Essential python libraries for single cell data analysis

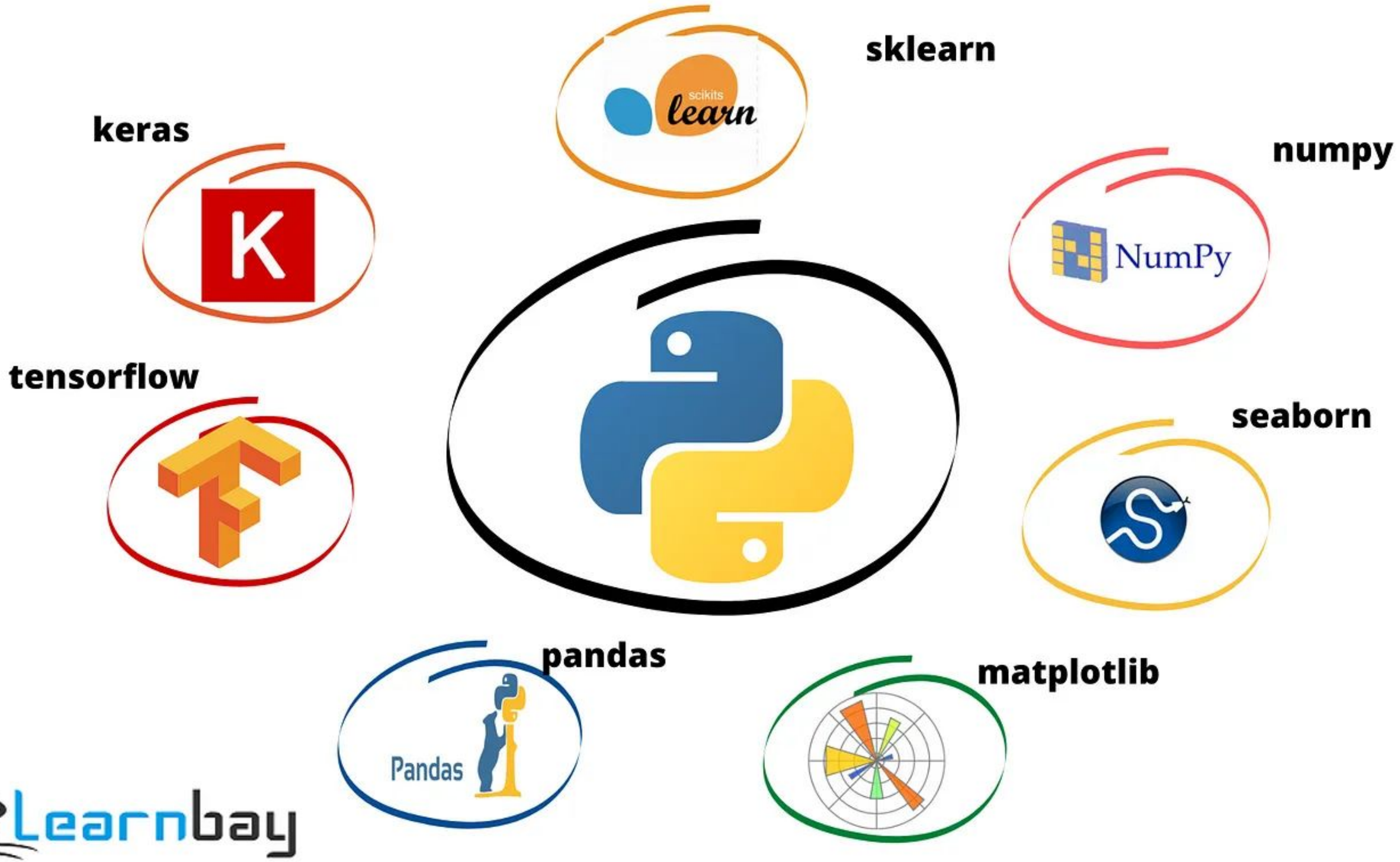
Bruno Puczko-Szymański

Warszawa 09.04.2024

UNIVERSITY OF WARSAW

WYDZIAŁ MATEMATYKI, INFORMATYKI I MECHANIKI

Essential libraries for everything



sklearn

keras

numpy

tensorflow

