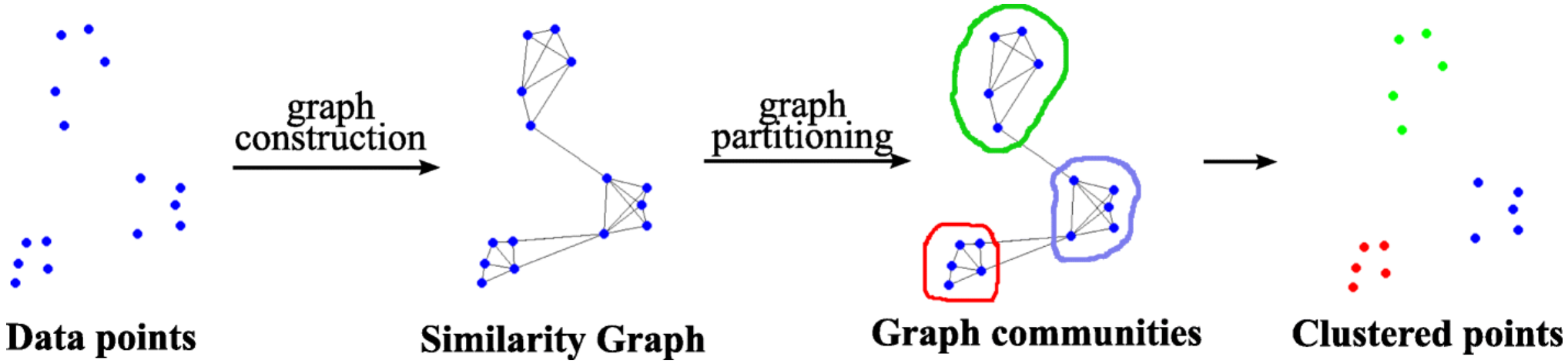
Clustering methods - Graph-based clustering

- Input data is organized into groups (clusters) according to similarity



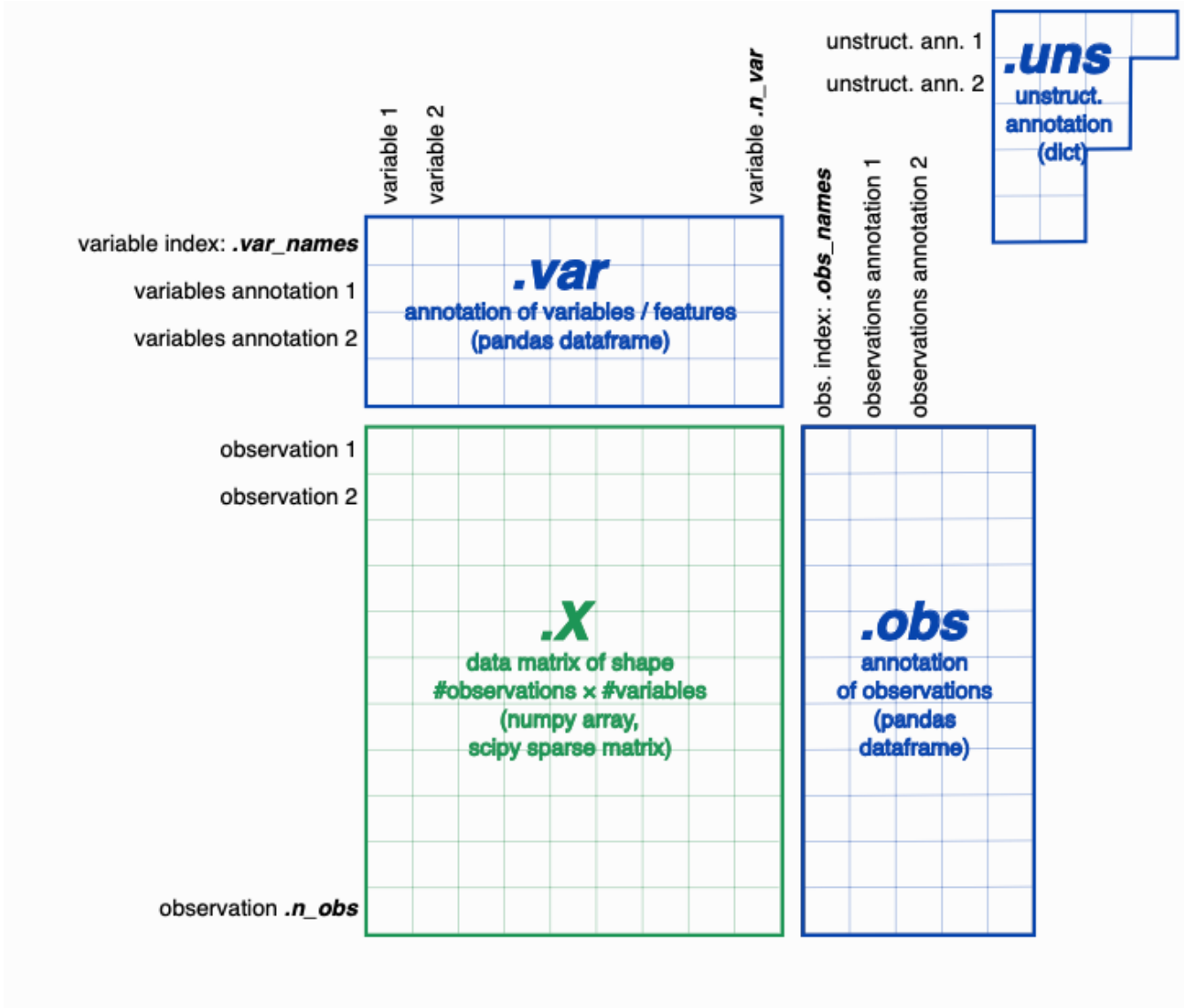
https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0248-7

