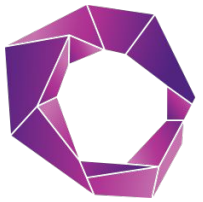
Stefano Mangiola

Maria Doyle

Tidy Transcriptomics for Single-cell RNA Sequencing Analyses



Peter Mac
Peter MacCallum Cancer Centre
Victoria Australia

Resources for #tidytranscriptomics

https://stemangiola.github.io/tidytranscriptomics/

Resources for #tidytranscriptomics

tidybulk: an R tidy framework for modular transcriptomic data analysis

Stefano Mangiola, Ramyar Molania, Ruining Dong, Maria A. Doyle & Anthony T. Papenfuss

Genome Biology **22**, Article number: 42 (2021) | Cite this article

Interfacing Seurat with the R tidy universe

Stefano Mangiola, Maria A Doyle, Anthony T Papenfuss

Bioinformatics, btab404, https://doi.org/10.1093/bioinformatics/btab404

https://stemangiola.github.io/tidytranscriptomics/

Tidy R tools

There are four basic principles to a tidy API:

- Reuse existing data structures.
- Compose simple functions with the pipe.
- Embrace functional programming.
- Design for humans.

```
# A tibble: 100 x 8
   observation    variable_1 variable_2
   <glue>         <chr>      <chr>
 1 observation 1  ...        ...
 2 observation 2  ...        ...
 3 observation 3  ...        ...
 4 observation 4  ...        ...
 5 observation 5  ...        ...
 6 observation 6  ...        ...
 7 observation 7  ...        ...
 8 observation 8  ...        ...
 9 observation 9  ...        ...
10 observation 10 ...        ...
# … with 90 more rows
```

Tidy R tools

There are four basic principles to a tidy API:

- Reuse existing data structures.
- Compose simple functions with the pipe.
- Embrace functional programming.
- Design for humans.

```
# A tibble: 100 x 8
   observation    variable_1 variable_2 variable_3
   <glue>         <chr>      <chr>      <list>
 1 observation 1  ...        ...        <gg>
 2 observation 2  ...        ...        <gg>
 3 observation 3  ...        ...        <gg>
 4 observation 4  ...        ...        <gg>
 5 observation 5  ...        ...        <gg>
 6 observation 6  ...        ...        <gg>
 7 observation 7  ...        ...        <gg>
 8 observation 8  ...        ...        <gg>
 9 observation 9  ...        ...        <gg>
10 observation 10 ...        ...        <gg>
# … with 90 more rows
```

Tidy R tools

There are four basic principles to a tidy API:

- Reuse existing data structures.
- Compose simple functions with the pipe.
- Embrace functional programming.
- Design for humans.

```
# A tibble: 100 x 8
   observation    variable_1 variable_2 variable_3 variable_4
   <glue>         <chr>      <chr>      <list>     <list>
 1 observation 1  ...        ...        <gg>       <tibble [10 × 2]>
 2 observation 2  ...        ...        <gg>       <tibble [10 × 2]>
 3 observation 3  ...        ...        <gg>       <tibble [10 × 2]>
 4 observation 4  ...        ...        <gg>       <tibble [10 × 2]>
 5 observation 5  ...        ...        <gg>       <tibble [10 × 2]>
 6 observation 6  ...        ...        <gg>       <tibble [10 × 2]>
 7 observation 7  ...        ...        <gg>       <tibble [10 × 2]>
 8 observation 8  ...        ...        <gg>       <tibble [10 × 2]>
 9 observation 9  ...        ...        <gg>       <tibble [10 × 2]>
10 observation 10 ...        ...        <gg>       <tibble [10 × 2]>
# … with 90 more rows
```

Tidy R tools

There are four basic principles to a tidy API:

- Reuse existing data structures.
- Compose simple functions with the pipe.
- Embrace functional programming.
- Design for humans.

```
# A tibble: 100 x 8
   observation    variable_1 variable_2 variable_3 variable_4        variable_5
   <glue>         <chr>      <chr>      <list>     <list>             <list>
 1 observation 1  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 2 observation 2  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 3 observation 3  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 4 observation 4  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 5 observation 5  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 6 observation 6  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 7 observation 7  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 8 observation 8  ...        ...        <gg>       <tibble [10 × 2]> <lm>
 9 observation 9  ...        ...        <gg>       <tibble [10 × 2]> <lm>
10 observation 10 ...        ...        <gg>       <tibble [10 × 2]> <lm>
# … with 90 more rows
```

Tidy R tools

There are four basic principles to a tidy API:

- Reuse existing data structures.
- Compose simple functions with the pipe.
- Embrace functional programming.
- Design for humans.

```
# A tibble: 100 x 8
   observation    variable_1 variable_2 variable_3 variable_4         variable_5 variable_6    variable_7
   <glue>         <chr>      <chr>      <list>     <list>             <list>     <list>        <list>
 1 observation 1  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 2 observation 2  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 3 observation 3  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 4 observation 4  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 5 observation 5  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 6 observation 6  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 7 observation 7  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 8 observation 8  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
 9 observation 9  ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
10 observation 10 ...        ...        <gg>       <tibble [10 × 2]> <lm>       <Seurat[,80]> <SnglCllE[,80…
# … with 90 more rows
```

Tidy R tools

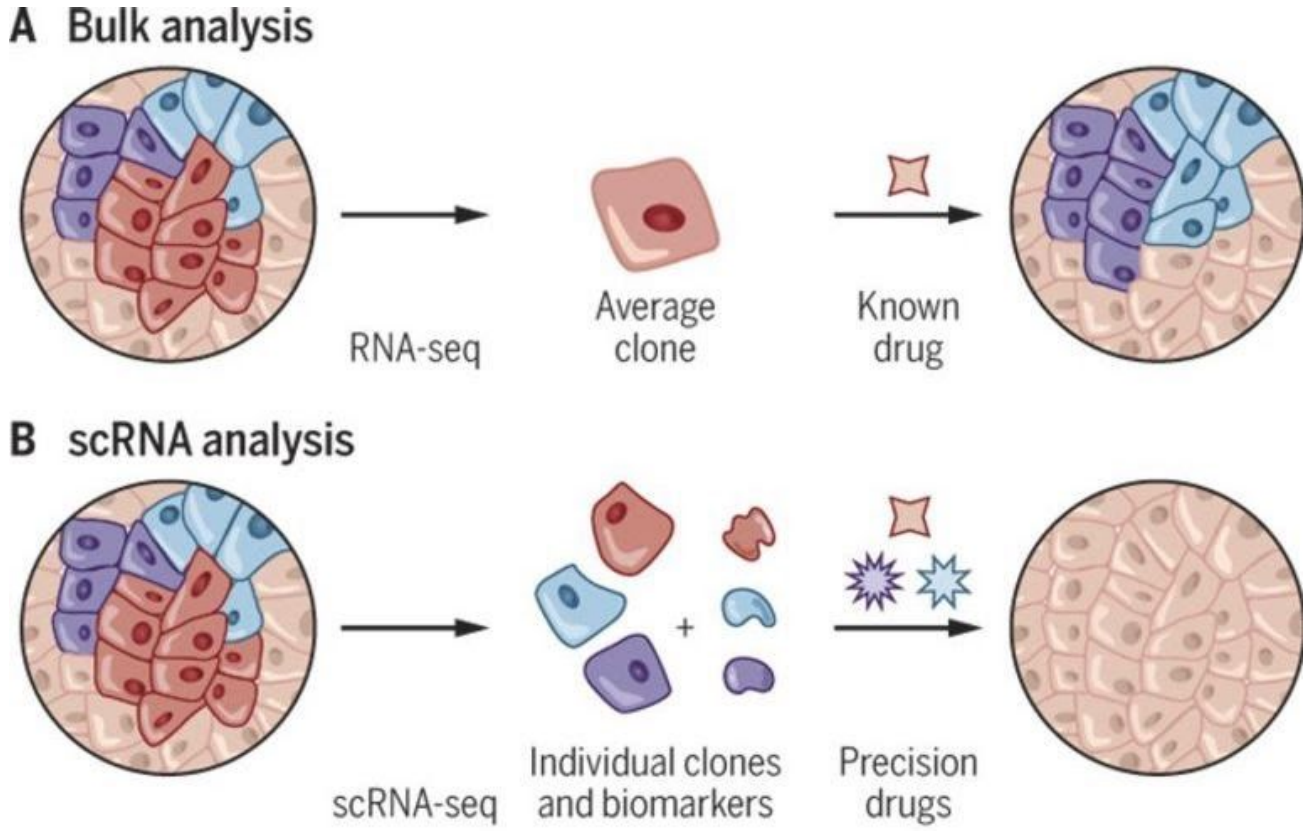
Base R

```
# Filter rows for A class
data_frame  = data_frame[data_frame$class == “A“,]

# Create column
data_frame$x = data_frame$a * data_frame$b

# Plot
plot(data_frame$x, data_frame$y)
```

