# Example of Meta-analysis Using Transcriptomic Data

Mengyuan Kan

December 11, 2018

## Prerequisite Files

### secondary analysis results

- A directory with all .csv files (microarray) and/or .txt files (RNA-Seq) of DE results
- A .csv file of sample information.

Required columns: GEO_ID, App (asthma or GC, representing disease or treatment), Tissue, Asthma (disease endotype or exposure), N_Condition0 (sample size in condition 0), N_Condition1 (sample size in condition 1), Total (total sample size), Unique_ID (DE filename without extention), and batchadj (if use SVA batch effect adjusted p value, define 'yes')

example of the sample info sheet: Microarray_data_infosheet_R.csv

| GEO_ID | App | Tissue | Asthma | PMID | Treatment |
|---|---|---|---|---|---|
| GSE44037 | asthma | BE | allergic_asthma | 24282527 | NA |
| GSE4917 | GC | MCF10A-Myc | GC | 16690749 | Dex 1uM 24h |

continued

| Daily_Dosage | steroid_resistance_sensitivity | N_Condition0 | N_Condition1 | Total |
|---|---|---|---|---|
| NA | no | 6 | 6 | 12 |
| NA | no | 3 | 3 | 6 |

continued

| Description |
|---|
| Bronchial epithelial cell brushings; n=12 (6 healthy control, 6 allergic asthma) |
| MCF10A-Myc cells; n=6 (3 dexamethasone treatments (1uM; 24h), 3 ethanol) |

continued

| Long_tissue_name | Unique_ID | batchadj |
|---|---|---|
| Bronchial epithelium | GSE44037_BE_asthma_vs_healthy | yes |
| MCF10A-Myc | GSE4917_MCF10A-Myc_nonasthma_dex_24hr_vs_control_24hr | corr_donor_scandate |

### scripts

- csv2rds.R: convert DE result files .csv/.txt to .RDS
- meta_analysis_geneexpr.py: generate command lines/job scripts for integration based on user-defined options
- integration_utility.R: R utilities and functions used for integration, will be called in meta_analysis_geneexpr.R and meta_analysis_RankProd.R
- meta_analysis_geneexpr.R: R scripts to run integration methods (i.e. fisher's sum-of-log, meta-analysis using random-effects model, and rank product)
- meta_analysis_RankProd.R: R scripts to run 1000 permutations for rank-based integration
- study.rankprod.combine.R: R scripts to compute expected rank product and p-values, and also combine p-value-based and effect size-based results in one file
- mk_bsub.py: use piped input to generate LSF file. Put under $PATH directory.

## Meta-analysis Example

### Convert csv output to RDS

Directly run R script `csv2rds.R` to convert recently added DE result in .csv format to RDS format

```
Rscript /home/mengykan//Projects/integration/scripts/csv2rds.R
```

**Note**

Change three variables in the scripts before running. Modify three variables under *Change Here*.

- `datainfor_fn_hpc`: sample info sheet with absolute path
- `resdir_hpc`: directory with all DE restuls .csv/.txt files
- `appdir_hpc`: directory where the .RDS files will be generated

How `csv2rds.R` works:

- If batchadj is *yes* in sample sheet, replace P.Value with pValuesBatch and replace adj.P.Val with qValuesBatch, otherwise use the original P.Value and adj.P.Val values without batch effect adjustment.
- Create SD column by computing SD=logFC/t
- Select results only have gene symbol available, and use the gene-based restuls with the smallest p-value.
- Create rank column based on the rank of p-values.
- If p-value is zero, assign the smallest non-zero p-value in the results

## Run meta-analysis for treatment comparison

In this example, perform meta-analysis for treatment studies of structural cell type

Check options in `meta_analysis_geneexpr.py`:

```
python /home/mengykan//Projects/integration/scripts/meta_analysis_geneexpr.py -h
```

