrUMAP` *Generate UMAP plots for an object of class SickleJr*

Description

Plot the first and second dimensions of a UMAP dimension reduction and color either by clustering results or metadata.

Usage

```
PlotSickleJrUMAP(
  SickleJr,
  umap.modality = "H",
  cluster = "kmeans",
  title = "",
  colorbymetadata = NULL,
  legendname = NULL
)
```

Arguments

`SickleJr`	An object of class SickleJr
`umap.modality`	String corresponding to the name of the UMAP of interest: defaults to `"H"`
`cluster`	String input that indicates which cluster to color by: defaults to `"kmeans"`
`title`	String input for optional `ggplot2{ggplot}` plot title
`colorbymetadata`	Name of metadata column if coloring by metadata
`legendname`	String input that to allow specification of a different legend name

Value

An object of class SickleJr with plots added to the `plots` slot

References

Wickham H (2016). *ggplot2: Elegant Graphics for Data Analysis*, 2 edition. Springer International Publishing, Cham, Switzerland. ISBN 978-3-319-24277-4, doi:10.1007/9783319242774, https://ggplot2.tidyverse.org/.

Examples

```
SimSickleJrSmall<-PlotSickleJrUMAP(SimSickleJrSmall,
title="K-Means Example")
SimSickleJrSmall<-PlotSickleJrUMAP(SimSickleJrSmall,umap.modality=1)
```

RunjrSiCKLSNMF *Run jrSiCKLSNMF on an object of class SickleJr*

Description

Wrapper function to run jrSiCKLSNMF on an object of class SickleJr. Performs jrSiCKLSNMF on the given SickleJr

Usage

```
RunjrSiCKLSNMF(
  SickleJr,
  rounds = 30000,
  differr = 1e-06,
  display_progress = TRUE,
  lossonsubset = FALSE,
  losssubsetsize = dim(SickleJr@H)[1],
  minibatch = FALSE,
  batchsize = 1000,
  random_W_updates = FALSE,
  seed = NULL,
  minrounds = 200,
  suppress_warnings = FALSE,
  subsample = 1:dim(SickleJr@normalized.count.matrices[[1]])[2]
)
```

Arguments

SickleJr	An object of class SickleJr
rounds	Number of rounds: defaults to 2000
differr	Tolerance for percentage change in loss between updates: defaults to 1e-6
display_progress	Boolean indicating whether to display the progress bar for jrSiCKLSNMF
lossonsubset	Boolean indicating whether to use a subset to calculate the loss function rather than the whole dataset
losssubsetsize	Size of the subset of data on which to calculate the loss
minibatch	Boolean indicating whether to use mini-batch updates
batchsize	Size of batch for mini-batch updates

`random_W_updates`
: Boolean indicating whether or not to use random_W_updates updates (i.e. only update $\mathbf{W}^v$ once per mini-batch epoch)

`seed`
: Number specifying desired random seed

`minrounds`
: Minimum number of rounds: most helpful for the mini-batch algorithm

`suppress_warnings`
: Boolean indicating whether to suppress warnings

`subsample`
: A numeric used primarily when finding an appropriate number of latent factors: defaults to total number of cells

Value

An object of class SickleJr with updated $\mathbf{W}^v$ matrices, updated $\mathbf{H}$ matrix, and a vector of values for the loss function added to the `Wlist`, `H`, and `loss` slots, respectively

References

Cai D, He X, Wu X, Han J (2008). "Non-negative matrix factorization on manifold." *Proceedings - IEEE International Conference on Data Mining, ICDM*, 63–72. ISSN 15504786, doi:10.1109/ICDM.2008.57.

Greene D, Cunningham P (2009). "A matrix factorization approach for integrating multiple data views." *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, **5781 LNAI**(PART 1), 423–438. ISSN 03029743, doi:10.1007/9783642041808_45/COVER, https://link.springer.com/chapter/10.1007/978-3-642-04180-8_45.

Eddelbuettel D, François R (2011). "Rcpp: Seamless R and C++ Integration." *Journal of Statistical Software*, **40**(8), 1–18. doi:10.18637/jss.v040.i08.

Eddelbuettel D, Sanderson C (2014). "RcppArmadillo: Accelerating R with high-performance C++ linear algebra." *Computational Statistics and Data Analysis*, **71**, 1054–1063. http://dx.doi.org/10.1016/j.csda.2013.02.005.

Elyanow R, Dumitrascu B, Engelhardt BE, Raphael BJ (2020). "NetNMF-SC: Leveraging gene-gene interactions for imputation and dimensionality reduction in single-cell expression analysis." *Genome Research*, **30**(2), 195–204. ISSN 15495469, doi:10.1101/gr.251603.119, https://pubmed.ncbi.nlm.nih.gov/31992614/.

