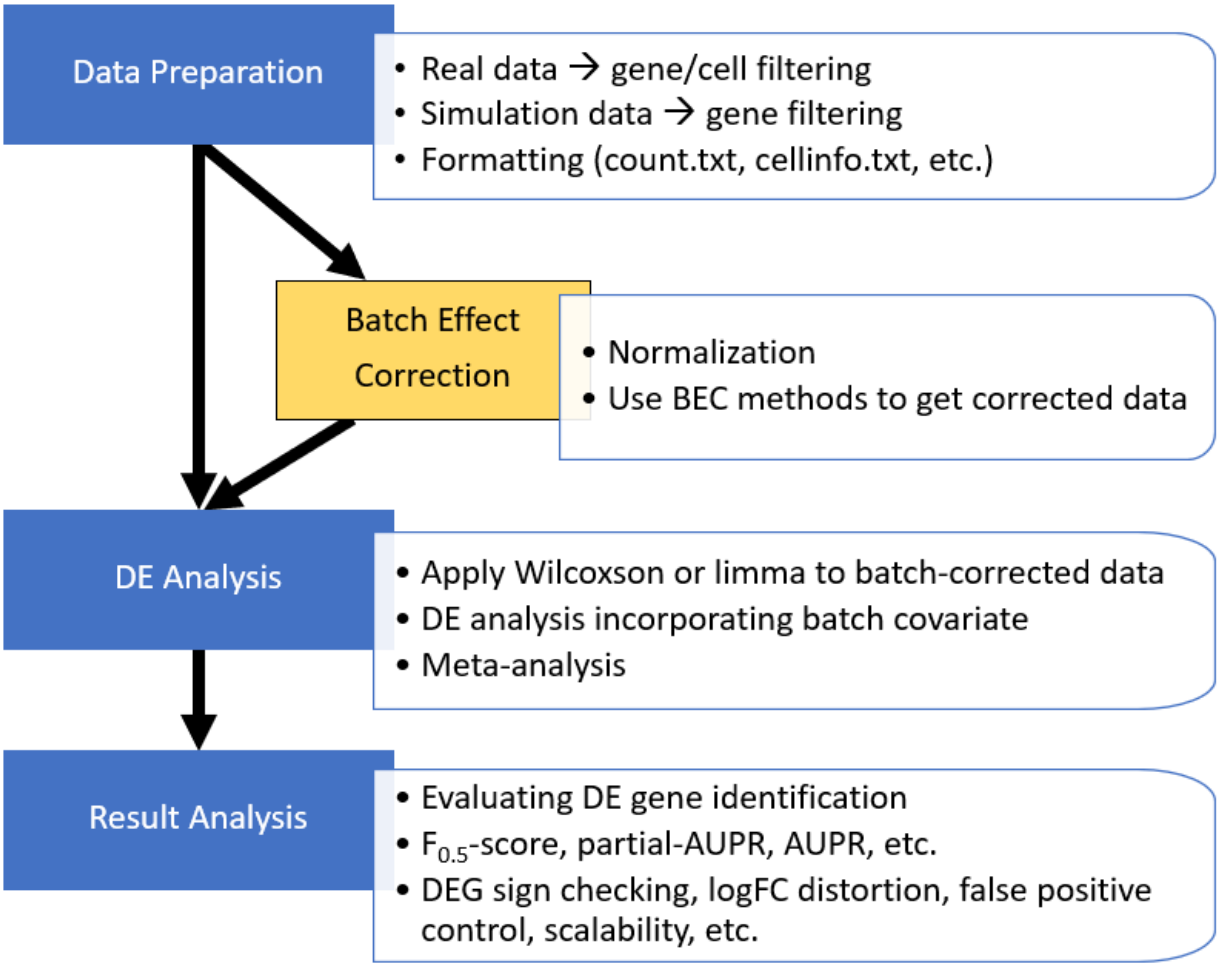
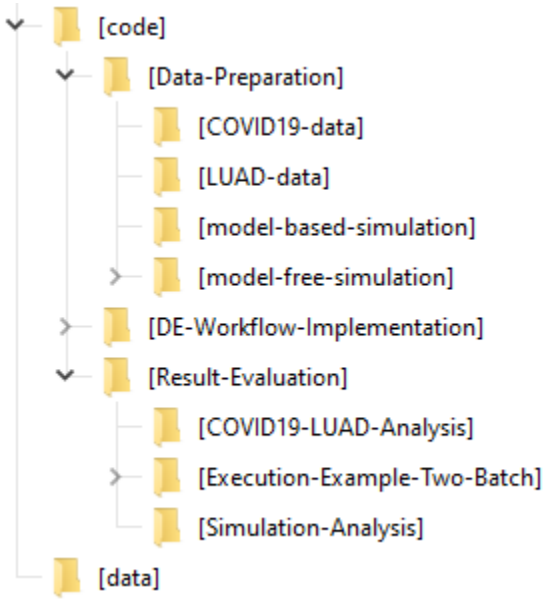
Benchmarking integration of single-cell differential expression

