# DCATS: Differential composition analysis of single-cell data

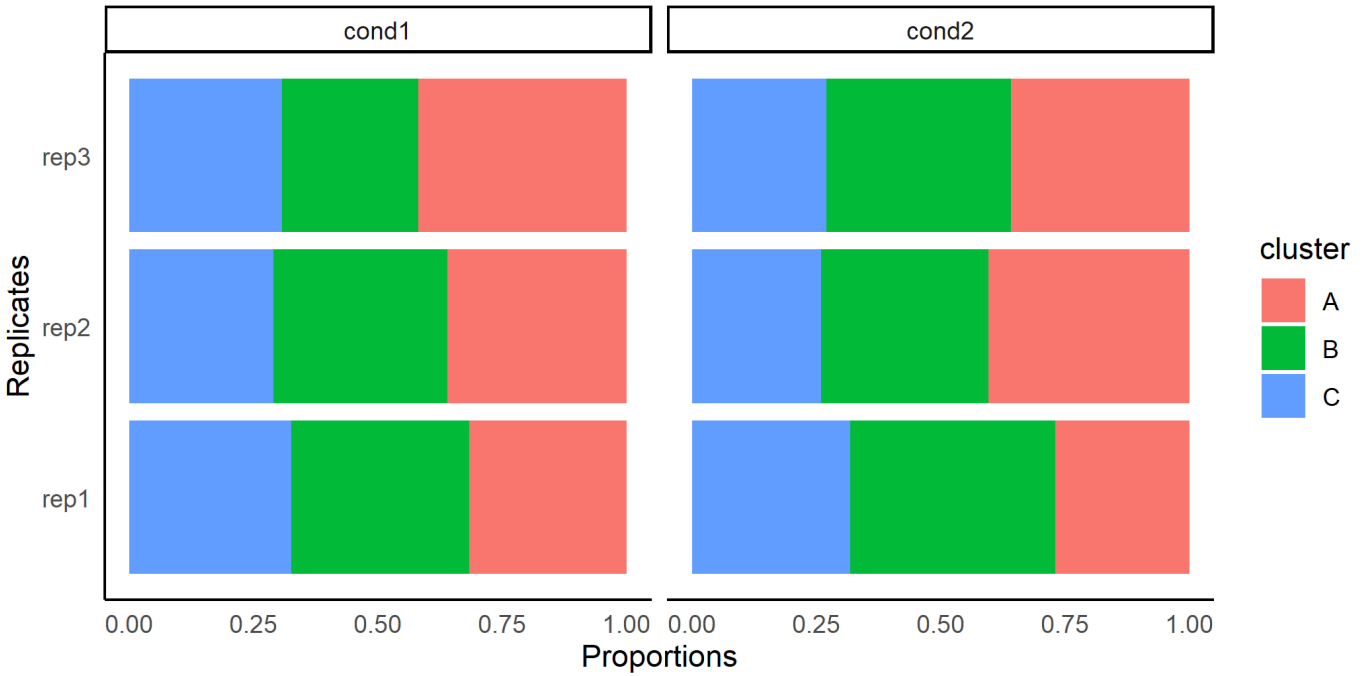
Xinyi Lin

The University of Hong Kong, Ho Lab

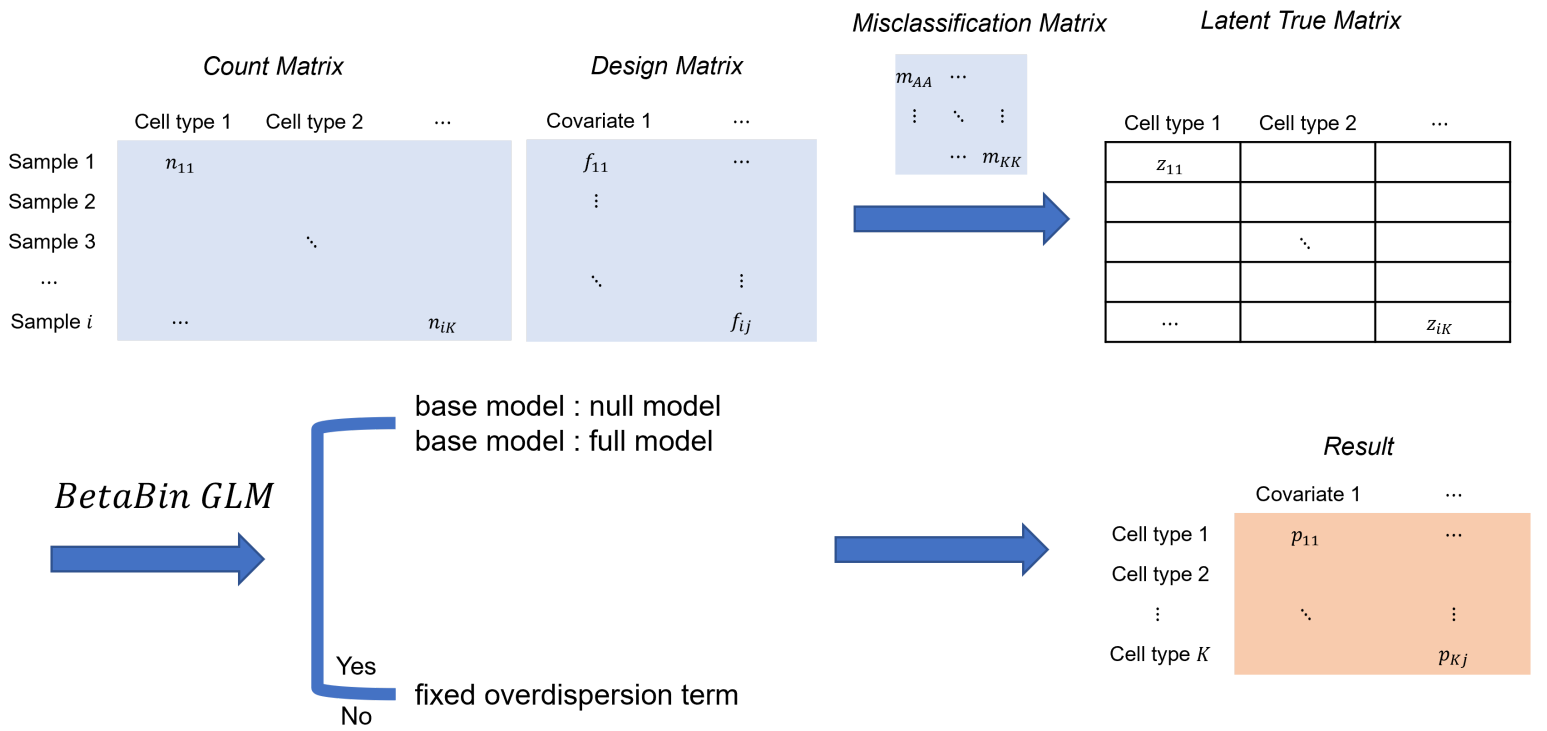
2021/11/01

