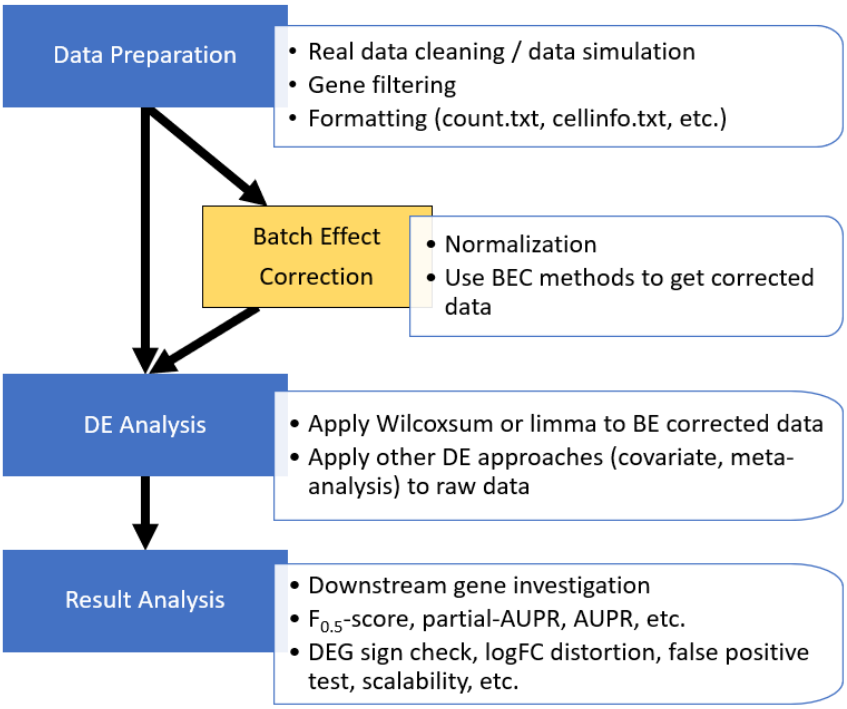
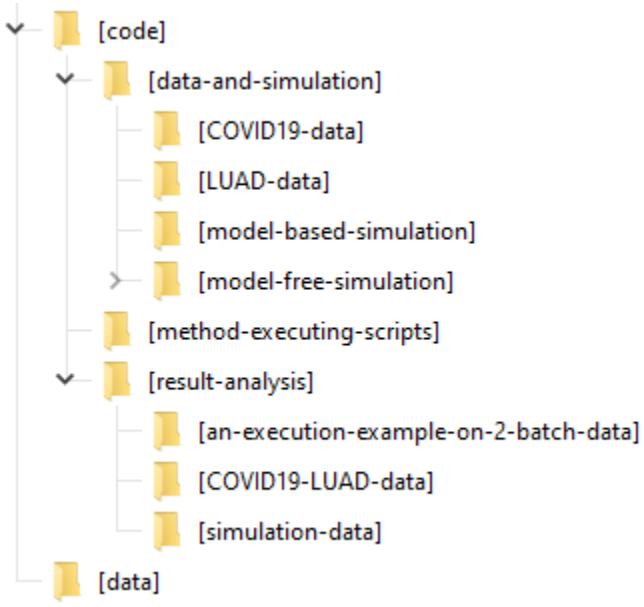
Benchmarking integration of single-cell differential expression

User Guide



1. GitHub folder structure



- **'code'** contains the core analysis Python & R scripts for this study
    - **'data-and-simulation'** contains sample data and scripts for data preparation step
        - **'COVID19-data'** gives scripts for data preparation using COVID-19 data
        - **'LUAD-data'** gives scripts for data preparation using LUAD data
        - **'model-based-simulation'** gives scripts for simulating data using MCA and Pancreas data
        - **'model-free-simulation'** gives scripts for simulating data using Splatter
    - **'method-executing-scripts'** contains implementation for each considered method
    - **'result-analysis'** contains <**ANA-function**> scripts for analyzing

- **'data'** contains figures and tables of the experimental results for illustration

2. Analysis code input-output

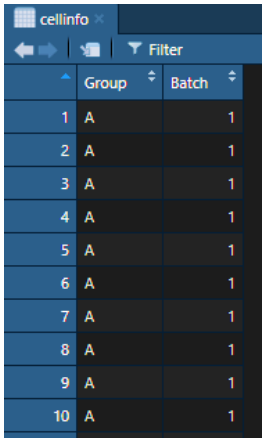
- BEC-limma
- BEC-pseudobulk_edger
- BEC-seurat3
- COV-deseq2
- COV-edger
- COV-edger_DetRate
- COV-limma_trend
- COV-limma_trend_Combat_false
- COV-limma_trend_False
- COV-limma_trend_mnnCorrect
- COV-limma_trend_scMerge
- COV-limma_voom
- COV-mast
- COV-zinbwave_deseq2
- DE-DEGs_from_Seurat_auc
- DE-Seurat_DEG_analysis_auc

- **‘<method-category>_<method-name>’:** script to specifically test a particular method
    - **< method-category >:** includes ‘BEC’, ‘COV’, ‘META’, ‘DE’, ‘ANA’ indicating the characteristic of a method. Note that all results of ‘BEC’, ‘COV’, ‘META’, ‘DE’ functions are **required** before running **‘ANA’** functions or else **code modification** is essential.
    - **< method-name >:** indicate the specific 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

- a data frame of cell descriptions (group, batch, … information)

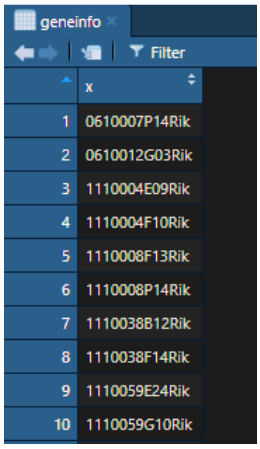


cellinfo

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

- a data frame of gene descriptions (id, name, code, …information)

