Package 'densvis'

November 21, 2024

Title Density-Preserving Data Visualization via Non-Linear Dimensionality Reduction

Version 1.17.0

Date 2024-09-06

Description Implements the density-preserving modification to t-SNE and UMAP described by Narayan et al. (2020) <doi:10.1101/2020.05.12.077776>.
The non-linear dimensionality reduction techniques t-SNE and UMAP enable users to summarise complex high-dimensional sequencing data such as single cell RNAseq using lower dimensional representations.
These lower dimensional representations enable the visualisation of discrete transcriptional states, as well as continuous trajectory (for example, in early development). However, these methods focus on the local neighbourhood structure of the data. In some cases, this results in misleading visualisations, where the density of cells in the low-dimensional embedding does not represent the transcriptional heterogeneity of data in the original high-dimensional space. den-SNE and densMAP aim to enable more accurate visual interpretation of high-dimensional datasets by producing lower-dimensional embeddings that accurately represent the heterogeneity of the original high-dimensional space, enabling the identification of homogeneous and heterogeneous cell states.
This accuracy is accomplished by including in the optimisation process a term which considers the local density of points in the original high-dimensional space. This can help to create visualisations that are more representative of heterogeneity in the original high-dimensional space.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports Rcpp, basilisk, assertthat, reticulate, Rtsne, irlba

Suggests knitr, rmarkdown, BiocStyle, ggplot2, uwot, testthat

BugReports https://github.com/Alanocallaghan/densvis/issues

LinkingTo Rcpp

biocViews DimensionReduction, Visualization, Software, SingleCell, Sequencing

VignetteBuilder knitr

URL https://bioconductor.org/packages/densvis

StagedInstall no

git_url https://git.bioconductor.org/packages/densvis

git_branch devel

git_last_commit af1be7b

git_last_commit_date 2024-10-29

Repository Bioconductor 3.21

Date/Publication 2024-11-21

Author Alan O'Callaghan [aut, cre], Ashwinn Narayan [aut], Hyunghoon Cho [aut]

Maintainer Alan O'Callaghan <alan.ocallaghan@outlook.com>

Contents

---

densne *Density-preserving t-SNE*

Description

Density-preserving t-SNE

Usage

```
densne(
  X,
  dims = 2,
  perplexity = 50,
  theta = 0.5,
  check_duplicates = TRUE,
  pca = FALSE,
  initial_dims = 50,
  partial_pca = FALSE,
  pca_center = TRUE,
```

```
  pca_scale = FALSE,
  verbose = getOption("verbose", FALSE),
  max_iter = 1000,
  Y_init = NULL,
  stop_lying_iter = if (is.null(Y_init)) 250L else 0L,
  mom_switch_iter = if (is.null(Y_init)) 250L else 0L,
  momentum = 0.5,
  final_momentum = 0.8,
  eta = 200,
  exaggeration_factor = 12,
  dens_frac = 0.3,
  dens_lambda = 0.1,
  num_threads = 1,
  normalize = TRUE
)
```

Arguments

X	Input data matrix, where rows are observations and columns are features.
dims	Integer output dimensionality.
perplexity	Perplexity parameter (should not be bigger than 3 * perplexity < nrow(X) - 1).
theta	Speed/accuracy trade-off (increase for less accuracy), set to 0.0 for exact TSNE
check_duplicates	logical; Checks whether duplicates are present. It is best to make sure there are no duplicates present and set this option to FALSE, especially for large datasets (default: TRUE)
pca	logical; Whether an initial PCA step should be performed (default: FALSE).
initial_dims	integer; the number of dimensions that should be retained in the initial PCA step (default: 50)
partial_pca	logical; Whether truncated PCA should be used to calculate principal components (requires the irlba package). This is faster for large input matrices (default: FALSE)
pca_center, pca_scale	Controls whether to centre and scale the data before applying PCA. Defaults: TRUE, FALSE.
verbose	Logical; Whether progress updates should be printed
max_iter	integer; Number of iterations
Y_init	matrix; Initial locations of the objects. If NULL, random initialization will be used
stop_lying_iter	integer; Iteration after which the perplexities are no longer exaggerated
mom_switch_iter	integer; Iteration after which the final momentum is used
momentum	numeric; Momentum used in the first part of the optimization
final_momentum	numeric; Momentum used in the final part of the optimization