Tidy R

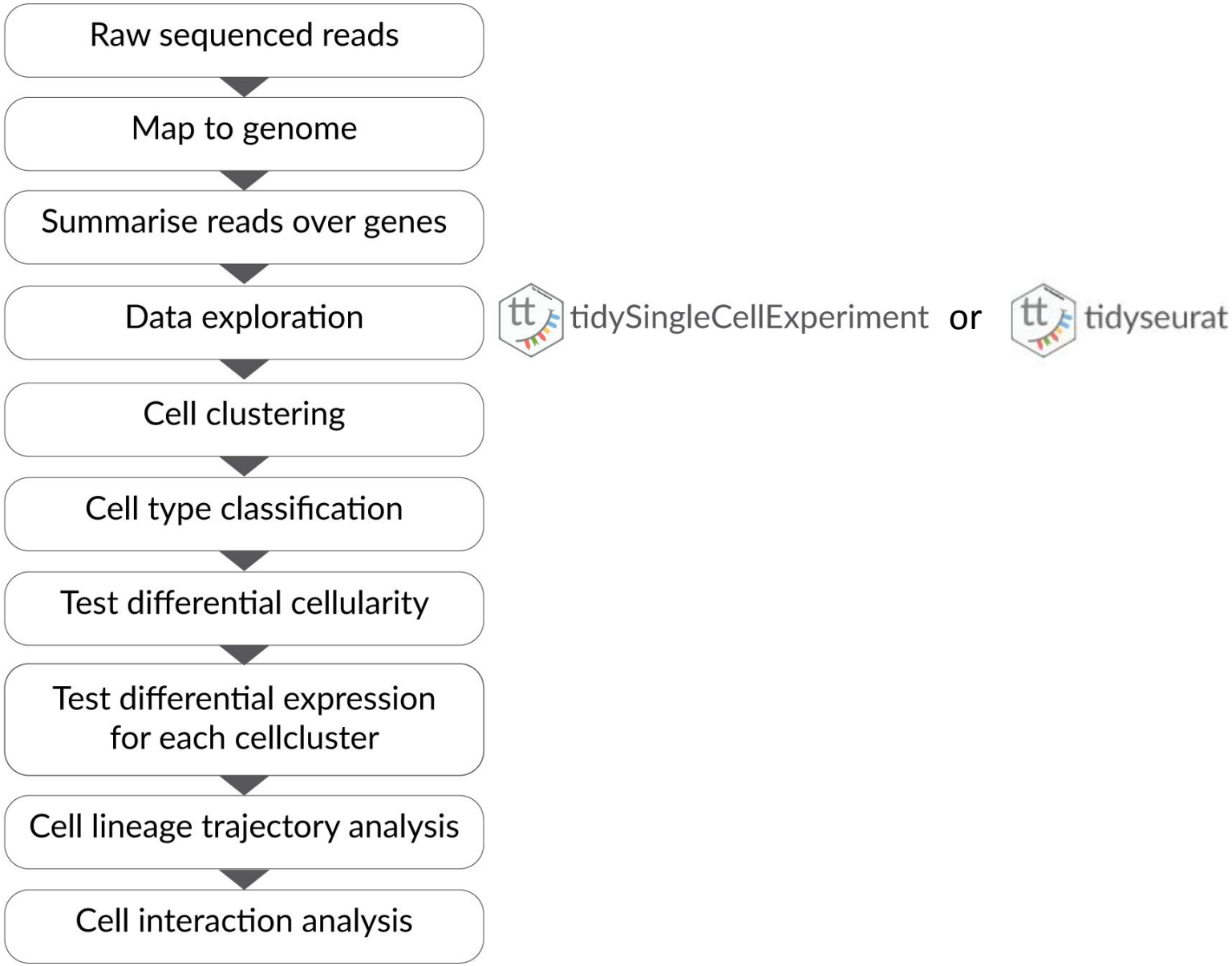
```
data_frame |>

  # Filter rows for A class
  filter(class == “A”) |>

  # Create column
  mutate(x = a * b) |>

  # Plot
  ggplot2.scatterplot(x, y)
```