Le Roux J, Weniger F, Hershey JR (2015). "Sparse NMF: half-baked or well done?" Mitsubishi Electric Research Laboratories (MERL), Cambridge.

Lee DD, Seung HS (2000). "Algorithms for Non-negative Matrix Factorization." In Leen T, Dietterich T, Tresp V (eds.), *Advances in Neural Information Processing Systems*, volume 13. https://proceedings.neurips.cc/paper/2000/file/f9d1152547c0bde01830b7e8bd60024c-Paper.pdf.

Liu J, Wang C, Gao J, Han J (2013). "Multi-view clustering via joint nonnegative matrix factorization." *Proceedings of the 2013 SIAM International Conference on Data Mining*, 252–260. doi:10.1137/1.9781611972832.28.

Examples

```
SimSickleJrSmall<-RunjrSiCKLSNMF(SimSickleJrSmall,rounds=5)
```

`SetLambdasandRowReg` *Set lambda values and type of row regularization for an object of class SickleJr*

Description

Provide the values for the graph regularization $\lambda_{\mathbf{W}^v}$ for each modality as a list, provide the value for the sparsity constraint $\lambda_{\mathbf{H}}$, and select whether to use L2 norm regularization.

Usage

```
SetLambdasandRowReg(
  SickleJr,
  lambdaWlist = list(10, 50),
  lambdaH = 500,
  rowReg = "None"
)
```

Arguments

`SickleJr`	An object of class SickleJr
`lambdaWlist`	A list of graph regularization constraints for the $\mathbf{W}^v$ matrices: defaults to 2 modalities with the RNA modality constraint equal to 10 and the ATAC modality constraint equal to 50
`lambdaH`	A numeric holding the sparsity constraint on $\mathbf{H}$: defaults to 500.
`rowReg`	A string that is equal to `"None"` for no constraints on the rows of $\mathbf{H}$ and `"L2Norm"` to set the L2 norms of the rows of $\mathbf{H}$ to be equal to 1: defaults to "None"

Value

An object of class SickleJr with the lambda hyperparameter values added to its `lambdaWlist` and `lambdaH` slots and with the row regularization value added to its `rowRegularization` slot

Examples

```
SimSickleJrSmall<-SetLambdasandRowReg(SimSickleJrSmall,
lambdaWlist=list(10,50),lambdaH=500,rowReg="None")
SimSickleJrSmall<-SetLambdasandRowReg(SimSickleJrSmall,
lambdaWlist=list(3,15),lambdaH=0,rowReg="L2Norm")
```

SetWandHfromWHinitials *Set* $\mathbf{W}$ *matrices and* $\mathbf{H}$ *matrix from pre-calculated values*

Description

Use values calculated in the step to determine number of latent factors in the initial steps for the jrSiCKLSNMF algorithm. If only a subset was calculated, this produces an error. In this case, please use `GenerateWmatricesandHmatrix` to generate new $\mathbf{W}^v$ matrices and a new $\mathbf{H}$ matrix.

Usage

```
SetWandHfromWHinitials(SickleJr, d)
```

Arguments

`SickleJr`	An object of class SickleJr
`d`	The number of desired latent factors

Value

An object of class SickleJr with the `Wlist` slot and the `H` slot filled from pre-calculated values.

Examples

```
SimSickleJrSmall<-SetWandHfromWHinitials(SimSickleJrSmall,d=5)
```

SickleJr-class *The SickleJr class*

Description

Defines the SickleJr class for use with jrSiCKLSNMF. This object contains all of the information required for analysis using jrSiCKLSNMF. This includes count matrices, normalized matrices, graph Laplacians, hyperparameters, diagnostic plots, and plots of cell clusters.

Value

An object of class SickleJr

Slots

`count.matrices` A list containing all of the quality controlled count matrices. Note that these count matrices should not use all features and should only include features that appear in at a minimum 10 cells.

`normalized.count.matrices` A list that holds the normalized count matrices

`graph.laplacian.list` A list of the graph Laplacians to be used for graph regularization

`rowRegularization` A string that indicates the type of row regularization to use. Types include "None" and "L2Norm"

`diffFunc` A string that holds the name of the function used to measure the discrepancy between data matrix X and WH for each modality; can be `"klp"` for the Poisson Kullback-Leibler divergence or `"fr"` for the Frobenius norm

`lambdaWlist` A list of lambda values to use as the hyperparameters for the corresponding $\mathbf{W}^v$ in the v^{th} modality

`lambdaH` A numeric value corresponding to the hyperparameter of the sparsity constraint on $\mathbf{H}$

`Wlist` A list of the generated $\mathbf{W}^v$ matrices, one for each modality

`H` The shared $\mathbf{H}$ matrix

`WHinitials` A list that if, when using `PlotLossvsLatentFactors`, all of the cells are used to calculate the initial values, stores these initial generated matrices; can be used as initializations when running `RunjrSiCKLSNMF` to save time

`lossCalcSubsample` A vector that holds the cell indices on which `PlotLossvsLatentFactors` was calculated

`latent.factor.elbow.values` A data frame that holds the relevant information to plot the latent factor elbow plot

`minibatch` Indicator variable that states whether the algorithm should use mini-batch updates.