eta	numeric; Learning rate
exaggeration_factor	numeric; Exaggeration factor used to multiply the affinities matrix P in the first part of the optimization
dens_frac	numeric; fraction of the iterations for which the full objective function (including the density-preserving term) is used. For the first 1 - dens_frac fraction of the iterations, only the original t-SNE objective function is used.
dens_lambda	numeric; the relative importanceof the density-preservation term compared to the original t-SNE objective function.
num_threads	Number of threads to be used for parallelisation.
normalize	logical; Should data be normalized internally prior to distance calculations with normalize_input?

Value

A numeric matrix corresponding to the t-SNE embedding

References

Density-Preserving Data Visualization Unveils Dynamic Patterns of Single-Cell Transcriptomic Variability Ashwin Narayan, Bonnie Berger, Hyunghoon Cho; bioRxiv (2020) doi:10.1101/2020.05.12.077776

Examples

```
x <- matrix(rnorm(1e3), nrow = 100)
d <- densne(x, perplexity = 5)
plot(d)
```

---

umap *Density-preserving and other implementations of UMAP*

Description

Density-preserving and other implementations of UMAP

Usage

```
umap(
  x,
  n_components = 2L,
  dens_frac = 0.3,
  dens_lambda = 0.1,
  dens_var_shift = 0.1,
  n_neighbors = 30L,
  metric = "euclidean",
```

```
  densmap = FALSE,
  n_epochs = 750L,
  learning_rate = 1,
  init = c("spectral", "random"),
  Y_init = NULL,
  min_dist = 0.1,
  spread = 1,
  low_memory = FALSE,
  set_op_mix_ratio = 1,
  local_connectivity = 1L,
  repulsion_strength = 1,
  negative_sample_rate = 5L,
  transform_queue_size = 4,
  random_state = NULL,
  angular_rp_forest = FALSE,
  target_n_neighbors = -1,
  target_weight = 0.5,
  disconnection_distance = NULL
)

densmap(...)
```

Arguments

x	A numeric matrix or matrix-like object.
n_components	The dimension of the space to embed into. This defaults to 2 to provide easy visualization, but can reasonably be set to any integer value in the range 2 to 100.
dens_frac	numeric; fraction of the iterations for which the full objective function (including the density-preserving term) is used. For the first `1 - dens_frac` fraction of the iterations, only the original t-SNE objective function is used. Only takes effect when `densmap=TRUE`.
dens_lambda	numeric; the relative importance of the density-preservation term compared to the original t-SNE objective function. Only takes effect when `densmap=TRUE`.
dens_var_shift	Regularization term added to the variance of embedding local radius for stability (float, non-negative); default 0.1. Only takes effect when `densmap=TRUE`.
n_neighbors	The size of local neighborhood (in terms of number of neighboring sample points) used for manifold approximation. Larger values result in more global views of the manifold, while smaller values result in more local data being preserved. In general values should be in the range 2 to 100.
metric	The metric to use to compute distances in high dimensional space. If a string is passed it must match one of: • "euclidean" • "manhattan" • "chebyshev" • "minkowski"

- "canberra"
- "braycurtis"
- "mahalanobis"
- "wminkowski"
- "seuclidean"
- "cosine"
- "correlation"
- "haversine"
- "hamming"
- "jaccard"
- "dice"
- "russelrao"
- "kulsinski"
- "rogerstanimoto"
- "sokalmichener"
- "sokalsneath"
- "yule"

`densmap` For `umap`, control whether the density-preserving UMAP algorithm described by Narayan et al. is used.

