

dplyr

plyxp: reimagining dplyr syntax for SummarizedExperiment objects.

plyxp

Summarized
Experiment

tidyverse

Justin Landis
Love Lab - UNC - Chapel Hill

Bioconductor

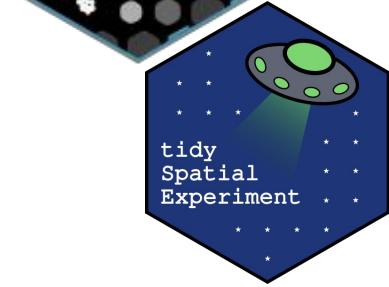
www.bioconductor.org

www.bioconductor.org

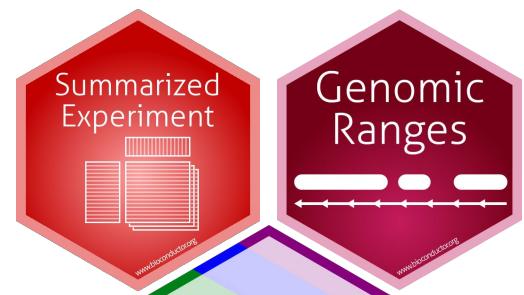
tidyverse



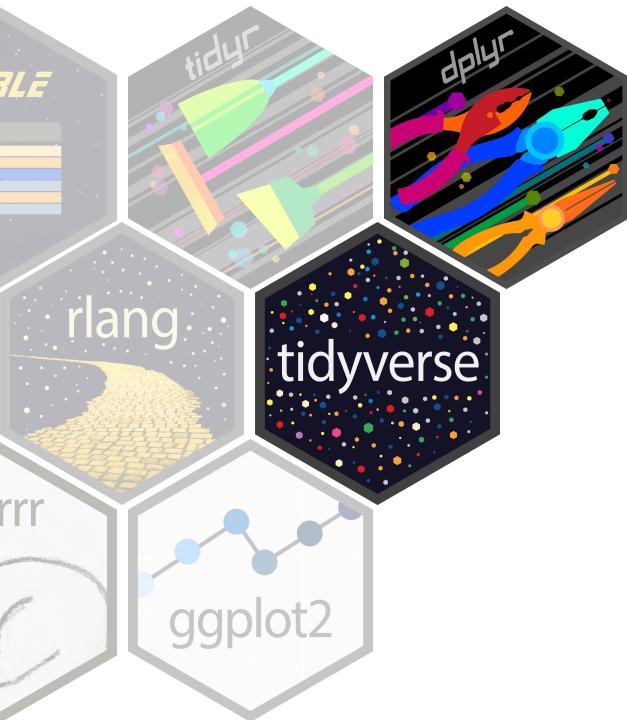
tidyomics



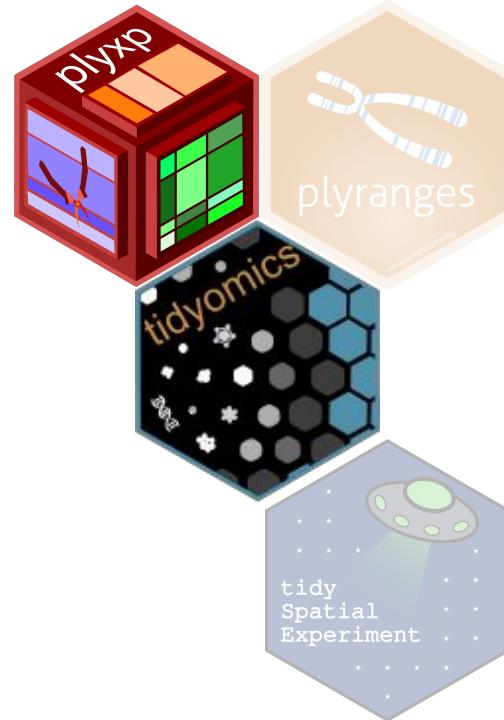
Bioconductor



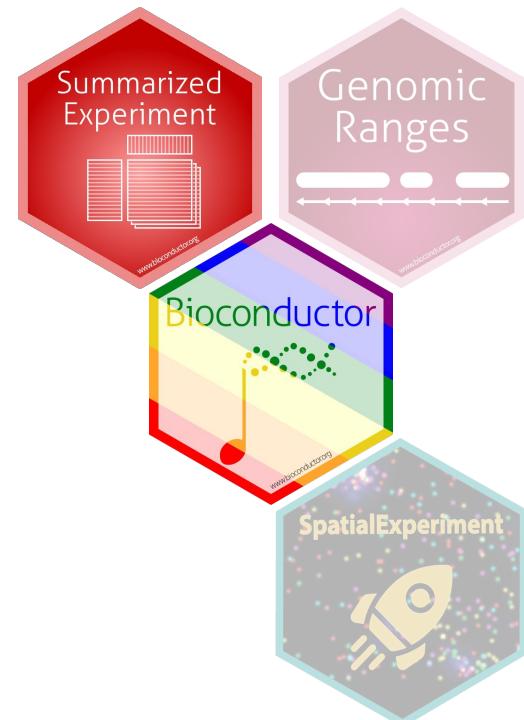
tidyverse

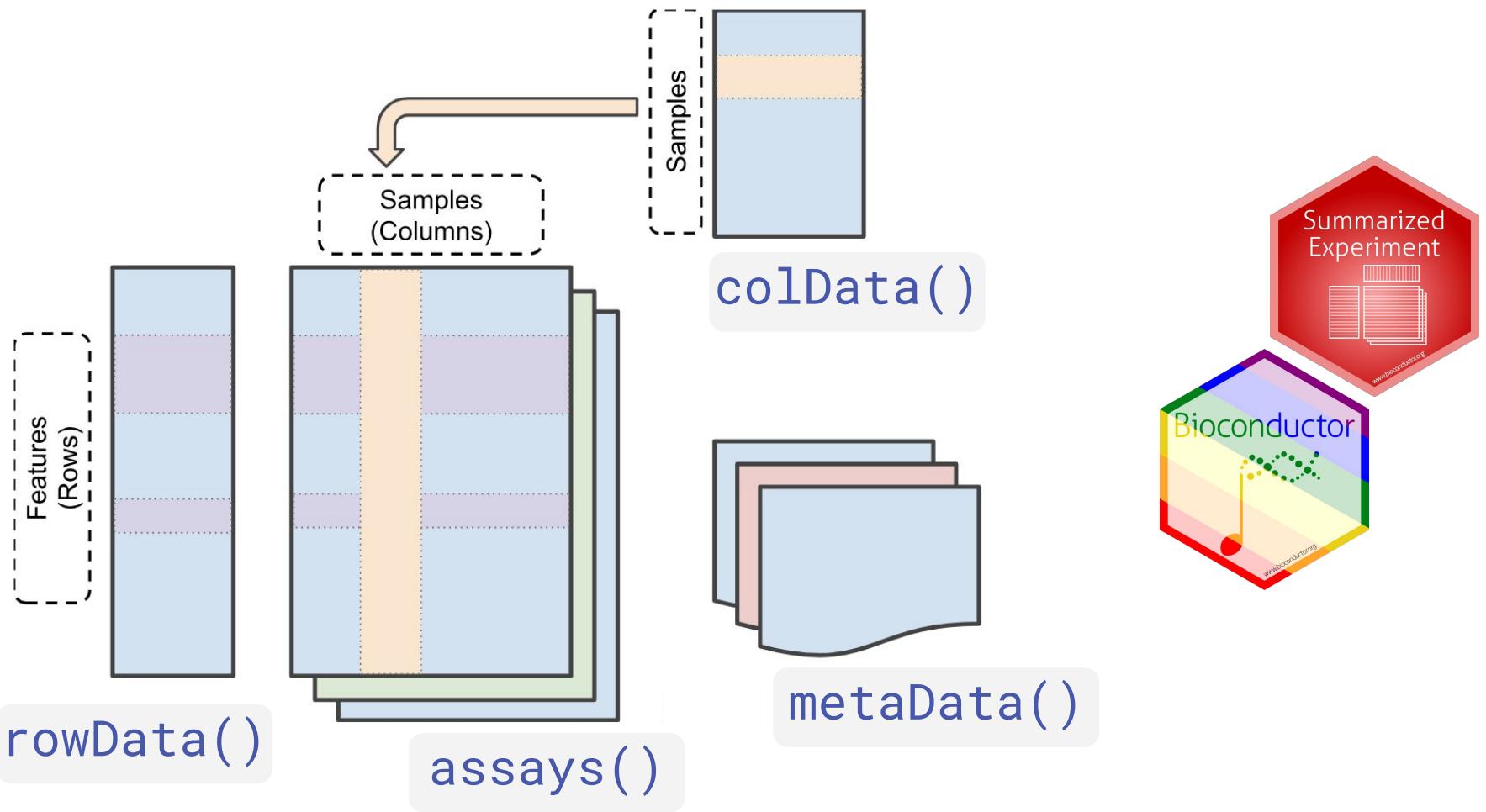


tidyomics



Bioconductor





Examples

```
1 df <- data.frame(  
2   x = 1:5,  
3   y = 6:10,  
4   z = c("a", "b", "c", "d", "e")  
5 )
```

Global:

Empty Environment

Examples

```
1 df <- data.frame(  
2   x = 1:5,  
3   y = 6:10,  
4   z = c("a", "b", "c", "d", "e")  
5 )
```

```
print("Hello World")
```



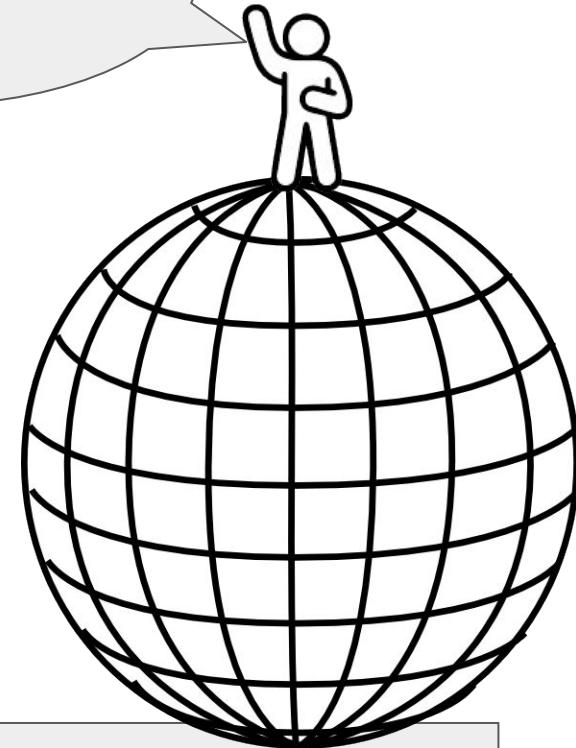
Global:

Empty Environment

Examples

```
1 df <- data.frame(  
2   x = 1:5,  
3   y = 6:10,  
4   z = c("a", "b", "c", "d", "e")  
5 )
```

print("Hello World")



Global:

Empty Environment

Examples

```
1 df <- data.frame(  
2   x = 1:5,  
3   y = 6:10,  
4   z = c("a", "b", "c", "d", "e")  