- `--script_dir`: directory for all the R scripts
- make selection:
  - `--tissue`: (required) select tissues
  - `--disease`: (required if treatment/study is not specified) select disease endotypes
  - `--treatment`: (required if disease/study is not specified) select treatments/exposures
  - `--study`: (optional) select studies corresponding to Unique_ID. If `study` is specified, tissue/disease/treatment types will be ignored
  - specify *entire* if use all tissues/disease endotypes/exposures
- `--out`: output prefix, including the absolute path. e.g. ~/GC_structural/GC_structural will generate files starting with GC_structural in the GC_structural directory

### Effect size-based integration

There are two ways to run:

1. directly use python to invoke R

```
python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databas\
es/AsthmaApp/databases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/mi\
croarray_results --script_dir /home/mengykan/Projects/integration/scripts --tissue 'ASM BE' --treatment GC --out /h\
ome/mengykan/Projects/integration/GC_structural/GC_structural --method metaranef
```

2. submit a job on HPC. This method is always preferable because this process will take a while.

1) create a directory for LSF scripts

```
mkdir /home/mengykan/Projects/integration/scripts/GC_structural
cd /home/mengykan/Projects/integration/scripts/GC_structural
```

2) create a command line

```
cmd="python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/da\
tabases/AsthmaApp/databases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databas\
es/microarray_results --script_dir /home/mengykan/Projects/integration/scripts --tissue 'ASM BE' --treatment GC --o\
ut /home/mengykan/Projects/integration/GC_structural/GC_structural --method metaranef"
```

3) mk_bsub.py is used for automatically generate LSF file by piping user's input command

```
echo $cmd | mk_bsub.py --line 1 --thread 1 --job_name GC_structural_metaranef --memory 24000 # generate a script GC\
_structural_metaranef.lsf
```

4) submit job

**Note**

It is recommended to exit the computational node and submit the job from headnode.

```
bsub < GC_structural_metaranef.lsf # submit job
```

You can check job status by issuing

```
bjobs
```

Show LSF script

```
cat GC_structural_metaranef.lsf
```

BASH

```
#!/bin/bash
#BSUB -L /bin/bash
#BSUB -J GC_structural_metaranef
#BSUB -q normal
#BSUB -outdir /home/mengykan/Projects/integration/scripts/GC_structural
#BSUB -o GC_structural_metaranef_%J.out
#BSUB -e GC_structural_metaranef_%J.screen
#BSUB -M 24000
#BSUB -n 1
python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp/data\
bases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir /hom\
e/mengykan/Projects/integration/scripts --tissue 'ASM BE' --treatment GC --out /home/mengykan/Projects/integration/GC_structural/GC_\
structural --method metaranef
```

Show standard outputs

```
cat /home/mengykan/Projects/integration/scripts/GC_structural/GC_structural_metaranef*.out
```

OUTPUT

```
5 treatment gene expression datasets have been selected
[1] "GSE13168_ASM_nonasthma_GFP_Flu_vs_GFP_basal"
[2] "GSE1815_BE_nonasthma_dex_8hr_vs_control_8hr"
[3] "SRP033351_healthy_dex_vs_healthy_untreated_full_DESeq2_results"
[4] "GSE34313_ASM_nonasthma_dex24hr_vs_nodex"
[5] "BUDResponse_non_asthma_BUD_vs_non_asthma_control_full_DESeq2_results"
Select genes shared within at least two studies
Obtain 19266 genes
Perform meta-analysis using random-effects model
```

Show results. Output files are under the pre-defined directory: /home/mengykan/Projects/integration/GC_structural.

```
head -6 /home/mengykan/Projects/integration/GC_structural/GC_structural.metaranef.txt
```

| Gene | logFC | SE | CI_lower | CI_upper | df | P.Value | qval | rank |
|---|---|---|---|---|---|---|---|---|
| KANK4 | -3.59 | 0.09 | -3.76 | -3.41 | 2 | 0 | 0 | 1.5 |
| PPP1R14A | 3.10 | 0.07 | 2.96 | 3.24 | 2 | 0 | 0 | 1.5 |
| SOST | -1.97 | 0.08 | -2.13 | -1.81 | 1 | 9.29E-128 | 5.96E-124 | 3 |
| FER1L6 | -2.84 | 0.13 | -3.09 | -2.58 | 1 | 4.54E-106 | 2.19E-102 | 4 |
| TBX18 | -1.79 | 0.08 | -1.95 | -1.63 | 1 | 5.18E-105 | 1.99E-101 | 5 |