User Guide



1. GitHub folder structure



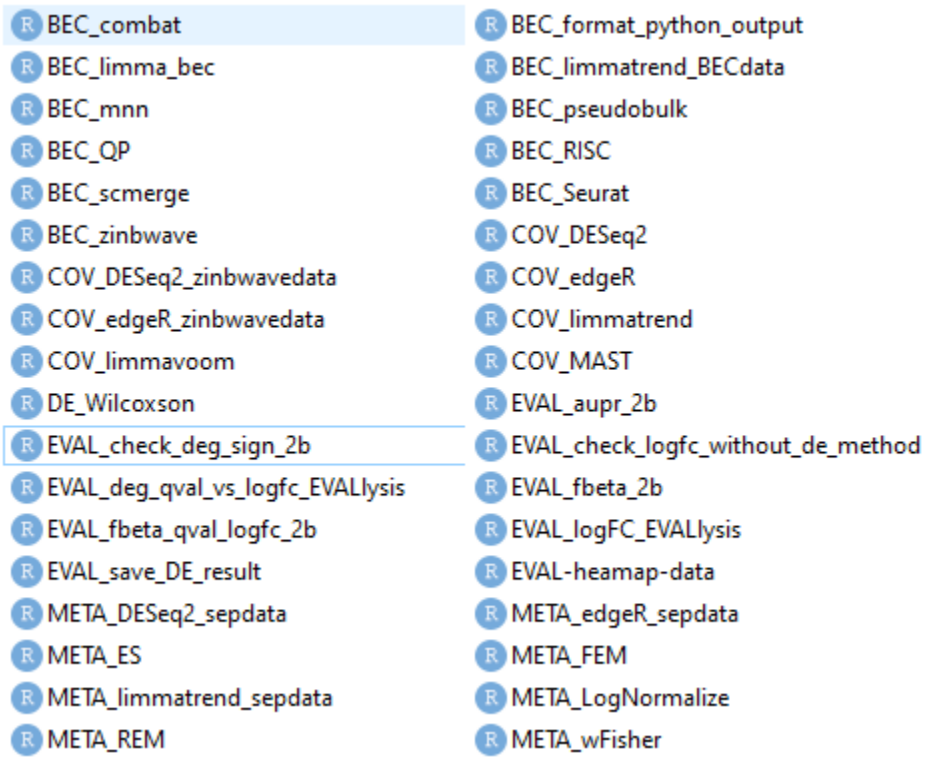
- **[code]** contains the core Python & R scripts used in this study
  - **'Data-Preparation'** contains the sample data and scripts for data preparation step
    - ✓ **'COVID19-data'** includes scripts for preparation of COVID-19 data
    - ✓ **'LUAD-data'** includes scripts for preparation of LUAD data
    - ✓ **'model-based-simulation'** includes scripts for simulating scRNA-seq data using Splatter R package
    - ✓ **'model-free-simulation'** includes scripts for simulating scRNA-seq data using MCA and Pancreatic data
  - **'DE-Workflow-Implementation'** contains implementation of each workflow used in this study
  - **'Result-Evaluation'** contains R scripts for evaluating DE analysis results
    - ✓ **'Execution-Example-Two-Batch'** includes scripts for reproducing the figures in the paper on $F_{0.5}$-score, partial AUPR, DEG sign checking, etc.

- ✓ **'COVID19-LUAD-Analysis'** gives scripts for evaluating COVID-19 and LUAD scRNA-seq data analyses (e.g., cumulative score curves, false positive, CPU times, etc.)
- ✓ **'Simulation-Analysis'** includes scripts for evaluating both model-based and model-free simulated scRNA-seq data analyses

- **[data]** contains figures and tables of the experimental results

2. Input and output of DE workflows

- **<method-category>_<DE-method-name>:** the format of the name of a DE workflow script



  - **<method-category>** includes 'BEC', 'COV', 'META', 'DE', 'EVAL' (for evaluation) indicating the characteristic of a method.
  - **< method-name >** indicates the specific DE method

- **input:**
  - A count matrix (genes × cells)