5 )
```

Global:

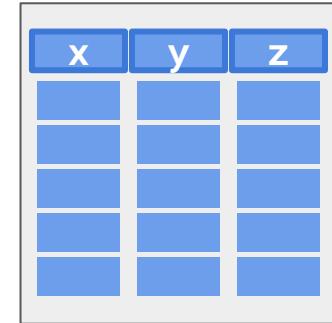
df

Empty Environment

Examples

```
1 df <- data.frame(  
2   x = 1:5,  
3   y = 6:10,  
4   z = c("a", "b", "c", "d", "e")  
5 )
```

“data.frame”



Global:

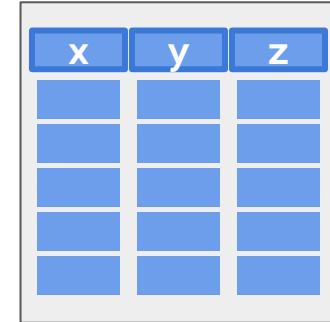
df

Empty Environment

Examples

```
1 x # error: no symbol 'x'  
2  
3 df$x # [5] 1 2 3 4 5
```

"`data.frame`"



Global:

`df`

Empty Environment

Examples

```
1 df$sum <- df$x + df$y
```

“`data.frame`”

x	y	z	sum

Global:

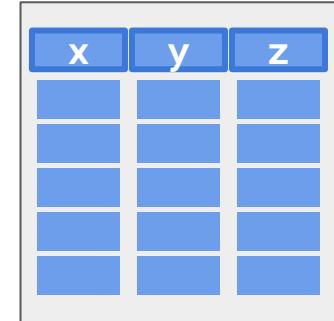
`df`

Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"`data.frame`"



Global:

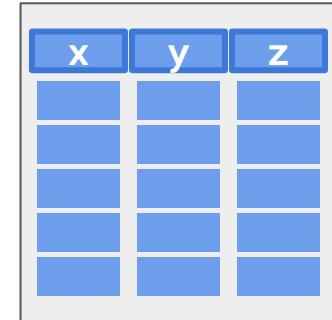
`df`

Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"data.frame"



mutate

Scope:



Global:



Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"`data.frame`"

x	y	z

Data Mask:



`mutate` Scope:



Global:

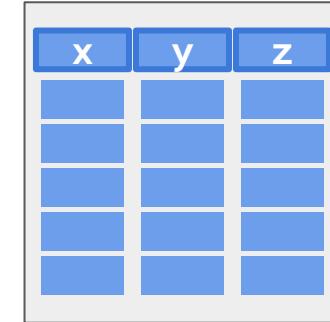


Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"`data.frame`"



Data Mask:



`mutate` Scope:



Global:

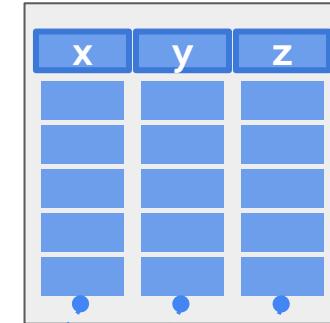


Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

“`data.frame`”