seaborn

pandas

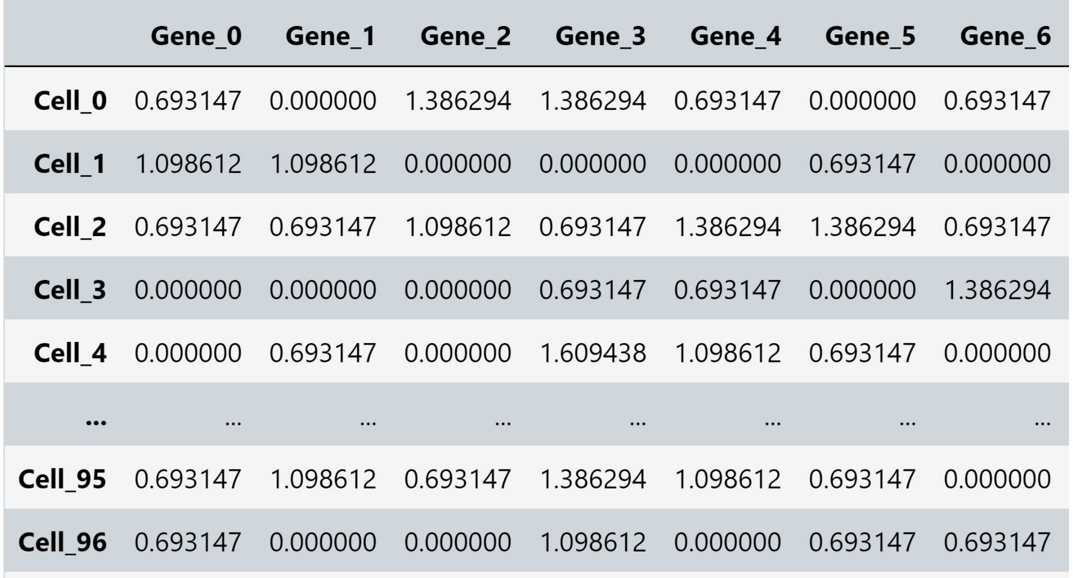
matplotlib

What informations do we need for single cell data analysis?

- Expression data

But also:

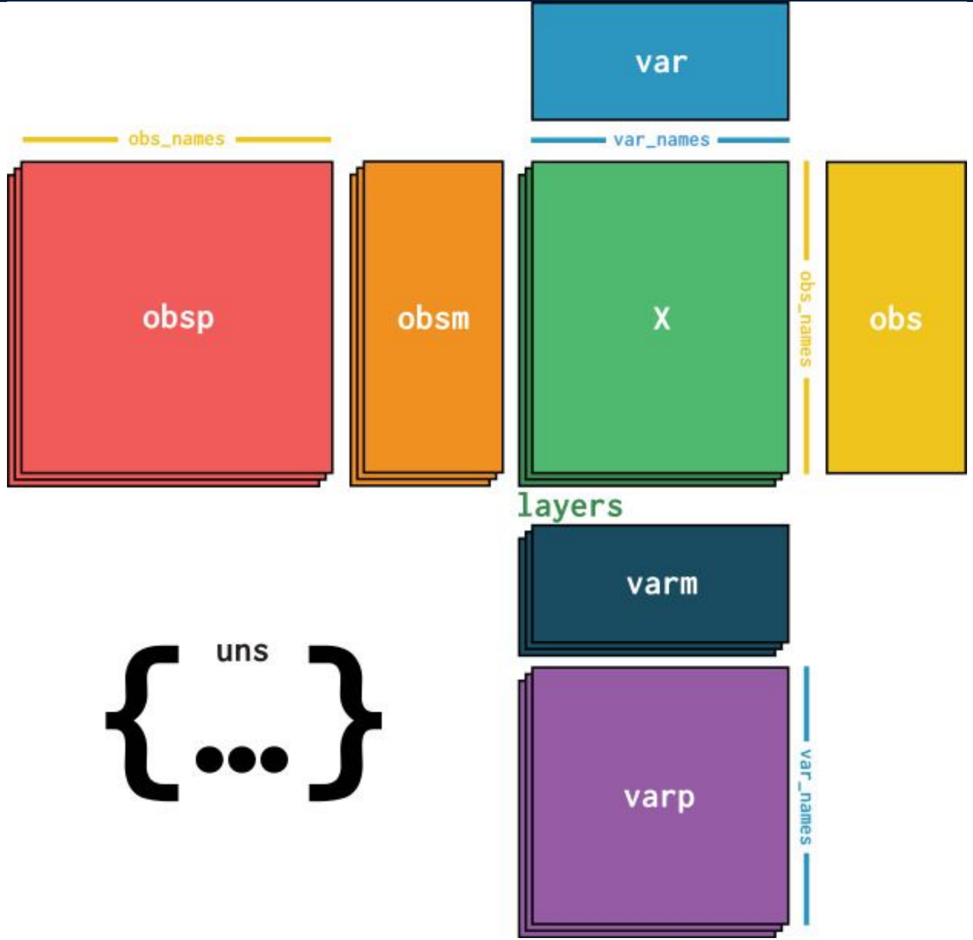
- patient ID
- sample ID
- condition
- cell type
- time
- and many many more…