`n_epochs` The number of training epochs to be used in optimizing the low dimensional embedding. Larger values result in more accurate embeddings. If None is specified a value will be selected based on the size of the input dataset (200 for large datasets, 500 for small). a valid predefined metric.

`learning_rate` The initial learning rate for the embedding optimization.

`init` How to initialize the low dimensional embedding. Valid options:

- "spectral": use a spectral embedding of the fuzzy 1-skeleton
- "random": assign initial embedding positions at random.

`Y_init` Numeric matrix specifying the initial locations of the objects in the embedding. If NULL, random or spectral initialization will be used, controlled by the `init` argument.

`min_dist` The effective minimum distance between embedded points. Smaller values will result in a more clustered/clumped embedding where nearby points on the manifold are drawn closer together, while larger values will result on a more even dispersal of points. The value should be set relative to the spread value, which determines the scale at which embedded points will be spread out.

`spread` The effective scale of embedded points. In combination with min_dist this determines how clustered/clumped the embedded points are.

`low_memory` For some datasets the nearest neighbor computation can consume a lot of memory. If you find that UMAP is failing due to memory constraints consider setting this option to True. This approach is more computationally expensive, but avoids excessive memory use.

`set_op_mix_ratio`
: Interpolate between (fuzzy) union and intersection as the set operation used to combine local fuzzy simplicial sets to obtain a global fuzzy simplicial sets. Both fuzzy set operations use the product t-norm. The value of this parameter should be between 0.0 and 1.0; a value of 1.0 will use a pure fuzzy union, while 0.0 will use a pure fuzzy intersection.

`local_connectivity`
: The local connectivity required – i.e. the number of nearest neighbors that should be assumed to be connected at a local level. The higher this value the more connected the manifold becomes locally. In practice this should be not more than the local intrinsic dimension of the manifold.

`repulsion_strength`
: Weighting applied to negative samples in low dimensional embedding optimization. Values higher than one will result in greater weight being given to negative samples.

`negative_sample_rate`
: The number of negative samples to select per positive sample in the optimization process. Increasing this value will result in greater repulsive force being applied, greater optimization cost, but slightly more accuracy.

`transform_queue_size`
: For transform operations (embedding new points using a trained model_ this will control how aggressively to search for nearest neighbors. Larger values will result in slower performance but more accurate nearest neighbor evaluation.

`random_state`
: The seed used by the random number generator.

`angular_rp_forest`
: Whether to use an angular random projection forest to initialise the approximate nearest neighbor search. This can be faster, but is mostly on useful for metric that use an angular style distance such as cosine, correlation etc. In the case of those metrics angular forests will be chosen automatically.

`target_n_neighbors`
: The number of nearest neighbors to use to construct the target simplcial set. If set to -1 use the n_neighbors value.

`target_weight`
: Weighting factor between data topology and target topology. A value of 0.0 weights entirely on data, a value of 1.0 weights entirely on target. The default of 0.5 balances the weighting equally between data and target.

`disconnection_distance`
: Numeric scalar. If specified, UMAP will disconnect any vertices of distance greater than or equal to disconnection_distance when approximating the manifold via our k-nn graph. This is particularly useful in the case that you have a bounded metric. The UMAP assumption that we have a connected manifold can be problematic when you have points that are maximally different from all the rest of your data. The connected manifold assumption will make such points have perfect similarity to a random set of other points. Too many such points will artificially connect your space.

`...`
: Passed from `densmap` to `umap`.

Value

A numeric matrix

References

Density-Preserving Data Visualization Unveils Dynamic Patterns of Single-Cell Transcriptomic Variability Ashwin Narayan, Bonnie Berger, Hyunghoon Cho; bioRxiv (2020) doi:10.1101/2020.05.12.077776

Examples

```
set.seed(42)
x <- matrix(rnorm(200), ncol=2)
densmap(x)
```

Index