A simple guide of ENIGMA

Weixu Wang & Xiaolan Zhou

2022/1/20

1. Construct ENIGMA object

To build ENIGMA object, user is required to prepare bulk gene expression matrix (RNA-seq) and reference profile matrix, the reference profile matrix could be learned from scRNA-seq or FACS RNA-seq datasets. Here, I used example Non Small Cell Lung Cancer (NSCLC) single cell bulk RNA-seq datasets and single cell RNA-seq datasets to illustrate how to build ENIGMA object and perform deconvolution.

ENIGMA used three ways to construct the object

1. Using single cell RNA-seq datasets as reference

```
egm = create_ENIGMA(bulk = Bulk, ref = ref_sc, ref_type = "single_cell",
                    meta_ref = metadata)
```

```
## Sun Jan 23 10:09:23 2022 Reference from Single Cell RNA-seq.

## Sun Jan 23 10:09:23 2022 Obtain reference from a matrix
```

The metadata contains the meta information of each cell in our reference datasets (which is required data.frame format). The reference profile (ref) need to be matrix.

```
head(metadata)
```

```
##                             V1        V2 V3 dbCluster t_sub cell_type
## AAACCTGAGTAACCCT_13 -27.09995  5.154716 NA    B_cell  <NA>    B_cell
## AAACCTGCAACGATGG_13  12.36870  4.525834 NA    B_cell  <NA>   Myeloid
## AAACCTGCATTAGCCA_13  20.98710  9.604408 NA    B_cell  <NA>   Myeloid
## AAACCTGGTCCATGAT_13 -19.22939 -7.118204 NA    B_cell T_CD4      T_CD4
## AAACCTGTCCTGTACC_13  27.25246  4.017953 NA    B_cell  <NA>   Myeloid
## AAACGGGCAAGAAAGG_13  30.32372  6.444399 NA    B_cell  <NA>   Myeloid
##                     CellFromTumor        ClusterName PatientID main_celltype
## AAACCTGAGTAACCCT_13             1 follicular B cells         3        B_cell
## AAACCTGCAACGATGG_13             1        macrophages         3       Myeloid
## AAACCTGCATTAGCCA_13             1        macrophages         3       Myeloid
## AAACCTGGTCCATGAT_13             1      CD4+ T cells          3        T cell
## AAACCTGTCCTGTACC_13             1        macrophages         3       Myeloid
## AAACGGGCAAGAAAGG_13             1        macrophages         3       Myeloid
```

```
class(metadata)
```

```
## [1] "data.frame"
```

```
class(ref_sc)
```

```
## [1] "matrix" "array"
```

```r
class(Bulk)
```

```
## [1] "matrix" "array"
```

ENIGMA also could adapt to the single cell object includes Seurat and SingleCellExperiment

```r
## Create Seurat object for single cell profile
ref_se = CreateSeuratObject(ref_sc)
ref_se@meta.data = metadata

egm = create_ENIGMA(bulk = Bulk, ref = ref_se, ref_type = "single_cell")
```

```
## Sun Jan 23 10:09:24 2022 Reference from Single Cell RNA-seq.

## Sun Jan 23 10:09:24 2022 Obtain reference from a Seurat object

## The `data` slot in the default assay is used. The default assay is RNA
## The `meta.data` slot in the Seurat object is used as metadata of reference
```

```r
## Create SingleCellExperiment for single cell profile
ref_se <- SingleCellExperiment(assays=list(counts = ref_sc))
colData(ref_se) = DataFrame(metadata)

egm = create_ENIGMA(bulk = Bulk, ref = ref_se, ref_type = "single_cell")
```

```
## Sun Jan 23 10:09:24 2022 Reference from Single Cell RNA-seq.

## Sun Jan 23 10:09:24 2022 Obtain reference from a SingleCellExperiment object

## The `counts` assay is used
## The `colData` assay in the SingleCellExperiment object is used as metadata of reference
```

