

scanpy.tl.dendrogram

scanpy.tl.dendrogram(adata, groupby, n_pcs=None, use_rep=None, var_names=None, use_raw=None, cor_method='pearson', linkage_method='complete', optimal_ordering=False, key_added=None, inplace=True)

Computes a hierarchical clustering for the given `groupby` categories.

By default, the PCA representation is used unless `.x` has less than 50 variables.

Alternatively, a list of `var_names` (e.g. genes) can be given.

Average values of either `var_names` or components are used to compute a correlation matrix.

The hierarchical clustering can be visualized using `scanpy.pl.dendrogram()` or multiple other visualizations that can include a dendrogram: `matrixplot()`, `heatmap()`, `dotplot()`, and `stacked_violin()`.

! Note

The computation of the hierarchical clustering is based on predefined groups and not per cell. The correlation matrix is computed using by default pearson but other methods are available.

Parameters: **adata :** `AnnData`

Annotated data matrix

n_pcs : `optional [int]` (default: `None`)

Use this many PCs. If `n_pcs==0` use `.x` if `use_rep` is `None`.

use_rep : `optional [str]` (default: `None`)

Use the indicated representation. `'x'` or any key for `.obs` is valid. If `None`, the representation is chosen automatically: For `.n_vars < 50`, `.x` is used, otherwise `'X_pca'` is used. If `'X_pca'` is not present, it's computed with default parameters.

var_names : `optional [Sequence [str]]` (default: `None`)

List of `var_names` to use for computing the hierarchical clustering. If `var_names` is given, then `use_rep` and `n_pcs` is ignored.

use_raw : `Optional[bool]` (default: `None`)

Only when `var_names` is not `None`. Use `raw` attribute of `adata` if present.

cor_method : `str` (default: `'pearson'`)

correlation method to use. Options are 'pearson', 'kendall', and 'spearman'

linkage_method : `str` (default: `'complete'`)

linkage method to use. See `scipy.cluster.hierarchy.linkage()` for more information.

optimal_ordering : `bool` (default: `False`)

Same as the `optimal_ordering` argument of `scipy.cluster.hierarchy.linkage()` which reorders the linkage matrix so that the distance between successive leaves is minimal.

key_added : `Optional[str]` (default: `None`)

By default, the dendrogram information is added to `.uns[f'dendrogram_{groupby}']`. Notice that the `groupby` information is added to the dendrogram.

inplace : `bool` (default: `True`)

If `True`, adds dendrogram information to `adata.uns[key_added]`, else this function returns the information.

Return type:

`Optional[Dict[str, Any]]`

Returns:

: If `inplace=False`, returns dendrogram information, else `adata.uns[key_added]` is updated with it.

Examples

```
>>> import scanpy as sc
>>> adata = sc.datasets.pbmc68k_reduced()
>>> sc.tl.dendrogram(adata, groupby='bulk_labels')
>>> sc.pl.dendrogram(adata)
>>> markers = ['C1QA', 'PSAP', 'CD79A', 'CD79B', 'CST3', 'LYZ']
>>> sc.pl.dotplot(adata, markers, groupby='bulk_labels', dendrogram=True)
```