Getting started

Source: `vignettes/getting_started.Rmd` (https://github.com/dynverse/anndata/blob/master/vignettes/getting_started.Rmd)

The API of anndata for R is very similar to its Python counterpart. Check out `?anndata` for a full list of the functions provided by this package.

`AnnData()` stores a data matrix `X` together with annotations of observations `obs` (`obsm`, `obsp`), variables `var` (`varm`, `varp`), and unstructured annotations `uns`.

Here is an example of how to create an AnnData object with 2 observations and 3 variables.

```
library(anndata)

ad <- AnnData(
  X = matrix(1:6, nrow = 2),
  obs = data.frame(group = c("a", "b"), row.names = c("s1", "s2")),
  var = data.frame(type = c(1L, 2L, 3L), row.names = c("var1", "var2", "var3")),
  layers = list(
    spliced = matrix(4:9, nrow = 2),
    unspliced = matrix(8:13, nrow = 2)
  ),
  obsm = list(
    ones = matrix(rep(1L, 10), nrow = 2),
    rand = matrix(rnorm(6), nrow = 2),
    zeros = matrix(rep(0L, 10), nrow = 2)
  ),
  varm = list(
    ones = matrix(rep(1L, 12), nrow = 3),
    rand = matrix(rnorm(6), nrow = 3),
    zeros = matrix(rep(0L, 12), nrow = 3)
  ),
  uns = list(
    a = 1,
    b = data.frame(i = 1:3, j = 4:6, value = runif(3)),
    c = list(c.a = 3, c.b = 4)
  )
)

ad
#> AnnData object with n_obs × n_vars = 2 × 3
#>     obs: 'group'
#>     var: 'type'
#>     uns: 'a', 'b', 'c'
#>     obsm: 'ones', 'rand', 'zeros'
#>     varm: 'ones', 'rand', 'zeros'
#>     layers: 'spliced', 'unspliced'
```

You can read the information back out using the `$` notation.

```
ad$X
#>    var1 var2 var3
#> s1    1    3    5
#> s2    2    4    6
ad$obs
#>    group
#> s1     a
#> s2     b
ad$obsm[["ones"]]
#>      [,1] [,2] [,3] [,4] [,5]
#> [1,]    1    1    1    1    1
#> [2,]    1    1    1    1    1
ad$layers[["spliced"]]
#>    var1 var2 var3
#> s1    4    6    8
#> s2    5    7    9
ad$uns[["b"]]
#>   i j     value
#> 1 1 4 0.6659395
#> 2 2 5 0.8000884
#> 3 3 6 0.6528542
```

Reading / writing AnnData objects

Read from h5ad format:

```
read_h5ad("pbmc_1k_protein_v3_processed.h5ad")
```

Creating a view

You can use any of the regular R indexing methods to subset the `AnnData` object. This will result in a 'View' of the underlying data without needing to store the same data twice.

```
view <- ad[, 2]
view
#> View of AnnData object with n_obs × n_vars = 2 × 1
#>     obs: 'group'
#>     var: 'type'
#>     uns: 'a', 'b', 'c'
#>     obsm: 'ones', 'rand', 'zeros'
#>     varm: 'ones', 'rand', 'zeros'
#>     layers: 'spliced', 'unspliced'
view$is_view
#> [1] TRUE
ad[,c("var1", "var2")]
#> View of AnnData object with n_obs × n_vars = 2 × 2
#>     obs: 'group'
#>     var: 'type'
#>     uns: 'a', 'b', 'c'
#>     obsm: 'ones', 'rand', 'zeros'
#>     varm: 'ones', 'rand', 'zeros'
#>     layers: 'spliced', 'unspliced'
ad[-1, ]
#> View of AnnData object with n_obs × n_vars = 1 × 3
#>     obs: 'group'
#>     var: 'type'
#>     uns: 'a', 'b', 'c'
#>     obsm: 'ones', 'rand', 'zeros'
#>     varm: 'ones', 'rand', 'zeros'
#>     layers: 'spliced', 'unspliced'
```

AnnData as a matrix

The `X` attribute can be used as an R matrix:

```
ad$X[,c("var1", "var2")]
#>    var1 var2
#> s1    1    3
#> s2    2    4
ad$X[-1, , drop = FALSE]
#>    var1 var2 var3
#> s2    2    4    6
ad$X[, 2] <- 10
```

You can access a different layer matrix as follows:

```
ad$layers["unspliced"]
#>    var1 var2 var3
#> s1    8   10   12
#> s2    9   11   13
ad$layers["unspliced"][,c("var2", "var3")]
#>    var2 var3
#> s1   10   12
#> s2   11   13
```

Note on state

If you assign an AnnData object to another variable and modify either, both will be modified:

```
ad2 <- ad

ad$X[,2] <- 10

list(ad = ad$X, ad2 = ad2$X)
#> $ad
#>    var1 var2 var3
#> s1    1   10    5
#> s2    2   10    6
#>
#> $ad2
#>    var1 var2 var3
#> s1    1   10    5
#> s2    2   10    6
```

This is standard Python behaviour but not R. In order to have two separate copies of an AnnData object, use the `$copy()` function:

```
ad3 <- ad$copy()

ad$X[,2] <- c(3, 4)

list(ad = ad$X, ad3 = ad3$X)
#> $ad
#>    var1 var2 var3
#> s1    1    3    5
#> s2    2    4    6
#>
#> $ad3
#>    var1 var2 var3
#> s1    1   10    5
#> s2    2   10    6
```