Single-cell transcriptomics

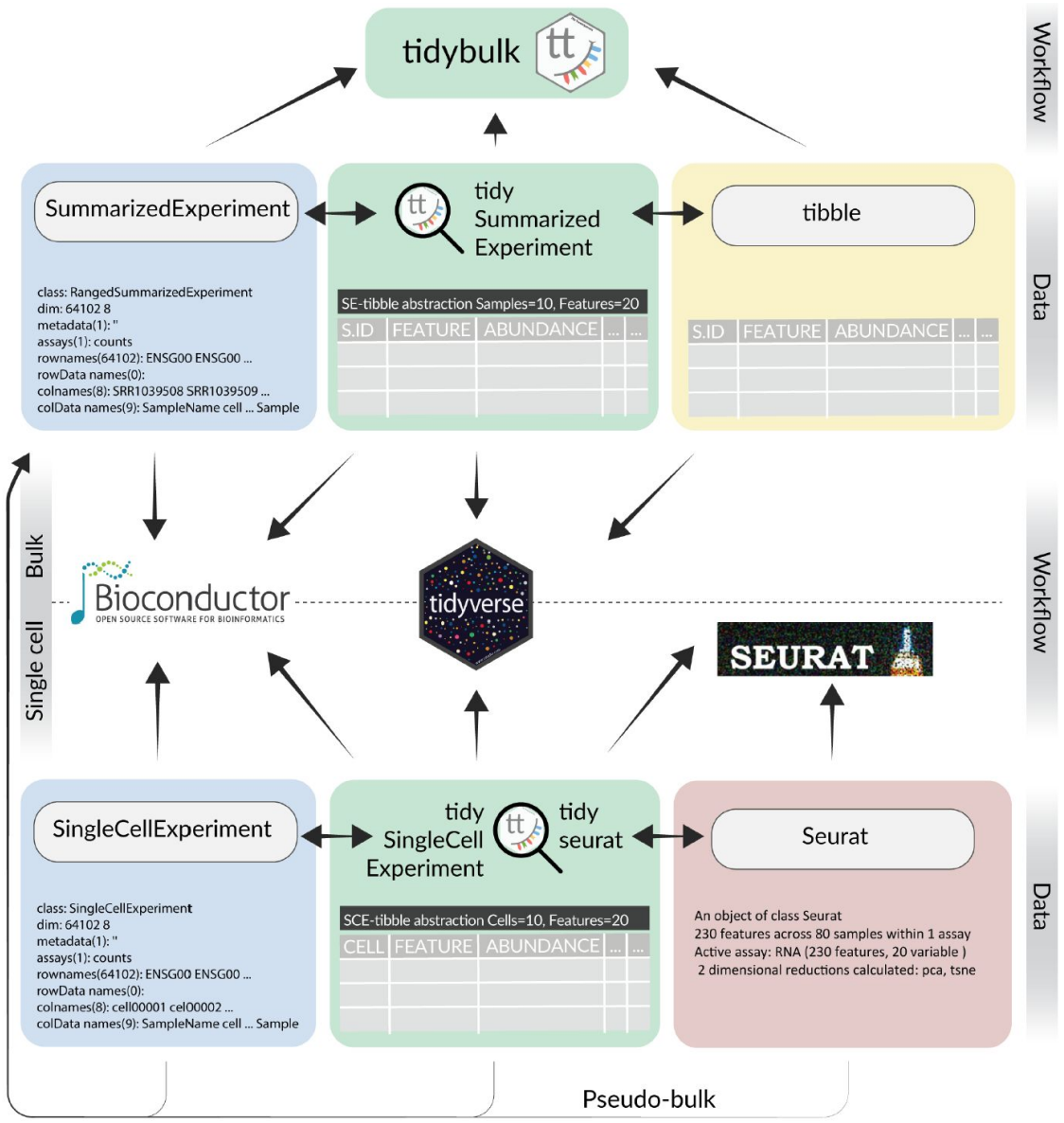


Shalek and Benson, 20174

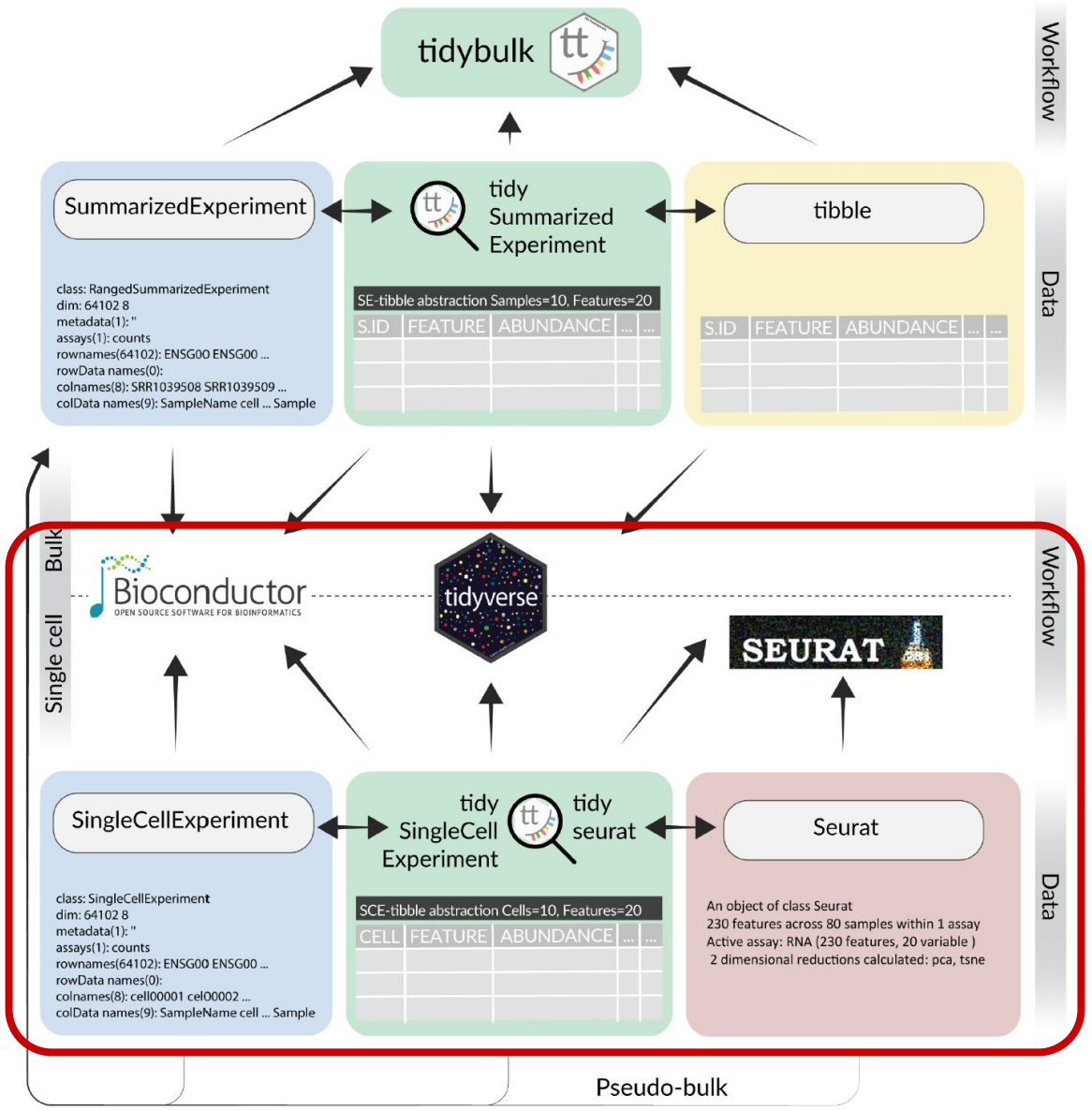
Single-cell transcriptomics workflow



The big picture

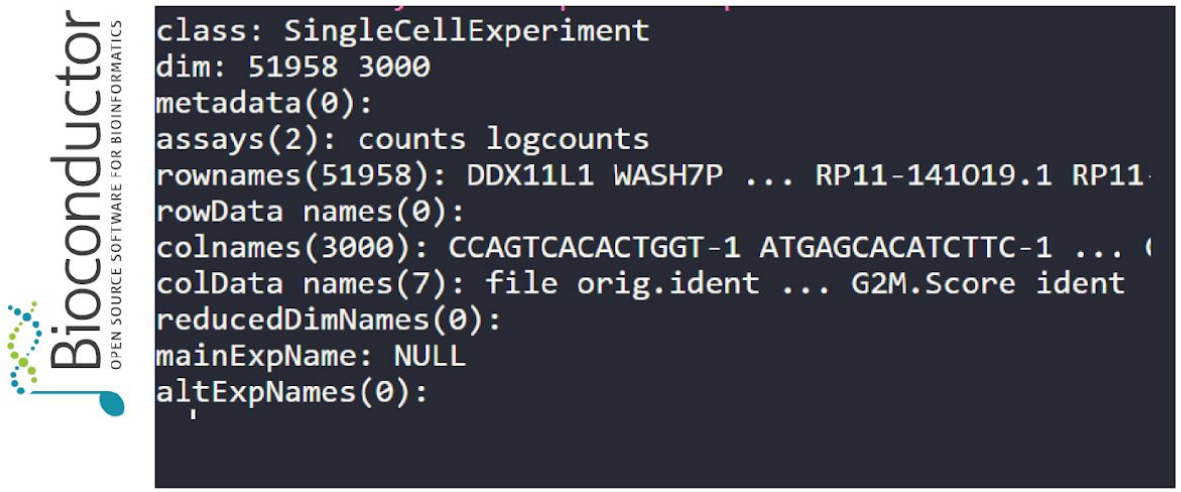


The big picture



Analysis infrastructure for single-cell data

Data container



Analysis infrastructure for single-cell data

Data container

```
class: SingleCellExperiment
dim: 51958 3000
metadata(0):
assays(2): counts logcounts
rownames(51958): DDX11L1 WASH7P ... RP11-141O19.1 RP11
rowData names(0):
colnames(3000): CCAGTCACACTGGT-1 ATGAGCACATCTTC-1 ...
colData names(7): file orig.ident ... G2M.Score ident
reducedDimNames(0):
mainExpName: NULL
altExpNames(0):
```