# Motivation

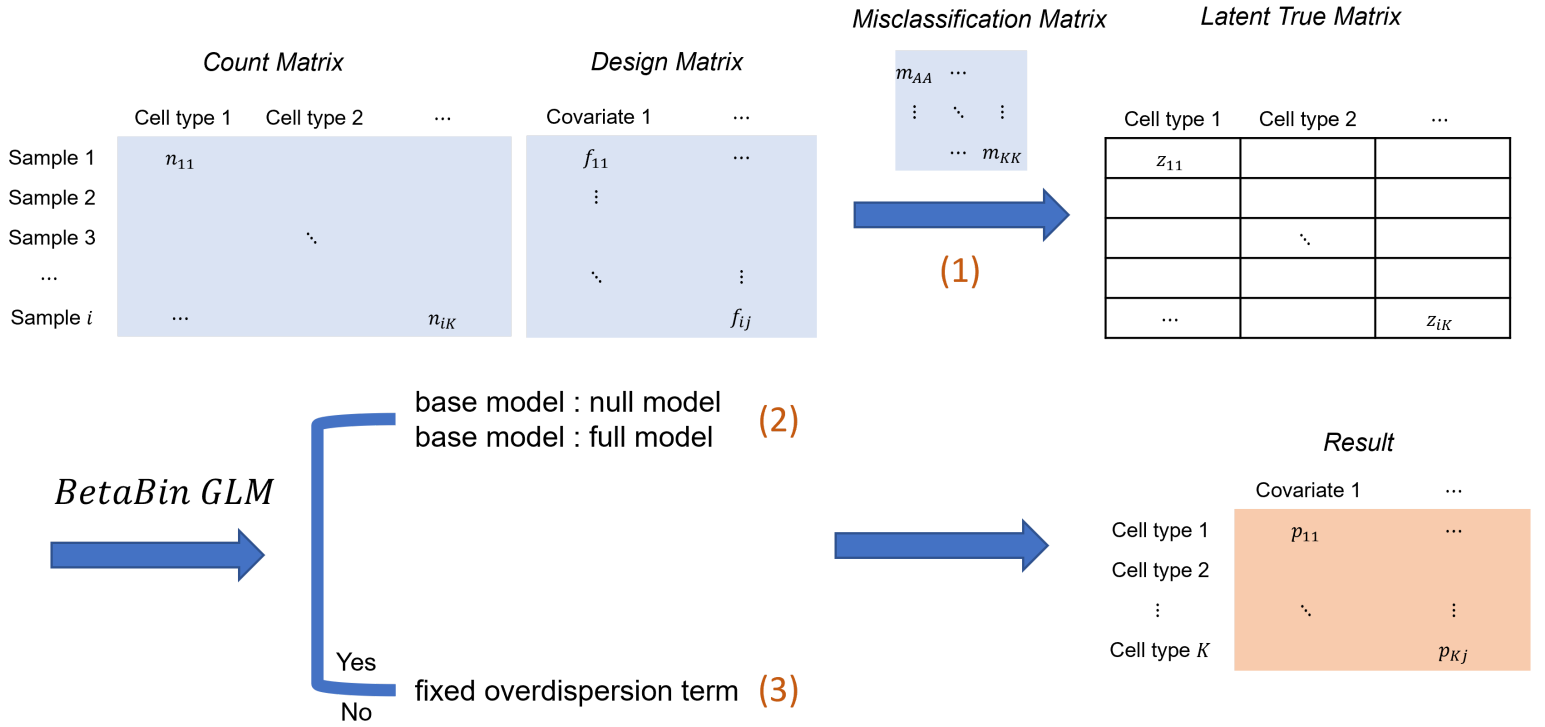
- R package designed for differential composition analysis on single cell data
- Basic assumptions
  - cell counts follow beta-binomial distribution
  - misclassification error exists