2. Using FACS/Sorted RNA-seq to construct reference

User could also input FACS/Sorted RNA-seq as the reference, and ENIGMA would perform directly aggregation of FACS/Sorted RNA-seq according to cell type annotation to each samples

```r
ref_m = cbind(Tumor,Immune,Endothelial,Fibroblast)
colnames(ref_m) = paste0("sample_",1:ncol(ref_m),sep="")
metadata = data.frame(celltype = c(rep("Tumor",ncol(Tumor)),rep("Immune",ncol(Immune)),
rep("Endothelial",ncol(Endothelial)),
rep("Fibroblast",ncol(Fibroblast))))
rownames(metadata) = colnames(ref_m)

# ref_type = "sort"
egm_sort = create_ENIGMA(bulk = Bulk, ref = ref_m, ref_type = "sort",meta_ref=metadata)
```

```
## Sun Jan 23 10:09:24 2022 Reference from FACS/Sort Bulk RNA-seq/microarray/scRNA-seq dataset.
```

```r
## the metadata is required as data.frame format with merely one column of cell type annotation inform
head(metadata)
```

```
##          celltype
## sample_1    Tumor
## sample_2    Tumor
## sample_3    Tumor
## sample_4    Tumor
## sample_5    Tumor
## sample_6    Tumor
```

3. Input reference profile matrix directly to build ENIGMA object

User could also input reference profile matrix directly to build ENIGMA object

```
ref_m = cbind(rowMeans(Tumor),rowMeans(Immune),
              rowMeans(Endothelial),rowMeans(Fibroblast))
colnames(ref_m) = c("Tumor","Immune","Endothelial","Fibroblast")

## ref_type = "aggre"
egm_aggre = ENIGMA::create_ENIGMA(bulk = Bulk, ref = ref_m, ref_type = "aggre")
```

```
## Sun Jan 23 10:09:24 2022 Reference from aggregated FACS/Sort Bulk RNA-seq/microarray/scRNA-seq.
```

2. Cross-platform batch effects corrections

In order to remove the batch effects between reference and bulk, ENIGMA implement B/S-mode batch effects correction methods for users. When the ref_type = "single_cell", ENIGMA would automatically use S-mode to correct batch effects. User is required to input the cluster id (e.g. macrostate or cell cluster, etc…) into varname_cell_type, and give a specific number of simulated pseudo-bulk samples (Default. 1000).

```
egm = batch_correct(egm, varname_cell_type = "main_celltype", n_pseudo_bulk=1000)
```

```
## Sun Jan 23 10:09:24 2022 Reference is from Single Cell RNA-seq, doing batch correction in S mode.

## Sun Jan 23 10:09:24 2022 Generating pseudo bulk...
## using 6 cores...
## Sun Jan 23 10:10:21 2022 Doing ComBat...
## Found 897 genes with uniform expression within a single batch (all zeros); these will not be adjuste

## Found2batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments

## Adjusting the Data

## Sun Jan 23 10:10:31 2022 Restore reference...
```

When the ref_type = "sort" or "aggre", ENIGMA would automatically use B-mode to correct batch effects. User doesn't need to input any parameter, just run

```
egm_sort = batch_correct(egm_sort)
```

```
## Sun Jan 23 10:10:38 2022 Reference is from FACS Bulk RNA-seq/microarray, doing batch correction in B

## Sun Jan 23 10:11:07 2022 Run B-mode to correct batch effect...
## Sun Jan 23 10:11:07 2022 Doing ComBat...
## Found 3633 genes with uniform expression within a single batch (all zeros); these will not be adjust

## Found2batches

## Adjusting for0covariate(s) or covariate level(s)

## Standardizing Data across genes

## Fitting L/S model and finding priors

## Finding parametric adjustments
## Adjusting the Data

## Sun Jan 23 10:11:09 2022 Done.
```

If user doesn't want to perform batch effects correction, just skip this step.

3. Estimating cell type fraction matrix through reference based method

ENIGMA provide both robust linear regression (RLR) and CIBERSORT (CBS) methods to deconvolve bulk gene expression profile. The estimated fractions would be used in following CSE profile inference.

```
###Robust Linear Regression
egm = ENIGMA::get_cell_proportion(egm, method = "RLR")
```

```
## Sun Jan 23 10:11:09 2022 Calculating cell type proportion of bulk samples...
## Using Robust Linear Regression...
```

```
###CIBERSORT
#egm = ENIGMA::get_cell_proportion(egm, method = "CBS", nu.v = 0.5)
#Note: the CIBERSORT would cost for a long time to run.
```