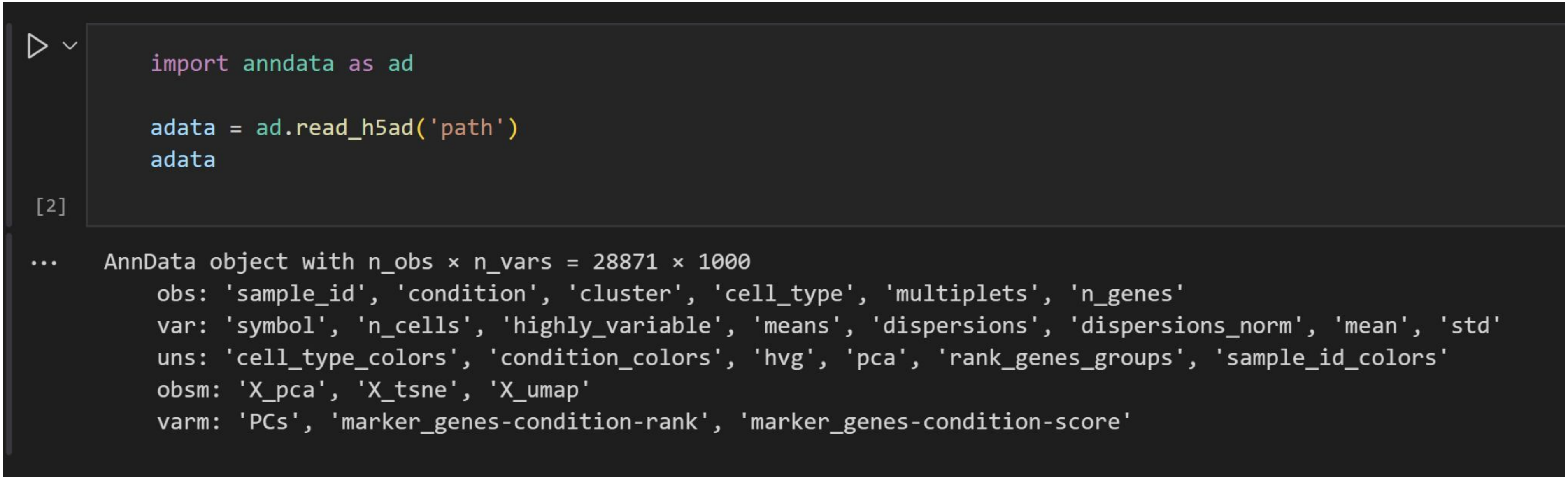
	Gene_0	Gene_1	Gene_2	Gene_3	Gene_4	Gene_5	Gene_6
Cell_0	0.693147	0.000000	1.386294	1.386294	0.693147	0.000000	0.693147
Cell_1	1.098612	1.098612	0.000000	0.000000	0.000000	0.693147	0.000000
Cell_2	0.693147	0.693147	1.098612	0.693147	1.386294	1.386294	0.693147
Cell_3	0.000000	0.000000	0.000000	0.693147	0.693147	0.000000	1.386294
Cell_4	0.000000	0.693147	0.000000	1.609438	1.098612	0.693147	0.000000
...	...	...	...	...	...	...	...
Cell_95	0.693147	1.098612	0.693147	1.386294	1.098612	0.693147	0.000000
Cell_96	0.693147	0.000000	0.000000	1.098612	0.000000	0.693147	0.693147

ANNDATA

Anndata


var
obs_names
var_names
obsp
obsm
X
obs_names
obs
layers
varm
uns
{ ••• }
varp
var_names