# Workflow


Count Matrix
Design Matrix
Misclassification Matrix
Latent True Matrix
Cell type 1
Cell type 2
Covariate 1
Sample 1
Sample 2
Sample 3
Sample i
$n_{11}$
$n_{iK}$
$f_{11}$
$f_{ij}$
$m_{AA}$
$m_{KK}$
$z_{11}$
$z_{iK}$
BetaBin GLM
base model : null model
base model : full model
Yes
No
fixed overdispersion term
Result
Covariate 1
Cell type 1
Cell type 2
Cell type K
$p_{11}$
$p_{Kj}$

# Workflow


Count Matrix
Design Matrix
Misclassification Matrix
Latent True Matrix
Cell type 1
Cell type 2
Covariate 1
Sample 1
Sample 2
Sample 3
Sample i
(1)
BetaBin GLM
base model : null model
base model : full model
(2)
Yes
No
fixed overdispersion term
(3)
Result
Cell type K

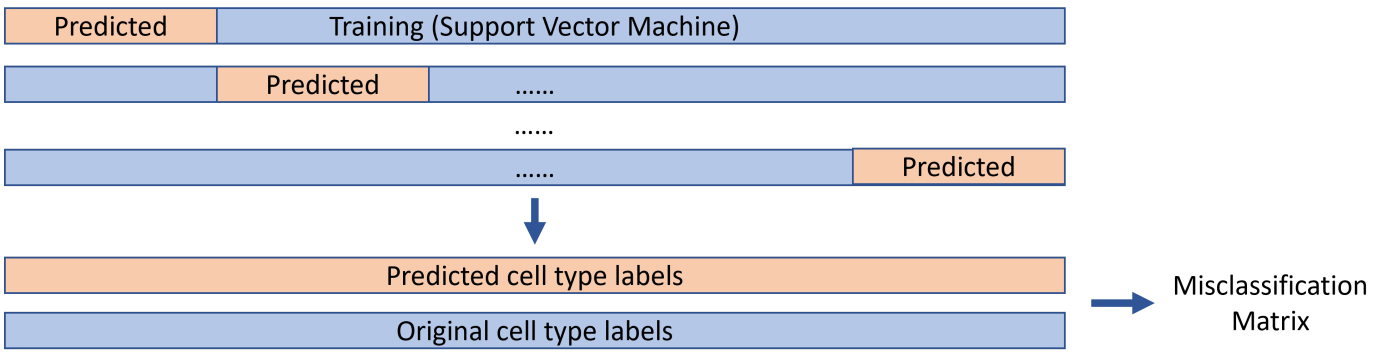
# Method: Misclassification Correction

- Uniform type:

$$\begin{bmatrix} a & (1-a)/(K-1) & \dots & (1-a)/(K-1) \\ (1-a)/(K-1) & a & \dots & (1-a)/(K-1) \\ \vdots & \vdots & \ddots & \vdots \\ (1-a)/(K-1) & (1-a)/(K-1) & \dots & a \end{bmatrix}$$

- KNN type: $m_{ij}$ = % of cluster $i$'s neighborhoods $\in$ cluster $j$

- SVM type:



# Method: Models Selection

- Type 1: Null model

Model 0: $g(\bar{y}) = \beta_0$

Model 1: $g(\bar{y}) = \beta_0 + \beta_1 * tested\_covariate$

- Type 2: Full model

Model 0: $g(\bar{y}) = \beta_0 + \beta * other\_covariates$

Model 1: $g(\bar{y}) = \beta_0 + \beta * other\_covariates + \beta_i * tested\_covariate$

| | age | sex | sample_type | condition |
|---|---|---|---|---|
| S-HC003 | 46 | M | fresh PBMC | control |
| S-HC004 | 34 | M | fresh PBMC | control |
| S-HC005 | 37 | F | fresh PBMC | control |
| S-HC006 | 27 | M | fresh PBMC | control |
| S-HC007 | 27 | M | fresh PBMC | control |
| S-HC008 | 44 | M | fresh PBMC | control |

# Method: Determine Over-dispersion

- Numbers of cells follow beta-binomial distribution:

$$P(Y = y|n, p) = \binom{n}{y} = p^y(1-p)^{n-y}$$

$$f(p|a, b) = \frac{1}{B(a, b)} p^{a-1}(1-p)^{b-1}$$

$$E(Y|n, \pi, \phi) = n\pi, Var(Y|n, \pi, \phi) = n\pi(1-\pi)[1 + (n-1) \times \phi]$$

- Without fixed over-dispersion term :
  - $\phi$ is estimated in each beta-binomial GLM for each cell type

- With fixed over-dispersion term :
  - $\phi$ is estimated across all cell types before testing
  - is given in each beta-binomial GLM for each cell type