4. Running ENIGMA

ENIGMA implement two type of model “trace norm” and “maximum L2 norm” (Default method). User could run maximum L2 norm model as follow

```
# random initialization
egm = ENIGMA::ENIGMA_L2_max_norm(egm)
```

```
## Sun Jan 23 10:11:14 2022 Optimizing cell type specific expression profile...
##    Ratio ranges from: 4270599.062249 - 4324844.261406
##    Ratio ranges from: 106388.285504 - 139869.452865
##    Ratio ranges from: 4006.640488 - 5152.592331
##    Ratio ranges from: 175.026074 - 200.176297
##    Ratio ranges from: 7.088569 - 10.621872
##    Ratio ranges from: 0.268721 - 0.867196
##    Ratio ranges from: 0.010118 - 0.086921
##    Ratio ranges from: 0.000382 - 0.009555
##    Ratio ranges from: 0.000014 - 0.001088
##    Ratio ranges from: 0.000001 - 0.000126
## Total loss: 20617781.7241509
## part1:1529623.02456231 part2:3624783.18974777 part3:78.4334553050786
## Perform Normalization...Sun Jan 23 10:11:21 2022 Done...
```

Also, user could run trace norm model as follow

```
# random initialization
egm = ENIGMA::ENIGMA_trace_norm(egm,gamma = 1)
```

```
## Using admm solver...
## Normalization...
## Converge in 231 steps
```

ENIGMA also could track the trained model through model_tracker.

```
# user also could specific the name of your trained model.
# If not, the model will be saved with the name as the format
```

```r
# "model type_the date you train the model".
# For example: maximum_L2_norm_model_Wed Jan 19 13:01:19 2022_trained

egm = ENIGMA::ENIGMA_L2_max_norm(egm,alpha = 0.9,beta = 0.1,model_tracker=TRUE,
model_name = "model1",preprocess="log")
```

```
## Sun Jan 23 10:13:50 2022 Optimizing cell type specific expression profile...
##    Ratio ranges from: 2178135.822319 - 2201906.790245
##    Ratio ranges from: 38173.157694 - 72033.693335
##    Ratio ranges from: 1228.459958 - 2665.092752
##    Ratio ranges from: 56.718716 - 101.895907
##    Ratio ranges from: 4.274993 - 11.748225
##    Ratio ranges from: 0.203518 - 3.806710
##    Ratio ranges from: 0.011097 - 1.989174
##    Ratio ranges from: 0.000684 - 1.068348
##    Ratio ranges from: 0.000046 - 0.582124
##    Ratio ranges from: 0.000003 - 0.321051
##    Ratio ranges from: 0.000000 - 0.178931
##    Ratio ranges from: 0.000000 - 0.100622
##    Ratio ranges from: 0.000000 - 0.057014
##    Ratio ranges from: 0.000000 - 0.032509
##    Ratio ranges from: 0.000000 - 0.018633
##    Ratio ranges from: 0.000000 - 0.010725
##    Ratio ranges from: 0.000000 - 0.006194
##    Ratio ranges from: 0.000000 - 0.003588
##    Ratio ranges from: 0.000000 - 0.002082
##    Ratio ranges from: 0.000000 - 0.001211
##    Ratio ranges from: 0.000000 - 0.000705
## Total loss: 5638679.45129871
## part1:1347332.88765161 part2:132223.273260213 part3:70.4512121014919
## Perform Normalization...Sun Jan 23 10:14:03 2022 Done...
```

```r
egm = ENIGMA::ENIGMA_L2_max_norm(egm,alpha = 0.9,beta = 0.2,model_tracker=TRUE,
model_name = "model2",preprocess="log")
```

```
## Sun Jan 23 10:14:03 2022 Optimizing cell type specific expression profile...
##    Ratio ranges from: 2649869.174368 - 2665204.737665
##    Ratio ranges from: 15431.843502 - 27253.858091
##    Ratio ranges from: 167.222199 - 311.839442
##    Ratio ranges from: 3.919249 - 7.724092
##    Ratio ranges from: 0.057744 - 1.398556
##    Ratio ranges from: 0.001011 - 0.465312
##    Ratio ranges from: 0.000021 - 0.161588
##    Ratio ranges from: 0.000000 - 0.057575
##    Ratio ranges from: 0.000000 - 0.020923
##    Ratio ranges from: 0.000000 - 0.007717
##    Ratio ranges from: 0.000000 - 0.002877
##    Ratio ranges from: 0.000000 - 0.001081
##    Ratio ranges from: 0.000000 - 0.000409
## Total loss: 5842254.93033892
## part1:1416539.12909585 part2:134715.971874071 part3:35.3807666983259
## Perform Normalization...Sun Jan 23 10:14:10 2022 Done...
```

```r
egm = ENIGMA::ENIGMA_L2_max_norm(egm,alpha = 0.9,beta = 0.3,model_tracker=TRUE,
model_name = "model3",preprocess="log")
```

```
## Sun Jan 23 10:14:10 2022 Optimizing cell type specific expression profile...
##    Ratio ranges from: 2843680.348073 - 2855267.894609
##    Ratio ranges from: 8139.698814 - 14044.699392
##    Ratio ranges from: 46.272501 - 77.571349
##    Ratio ranges from: 0.490106 - 1.919599
##    Ratio ranges from: 0.003707 - 0.366935
##    Ratio ranges from: 0.000034 - 0.087466
##    Ratio ranges from: 0.000000 - 0.021706
##    Ratio ranges from: 0.000000 - 0.005537
##    Ratio ranges from: 0.000000 - 0.001440
##    Ratio ranges from: 0.000000 - 0.000379
## Total loss: 5911392.83223765
## part1:1440104.69359872 part2:135555.646952621 part3:23.6203718209044
## Perform Normalization...Sun Jan 23 10:14:17 2022 Done...
```

```
egm = ENIGMA::ENIGMA_L2_max_norm(egm,alpha = 0.9,beta = 0.5,model_tracker=TRUE,
model_name = "Ken Wang",preprocess="log")
```

```
## Sun Jan 23 10:14:17 2022 Optimizing cell type specific expression profile...
##    Ratio ranges from: 3015676.460081 - 3023362.085946
##    Ratio ranges from: 3376.484507 - 5714.491417
##    Ratio ranges from: 8.968655 - 12.543409
##    Ratio ranges from: 0.031455 - 0.358873
##    Ratio ranges from: 0.000098 - 0.045663
##    Ratio ranges from: 0.000000 - 0.006364
##    Ratio ranges from: 0.000000 - 0.000924
## Total loss: 5967163.56766445
## part1:1459135.98295061 part2:136230.539265652 part3:14.1878972461466
## Perform Normalization...Sun Jan 23 10:14:21 2022 Done...
```