Analysis

Bioconductor community

scran/scater

Manipulation

```
colData(data)
reducedDims(data, “umap”)
subset(data, , class==”A”)
data$info = info

data = data |> cbind(cohort_info[
    match(data$sample, cohort_info$sample)
    ,])
subset(data, , !is.na(sample_id))
```

Tidy data representation

Data container

Bioconductor
OPEN SOURCE SOFTWARE FOR BIOINFORMATICS

```
# A SingleCellExperiment-tibble abstraction: 80 x 15
# Features=230 | Assays=counts, logcounts
   cell             orig.ident    nCount_RNA nFeature_RNA RNA_snn_res.0.8 letter.idents groups RNA_snn_res.1    PC_1   PC_2   PC_3  PC_4   PC_5 tSNE_1 tSNE_2
   <chr>            <fct>              <dbl>        <int> <fct>           <fct>         <chr>  <fct>           <dbl>  <dbl>  <dbl> <dbl>  <dbl>  <dbl>  <dbl>
 1 ATGCCAGAACGACT   SeuratProject         70           47 0               A             g2     0             -0.774  -0.900 -0.249 0.559  0.465  0.868  -8.10
 2 CATGGCCTGTGCAT   SeuratProject         85           52 0               A             g1     0             -0.0260 -0.347  0.665 0.418  0.585 -7.39   -8.77
 3 GAACCTGATGAACC   SeuratProject         87           50 1               B             g2     0             -0.457   0.180  1.32  2.01  -0.482 -28.2    0.241
 4 TGACTGGATTCTCA   SeuratProject        127           56 0               A             g2     0             -0.812  -1.38  -1.00  0.139 -1.60   16.3   -11.2
 5 AGTCAGACTGCACA   SeuratProject        173           53 0               A             g2     0             -0.774  -0.900 -0.249 0.559  0.465  1.91   -11.2
 6 TCTGATACACGTGT   SeuratProject         70           48 0               A             g1     0             -0.774  -0.900 -0.249 0.559  0.465  3.15   -9.94
 7 TGGTATCTAAACAG   SeuratProject         64           36 0               A             g1     0             -0.460  -1.19  -0.312 0.716 -1.65   17.9    -9.90
 8 GCAGCTCTGTTTCT   SeuratProject         72           45 0               A             g1     0             -0.900  -0.388  0.693 0.404  0.536 -6.49   -8.39
 9 GATATAACACGCAT   SeuratProject         52           36 0               A             g1     0             -0.774  -0.900 -0.249 0.559  0.465  1.33   -9.68
10 AATGTTGACAGTCA   SeuratProject        100           41 0               A             g1     0             -0.488  -1.16  -0.306 0.702 -1.47   17.0    -9.43
# … with 70 more rows
```

Tidy analysis infrastructure

Data container

```
# A SingleCellExperiment-tibble abstraction: 80 x 15
# Features=230 | Assays=counts, logcounts
   cell             orig.ident    nCount_RNA nFeature_RNA RNA_snn_res.0.8
   <chr>            <fct>              <dbl>        <int> <fct>
 1 ATGCCAGAACGACT   SeuratProject         70           47 0
 2 CATGGCCTGTGCAT   SeuratProject         85           52 0
 3 GAACCTGATGAACC   SeuratProject         87           50 1
 4 TGACTGGATTCTCA   SeuratProject        127           56 0
 5 AGTCAGACTGCACA   SeuratProject        173           53 0
 6 TCTGATACACGTGT   SeuratProject         70           48 0
 7 TGGTATCTAAACAG   SeuratProject         64           36 0
 8 GCAGCTCTGTTTCT   SeuratProject         72           45 0
 9 GATATAACACGCAT   SeuratProject         52           36 0
10 AATGTTGACAGTCA   SeuratProject        100           41 0
# … with 70 more rows
```

Analysis

Bioconductor community

Manipulation

```
data
data |> select(contains("UMAP"))
data |> filter(class=="A")
data |> mutate(info = info)

data |> inner_join(cohort_info, by="sample")
```