Anndata

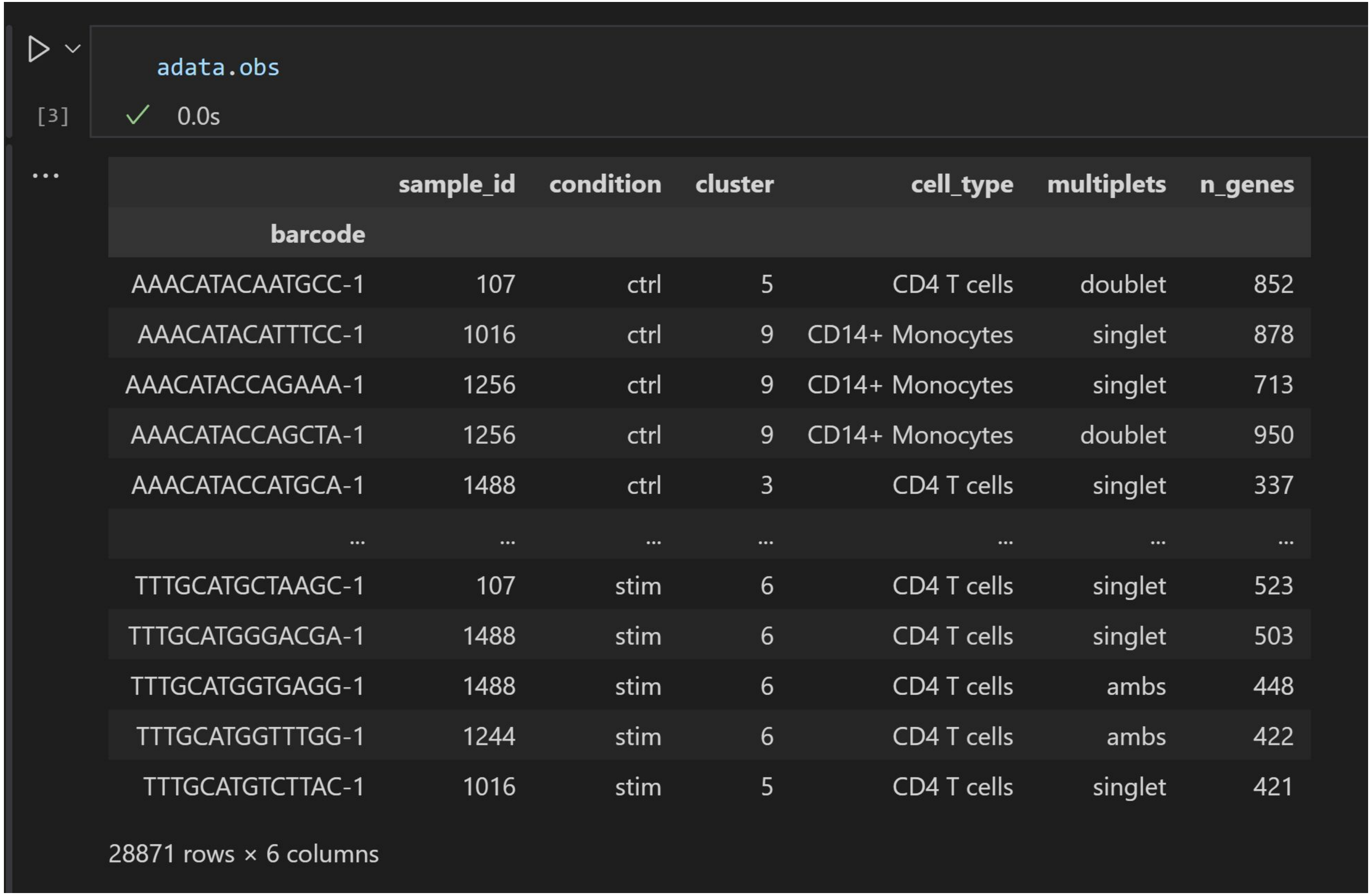


```
import anndata as ad

adata = ad.read_h5ad('path')
adata
```

```
AnnData object with n_obs × n_vars = 28871 × 1000
    obs: 'sample_id', 'condition', 'cluster', 'cell_type', 'multiplets', 'n_genes'
    var: 'symbol', 'n_cells', 'highly_variable', 'means', 'dispersions', 'dispersions_norm', 'mean', 'std'
    uns: 'cell_type_colors', 'condition_colors', 'hvg', 'pca', 'rank_genes_groups', 'sample_id_colors'
    obsm: 'X_pca', 'X_tsne', 'X_umap'
    varm: 'PCs', 'marker_genes-condition-rank', 'marker_genes-condition-score'
```