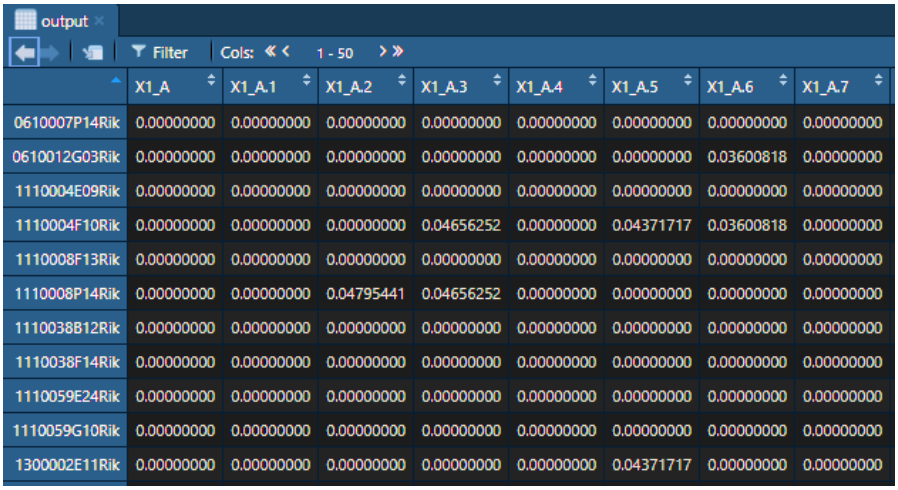


geneinfo

Filter

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

- **output:**
  - batch effect correction methods: a matrix of 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 rank sum test: a data frame of gene ranking analysis

	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

- parametric and integration methods: a data frame of gene ranking analysis

	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 gene ranking analysis

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. Sample code usage

- All requirement libraries used 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 includes **'BEC_scanorama.py'**, **'BEC_scgen.py'**, **'BEC_scvi.py'** for corresponding methods **'scanorama'**, **'scgen'**, and **'scvi'**
    - ✓
    - ✓ After installing library dependencies, a Python code can be used directly in the command line as:

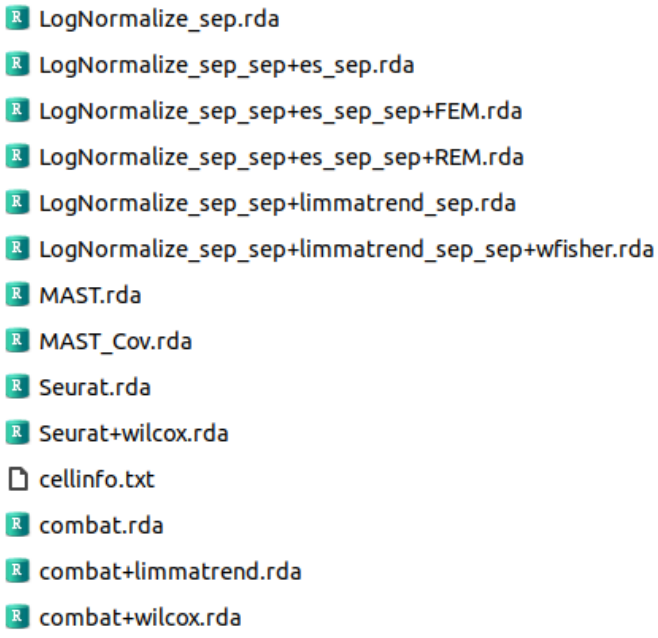
      ```
      $>python BEC_scanorama.py
      ```

    - ✓ 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 separate method provided in the **'method-executing-scripts'**

```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 = T,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 = T,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')
```

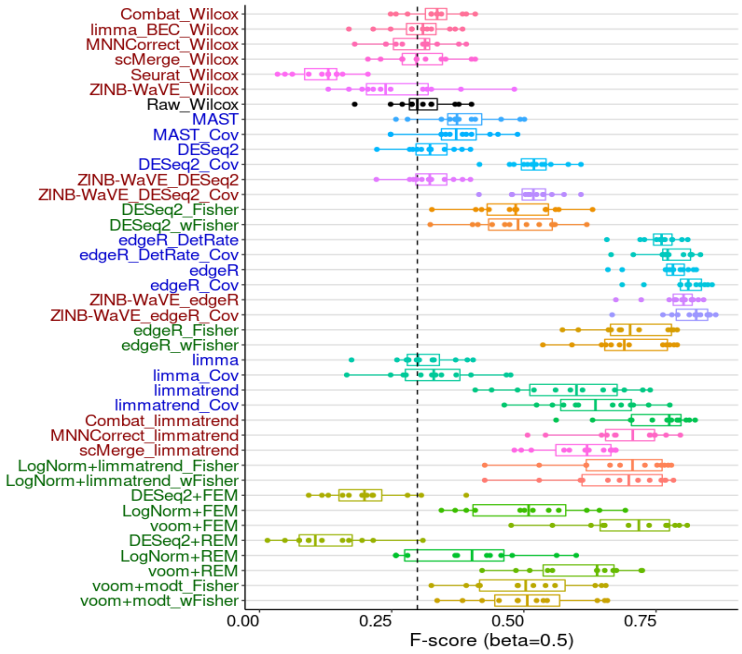
- Results from separate scripts will be looked like the following:



4. Visualization

- **'ANA_fbeta_2b.R'**

  ✓ Aggerate all output results and visualize F-beta performance



- **'ANA_aupr_2b.R'**

  ✓ Aggerate all output results and illustrate the partial AUPR curve