## p-value-based integration

Here use the second way to run, i.e. submit a job to HPC

BASH

```
cd /home/mengykan/Projects/integration/scripts/GC_structural
cmd="python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp\
/databases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir\
 /home/mengykan/Projects/integration/scripts --tissue 'ASM BE' --treatment GC --out /home/mengykan/Projects/integration/GC_structura\
l/GC_structural --method fisherp"
echo $cmd | mk_bsub.py --line 1 --thread 10 --job_name GC_structural_fisherp --memory 24000 # generate a script GC_structural_fisher\
p.lsf
bsub < GC_structural_fisherp.lsf
```

output file: /home/mengykan/Projects/integration/GC_structural/GC_structural.fisherp.txt

```
head -6 /home/mengykan/Projects/integration/GC_structural/GC_structural.fisherp.txt
```

| Gene | P.Value | qval | rank |
|---|---|---|---|
| GLUL | 0 | 0 | 2 |
| SAMHD1 | 0 | 0 | 2 |
| TSC22D3 | 0 | 0 | 2 |
| NKD1 | 2.66E-322 | 1.29E-318 | 4 |
| FKBP5 | 2.14E-304 | 8.24E-301 | 5 |

## rank-based integration

To perform 1,000,000 permutations, create 20 .lsf files with R command to run 50,000 permutations each based on random seeds from 1 to 1,000,000

1. Generate 20 .lsf files to run 1,000,000 permutations

BASH

```
cd /home/mengykan/Projects/integration/scripts/GC_structural
python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp/data\
bases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir /hom\
e/mengykan/Projects/integration/scripts --tissue 'ASM BE' --treatment GC --out /home/mengykan/Projects/integration/GC_structural/GC_\
```

```
structural --method rankprodt # generate 20 LSF file
# These LSF are named by rank_[seed].lsf. Run these files in parallele. Each may take >40 minutes on HPC.
for i in rank_*.lsf; do bsub < $i; done
```

Output files:

- Under output directory: /home/mengykan/Projects/integration/GC_structural/
  - GC_structural.rankprodt.txt: gene and its observed rank product
  - GC_structural.ranklist.RDS: a list object with the gene and rank columns from each study
- The directory `GC_structural_rankperm` the under output directory
  - Containing 20 RDS objects. Each has permutation results after 50,000 permutations.

**Note**

To run step 2, make sure all 20 jobs are completed. Two ways to check:

1) the number of RDS files in the output directory

```
ll rank*.RDS | wc # should be 20
```

2) the standard error in script directory. If the job is halted, 'Execution halted' can be found in .screen files.

```
grep "Execution halted" rank*screen
```

If any LFS script reports an error, re-run the corresponding script.

2. Combine permutation results and obtain expected RP and p-values, then combine the results from the other two methods

Directly run R script `study.rankprod.combine.R`. V1: output prefix; V2: directory with previous integration results

```
Rscript /home/mengykan/Projects/integration/scripts/study.rankprod.combine.R "GC_structural" "/home/mengykan/Projec\
ts/integration/GC_structural"
```

generate a combined output file

```
cat /home/mengykan//Projects/integration/GC_structural/GC_structural.txt
```

| Gene | logFC | SE | CI_lower | CI_upper | df |
|---|---|---|---|---|---|
| KANK4 | -3.59 | 0.09 | -3.76 | -3.41 | 2 |
| PPP1R14A | 3.10 | 0.07 | 2.96 | 3.24 | 2 |

Continued

| metaranef_P.Value | metaranef_qval | metaranef_rank | fisherp_P.Value | fisherp_qval | fisherp_rank |
|---|---|---|---|---|---|

| | | | | | |
|---|---|---|---|---|---|
| 0 | 0 | 1.5 | 1.69E-22 | 3.04E-21 | 1071 |
| 0 | 0 | 1.5 | 5.02E-89 | 8.95E-87 | 108 |