Data Mask:



`mutate` Scope:



Global:



Empty Environment

Examples

“`data.frame`”

```
1 df <- mutate(df, sum = x + y)
```

Data Mask:



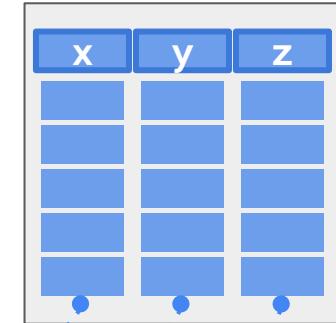
`mutate` Scope:



Global:



Empty Environment



Examples

```
1 df <- mutate(df, sum = x + y)
```

“`data.frame`”

x	y	z	sum

Data Mask:



`mutate` Scope:



Global:



Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"data.frame"

x	y	z	sum

mutate

Scope:

df

Global:

df

Empty Environment

Examples

```
1 df <- mutate(df, sum = x + y)
```

"`data.frame`"

x	y	z	sum

Global:

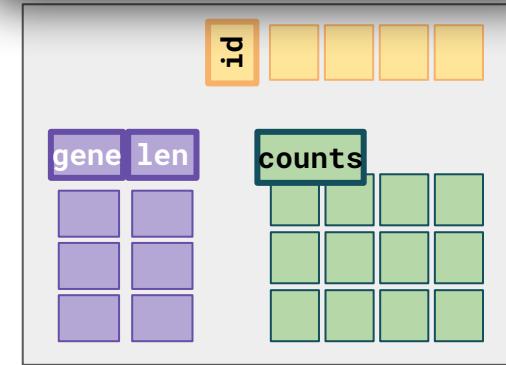
df

Empty Environment

Examples

```
1 se <- mutate(se, ...) #???
```

“SummarizedExperiment”



Global:

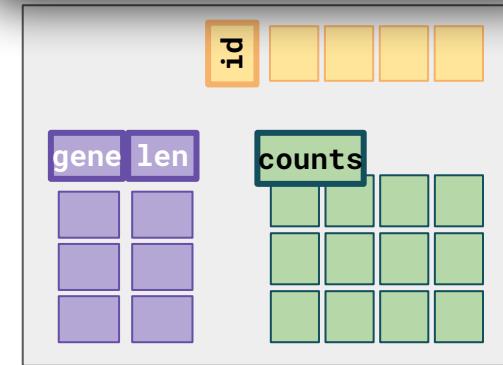
se

Empty Environment

Examples

```
1 se <- mutate(se, ...)
```

“SummarizedExperiment”



`mutate`

Scope:



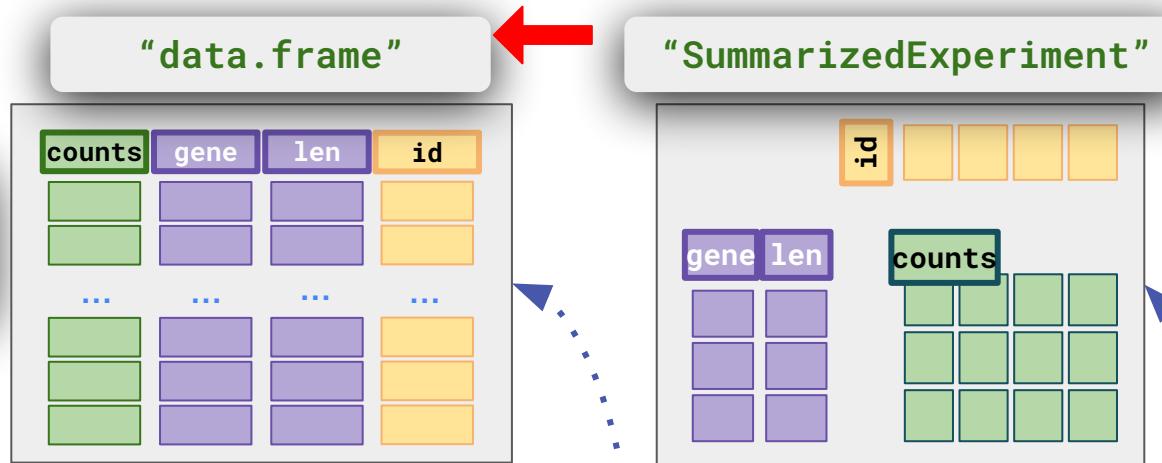
Global:



Empty Environment

Examples

```
1 se <- mutate(se, ...)
```



`mutate` Scope:



Global:

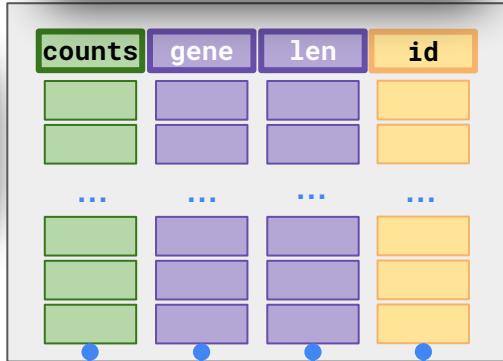


Empty Environment

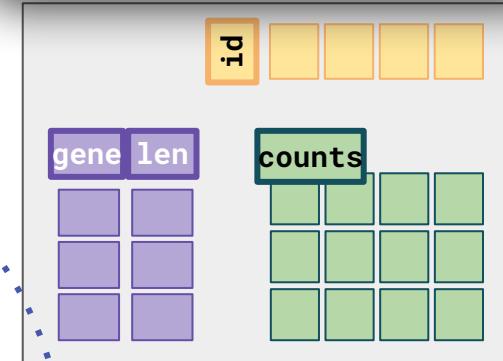
Examples

```
1 se <- mutate(se, ...)
```

“data.frame”



“SummarizedExperiment”



Data Mask:



mutate Scope:



Global:

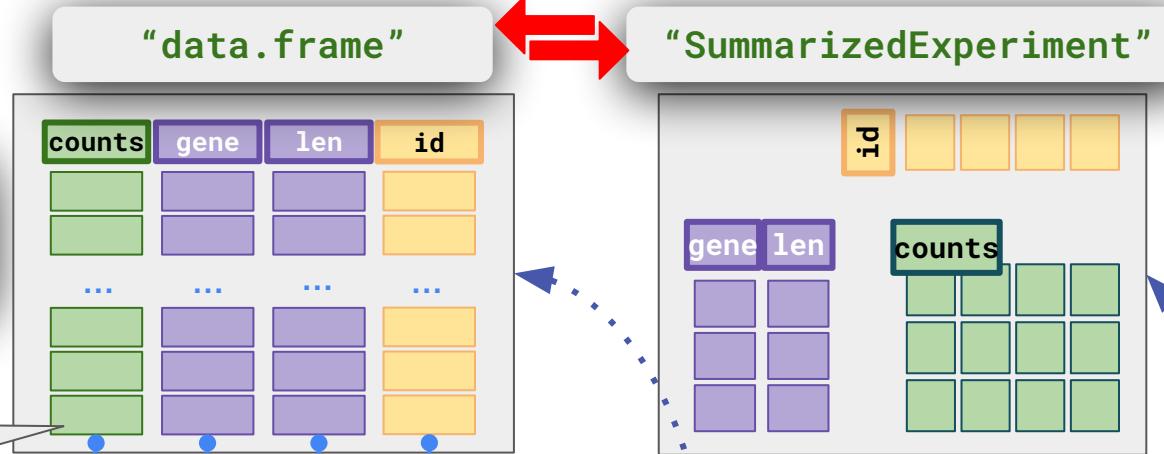


Empty Environment

Examples

```
1 se <- mutate(se, ...)
```

Inefficient with
increasing
complexity!



