scanpy.tl.louvain

scanpy.tl.louvain(*adata, resolution=None, random_state=0, restrict_to=None, key_added='louvain', adjacency=None, flavor='vtraag', directed=True, use_weights=False, partition_type=None, partition_kwargs=mappingproxy({}), neighbors_key=None, obsp=None, copy=False*)

Cluster cells into subgroups [Blondel08] [Levine15] [Traag17].

Cluster cells using the Louvain algorithm [Blondel08] in the implementation of [Traag17]. The Louvain algorithm has been proposed for single-cell analysis by [Levine15].

This requires having ran `neighbors()` or `bbknn()` first, or explicitly passing a `adjacency` matrix.

Parameters:

adata : `AnnData`

The annotated data matrix.

resolution : `Optional` [`float`] (default: `None`)

For the default flavor (`'vtraag'`) or for `` `RAPIDS` ``, you can provide a resolution (higher resolution means finding more and smaller clusters), which defaults to 1.0. See “Time as a resolution parameter” in [Lambiotte09].

random_state : `Union` [`None` , `int` , `RandomState`] (default: `0`)

Change the initialization of the optimization.

restrict_to : `Optional` [`Tuple` [`str` , `Sequence` [`str`]]] (default: `None`)

Restrict the clustering to the categories within the key for sample annotation, tuple needs to contain `(obs_key, list_of_categories)` .

key_added : `str` (default: `'louvain'`)

Key under which to add the cluster labels. (default: `'louvain'`)

adjacency : `Optional` [`spmatrix`] (default: `None`)

Sparse adjacency matrix of the graph, defaults to neighbors connectivities.

flavor : `Literal` [`'vtraag'` , `'igraph'` , `'rapids'`] (default: `'vtraag'`)

Choose between to packages for computing the clustering. `'vtraag'` is much more powerful, and the default.

directed : `bool` (default: `True`)

Interpret the `adjacency` matrix as directed graph?

use_weights : `bool` (default: `False`)

Use weights from knn graph.

partition_type : `Optional` [`Type` [`MutableVertexPartition`]] (default: `None`)

Type of partition to use. Only a valid argument if `flavor` is `'vtraag'` .

partition_kwargs : `Mapping` [`str` , `Any`] (default: `mappingproxy({})`)

Key word arguments to pass to partitioning, if `vtraag` method is being used.

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

Use neighbors connectivities as adjacency. If not specified, louvain looks .obsp[‘connectivities’] for connectivities (default storage place for pp.neighbors). If specified, louvain looks .obsp[.uns[neighbors_key][‘connectivities_key’]] for connectivities.

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

Use .obsp[obsp] as adjacency. You can’t specify both `obsp` and `neighbors_key` at the same time.

copy : `bool` (default: `False`)

Copy adata or modify it inplace.

Return type: `Optional` [`AnnData`]

Returns:

: `None`

By default (`copy=False`), updates `adata` with the following fields:

`adata.obs['louvain']` (**`pandas.Series`**, dtype `category`)

Array of dim (number of samples) that stores the subgroup id (`'0'`, `'1'`, ...) for each cell.

`AnnData`

When `copy=True` is set, a copy of `adata` with those fields is returned.