The list of trained models are saved in model_name, according to the name of models, users could check the basic information of each model and their outputs

```
egm@model_name
```

```
##        Model Name            Model Type
## model1     model1 maximum L2 norm model
## model2     model2 maximum L2 norm model
## model3     model3 maximum L2 norm model
## model4   Ken Wang maximum L2 norm model
```

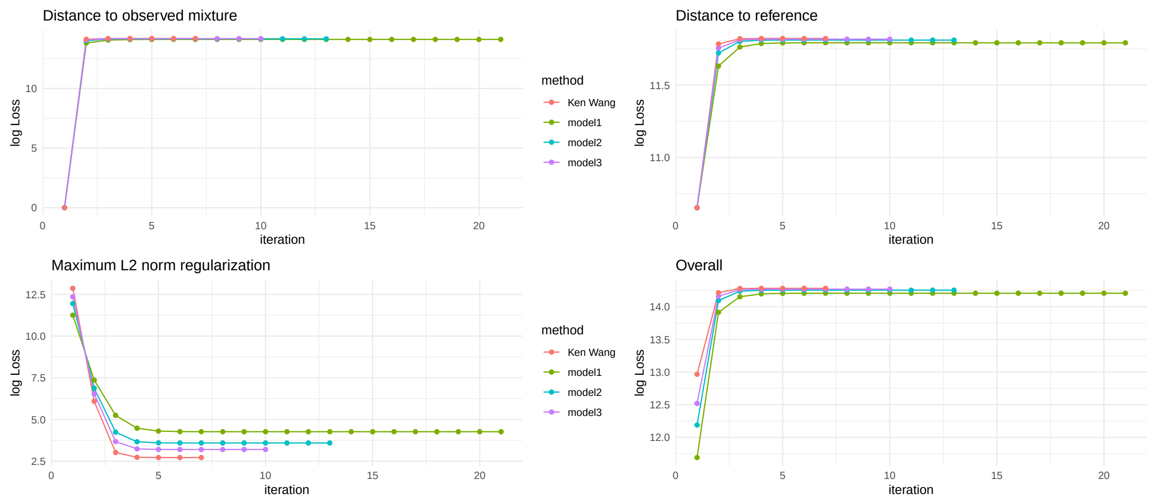
```
egm@model["Ken Wang"]
```

```
## $`Ken Wang`
## $`Ken Wang`$basic_infor
##   alpha beta step_size epsilon max_iter Normalize Normalize_method preprocess
## 1   0.9  0.5      0.01   0.001     1000      TRUE             frac        log
##    pos
## 1 TRUE
##
## $`Ken Wang`$result_CSE_normalized
## class: SingleCellExperiment
## dim: 17052 144
## metadata(0):
## assays(1): logcounts
## rownames(17052): A1BG A1CF ... ZZEF1 ZZZ3
## rowData names(0):
```

```
## colnames: NULL
## colData names(3): label sample cell_type
## reducedDimNames(0):
## altExpNames(0):
##
## $`Ken Wang`$loss_his
##                [,1]      [,2]         [,3]
## [1,] 1.209589e-26  42327.59 385391.98878
## [2,] 1.359938e+06 131155.10    440.72755
## [3,] 1.454532e+06 136008.85     19.59120
## [4,] 1.458932e+06 136221.86     14.39720
## [5,] 1.459127e+06 136230.36     14.19669
## [6,] 1.459136e+06 136230.57     14.18823
## [7,] 1.459136e+06 136230.54     14.18790
##
## $`Ken Wang`$result_CSE
## class: SingleCellExperiment
## dim: 17052 144
## metadata(0):
## assays(1): logcounts
## rownames(17052): A1BG A1CF ... ZZEF1 ZZZ3
## rowData names(0):
## colnames: NULL
## colData names(3): label sample cell_type
## reducedDimNames(0):
## altExpNames(0):
```