Anndata - cells



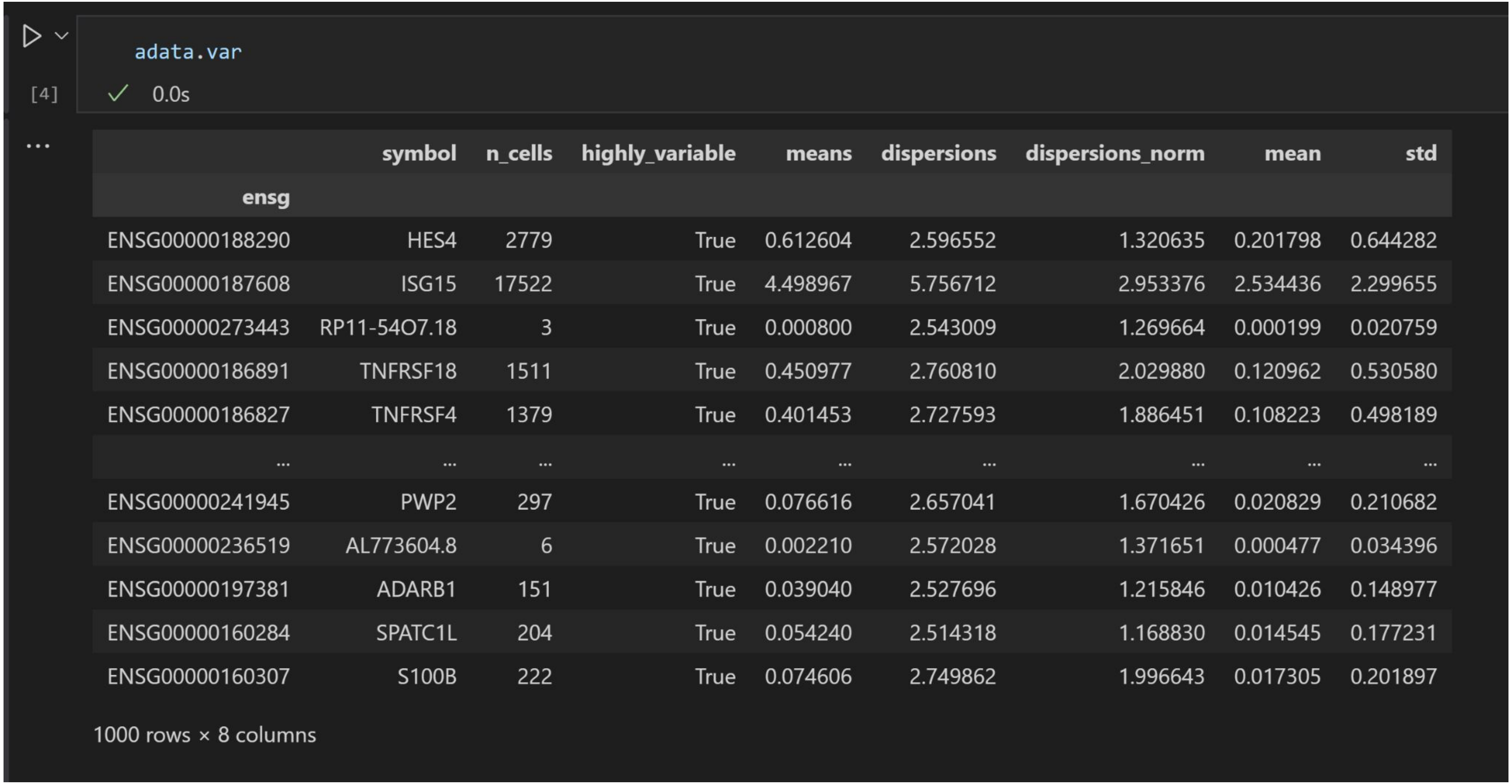
```
adata.obs
```

[3] ✓ 0.0s

	sample_id	condition	cluster	cell_type	multiplets	n_genes
barcode						
AAACATACAATGCC-1	107	ctrl	5	CD4 T cells	doublet	852
AAACATACATTTCC-1	1016	ctrl	9	CD14+ Monocytes	singlet	878
AAACATACCAGAAA-1	1256	ctrl	9	CD14+ Monocytes	singlet	713
AAACATACCAGCTA-1	1256	ctrl	9	CD14+ Monocytes	doublet	950
AAACATACCATGCA-1	1488	ctrl	3	CD4 T cells	singlet	337
...	...	...	...	...	...	...
TTTGCATGCTAAGC-1	107	stim	6	CD4 T cells	singlet	523
TTTGCATGGGACGA-1	1488	stim	6	CD4 T cells	singlet	503
TTTGCATGGTGAGG-1	1488	stim	6	CD4 T cells	ambs	448
TTTGCATGGTTTGG-1	1244	stim	6	CD4 T cells	ambs	422
TTTGCATGTCTTAC-1	1016	stim	5	CD4 T cells	singlet	421