Example of functions for clustering and graph communities using different data structures:

- *SCE: scran::buildSNNGraph and igraph::cluster_louvain (tutorial)*
- *Seurat: Seurat::FindClusters and Louvain clustering*
- *Anndata: scanpy.pp.neighbors and Leiden clustering*

Example of non-linear, graph-based clustering algorithms: t-SNE and UMAP

Conversion from SCE to anndata



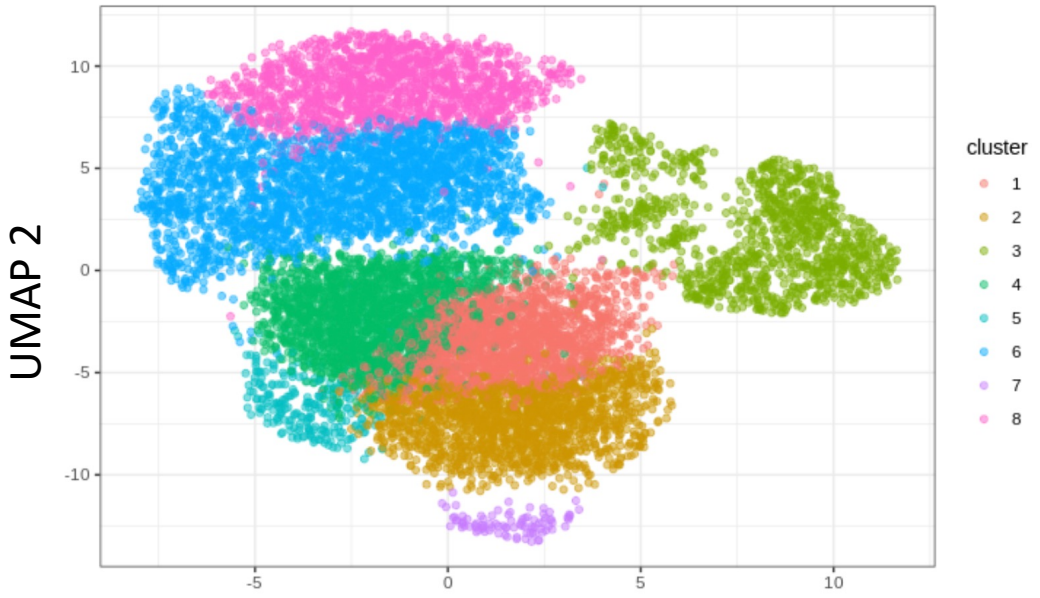
1. Rows -> cells (obs), Cols -> rows (var)! (*transpose the matrix*)
2. Metadata of cells -> obs
3. Metadata of genes -> var
4. Anndata.layer [‘counts’] = raw counts
5. Anndata is good for storing large datasets and for python tools (e.g. Celltypist)