`clusterdiagnostics` List of the cluster diagnostic results for the SickleJr object. Includes diagnostic plots from `fviz_nbclust` and and diagnostics from `clValid`

`clusters` List of results of different clustering methods performed on the SickleJr object

`metadata` List of metadata

`loss` Vector of the value for the loss function

`umap` List of different UMAP-based dimension reductions using `umap`

`plots` Holds various `ggplot` results for easy access of diagnostics and cluster visualizations

---

`SimData` *A simulated dataset for use with jrSiCKLSNMF*

Description

A simulated dataset with $\mathcal{U}(1, 1.25)$ multiplicative noise for the scRNA-seq variability parameter in SPARSim for the simulated scRNA-seq data and with $\mathcal{N}(-0.25, 0.25)$ additive noise to the expression levels of the scATAC-seq data for data simulated via SimATAC. The simulated matrices are located in SimData$Xmatrices and the identities for the cell types are contained in SimData$cell_type. This corresponds to the Xmatrix data found in both

XandLmatrices25/XandindividLKNNLmatrices1Sparsity5.RData and

XandBulkLmatrix25/XandBulkLKNNmatrices1Sparsity5.RData on our Github

ellisdoro/jrSiCKLSNMF_Simulations

Usage

```
data(SimData)
```

Format

A list made up of a two items. The first is list of 2 simulated sparse matrices and the second is a vector containing cell identities.

Xmatrices A list of 2 sparse matrices, each containing a different simulated omics modality measured on the same set of single cells: the first entry in the list corresponds to simulated scRNA-seq data and has 1000 genes and 300 cells; the second entry in the list corresponds to simulated scATAC-seq data and has 5910 peaks and 300 cells.

cell_type A vector containing the cell-type identities of the simulated data

Source

jrSicKLSNMF Simulations

`SimSickleJrSmall` *A small SickleJr object containing a subset of data from the* `SimData` *data object. Contains the completed analysis from the 'Getting Started' vignette for a small subset of 10 cells with 150 genes and 700 peaks. The clusters derived from this dataset are not accurate; this dataset is intended for use with code examples.*

Description

A small SickleJr object containing a subset of data from the `SimData` data object. Contains the completed analysis from the 'Getting Started' vignette for a small subset of 10 cells with 150 genes and 700 peaks. The clusters derived from this dataset are not accurate; this dataset is intended for use with code examples.

Usage

```
data(SimSickleJrSmall)
```

Format

A SickleJr object containing a completed analysis using jrSiCKLSNMF

count.matrices Contains a list of 2 sparse matrices, each containing a different simulated omics modality measured on the same set of single cells

normalized.count.matrices The normalized versions of the count matrices contained in slot `count.matrices`

graph.laplacian.list A list of sparse matrices containing the graph Laplacians corresponding to the KNN feature-feature similarity graphs constructed for each omics modality

rowRegularization A string indicating the row regularization: here it is set to `"None"`

diffFunc A string specifying the function to measure the discrepancy between the normalized data and the fitted matrices: here, it is set to "klp" for the Poisson Kullback-Leibler divergence

lambdaWlist A list holding the graph regularization parameters: here, they are 10 and 50

lambdaH A numeric indicating the value for the sparsity parameter. Here it is equals 500

Wlist A list holding the fitted $\mathbf{W}^v$ matrices

H A matrix holding $\mathbf{H}$

WHinitials A list of initial values for $\mathbf{W}^v$ and $\mathbf{H}$

lossCalcSubsample A vector containing a subset on which to calculate the loss

latent.factor.elbow.values A data frame holding the loss and the number of latent factor that is used for diagnostic plots

minibatch A Boolean indicating whether or not to use the mini-batch algorithm: `FALSE` here

clusterdiagnostics Diagnostic plots and results

clusters A list holding the `"kmeans"` clustering results

metadata A list holding metadata; here this is just cell type information

loss A list holding a vector called "Loss"

umap A list holding various UMAP approximations

plots A list holding ggplots corresponding to different diagnostics and visualizations

Source

jrSicKLSNMF Simulations

Index