README

The goal of this repo is to design efficient abstractions to the `SummarizedExperiment` class such that using common dplyr functions feels as natural to operating on a `data.frame` or `tibble`. While the overall goal is for it to **feel** like a tibble operation, it would be smart to emphasize that certain data wrangling pipelines do not translate well to the structure of the `SummarizedExperiment` class.

Example Data

I will be using the following example data throughout this document:

Listing 1 reproducible example data

```
library(SummarizedExperiment)
set.seed(1234)
se <- SummarizedExperiment(
  list(counts = matrix(sample(1:20, 20), nrow = 5, ncol = 4)),
  rowData = data.frame(gene = sprintf("g%i", 1:5),
                       length = rbinom(5, 100, runif(5)),
                       direction = sample(c("-","+"), 5, T)),
  colData = data.frame(sample = sprintf("s%i", 1:4),
                       condition = rep(c("cntrl","drug"), each =2))
)
rownames(se) <- sprintf("row_%s", letters[1:5])
colnames(se) <- sprintf("col_%s", LETTERS[1:4])
assay(se, 'logcounts') <- log(assay(se, 'counts'))
se
```

```
class: SummarizedExperiment
dim: 5 4
metadata(0):
assays(2): counts logcounts
rownames(5): row_a row_b row_c row_d row_e
rowData names(3): gene length direction
colnames(4): col_A col_B col_C col_D
colData names(2): sample condition
```

The abstraction

In order to access parts of the `SummarizedExperiment` as if it were a tibble, I propose we use some data masking concepts from the `rlang` package.

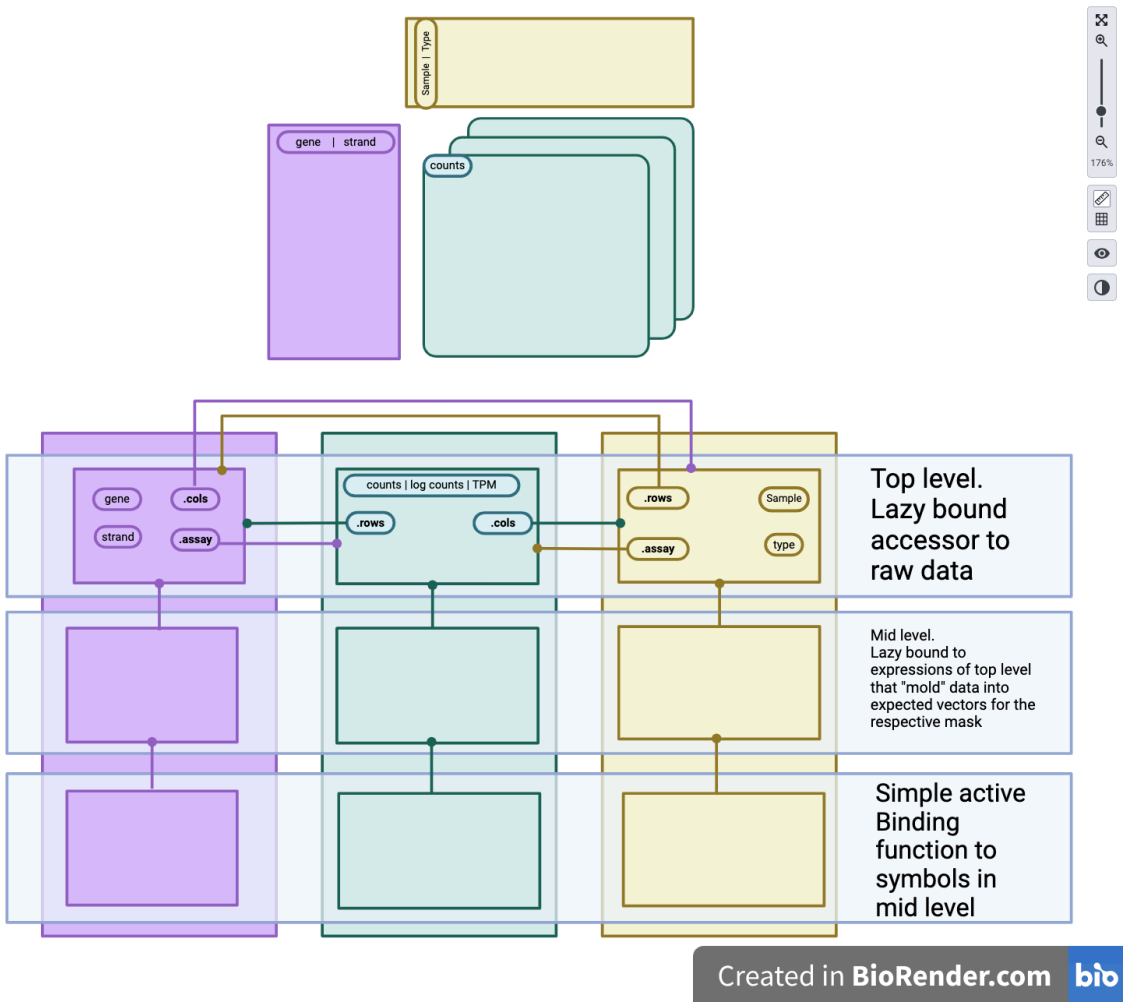

Figure 1: Figure created with BioRender.com

In Figure 1, we an abstract a `SummarizedExperiment` object (top portion) into three distinct data masks (the bottom portion) that represent different evaluation contexts for our object. We are either evaluating on the `assay_mask`, `rowData_mask`, or the `colData_mask`. Data will be lazily bound to the top level of each mask "as is" from the `SummarizedExperiment` object's data context.

For example, for `se` from Listing 1

data mask

To quote the documentation of `?rlang::new_data_mask`:

> A data mask is an environment (or possibly multiple environments forming an ancestry) containing user-supplied objects. Objects in the mask have precedence over objects in the environment (i.e. they mask those objects). Many R functions evaluate quoted expressions in a data mask so these expressions can refer to objects within the user data.

`dplyr` verbs

`mutate`

`mutate` is one of the more common `dplyr` verbs used and is likely the most compatible.