# How to use DCATS

- **Count Matrix**

```
library(DCATS)
data("Haber2017")
rbind(Haber2017$count_ctrl, Haber2017$count_Hpoly3)
```

```
##     Endocrine Enterocyte Enterocyte.Progenitor Goblet Stem  TA TA.Ear
## B1         36         59                   136     36  239 125      1
## B2          5         46                    23     20   50  11      
## B3         45         98                   188    124  250 155      3
## B4         26        221                   198     36  131 130      1
## B5         52         75                   347     66  323 263      3
## B6         65        126                   115     33   65  39      1
```

# How to use DCATS

- **Design Matrix**

```
sim_design = data.frame(condition = c("control", "control", "control",
print(sim_design)
```

```
##   condition
## 1   control
## 2   control
## 3   control
## 4   control
## 5    Hpoly3
## 6    Hpoly3
```

# How to use DCATS

- **Design Matrix**

```
data("Ren2021")
print(head(Ren2021$designM, 10))
```

```
##          age sex sample_type   state
## S-HC003   46   M  fresh PBMC control
## S-HC004   34   M  fresh PBMC control
## S-HC005   37   F  fresh PBMC control
## S-HC006   27   M  fresh PBMC control
## S-HC007   27   M  fresh PBMC control
## S-HC008   44   M  fresh PBMC control
## S-HC009   29   M  fresh PBMC control
## S-HC010   58   M  fresh PBMC control
## S-HC011   35   M  fresh PBMC control
## S-HC012   33   M  fresh PBMC control
```

# How to use DCATS

- **Misclassification Matrix (a K $\times$ K matrix)**

```
Haber2017$svm_mat
```

| | Endocrine | Enterocyte | Enterocyte.Progenitor | Goblet | Stem |
|---|---|---|---|---|---|
| Endocrine | 0.9785932722 | 0.0006868132 | 0.0005494505 | 0.0093209055 | 0.0102933608 |
| Enterocyte | 0.0000000000 | 0.9800824176 | 0.0170329670 | 0.0000000000 | 0.0000000000 |
| Enterocyte.Progenitor | 0.0000000000 | 0.0185439560 | 0.9401098901 | 0.0000000000 | 0.0005146680 |
| Goblet | 0.0091743119 | 0.0000000000 | 0.0000000000 | 0.9826897470 | 0.0056613484 |
| Stem | 0.0030581040 | 0.0006868132 | 0.0000000000 | 0.0026631158 | 0.8625836336 |
| TA | 0.0000000000 | 0.0000000000 | 0.0335164835 | 0.0000000000 | 0.0761708698 |
| TA.Early | 0.0000000000 | 0.0000000000 | 0.0082417582 | 0.0039946738 | 0.0303654143 |
| Tuft | 0.0091743119 | 0.0000000000 | 0.0005494505 | 0.0013315579 | 0.0144107051 |

| | TA | TA.Early | Tuft |
|---|---|---|---|
| Endocrine | 0.0114537445 | 0.0087548638 | 0.0000000000 |
| Enterocyte | 0.0000000000 | 0.0000000000 | 0.0000000000 |
| Enterocyte.Progenitor | 0.0572687225 | 0.0087548638 | 0.0000000000 |
| Goblet | 0.0017621145 | 0.0019455253 | 0.0000000000 |
| Stem | 0.0933920705 | 0.0345330739 | 0.0000000000 |
| TA | 0.7506607930 | 0.0617704280 | 0.0000000000 |
| TA.Early | 0.0748898678 | 0.8793774319 | 0.0000000000 |
| Tuft | 0.0105726872 | 0.0048638132 | 1.0000000000 |

# How to use DCATS

- **Misclassification Matrix**

```
data("Kang2017")
data("simulation")

# Three ways to calculate similarity matrices
## Uniform type
simil_mat = create_simMat(K = 3, confuse_rate = 0.2)

## KNN type
knn_mat = knn_simMat(KNN_matrix = simulation$knnGraphs, clusters = simu

## SVM type
svm_mat = svm_simMat(dataframe = Kang2017$svmDF)
```

# How to use DCATS

- **Main Function**

```
print(sim_count)
```

```
##      [,1] [,2] [,3]
## [1,]   36   35   29
## [2,]  271  279  250
## [3,]  518  379  403
## [4,]  152  220  228
## [5,]   84   87   79
## [6,]  259  203  238
## [7,]  345  376  379
```

```
print(sim_design)
```

```
##   condition gender
## 1        g1 Female
## 2        g1 Female
## 3        g1 Female
## 4        g1 Female
## 5        g2   Male
## 6        g2 Female
## 7        g2 Female
```

```
## null model, flexible phi
res = dcats_GLM(sim_count, sim_design, similarity_mat = simil_mat)
## full model, flexible phi
res = dcats_GLM(sim_count, sim_design, simil_mat, base_model='FULL')
## null model, fixed phi
phi = getPhi(sim_count, sim_design)
res = dcats_GLM(sim_count, sim_design, simil_mat, fix_phi = phi)
```