Data Mask:



`mutate` Scope:



Global:

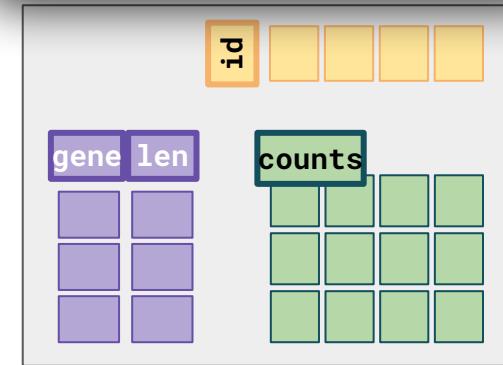


Empty Environment

Examples

```
1 se <- mutate(se, ...)
```

“SummarizedExperiment”



`mutate`

Scope:

`se`

Global:

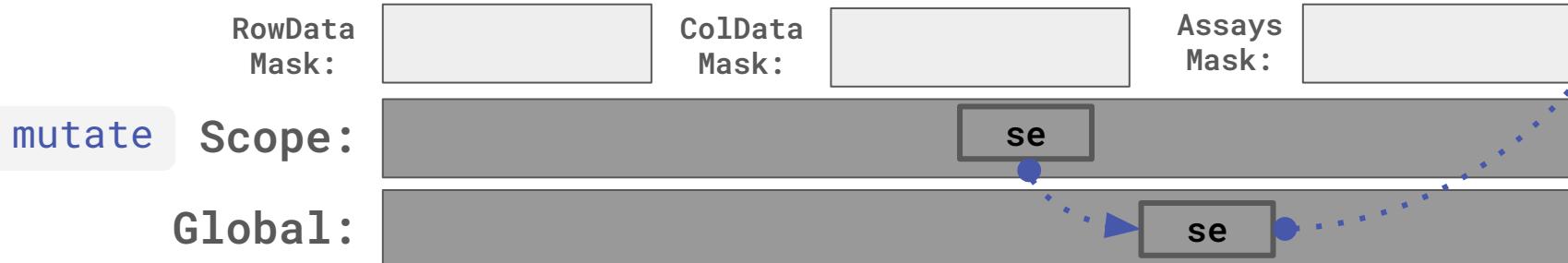
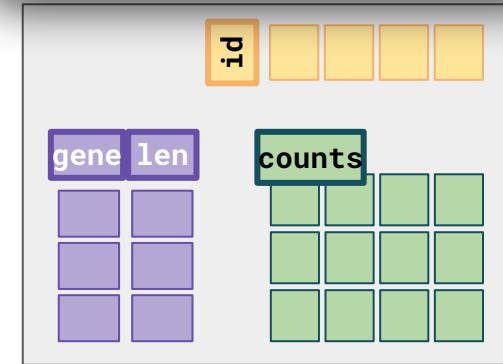
`se`

Empty Environment

Examples

```
1 se <- mutate(se, ...)
```

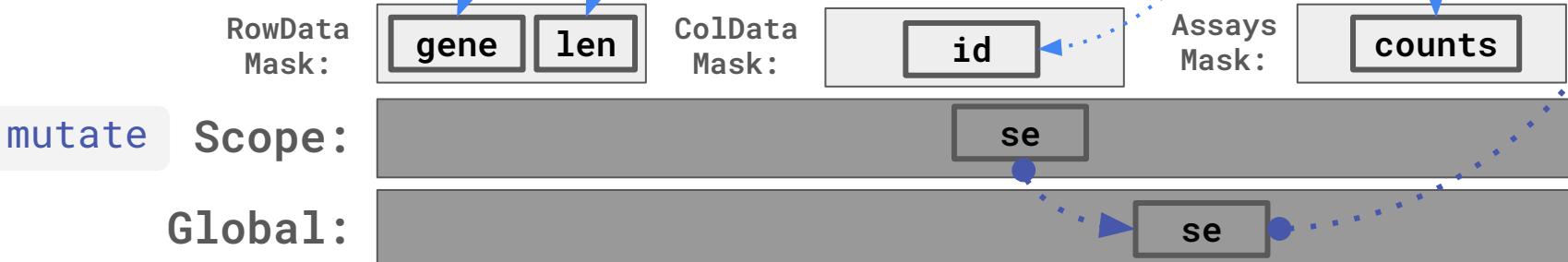
“SummarizedExperiment”



Empty Environment

Examples

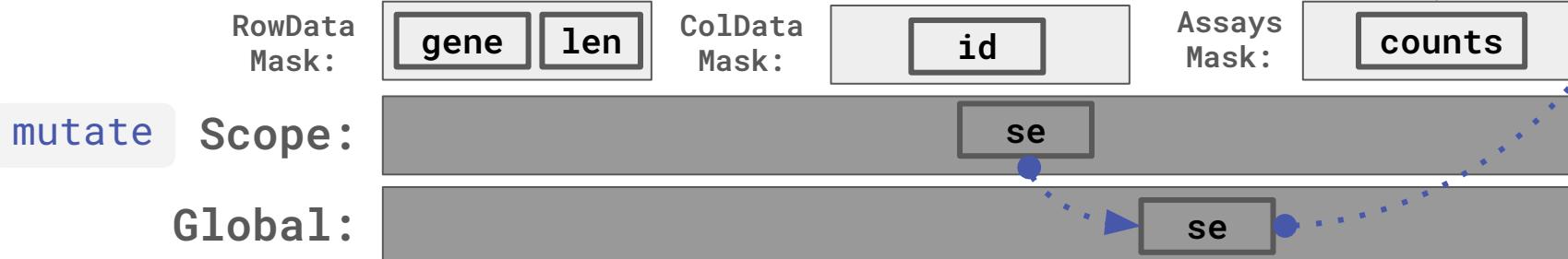
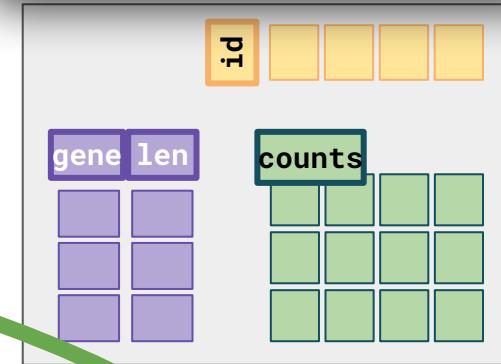
```
1 se <- mutate(se, ...)
```



