scanpy.tl.draw_graph

scanpy.tl.draw_graph(*adata, layout='fa', init_pos=None, root=None, random_state=0, n_jobs=None, adjacency=None, key_added_ext=None, neighbors_key=None, obsp=None, copy=False, **kwds*)

Force-directed graph drawing [Islam11] [Jacomy14] [Chippada18].

An alternative to tSNE that often preserves the topology of the data better. This requires to run `neighbors()`, first.

The default layout ('fa', `ForceAtlas2`) [Jacomy14] uses the package `fa2` [Chippada18], which can be installed via `pip install fa2`.

Force-directed graph drawing describes a class of long-established algorithms for visualizing graphs. It has been suggested for visualizing single-cell data by [Islam11]. Many other layouts as implemented in igraph [Csardi06] are available. Similar approaches have been used by [Zunder15] or [Weinreb17].

Parameters:

adata : `AnnData`

Annotated data matrix.

layout : `Literal` [`'fr'` , `'drl'` , `'kk'` , `'grid_fr'` , `'lgl'` , `'rt'` , `'rt_circular'` , `'fa'`] (default: `'fa'`)

'fa' (`ForceAtlas2`) or any valid igraph layout. Of particular interest are 'fr' (Fruchterman Reingold), 'grid_fr' (Grid Fruchterman Reingold, faster than 'fr'), 'kk' (Kamadi Kawai', slower than 'fr'), 'lgl' (Large Graph, very fast), 'drl' (Distributed Recursive Layout, pretty fast) and 'rt' (Reingold Tilford tree layout).

root : `Optional` [`int`] (default: `None`)

Root for tree layouts.

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

For layouts with random initialization like 'fr', change this to use different intial states for the optimization. If `None`, no seed is set.

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

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

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

By default, append `layout`.

proceed

Continue computation, starting off with ‘X_draw_graph_`layout`’.

init_pos : `Union` [`str` , `bool` , `None`] (default: `None`)

`'paga'` / `True` , `None` / `False` , or any valid 2d-`.obsm` key. Use precomputed coordinates for initialization. If `False` / `None` (the default), initialize randomly.

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

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

Return a copy instead of writing to adata.

****kwds**

Parameters of chosen igraph layout. See e.g. fruchterman-reingold [Fruchterman91]. One of the most important ones is `maxiter`.

Returns:

: Depending on `copy` , returns or updates `adata` with the following field.

X_draw_graph_layout : `adata.obsm`

Coordinates of graph layout. E.g. for layout=’fa’ (the default), the field is called ‘X_draw_graph_fa’