Tidyseurat and tidySingleCellExperiment

Data container

Analysis

Manipulation


```
# A SingleCellExperiment-tibble abstraction: 80 x 15
# Features=230 | Assays=counts, logcounts
   cell            orig.ident    nCount_RNA nFeature_RNA RNA_snn_res.0.8
   <chr>           <fct>              <dbl>        <int> <fct>
 1 ATGCCAGAACGACT SeuratProject         70           47 0
 2 CATGGCCTGTGCAT SeuratProject         85           52 0
 3 GAACCTGATGAACC SeuratProject         87           50 1
 4 TGACTGGATTCTCA SeuratProject        127           56 0
 5 AGTCAGACTGCACA SeuratProject        173           53 0
 6 TCTGATACACGTGT SeuratProject         70           48 0
 7 TGGTATCTAAACAG SeuratProject         64           36 0
 8 GCAGCTCTGTTTCT SeuratProject         72           45 0
 9 GATATAACACGCAT SeuratProject         52           36 0
10 AATGTTGACAGTCA SeuratProject        100           41 0
# … with 70 more rows
```

Bioconductor community

```
data
data |> select(contains("UMAP"))
data |> filter(class=="A")
data |> mutate(info = info)

data |> inner_join(cohort_info, by="sample")
```


```
# A Seurat-tibble abstraction: 80 x 15
# Features=230 | Active assay=RNA | Assays=RNA
   cell            orig.ident    nCount_RNA nFeature_RNA RNA_snn_res.0.8
   <chr>           <fct>              <dbl>        <int> <fct>
 1 ATGCCAGAACGACT SeuratProject         70           47 0
 2 CATGGCCTGTGCAT SeuratProject         85           52 0
 3 GAACCTGATGAACC SeuratProject         87           50 1
 4 TGACTGGATTCTCA SeuratProject        127           56 0
 5 AGTCAGACTGCACA SeuratProject        173           53 0
 6 TCTGATACACGTGT SeuratProject         70           48 0
 7 TGGTATCTAAACAG SeuratProject         64           36 0
 8 GCAGCTCTGTTTCT SeuratProject         72           45 0
 9 GATATAACACGCAT SeuratProject         52           36 0
10 AATGTTGACAGTCA SeuratProject        100           41 0
# … with 70 more rows
```

Seurat
SeuratWrappers
community

```
data
data |> select(contains("UMAP"))
data |> filter(class=="A")
data |> mutate(info = info)

data |> inner_join(cohort_info, by="sample")
```

Tidy operators available

as_tibble()
mutate()
bind_rows()
left_join() inner_join() *_join()
select() **distinct()**
count() add_count() **summarise()**
pull() slice()
filter() sample_n() sample_frac()
rename()
separate() unite() extract()
nest() unnest() map_*()
pivot_longer()
join_features()
ggplot()
plotly()

```
pbmc %>% select(cell, nCount_RNA , ident)
#> # A SingleCellExperiment-tibble abstraction: 3,000 × 3
#> [90m# Features=51958 | Assays=counts, logcounts[39m
#>    cell              nCount_RNA ident
#>    <chr>                  <dbl> <fct>
#>  1 CCAGTCACACTGGT-1        3421 SingleCellExperiment
#>  2 ATGAGCACATCTTC-1        2752 SingleCellExperiment
#>  3 TATGAATGGAGGAC-1        2114 SingleCellExperiment
#>  4 CATATAGACTAAGC-1        3122 SingleCellExperiment
#>  5 GAGGCAGACTTGCC-1        2341 SingleCellExperiment
#>  6 AGCTGCCTTTCATC-1        5472 SingleCellExperiment
#>  7 TGATTAGATGACTG-1        1258 SingleCellExperiment
#>  8 ACGAAGCTCTGAGT-1        7683 SingleCellExperiment
#>  9 CGGCATCTTCGTAG-1        3500 SingleCellExperiment
#> 10 ATAGCGTGCCCTTG-1        3092 SingleCellExperiment
#> # … with 2,990 more rows
```

Tidy operators available

as_tibble()
mutate()
bind_rows()
left_join() inner_join() *_join()
select() **distinct()**
count() add_count() **summarise()**
pull() slice()
filter() sample_n() sample_frac()
rename()
separate() unite() extract()
nest() unnest() map_*()
pivot_longer()
join_features()
ggplot()
plotly()

```
pbmc %>% bind_rows(pbmc)
#> Warning in bind_rows.SingleCellExperiment(., pbmc): tidySingleCel
#> says: you have duplicated cell names, they will be made unique.
#> # A SingleCellExperiment-tibble abstraction: 6,000 × 8
#> [90m# Features=51958 | Assays=counts, logcounts[39m
#>    cell   file      orig.ident nCount_RNA nFeature_RNA  S.Score G
#>    <chr>  <chr>     <chr>           <dbl>        <int>    <dbl>
#>  1 CCAGT… ../data/… SeuratPro…       3421          979 -5.42e-2
#>  2 ATGAG… ../data/… SeuratPro…       2752          898 -5.01e-2
#>  3 TATGA… ../data/… SeuratPro…       2114          937 -2.95e-5
#>  4 CATAT… ../data/… SeuratPro…       3122         1086 -6.65e-2
#>  5 GAGGC… ../data/… SeuratPro…       2341          957 -3.74e-3
#>  6 AGCTG… ../data/… SeuratPro…       5472         1758 -5.88e-2
#>  7 TGATT… ../data/… SeuratPro…       1258          542 -2.51e-2
#>  8 ACGAA… ../data/… SeuratPro…       7683         1926 -1.33e-1
#>  9 CGGCA… ../data/… SeuratPro…       3500         1092 -6.87e-2
#> 10 ATAGC… ../data/… SeuratPro…       3092          974 -1.24e-2
#> # … with 5,990 more rows
```

