scanpy.tl.marker_gene_overlap

scanpy.tl.marker_gene_overlap(*adata*, *reference_markers*, ***, *key='rank_genes_groups'*, *method='overlap_count'*, *normalize=None*, *top_n_markers=None*, *adj_pval_threshold=None*, *key_added='marker_gene_overlap'*, *inplace=False*)

Calculate an overlap score between data-deriven marker genes and provided markers

Marker gene overlap scores can be quoted as overlap counts, overlap coefficients, or jaccard indices. The method returns a pandas dataframe which can be used to annotate clusters based on marker gene overlaps.

This function was written by Malte Luecken.

Parameters:

adata : `AnnData`

The annotated data matrix.

reference_markers : `Union` [`Dict` [`str` , `set`], `Dict` [`str` , `list`]]

A marker gene dictionary object. Keys should be strings with the cell identity name and values are sets or lists of strings which match format of `adata.var_name` .

key : `str` (default: `'rank_genes_groups'`)

The key in `adata.uns` where the rank_genes_groups output is stored. By default this is `'rank_genes_groups'` .

method : `Literal` [`'overlap_count'` , `'overlap_coef'` , `'jaccard'`] (default: `'overlap_count'`)

(default: `overlap_count`) Method to calculate marker gene overlap. `'overlap_count'` uses the intersection of the gene set, `'overlap_coef'` uses the overlap coefficient, and `'jaccard'` uses the Jaccard index.

normalize : `Optional` [`Literal` [`'reference'` , `'data'`]] (default: `None`)

Normalization option for the marker gene overlap output. This parameter can only be set when `method` is set to `'overlap_count'` . `'reference'` normalizes the data by the total number of marker

genes given in the reference annotation per group. `'data'` normalizes the data by the total number of marker genes used for each cluster.

---

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

The number of top data-derived marker genes to use. By default the top 100 marker genes are used. If `adj_pval_threshold` is set along with `top_n_markers`, then `adj_pval_threshold` is ignored.

---

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

A significance threshold on the adjusted p-values to select marker genes. This can only be used when adjusted p-values are calculated by `sc.tl.rank_genes_groups()`. If `adj_pval_threshold` is set along with `top_n_markers`, then `adj_pval_threshold` is ignored.

---

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

Name of the `.uns` field that will contain the marker overlap scores.

---

inplace : `bool` (default: `False`)

Return a marker gene dataframe or store it inplace in `adata.uns`.

Returns:

: A pandas dataframe with the marker gene overlap scores if `inplace=False`. For `inplace=True` `adata.uns` is updated with an additional field specified by the `key_added` parameter (default = 'marker_gene_overlap').

Examples

```
>>> import scanpy as sc
>>> adata = sc.datasets.pbmc68k_reduced()
>>> sc.pp.pca(adata, svd_solver='arpack')
>>> sc.pp.neighbors(adata)
>>> sc.tl.louvain(adata)
>>> sc.tl.rank_genes_groups(adata, groupby='louvain')
>>> marker_genes = {
...     'CD4 T cells': {'IL7R'},
...     'CD14+ Monocytes': {'CD14', 'LYZ'},
...     'B cells': {'MS4A1'},
...     'CD8 T cells': {'CD8A'},
...     'NK cells': {'GNLY', 'NKG7'},
...     'FCGR3A+ Monocytes': {'FCGR3A', 'MS4A7'},
...     'Dendritic Cells': {'FCER1A', 'CST3'},
...     'Megakaryocytes': {'PPBP'}
... }
>>> marker_matches = sc.tl.marker_gene_overlap(adata, marker_genes)
```