Clustering methods - Graph-based clustering

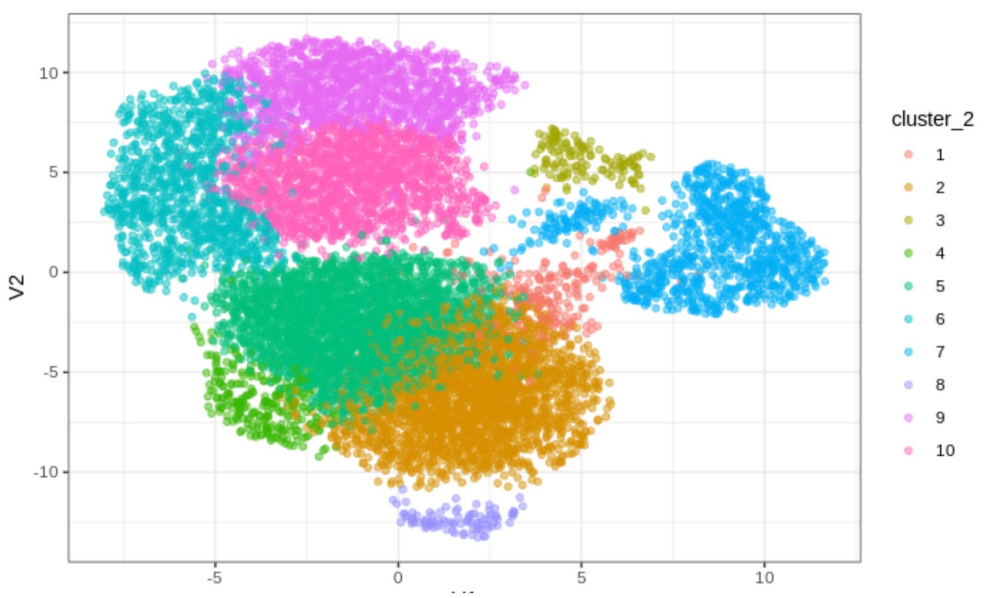
A graph-based clustering method has several key parameters (SCE):

- How many neighbors are considered when constructing the graph – **k**.
- What scheme is used to weight the edges – number and jaccard.
- Which community detection algorithm is used to define the clusters -
  e.g. walktrap, **louvain**, infomap, fast_greedy, label_prop, leading_eigen.

k = 25



k = 10



```
In [27]: adata.obs.head(10)
         adata.var.head(5)
```

Out[27]:

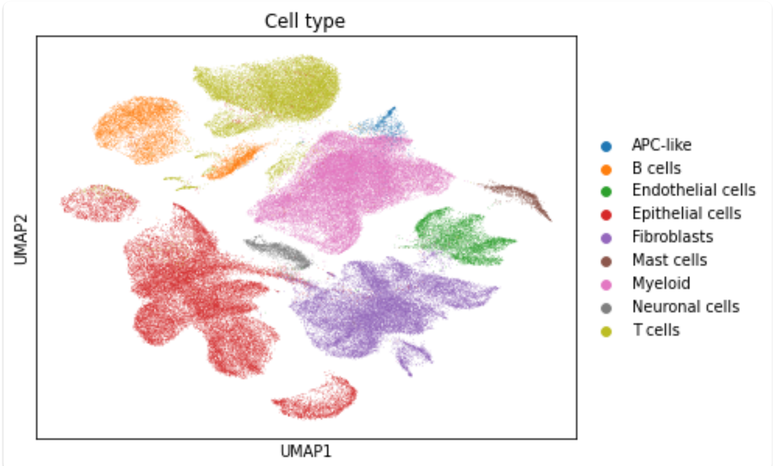
	n_cells_by_counts	mean_counts	pct_dropout_by_counts	total_counts	n_cells	highly_variable	means	dispersions	dispersions_norm	mean
ISG15	1575	0.017943	98.645895	2087.0	1575	True	0.091402	2.709469	1.100097	0.024371
TNFRSF18	308	0.002915	99.735197	339.0	308	True	0.028312	2.697045	1.050803	0.006090
TNFRSF4	268	0.002467	99.769587	287.0	268	True	0.023890	2.748348	1.254348	0.005108
CFAP74	1272	0.015811	98.906399	1839.0	1272	True	0.081199	2.561168	0.511711	0.021531
PLCH2	479	0.004307	99.588180	501.0	479	True	0.037998	2.630530	0.786903	0.008724

Metadata of cells -> obs
Metadata of genes -> var

```
In [20]: adata.obs[['Cell type']].head(23)
```

