anndata test

December 20, 2024

0.1 The anndata package

0.1.1 Students notebook

Organizing single cell data in Python This notebook is the first one of a series where you will replicate some of the analysis presented in the coding lectures on the T-cell use case, this time using iTReg-activated T-cells rather than the resting ones. Let's start!

[ ]:
```
# importing anndata, numpy and pandas as we did in the coding lectures
import anndata as ad
import numpy as np
import pandas as pd
```

[ ]:
```
# exercise: read the scRNA-seq dataset from the './data/scde.h5ad' file and␣
 ↪name it "scde"
scde = None
# your code here

print(scde)
```

[ ]:
```
# tip: use the scanpy function for reading h5ad files!
```

Note that we have only 500 cells; this will allow us to proceed with the analysis faster. Let's check the `obs` data frame:

[ ]:
```
# obs data frame
scde.obs
```

It appears that we do not have any information on the selected cells in the scanpy object. However, we know that this information is stored in a csv file, namely 'obs.csv'. Let's load it and assign the resulting data frame to `scde.obs`.

[ ]:
```
# exercise: read the ./data/obs.csv file into a pandas data frame called obs
obs = None
# your code here

print(obs)
```

[ ]:
```
# tip: use the pandas function for reading csv files! The first column is the␣
↪index column
```

[ ]:
```
# exercise: assign the obs data frame to scde.obs
# your code here

print(scde.obs)
```

[ ]:
```
# tip: it's a simple assignment! :-)
```

Now that we have the cell annotation from the original study, we can check what types of cells are included.

[ ]:
```
# exercise: create a list indicating the cell types contained in the cluster.id␣
↪column of obs.
cell_types = []
# your code here

print(cell_types)
```

[ ]:
```
# tip: each cell type should be present once. Consider using the `unique()` and␣
↪`tolist()` methods
```

We now want to ensure that all cells have a sufficient amount of reads before continuing our analyses. Thus, let's compute the sum of reads for each cell and let's story it in a numpy array.

[ ]:
```
# exercise: compute the number of reads for each cell
num_reads = np.nan
# your code here

print(num_reads[0:10])
```

[ ]:
```
# tip: simple sum across all genes for each cell
```

[ ]:
```
# exercise: remove cells with less than 10000 reads in total. Name the new␣
↪anndata object 'scder'
scder = None
# your code here

print(scder.n_obs)
```

[ ]:
```
# tip: recall how anndata object can be subsetted
```

We are almost done with this first phase of the analysis. We just need to save the reduced scanpy object in a new file, 'scdr.h5ad'

[ ]:
```
# making sure the 'scdr.h5ad' file is not on the disk already
import os
```

```
if os.path.isfile('./data/scder.h5ad'):
    os.remove('./data/scder.h5ad')
```

[ ]:
```
# exercise: save the new scanpy object in a file named './data/scder.h5ad'
# your code here
```

[ ]:
```
# tip: use the scanpy method for writing h5ad files
```