28871 rows × 6 columns

Anndata - genes



```
adata.var
```

[4] ✓ 0.0s

	symbol	n_cells	highly_variable	means	dispersions	dispersions_norm	mean	std
ensg								
ENSG00000188290	HES4	2779	True	0.612604	2.596552	1.320635	0.201798	0.644282
ENSG00000187608	ISG15	17522	True	4.498967	5.756712	2.953376	2.534436	2.299655
ENSG00000273443	RP11-54O7.18	3	True	0.000800	2.543009	1.269664	0.000199	0.020759
ENSG00000186891	TNFRSF18	1511	True	0.450977	2.760810	2.029880	0.120962	0.530580
ENSG00000186827	TNFRSF4	1379	True	0.401453	2.727593	1.886451	0.108223	0.498189
...	...	...	...	...	...	...	...	...
ENSG00000241945	PWP2	297	True	0.076616	2.657041	1.670426	0.020829	0.210682
ENSG00000236519	AL773604.8	6	True	0.002210	2.572028	1.371651	0.000477	0.034396
ENSG00000197381	ADARB1	151	True	0.039040	2.527696	1.215846	0.010426	0.148977
ENSG00000160284	SPATC1L	204	True	0.054240	2.514318	1.168830	0.014545	0.177231
ENSG00000160307	S100B	222	True	0.074606	2.749862	1.996643	0.017305	0.201897

1000 rows × 8 columns

SCANPY

Scanpy

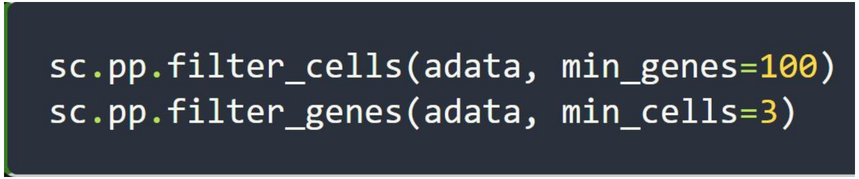
Scanpy is a scalable toolkit for analyzing single-cell gene expression data built jointly with anndata.

It includes preprocessing, visualization, clustering, trajectory inference and differential expression testing.

The Python-based implementation efficiently deals with datasets of more than one million cells.

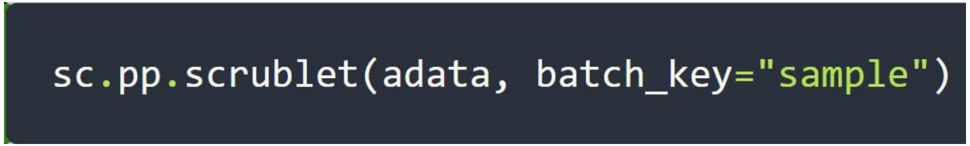
Scanpy - preprocessing

Filtering:



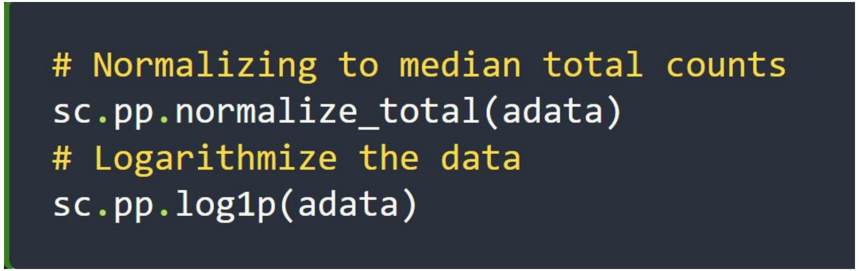
```python
sc.pp.filter_cells(adata, min_genes=100)
sc.pp.filter_genes(adata, min_cells=3)
```

Doublet detection:



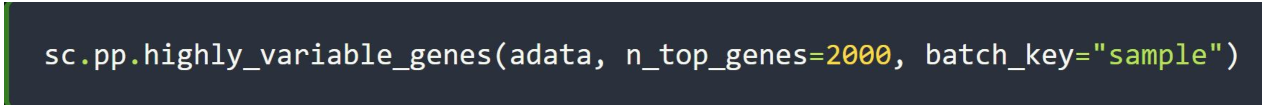
```python
sc.pp.scrublet(adata, batch_key="sample")
```