	X1_A	X1_A.1	X1_A.2	X1_A.3	X1_A.4	X1_A.5	X1_A.6	X1_A.7	X1_A.8	X1_A.9	X1_A.10
0610007P14Rik	0	0	0	0	0	0	0	0	1	0	0
0610012G03Rik	0	0	0	1	0	0	0	0	0	0	0
1110004E09Rik	0	0	0	0	0	0	0	0	2	0	0
1110004F10Rik	0	0	1	0	0	0	0	0	1	0	0
1110008F13Rik	0	1	0	0	0	0	0	0	2	0	0
1110008P14Rik	0	0	0	0	0	0	0	0	0	0	1
1110038B12Rik	0	0	0	0	1	0	0	0	0	0	0
1110038F14Rik	0	0	1	0	0	0	0	0	0	0	0
1110059E24Rik	0	1	0	0	0	0	0	0	1	0	0
1110059G10Rik	0	0	0	0	0	0	0	0	0	0	1
1300002E11Rik	0	0	0	0	0	0	0	0	0	0	0
1600020E01Rik	0	0	0	0	0	0	1	0	0	0	0
1700037H04Rik	1	0	0	0	0	0	0	0	0	0	0
1700097N02Rik	0	0	0	0	1	0	0	0	0	0	0

  - Data frame of cell information (e.g., group, batch)

cellinfo

Filter

	Group	Batch
1	A	1
2	A	1
3	A	1
4	A	1
5	A	1
6	A	1
7	A	1
8	A	1
9	A	1
10	A	1

  - Data frame of gene information (e.g., gene IDs, index)

geneinfo

Filter

	x
1	0610007P14Rik
2	0610012G03Rik
3	1110004E09Rik
4	1110004F10Rik
5	1110008F13Rik
6	1110008P14Rik
7	1110038B12Rik
8	1110038F14Rik
9	1110059E24Rik
10	1110059G10Rik

- **output:**
  - BEC methods: a matrix of batch-corrected values (genes × cells)

output

Filter Cols: 1 - 50

	X1_A	X1_A.1	X1_A.2	X1_A.3	X1_A.4	X1_A.5	X1_A.6	X1_A.7
0610007P14Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
0610012G03Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.03600818	0.00000000
1110004E09Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1110004F10Rik	0.00000000	0.00000000	0.00000000	0.04656252	0.00000000	0.04371717	0.03600818	0.00000000
1110008F13Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1110008P14Rik	0.00000000	0.00000000	0.04795441	0.04656252	0.00000000	0.00000000	0.00000000	0.00000000
1110038B12Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1110038F14Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1110059E24Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1110059G10Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000
1300002E11Rik	0.00000000	0.00000000	0.00000000	0.00000000	0.00000000	0.04371717	0.00000000	0.00000000

  - Wilcoxson test: a data frame of DE analysis results

	X	p_val	avg_log2FC	pct.1	pct.2	p_val_adj
1	Rpl41	3.361860e-32	0.8112850	0.989	0.934	1.028393e-28
2	Gzma	7.953368e-01	0.7430536	0.093	0.079	1.000000e+00
3	H3f3b	1.874244e-14	0.7032320	0.956	0.968	5.733313e-11
4	Rpl4	1.286567e-18	0.6976022	0.995	0.984	3.935608e-15
5	Rps29	4.303247e-23	0.6938563	0.989	0.950	1.316363e-19
6	Rpl35a	4.384710e-18	0.6661287	0.913	0.807	1.341283e-14
7	Rps15	6.610199e-16	0.6603338	0.934	0.882	2.022060e-12
8	Rpl27a	2.439341e-23	0.6384096	0.973	0.825	7.461943e-20
9	Rpl10a	1.966441e-16	0.6215998	0.896	0.744	6.015343e-13
10	Nkg7	4.653614e-02	0.6019261	0.765	0.794	1.000000e+00

  - DE workflows for covariate models and BEC data analysis: a data frame of DE analysis results

	pvalue	adjpvalue	logFC
0610007P14Rik	2.152735e-03	2.374718e-02	1.223651041
0610012G03Rik	4.866872e-01	8.331147e-01	-0.248451779
1110004E09Rik	8.736544e-01	9.750514e-01	-0.060952425
1110004F10Rik	6.314026e-01	8.961051e-01	0.166795670
1110008F13Rik	5.896157e-01	8.806809e-01	0.185851914
1110008P14Rik	1.379971e-01	4.504801e-01	0.575286590
1110038B12Rik	5.304236e-04	7.953754e-03	1.536010377
1110038F14Rik	4.149591e-02	2.073660e-01	-0.725644048
1110059E24Rik	2.805607e-01	6.566887e-01	-0.412511845
1110059G10Rik	4.577728e-02	2.218105e-01	-0.708648964