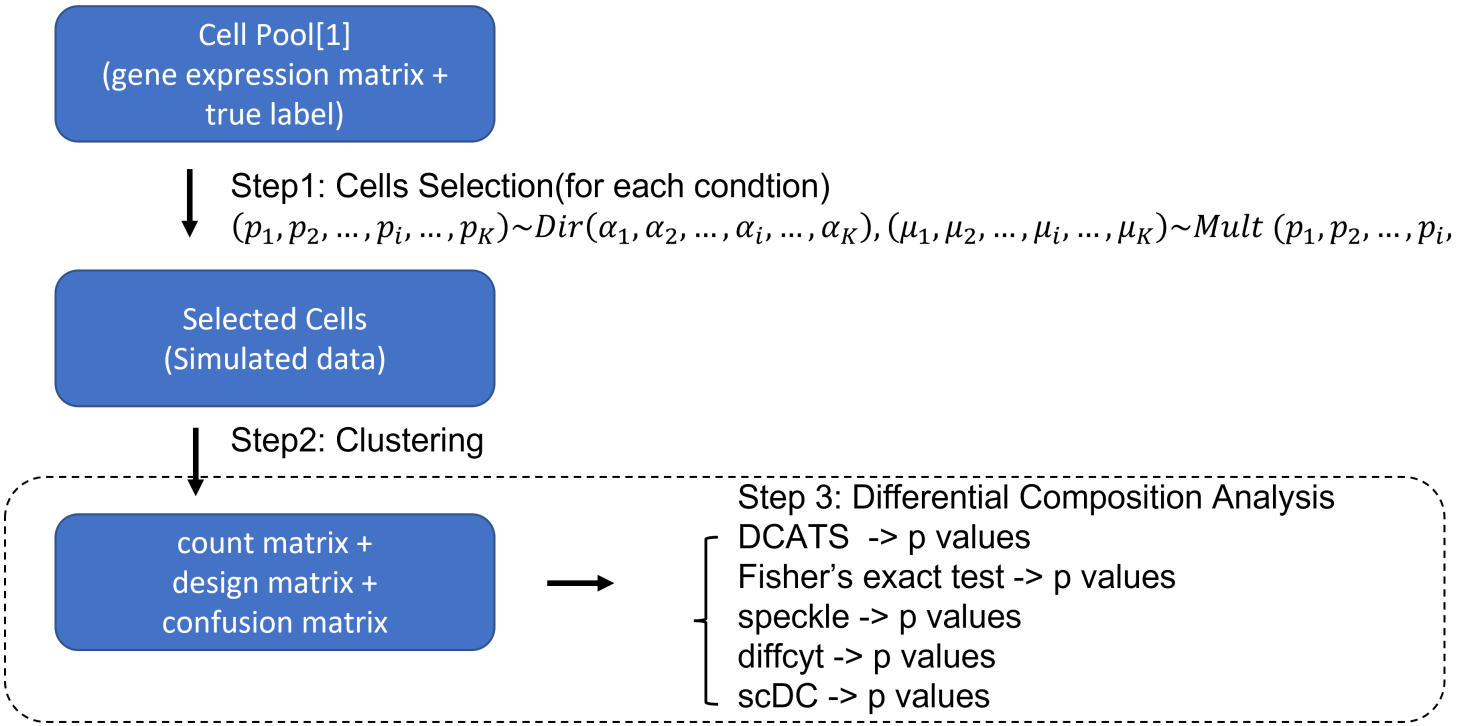
# How to use DCATS

- **Main Function**

```
res = dcats_GLM(sim_count, sim_design, similarity_mat = simil_mat)
print(res$LRT_pvals)
```

```
##              condition    gender
## cell_type_1 0.8939193 0.9903591
## cell_type_2 0.6906093 0.6115926
## cell_type_3 0.5738177 0.6273852
```