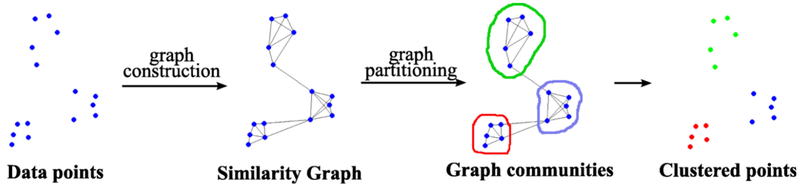
Out[20]:

	Cell type
TAGACTGAGGCATGCA-1_6	Epithelial cells
TCATTTGGTCACCGCA-1_6	Epithelial cells
GACCTTCAGGACTTCT-1_6	Epithelial cells
ACACGCGAGTACAACA-1_6	Epithelial cells
TTCTAACTCGGTATGT-1_6	Epithelial cells
TGCTCCAAGTGCCGAA-1_6	Epithelial cells
AAGCGTTCAGAGAATT-1_6	T cells
ATATCCTTCCGGCTTT-1_6	T cells
GACACGCCACATATGC-1_6	Epithelial cells
GATCAGTGTTGAAGTA-1_6	Epithelial cells
CAGGTATCACCCTAGG-1_6	Epithelial cells
TGACTCCTCCCGAGTG-1_6	Epithelial cells



Main ideas of UMAP

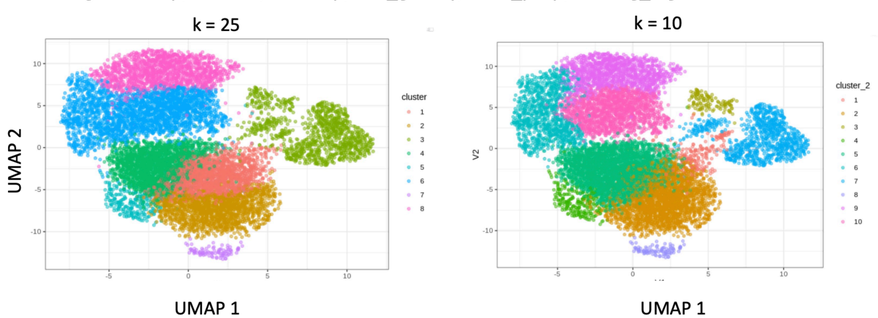
- Input data is organized into groups (clusters) according to similarity



https://appliednetsci.springeropen.com/articles/10.1007/s41109-019-0248-7

A graph-based clustering method has several key parameters (SCE):

- How many neighbors are considered when constructing the graph – **k**.
- What scheme is used to weight the edges – number and jaccard.
- Which community detection algorithm is used to define the clusters -
  e.g. walktrap, **louvain**, infomap, fast_greedy, label_prop, leading_eigen.




```
In [25]: import pandas as pd

         # Extract UMAP coordinates and cell type annotations
         umap_coords = adata.obsm['X_umap']
         cell_types = adata.obs['Cell type']

         df = pd.DataFrame({
             'UMAP1': umap_coords[:, 0], # UMAP1 (X-axis)
             'UMAP2': umap_coords[:, 1], # UMAP1 Y-axis)
             'Cell_Type': cell_types
         })

         print(df)
```

```
                          UMAP1      UMAP2         Cell_Type
TAGACTGAGGCATGCA-1_6  -3.397396  11.426294  Epithelial cells
TCATTTGGTCACCGCA-1_6  -4.017027  11.315504  Epithelial cells
GACCTTCAGGACTTCT-1_6  -3.180807  11.725115  Epithelial cells
ACACGCGAGTACAACA-1_6  -2.980008  11.802874  Epithelial cells
TTCTAACTCGGTATGT-1_6  -3.828823  10.371683  Epithelial cells
...                         ...        ...              ...
TGAATGCGTTATGGTC-1_16 10.921441  12.409225          Myeloid
TACGGGCTCATACAGC-1_16  4.223805   6.490491  Epithelial cells
CTATAGGAGTGATAGT-1_16  4.850944   6.119487  Epithelial cells
AAGTCGTAGGCACCAA-1_16  4.052368   8.995618  Epithelial cells
TCACTCGTCGTTGTGA-1_16  4.646813   5.651470  Epithelial cells

[116313 rows x 3 columns]
```

Thank you – Keep Calm and Annotate!