

scanpy.tl.embedding_density

scanpy.tl.embedding_density(adata, basis='umap', groupby=None, key_added=None, components=None)

Calculate the density of cells in an embedding (per condition).

Gaussian kernel density estimation is used to calculate the density of cells in an embedded space. This can be performed per category over a categorical cell annotation. The cell density can be plotted using the `pl.embedding_density` function.

Note that density values are scaled to be between 0 and 1. Thus, the density value at each cell is only comparable to densities in the same category.

Beware that the KDE estimate used (`scipy.stats.gaussian_kde`) becomes unreliable if you don't have enough cells in a category.

This function was written by Sophie Tritschler and implemented into Scanpy by Malte Luecken.

Parameters:

adata : `AnnData`

The annotated data matrix.

basis : `str` (default: `'umap'`)

The embedding over which the density will be calculated. This embedded representation should be found in

`adata.obsm['X_[basis]']`.

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

Key for categorical observation/cell annotation for which densities are calculated per category.

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

Name of the `.obs` covariate that will be added with the density estimates.

components : `Union [str, Sequence [str], None]` (default: `None`)

The embedding dimensions over which the density should be calculated. This is limited to two components.

Return type:

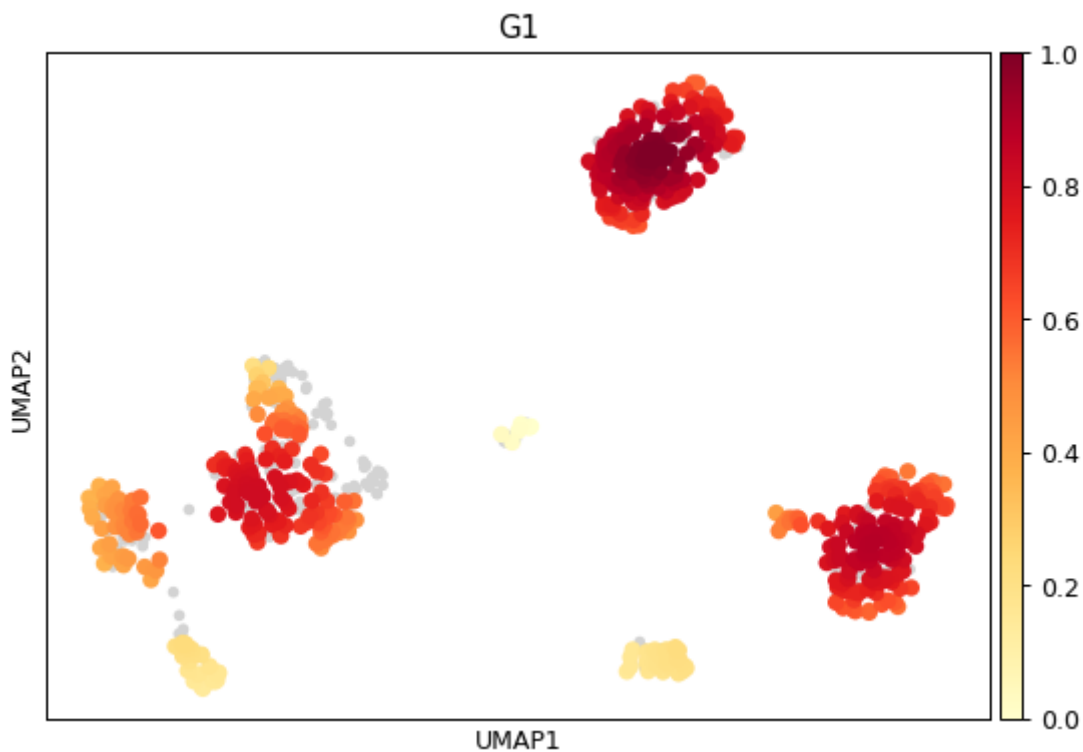
None

Returns:

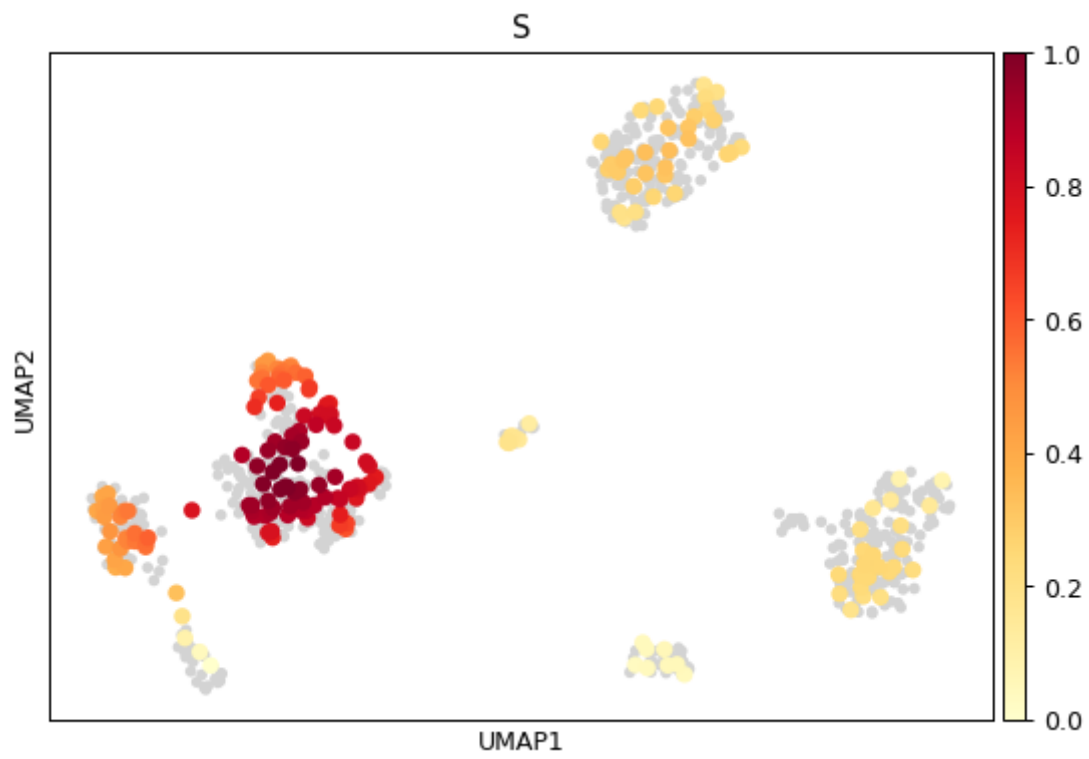
: Updates `adata.obs` with an additional field specified by the `key_added` parameter. This parameter defaults to `[basis]_density_[groupby]`, where `[basis]` is one of `umap`, `diffmap`, `pca`, `tsne`, or `draw_graph_fa` and `[groupby]` denotes the parameter input. Updates `adata.uns` with an additional field `[key_added]_params`.

Examples

```
import scanpy as sc
adata = sc.datasets.pbmc68k_reduced()
sc.tl.umap(adata)
sc.tl.embedding_density(adata, basis='umap', groupby='phase')
sc.pl.embedding_density(
    adata, basis='umap', key='umap_density_phase', group='G1'
)
```



```
sc.pl.embedding_density(
    adata, basis='umap', key='umap_density_phase', group='S'
)
```



! See also

`pl.embedding_density`