Tidy operators available

as_tibble()
mutate()
bind_rows()
left_join() inner_join() *_join()
select() **distinct()**
count() add_count() **summarise()**
pull() slice()
filter() sample_n() sample_frac()
rename()
separate() unite() extract()
nest() unnest() map_*()
pivot_longer()
join_features()
ggplot()
plotly()

```
seurat_obj |>
  join_features(features = c("CD3D",  "CD8A"), shape = "wide", assay = "SCT")
```

```
# A Seurat-tibble abstraction: 36,683 x 18
# Features=3000 | Active assay=integrated | Assays=RNA, SCT, integrated
   cell  file  Barcode batch BCB    S.Score G2M.Score Phase curated_cell_ty…  CD3D  CD8A     PC_1
   <chr> <chr> <chr>   <chr> <chr>    <dbl>     <dbl> <chr> <chr>            <dbl> <dbl>    <dbl>
 1 1_AA… ../d… AAACCC… 1     BCB1… -0.00277   -0.103  G1    CD4+_ribosome_r… 1.10  0       1.19
 2 1_AA… ../d… AAACGA… 1     BCB1…  0.0620    -0.0972 S     CD4+_Tcm_S100A4… 1.10  0      -2.06
 3 1_AA… ../d… AAACGC… 1     BCB1…  0.0943    -0.193  S     CD8+_high_ribon… 1.79  0.693   1.11
 4 1_AA… ../d… AAACGC… 1     BCB1… -0.0676    -0.205  G1    CD4+_ribosome_r… 1.39  0      -2.57
 5 1_AA… ../d… AAACGC… 1     BCB1…  0.0216    -0.124  S     CD4+_Tcm_S100A4… 1.61  0      -2.38
 6 1_AA… ../d… AAAGAA… 1     BCB1…  0.0598    -0.196  S     MAIT             0.693 0      -2.31
 7 1_AA… ../d… AAAGGA… 1     BCB1…  0.0281    -0.0540 S     CD8+_high_ribon… 1.10  0       3.34
 8 1_AA… ../d… AAAGGG… 1     BCB1…  0.0110    -0.0998 S     MAIT             1.39  0      -0.00363
 9 1_AA… ../d… AAAGGG… 1     BCB1…  0.0341    -0.143  S     CD8+_high_ribon… 0     0       1.01
10 1_AA… ../d… AAAGTC… 1     BCB1…  0.0425    -0.183  S     MAIT             0.693 0      -0.860
# … with 36,673 more rows, and 1 more variable: UMAP_2 <dbl>
```

Tidy operators available

as_tibble()
mutate()
bind_rows()
left_join() inner_join() *_join()
select() **distinct()**
count() add_count() **summarise()**
pull() slice()
filter() sample_n() sample_frac()
rename()
separate() unite() extract()
nest() unnest() map_*()
pivot_longer()
join_features()
ggplot()
plotly()

```
seurat_obj |>
  nest(data = -curated_cell_type)
```

```
# A tibble: 14 x 2
   curated_cell_type                       data
   <chr>                                   <list>
 1 CD4+_ribosome_rich                      <tidysert[,3468]>
 2 CD4+_Tcm_S100A4_IL32_IL7R_VIM           <tidysert[,4978]>
 3 CD8+_high_ribonucleosome                <tidysert[,4885]>
 4 MAIT                                    <tidysert[,1909]>
 5 CD8+_transitional_effector_GZMK_KLRB1_LYAR4 <tidysert[,4138]>
 6 TCR_V_Delta_2                           <tidysert[,355]>
 7 T_cell:CD8+_other                       <tidysert[,2109]>
 8 NK_cells                                <tidysert[,4535]>
```

What tidy data frameworks are and what are not

NO: data containers
NO: analysis tools

YES: data interface
YES: manipulation, integration, visualisation tools

Therefore, the question "**can we go from tidyseurat to Seurat and vice versa**" is not relevant, as we never leave
Seurat
SingleCellExperiment