scanpy.tl.leiden

scanpy.tl.leiden(*adata*, *resolution=1*, ***, *restrict_to=None*, *random_state=0*, *key_added='leiden'*, *adjacency=None*, *directed=True*, *use_weights=True*, *n_iterations=-1*, *partition_type=None*, *neighbors_key=None*, *obsp=None*, *copy=False*, ***partition_kwargs*)

Cluster cells into subgroups [Traag18].

Cluster cells using the Leiden algorithm [Traag18], an improved version of the Louvain algorithm [Blondel08]. It has been proposed for single-cell analysis by [Levine15].

This requires having ran `neighbors()` or `bbknn()` first.

Parameters:

adata : `AnnData`

The annotated data matrix.

resolution : `float` (default: `1`)

A parameter value controlling the coarseness of the clustering. Higher values lead to more clusters. Set to `None` if overriding `partition_type` to one that doesn't accept a `resolution_parameter`.

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: `'leiden'`)

`adata.obs` key under which to add the cluster labels.

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

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

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

Whether to treat the graph as directed or undirected.

use_weights : `bool` (default: `True`)

If `True`, edge weights from the graph are used in the computation (placing more emphasis on stronger edges).

n_iterations : `int` (default: `-1`)

How many iterations of the Leiden clustering algorithm to perform. Positive values above 2 define the total number of iterations to perform, -1 has the algorithm run until it reaches its optimal clustering.

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

Type of partition to use. Defaults to `RBConfigurationVertexPartition`. For the available options, consult the documentation for `find_partition()`.

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

Use neighbors connectivities as adjacency. If not specified, leiden looks .obsp[‘connectivities’] for connectivities (default storage place for pp.neighbors). If specified, leiden 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`)

Whether to copy `adata` or modify it inplace.

****partition_kwargs**

Any further arguments to pass to `~leidenalg.find_partition` (which in turn passes arguments to the `partition_type`).

Return type: `Optional`[`AnnData`]

Returns:

: `adata.obs[key_added]`

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

`adata.uns['leiden']['params']`

A dict with the values for the parameters `resolution`, `random_state`, and `n_iterations`.