Users also could compare different models through inspecting their loss curves

```
plotLossCurve(egm,rlgType = "maximum L2 norm")
```



We could notice that the loss curve is not always decreasing, but this phenomena is relevant with the initialization point. Suppose we initialize the CSE as zero matrices

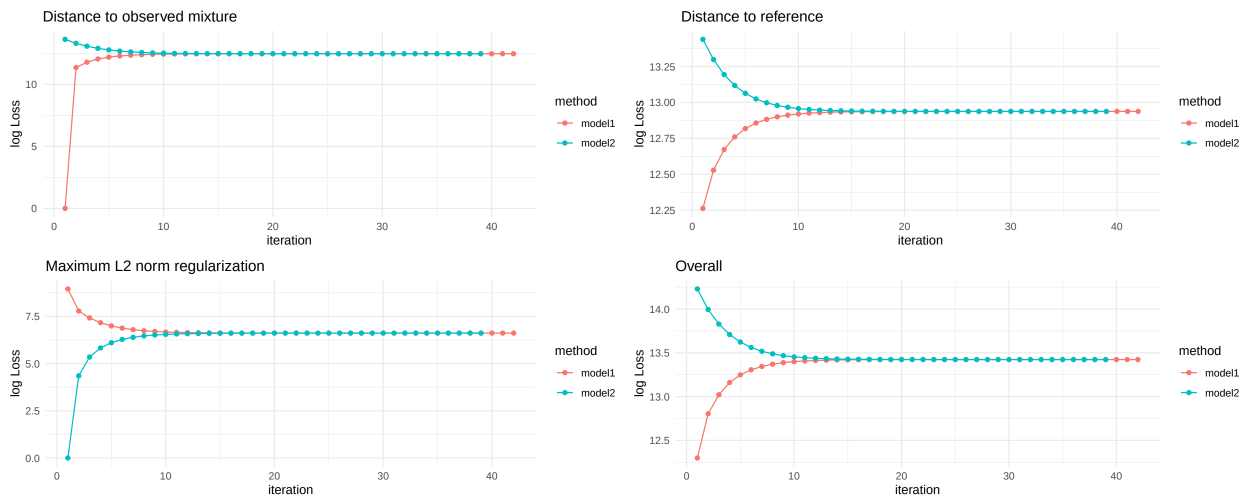
```
##clean all trained model
egm = clean_model(egm, NULL)

egm = ENIGMA::ENIGMA_L2_max_norm(egm,beta = 0.01,model_tracker=TRUE,
                                 verbose=FALSE,model_name = "model1",
                                          preprocess="log")

##
X_int = matrix(0, nrow = length(intersect(rownames(egm@bulk),rownames(egm@ref))),
                 ncol = ncol(egm@bulk))
egm = ENIGMA::ENIGMA_L2_max_norm(egm,beta = 0.01,model_tracker=TRUE,
                                          verbose=FALSE,model_name = "model2",
                                          preprocess="log",
                                          X_int = X_int)
```

Then, we could plot the loss curves as follow

```
plotLossCurve(egm,rlgType = "maximum L2 norm")
```



Therefore, users could use loss curve to see if their models' convergence. And all trained model would be saved in their ENIGMA object.