- Meta-analysis methods: a data frame of DE analysis results

Name	Type	Value
MetaDE.Res$`voom+FEM`	list [5] (S3: MetaDE.ES)	List of length 5
mu.hat	double [2609]	0.1070 -0.0710 -0.1316 -0.1213 0.0643 0.0697 ...
mu.var	double [2609]	0.00920 0.00920 0.00919 0.00919 0.00918 0.00919 ...
zval	double [2609]	1.116 -0.741 -1.373 -1.266 0.671 0.727 ...
pval	double [2609]	0.8677 0.2294 0.0849 0.1028 0.7489 0.7664 ...
FDR	double [2609 x 1]	0.939 0.592 0.416 0.433 0.881 0.893 ...

3. How to use the sample codes

All libraries required for testing Python (version >= 3.8) code are listed in the file **'requirements.txt'** including:

- anndata==0.8.0
- helpers==0.2.0
- matplotlib==3.5.3
- numpy==1.23.1
- pandas==1.4.4
- scanorama==1.7.2
- scanpy==1.9.1
- scgen==2.1.0
- scipy==1.9.1
- scvi==0.6.8
- seaborn==0.12.1
- torch==1.12.1

✓ Python codes include **'BEC_scanorama.py'**, **'BEC_scgen.py'**, **'BEC_scvi.py'** for implementing **'scanorama'**, **'scgen'**, and **'scvi'** methods, respectively

✓ After installing the required libraries, a Python code can be used directly in the command line as follows:

```
$>python BEC_scanorama.py
```

✓ Python codes are run separately, and the results will be integrated later via an R wrapper function from **'BEC_format_python_output.R'**

- **'_sample_run.R'** shows how to use each method provided in the **'DE-Workflow-Implementation'**

```r
dir_refscript='ref_script'
files.sources = list.files(dir_refscript,full.names = T)

sapply(files.sources, source)

# example-1a on processing a BEC method
run_combat(count,cellinfo)
load('combat.rda')
run_wilcox(processed, cellinfo=cellinfo,is.log=T,former.meth = 'combat')
load('combat+wilcox.rda')

# example-1b on processing a BEC method using Python including 'scvi', 'scgen', 'scanorama'
run_format_python(cellinfo, meth='scvi')
run_wilcox(processed, cellinfo=cellinfo,is.log=T,former.meth = 'scvi')
load('scvi+wilcox.rda')

#example-2 on processing a COV method
run_limmavoom(count,cellinfo,cov=T)
load('limmavoom.rda')

#example-3 on processing a META method
run_LogNormalize(count,cellinfo,separate = T,former.meth = '')
load('LogNormalize_sep.rda')
run_limmatrend_sep(processed=processed,cellinfo=cellinfo,former.meth = 'LogNormalize')
load('LogNormalize_sep+limmatrend_sep.rda')
run_wFisher(res,processed=processed,cellinfo=cellinfo,former.meth='LogNormalize_sep+limmatrend')
load('LogNormalize_sep+limmatrend_sep+wfisher.rda')
```

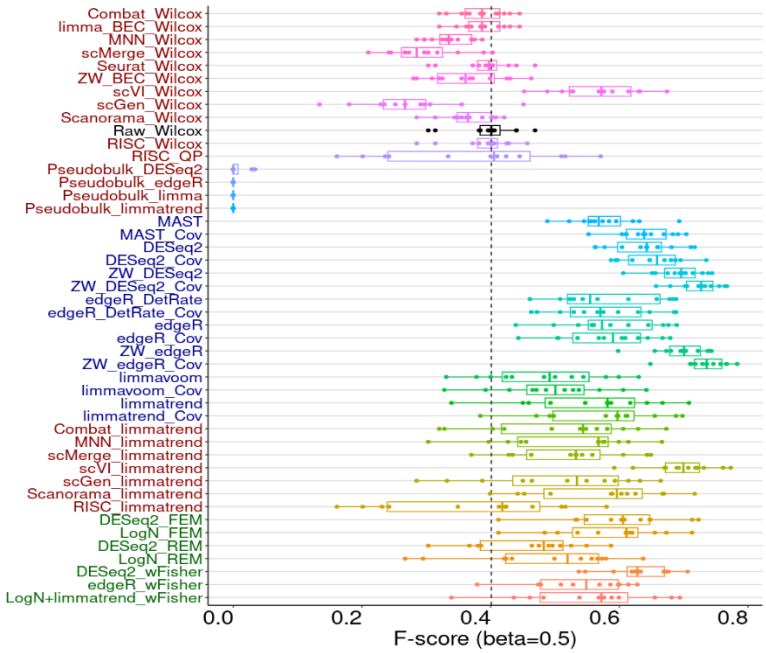
- Results from each DE workflow are shown as follows:

  - LogNormalize_sep.rda
  - LogNormalize_sep_sep+es_sep.rda
  - LogNormalize_sep_sep+es_sep_sep+FEM.rda
  - LogNormalize_sep_sep+es_sep_sep+REM.rda
  - LogNormalize_sep_sep+limmatrend_sep.rda
  - LogNormalize_sep_sep+limmatrend_sep_sep+wfisher.rda
  - MAST.rda
  - MAST_Cov.rda
  - Seurat.rda
  - Seurat+wilcox.rda
  - cellinfo.txt
  - combat.rda
  - combat+limmatrend.rda
  - combat+wilcox.rda

4. Visualization

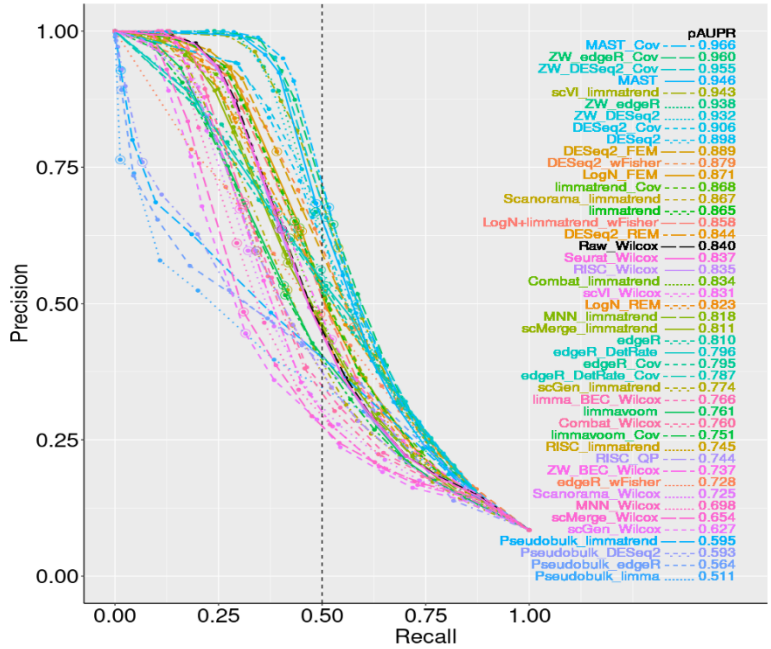
- **'EVAL_fbeta_2b.R'**

  ✓ Gather all the output results and visualize the F-beta results



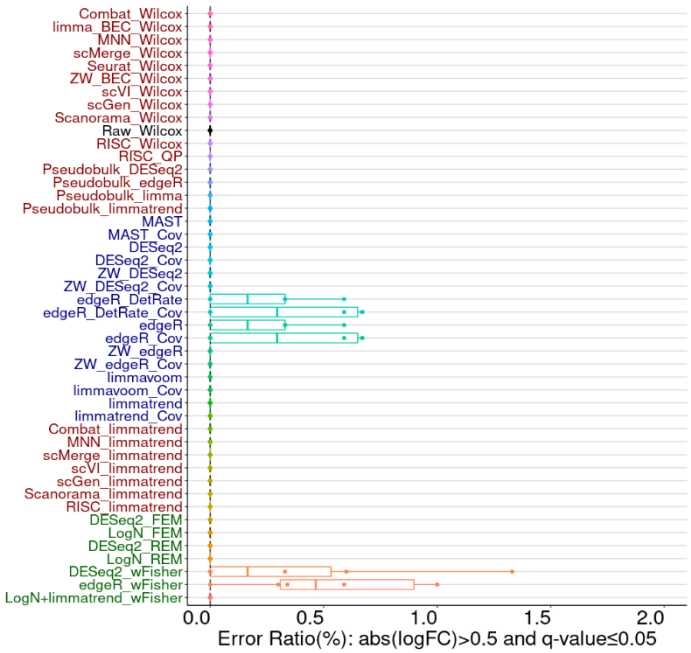
- **'EVAL _aupr_2b.R'**

  ✓ Gather all the output results and illustrate the partial AUPR curves



- **'EVAL_check_deg_sign_2b.R'**

  - ✓ Gather all the output results and illustrate the ratio of DEGs altered their signs



- **'EVAL _fbeta_qval_logfc_2b.R'**

  - ✓ Gather all the output results and visualize the F-beta results using both *q*-value and fold change cutoffs