Continued

| RP | Count | rankprodperm_P.Value | rankprodperm_rank | FDR |
|---|---|---|---|---|
| 0.67 | 1194 | 1.19E-03 | 294 | 7.82E-02 |
| 0.70 | 11 | 1.10E-05 | 53.5 | 3.96E-03 |

## Run meta-analysis for disease comparison

In this example: Perform meta-analysis for severe asthma studies of all cell types

### Effect size-based integration

BASH

```
mkdir /home/mengykan/Projects/integration/scripts/severe_asthma_entire
cd /home/mengykan/Projects/integration/scripts/severe_asthma_entire
cmd="python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp\
/databases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir\
 /home/mengykan/Projects/integration/scripts --tissue 'entire' --disease 'severe_asthma' --out /home/mengykan/Projects/integration/s\
evere_asthma_entire/severe_asthma_entire --method metaranef"
echo $cmd | mk_bsub.py --line 1 --thread 1 --job_name severe_asthma_entire_metaranef --memory 24000
bsub < severe_asthma_entire_metaranef.lsf
```

**Note**

`--disease` option is specified to select severe asthma; `--tissue 'entire'` to use all cell and tissue types

### p-value-based integration

BASH

```
cd /home/mengykan/Projects/integration/scripts/severe_asthma_entire
cmd="python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp\
/databases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir\
 /home/mengykan/Projects/integration/scripts --tissue 'entire' --disease 'severe_asthma' --out /home/mengykan/Projects/integration/s\
evere_asthma_entire/severe_asthma_entire --method fisherp"
echo $cmd | mk_bsub.py --line 1 --thread 10 --job_name severe_asthma_entire_fisherp --memory 24000
bsub < severe_asthma_entire_fisherp.lsf
```

### rank-based integration

To perform 1,000,000 permutations, create 20 .lsf files with R command to run 50,000 permutations each based on random seeds from 1 to 1,000,000

1. Generate 20 .lsf files to run 1,000,000 permutations

BASH

```
cd /home/mengykan/Projects/integration/scripts/severe_asthma_entire
python /home/mengykan/Projects/integration/scripts/meta_analysis_geneexpr.py --samp_info /project/bhimeslab/databases/AsthmaApp/data\
bases/Microarray_data_infosheet_R.RDS --DE_dir /project/bhimeslab/databases/AsthmaApp/databases/microarray_results --script_dir /hom\
e/mengykan/Projects/integration/scripts --tissue 'entire' --disease 'severe_asthma' --out /home/mengykan/Projects/integration/severe\
_asthma_entire/severe_asthma_entire --method rankprodt # generate 20 LSF file
for i in rank_*.lsf; do bsub < $i; done
```

2. Combine results

```
Rscript /home/mengykan/Projects/integration/scripts/study.rankprod.combine.R "severe_asthma_entire" "/home/mengykan\
/Projects/integration/severe_asthma_entire"
```

## show results

datasets selected:

OUTPUT

```
[1] "9 disease gene expression datasets have been selected"
[1] "GSE31773_CD4_severe_asthma_vs_healthy"
[2] "GSE31773_CD8_severe_asthma_vs_healthy"
[3] "GSE27011_WBC_severe_asthma_vs_healthy"
[4] "GSE63142_ScanYear_BE_Baseline_severe_asthma_vs_healthy"
[5] "GSE89809_Epithelial_Baseline_severe_vs_healthy"
[6] "GSE89809_BAL_Baseline_severe_vs_healthy"
[7] "GSE89809_Spm_Baseline_severe_vs_healthy"
[8] "GSE69683_Blood_Baseline_severe_asthma_vs_healthy"
[9] "GSE64913_Central_airway_epithelium_Baseline_severe_asthma_vs_healthy"
...
Select genes shared within at least two studies
Obtain 21632 genes
```

```
cat /home/mengykan/Projects/integration/severe_asthma_entire/severe_asthma_entire.txt
```