Normalization:



```python
# Normalizing to median total counts
sc.pp.normalize_total(adata)
# Logarithmize the data
sc.pp.log1p(adata)
```

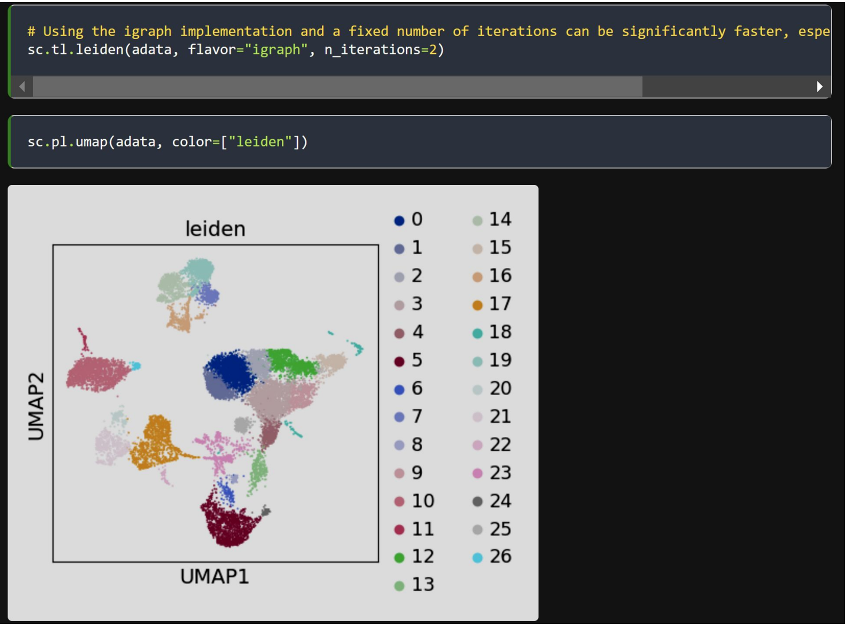
Feature selection:



```python
sc.pp.highly_variable_genes(adata, n_top_genes=2000, batch_key="sample")
```

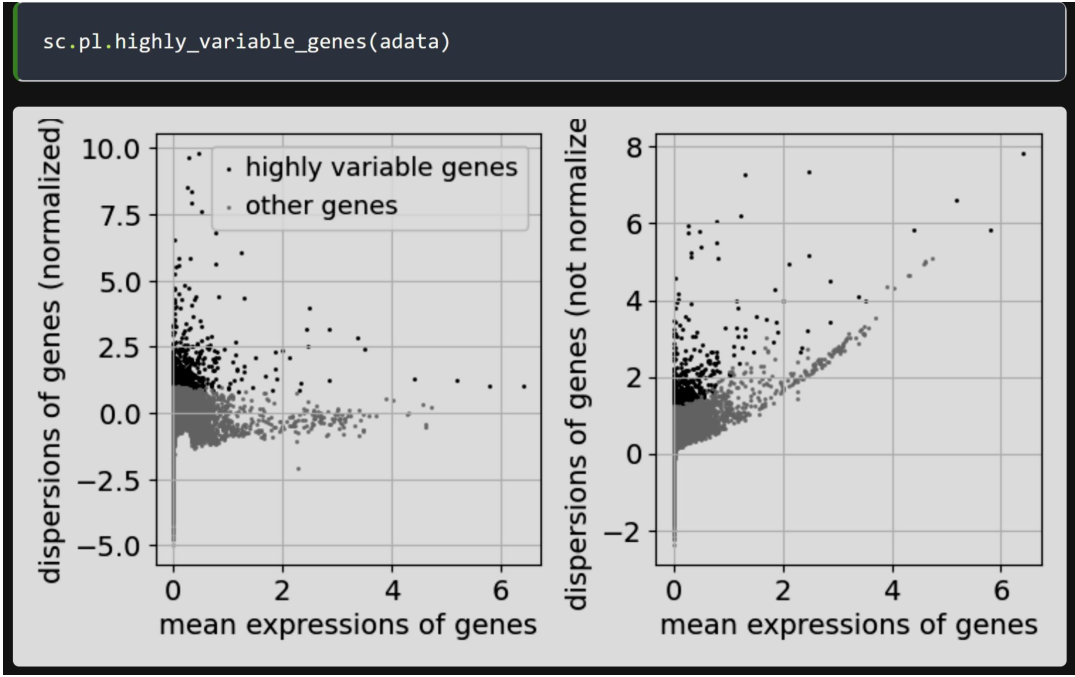
Scanpy - visualization

Many already implemented dimensionality reduction, and clustering algorithms like pca, t-sne, umap, laiden and many more.



```
# Using the igraph implementation and a fixed number of iterations can be significantly faster, espe
sc.tl.leiden(adata, flavor="igraph", n_iterations=2)
```