Examples

```
1 se <- mutate(se,  
2     ...  
3 )
```

“SummarizedExperiment”

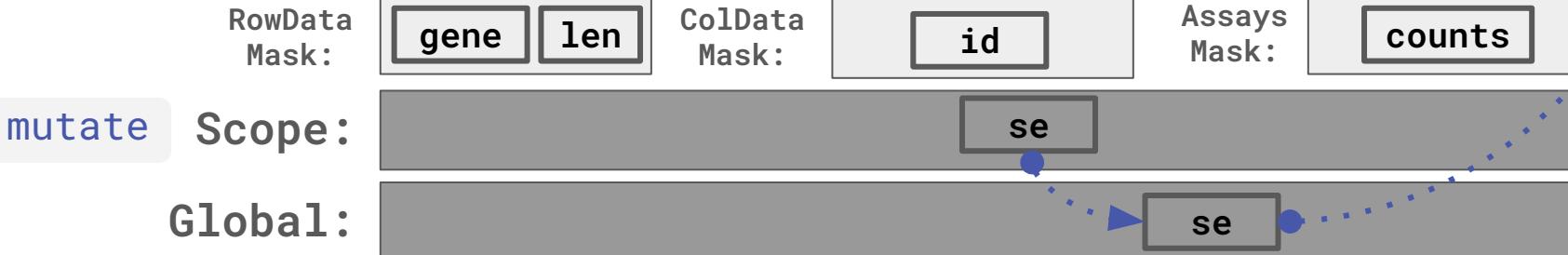
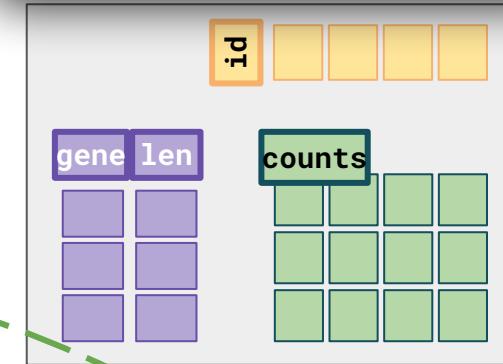


Empty Environment

Examples

```
1 se <- mutate(se,  
2   ...,  
3   cols(...))  
4 )
```

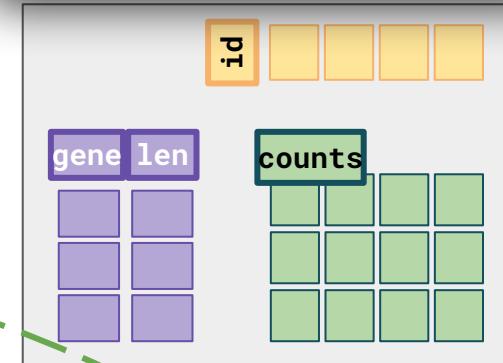
“SummarizedExperiment”



Examples

```
1 se <- mutate(se,  
2   ...,  
3   cols(...),  
4   rows(...)  
5 )
```

“SummarizedExperiment”



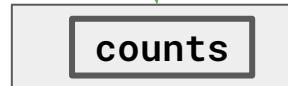
RowData
Mask:



ColData
Mask:

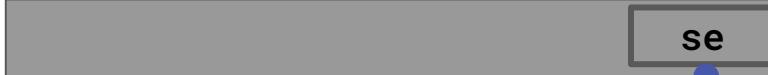


Assays
Mask:

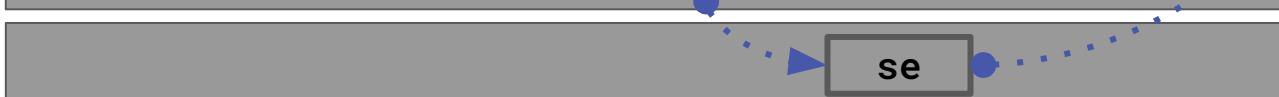


mutate

Scope:



Global:

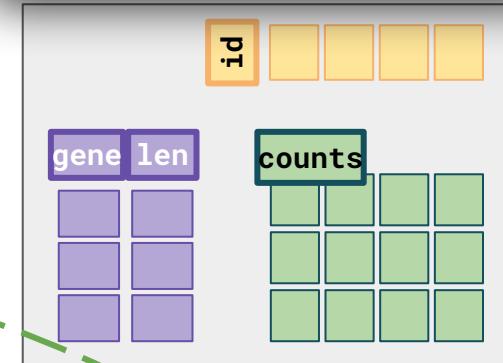


Empty Environment

Examples

```
1 se <- mutate(se,  
2   ....,  
3   cols(...),  
4   rows(...)  
5 )
```

“SummarizedExperiment”



RowData
Mask:



ColData
Mask:

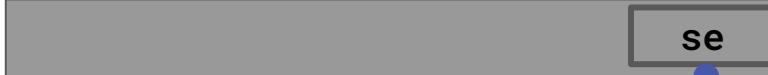


Assays
Mask:

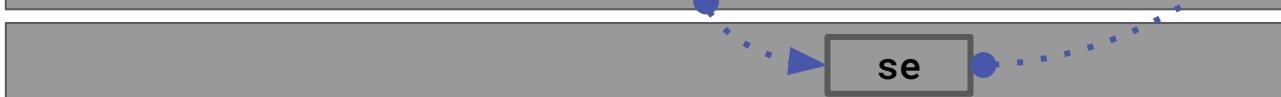


`mutate`

Scope:



Global:

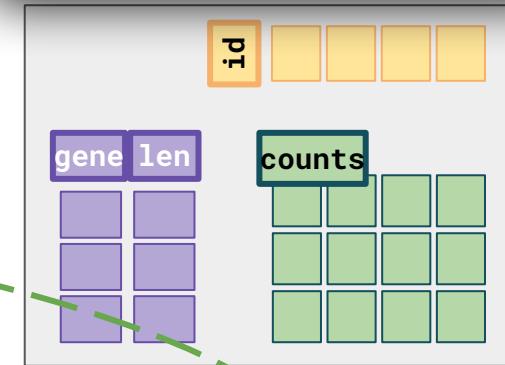


Empty Environment

Examples: Pronouns

```
1 se <- mutate(se,  
2     ....,  
3     cols(...),  
4     rows(...)  
5 )
```

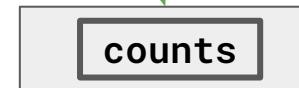
“SummarizedExperiment”



RowData
Mask:



ColData
Mask:



Assays
Mask:

mutate

Scope:

se

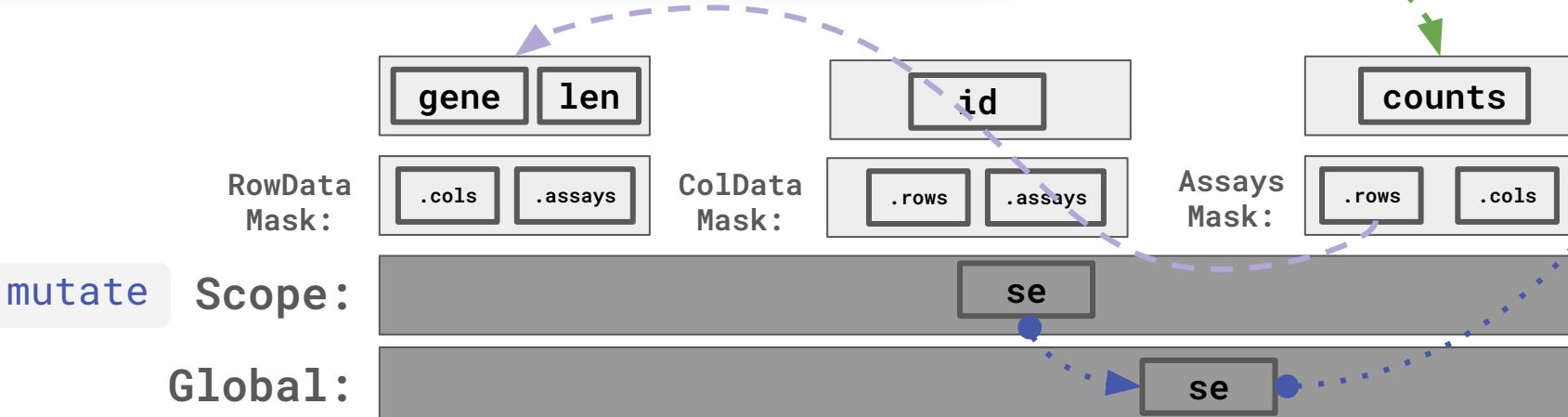
Global:

se

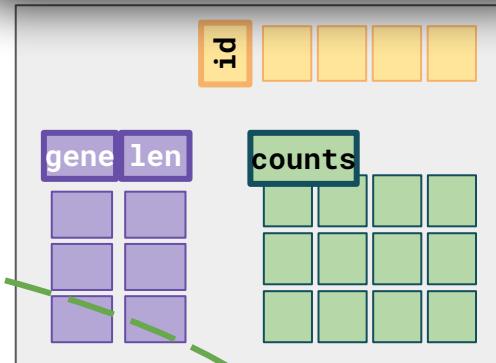
Empty Environment

Examples: Pronouns - Compute TPM

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4
5 )
```



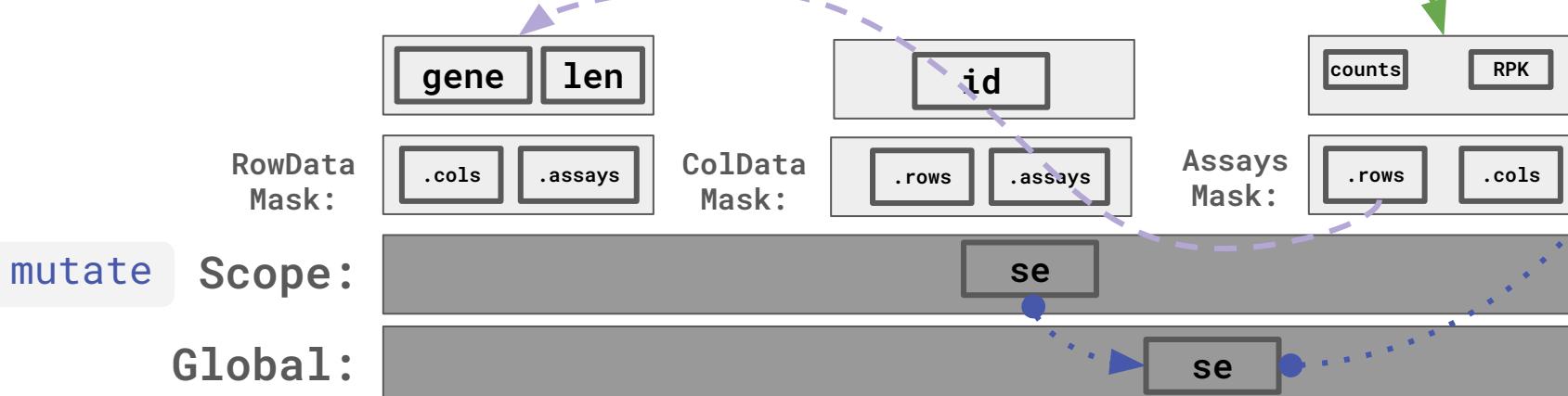
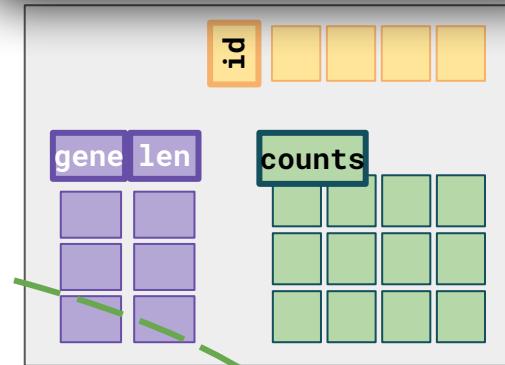
“SummarizedExperiment”



Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4
5 )
```

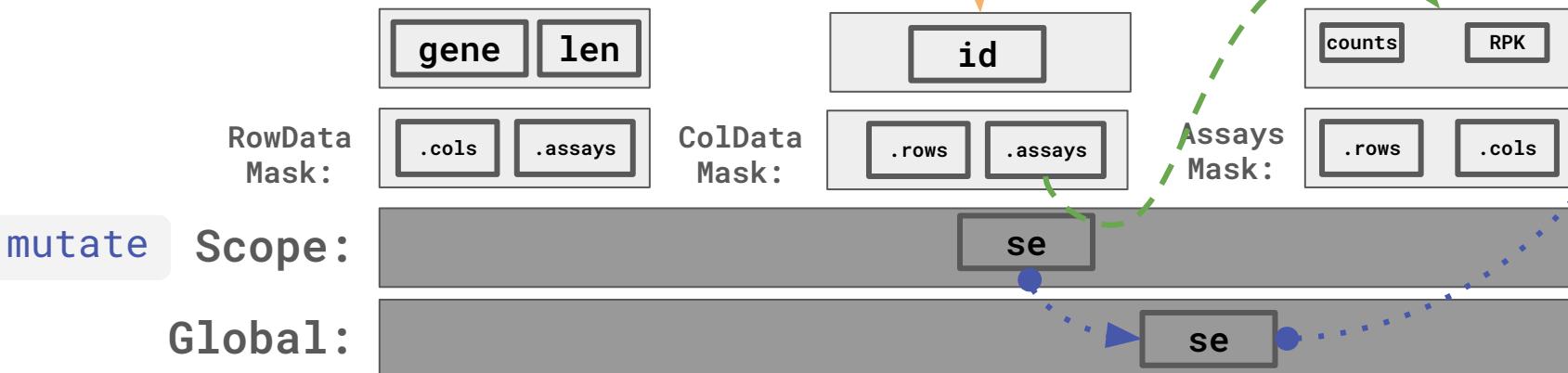
“SummarizedExperiment”