# Method: Simulation Design

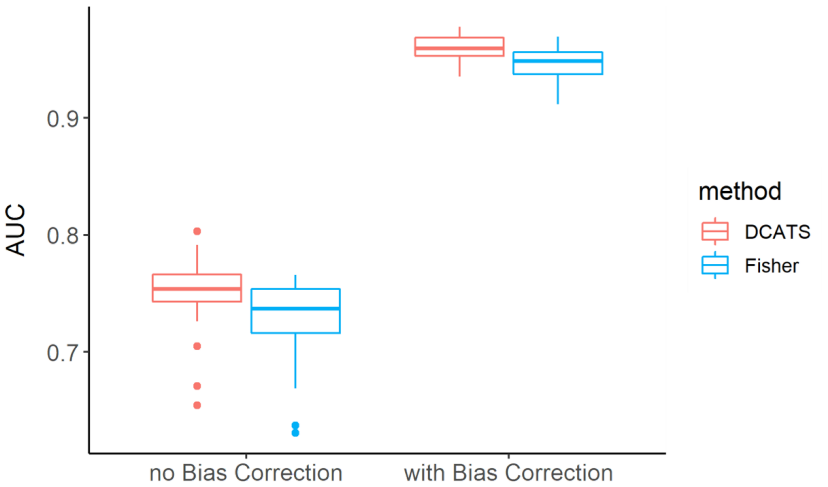


# Results: Simulation

- **Theoretical Simulation**

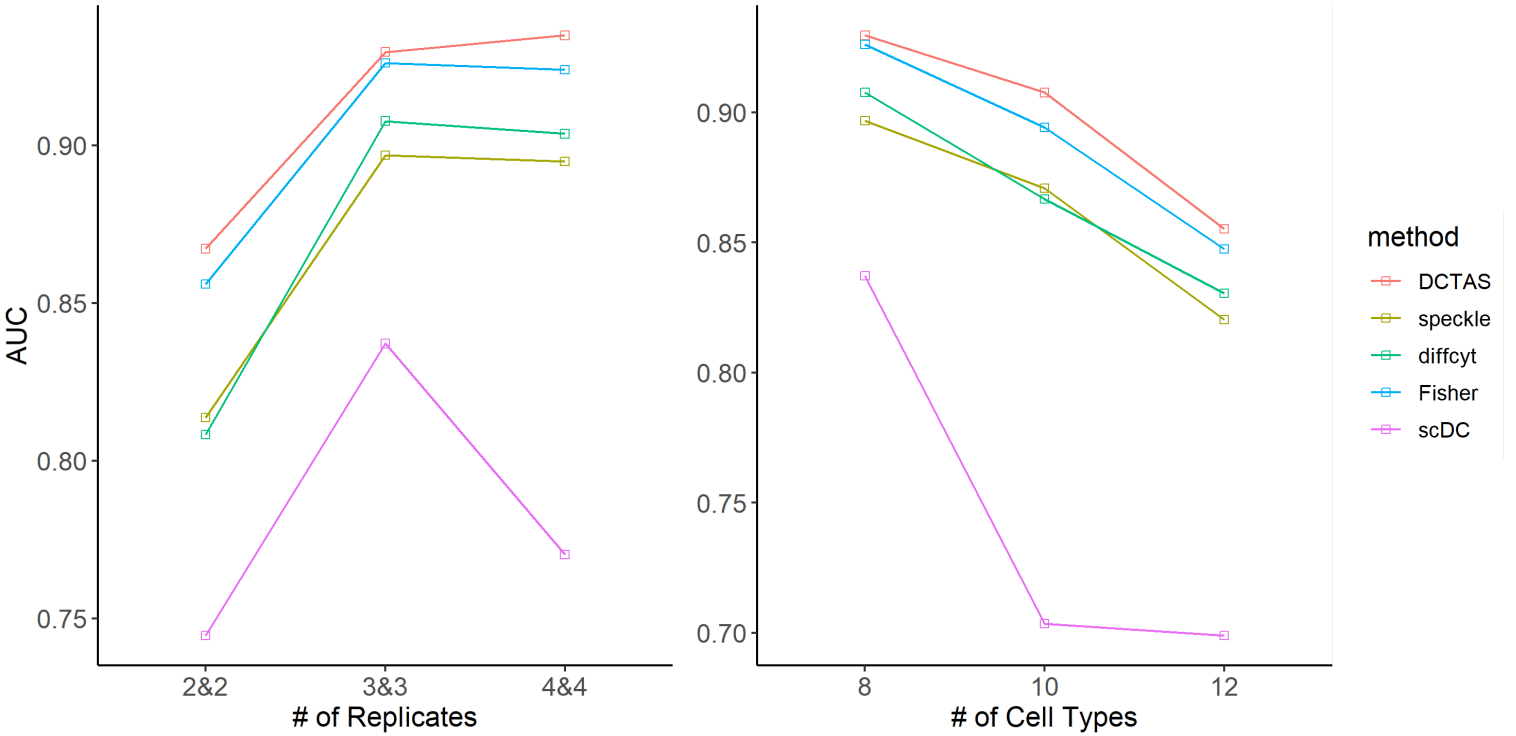
**similarity/misclassification matrix**

| | Cell Type 1 | Cell Type 2 | Cell Type 3 |
|---|---|---|---|
| Cell Type 1 | 1.0 | 0.0 | 0.0 |
| Cell Type 2 | 0.0 | 0.7 | 0.3 |
| Cell Type 3 | 0.0 | 0.3 | 0.7 |