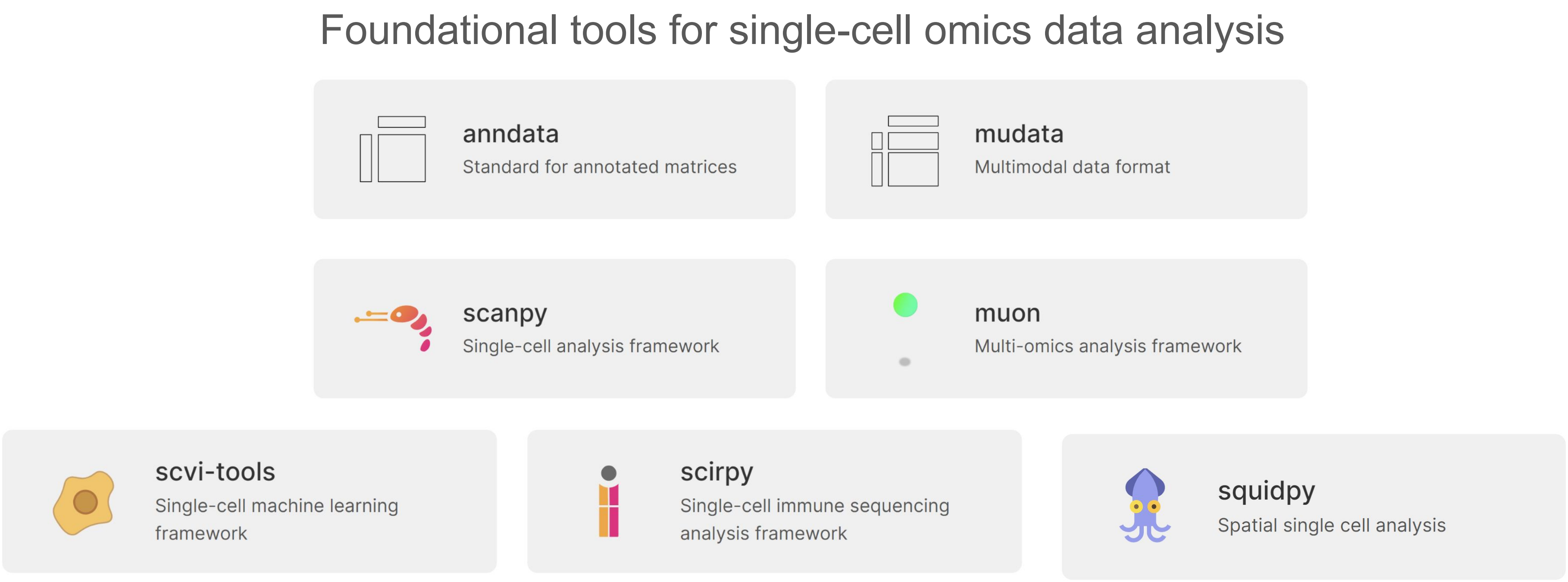
```
sc.pl.umap(adata, color=["leiden"])
```



```
sc.pl.highly_variable_genes(adata)
```

highly variable genes
other genes
dispersions of genes (normalized)
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
0
2
4
6
mean expressions of genes
dispersions of genes (not normalize
8
6
4
2
0
−2
0
2
4
6
mean expressions of genes

Scverse



Foundational tools for single-cell omics data analysis

anndata
Standard for annotated matrices

mudata
Multimodal data format

scanpy
Single-cell analysis framework

muon
Multi-omics analysis framework

scvi-tools
Single-cell machine learning framework

scirpy
Single-cell immune sequencing analysis framework

squidpy
Spatial single cell analysis

Bibliography

- https://www.sc-best-practices.org/introduction/analysis_tools.html
- https://anndata.readthedocs.io/en/latest/tutorials/notebooks/getting-started.html
- https://scverse.org/
- https://scanpy.readthedocs.io/en/latest/tutorials/basics/clustering.html