Empty Environment

Examples: Pronouns

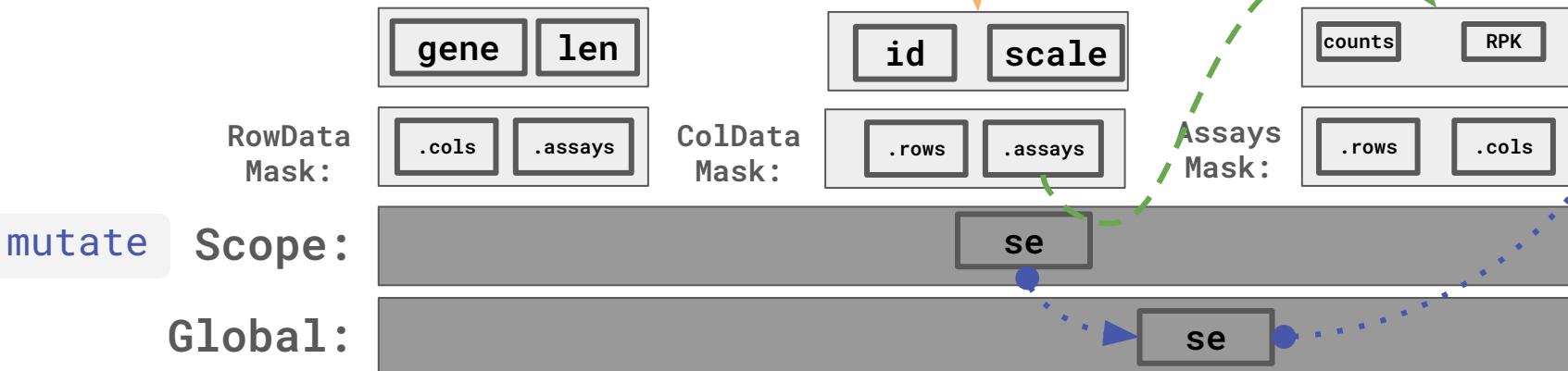
```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5 )
```



“SummarizedExperiment”

Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5 )
```



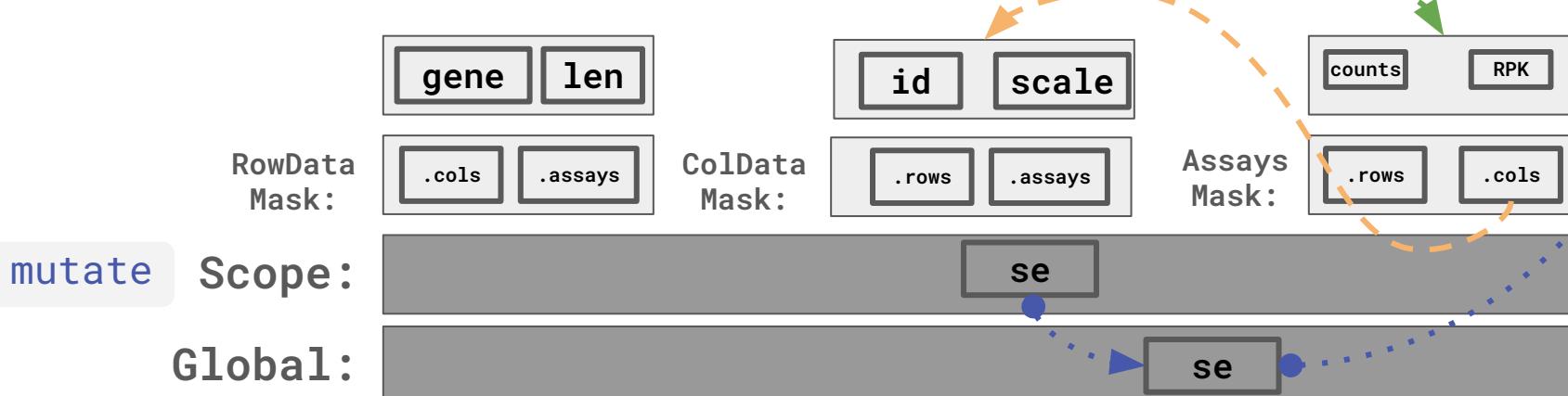
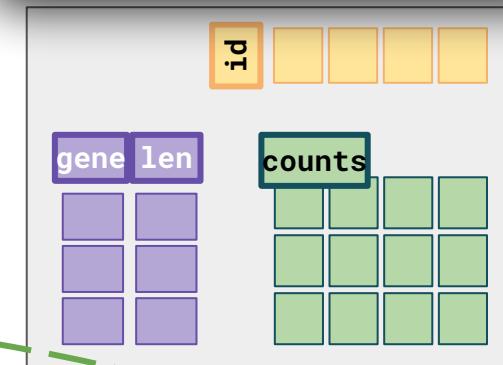
Empty Environment

“SummarizedExperiment”

Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5   TPM = RPK / .cols$scale)
```

“SummarizedExperiment”

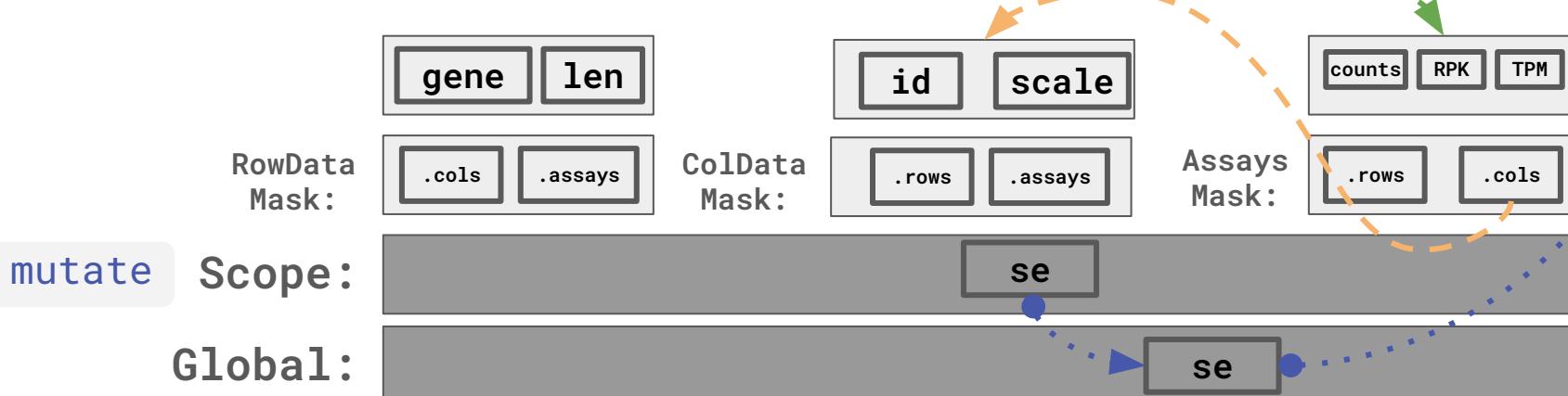
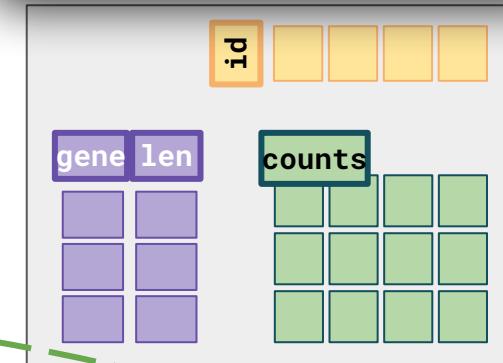