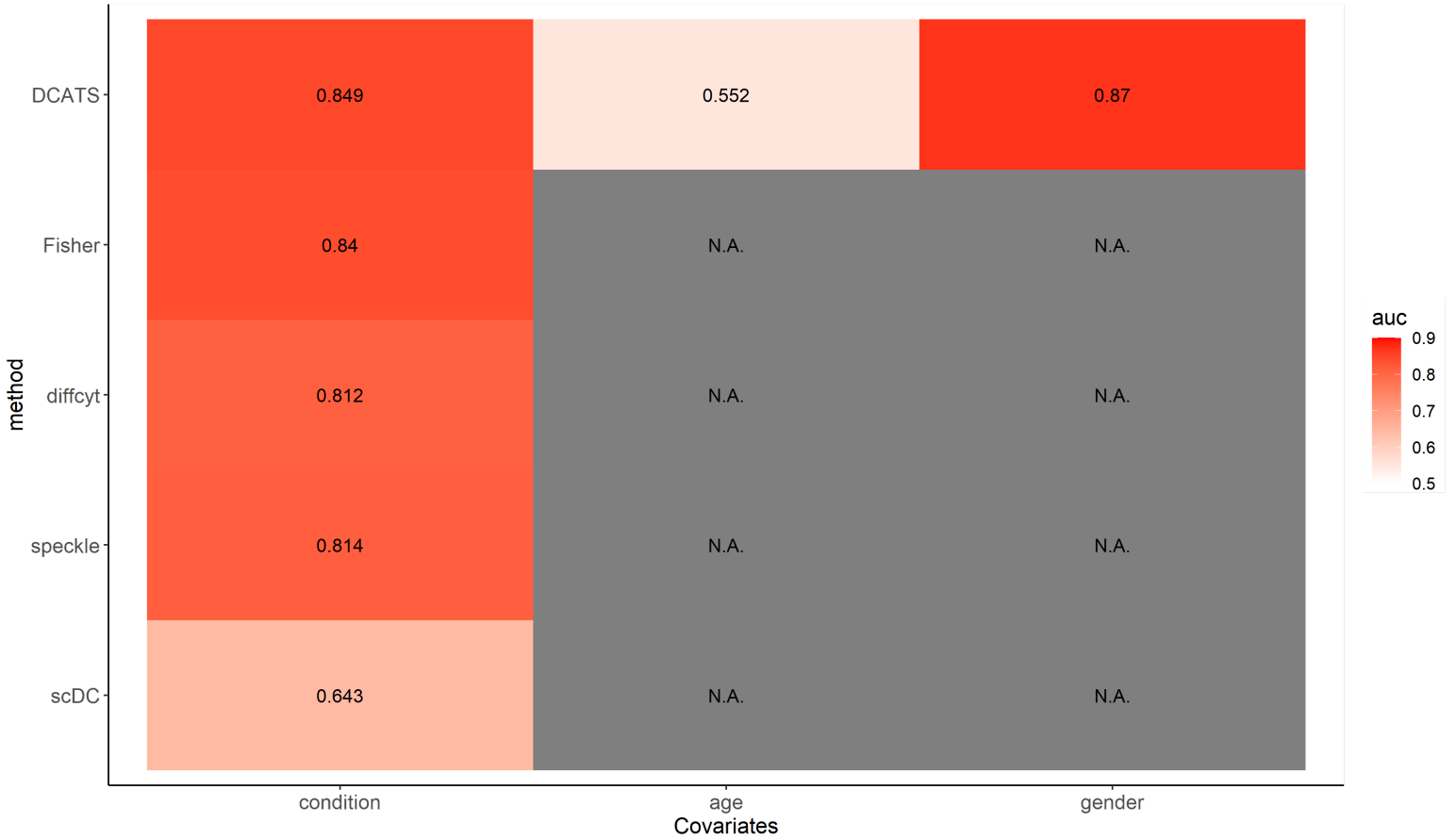
# Results: Simulation

- **Simulation with Expression Info**

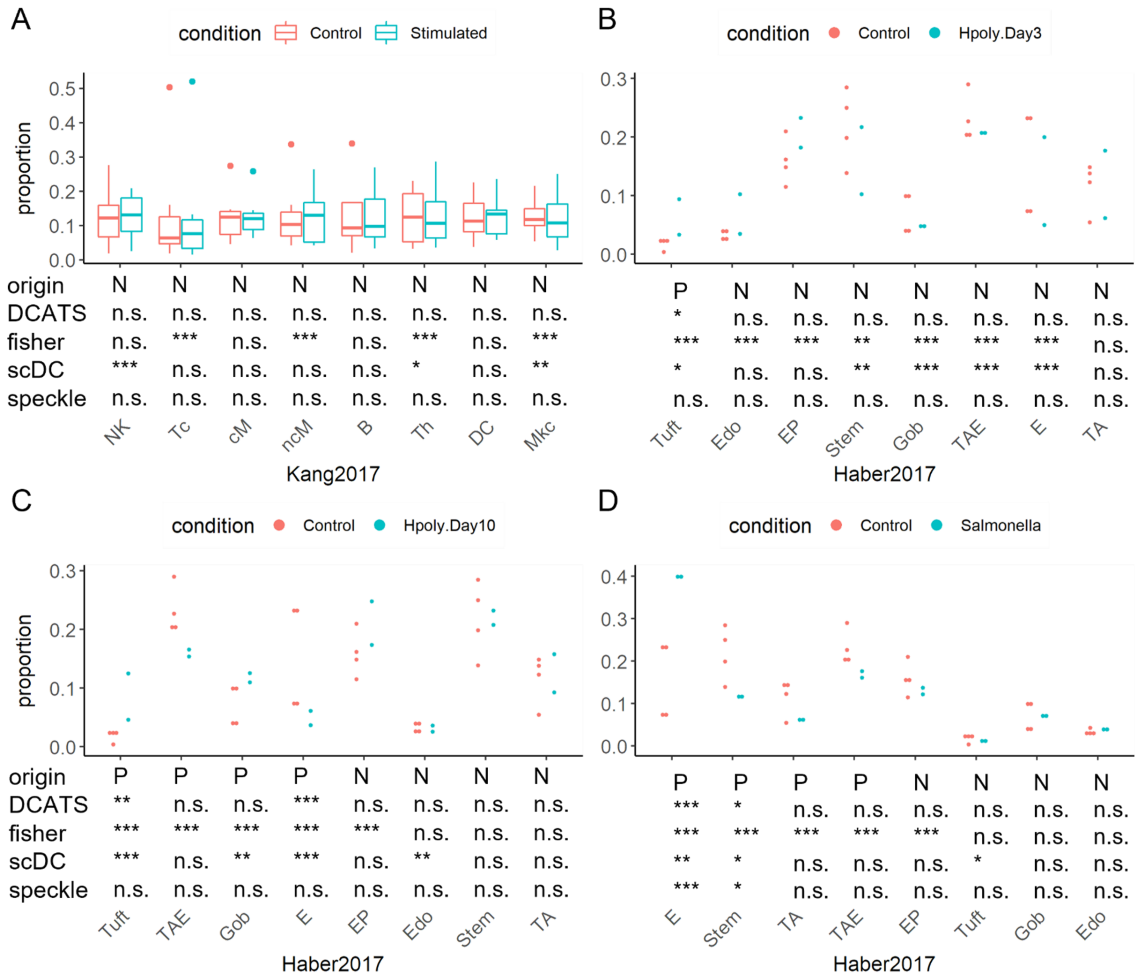


# Results: Simulation

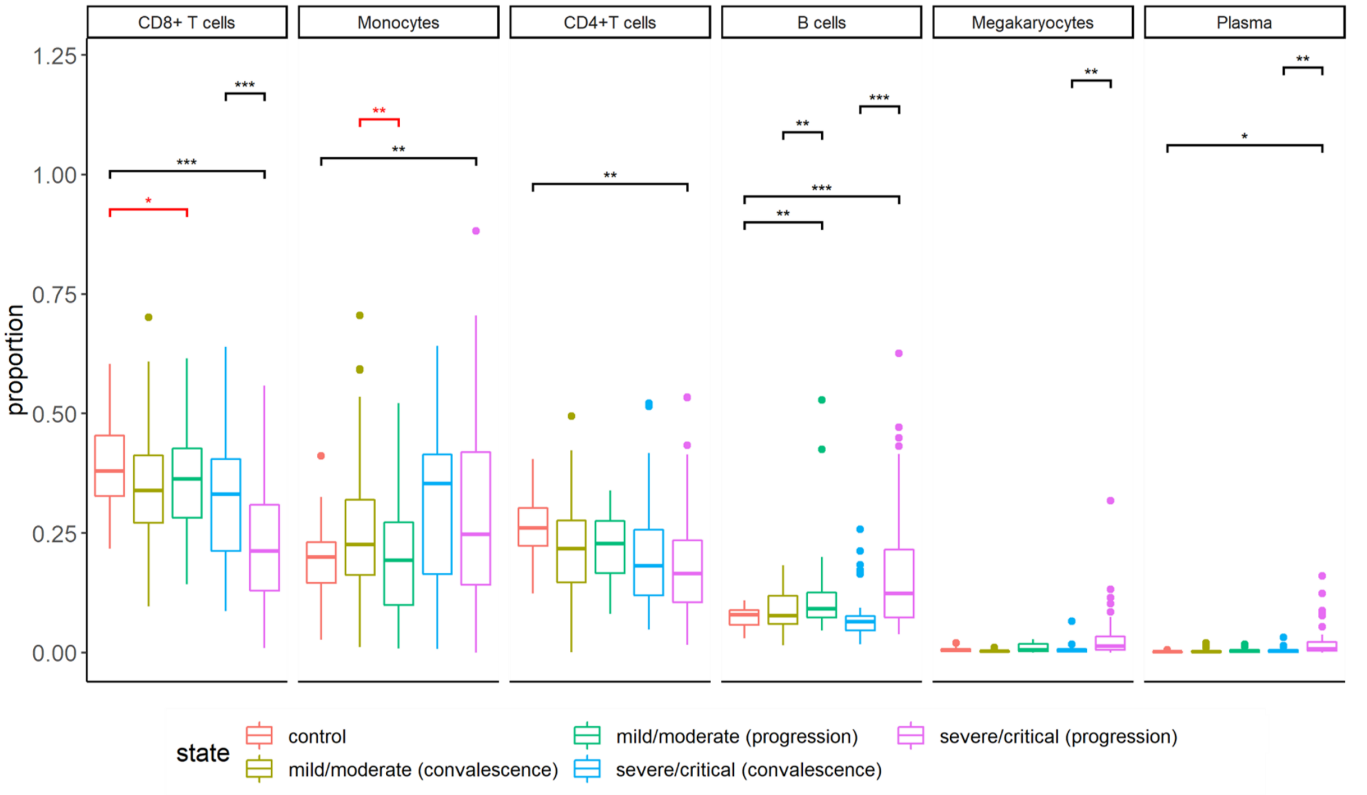
- **Simulation with Covariates**



# Results: Real World Data



# Results: Real World Data



# Thanks

# Q&A

# References

[1] L. Zappia, B. Phipson, and A. Oshlack. "Splatter: simulation of single-cell RNA sequencing data". In: *Genome biology* 18.1 (2017), pp. 1-15.

[2] H. M. Kang, M. Subramaniam, S. Targ, et al. "Multiplexed droplet single-cell RNA-sequencing using natural genetic variation". In: *Nature biotechnology* 36.1 (2018), p. 89.

[3] A. L. Haber, M. Biton, N. Rogel, et al. "A single-cell survey of the small intestinal epithelium". In: *Nature* 551.7680 (2017), pp. 333-339.

[4] X. Ren, W. Wen, X. Fan, et al. "COVID-19 immune features revealed by a large-scale single-cell transcriptome atlas". In: *Cell* 184.7 (2021), pp. 1895-1913.