Empty Environment

Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5   TPM = RPK / .cols$scale)
```

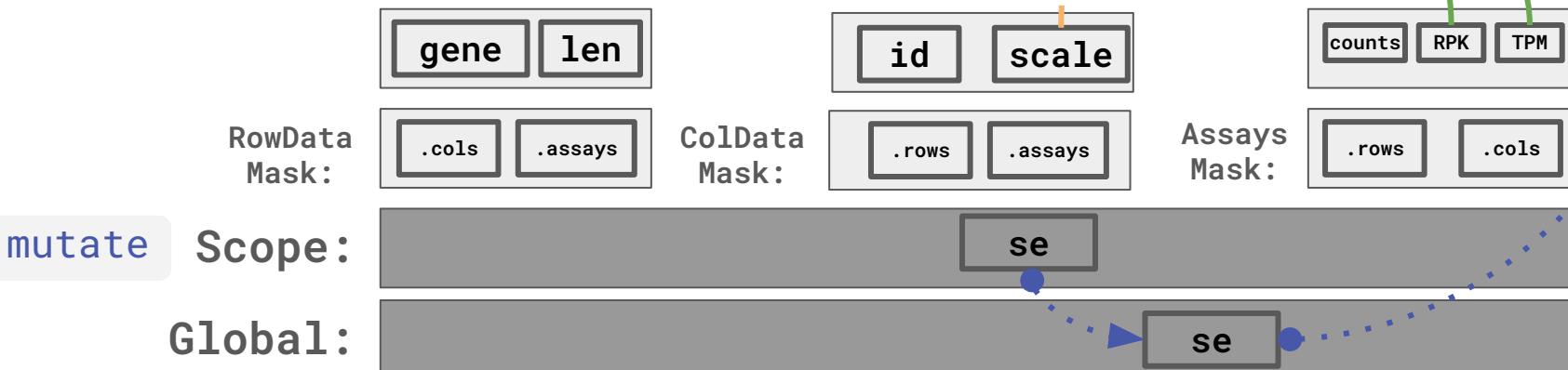
“SummarizedExperiment”



Empty Environment

Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5   TPM = RPK / .cols$scale)
```

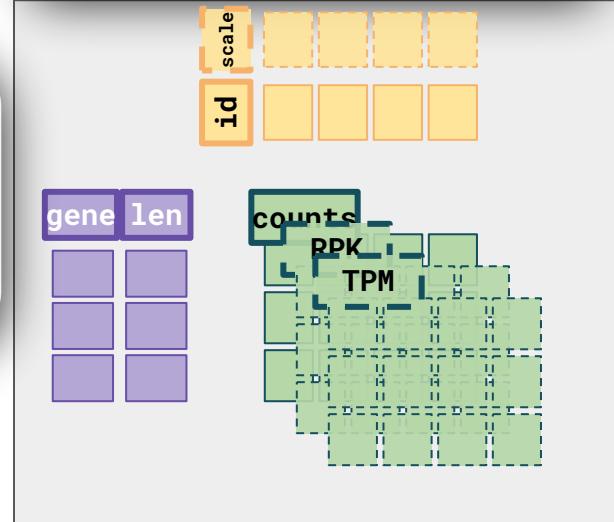


Empty Environment

Examples: Pronouns

```
1 xp <- new_plyxp(se)
2 xp <- mutate(xp,
3   RPK = counts / (.rows$len / 1e3),
4   cols(scale = colSums(.assays_asis$RPK) / 1e6)
5   TPM = RPK / .cols$scale)
```

“SummarizedExperiment”



`mutate`

Scope:

se

Global:

se

Empty Environment

In Summary

```
1 xp <- new_plyxp(se)
2 xp <- group_by(xp, rows(...), cols(...)) %>%
3   filter(cols(...)) %>%
4   summarise(...) %>%
5   mutate(...) %>%
6   arrange(rows(...))
```

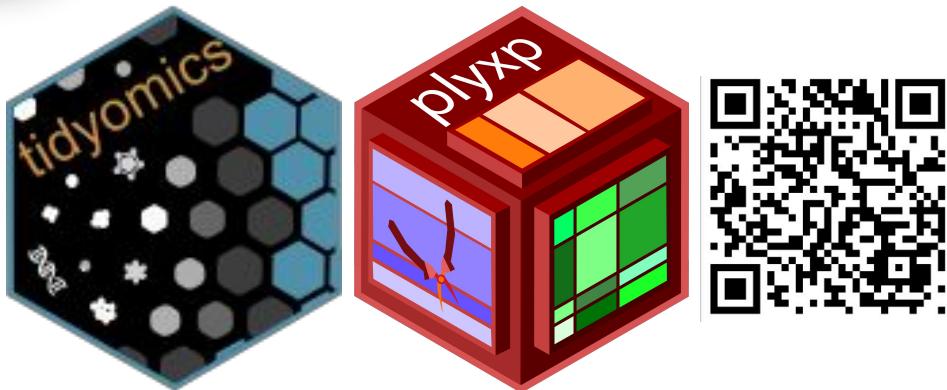
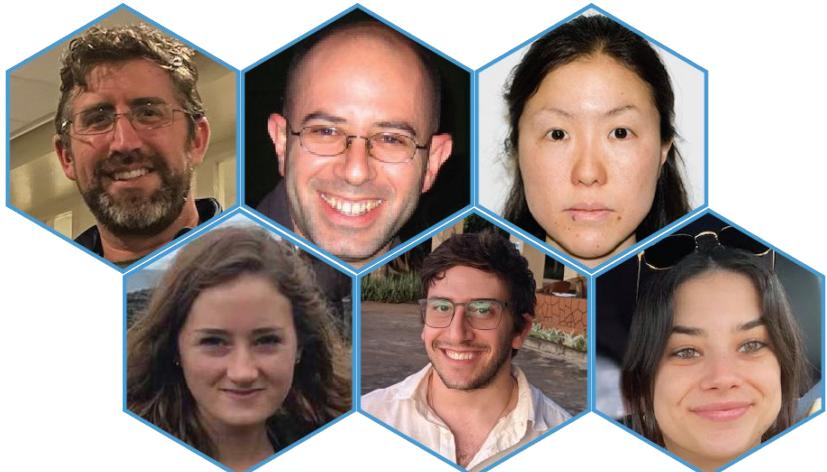
- `plyxp` is an R-based implementation allowing for `tidyverse`-style grammar applied to annotated matrix data
- No ambiguity: evaluation contexts are explicitly annotated, or derived from pronouns.
- Computational flexibility: pronoun accessibility may provide either reshaped or raw data

Looking For Feedback!

<https://github.com/tidyomics>

<https://jtlandis.github.io/plyxp/>

Thanks to the Love Lab!



This work was supported by NHGRI R01 HG009937 and the Wellcome Trust [313919/Z/24/Z].