Cell Proportion Estimation and Adjustment Methods

Companion Tutorial for Kaushal *et al.*, BMC Bioinformatics (2017)

April 14, 2017

How to cite

This tutorial accompanies: Kaushal, A., Zhang, H., Karmaus, W.J.J. *et al.* Comparison of different cell type correction methods for genome-scale epigenetics studies. *BMC Bioinformatics* **18**, 216 (2017). https://doi.org/10.1186/s12859-017-1611-2. Please cite the article when using these workflows.

Overview

Cellular composition is a major source of confounding in epigenome-wide association studies (EWAS). This companion tutorial provides a practical, reproducible overview of approaches to *estimate* and *adjust* cell type proportions in peripheral blood DNA methylation data, mirroring the methods compared in Kaushal *et al.* (2017). Covered categories include **reference-free** methods (RefFreeEWAS, SVA, ReFACTor, RefFreeCellMix), **reference-based** methods (Houseman et al.; extended Houseman via `minfi`), and a factor-based strategy (RUV) that removes unwanted variation without explicitly estimating fractions.

Unless noted, methylation data are in β-values (rows = CpG sites, columns = samples), with primary and clinical covariates provided separately.

1 Example Data

To reproduce the code patterns below:

1. Download `FaSTLMM.Py.102.zip`.
2. Unzip and navigate to `FaSTLMM.Py.102/FastLmm.Py/doc/FastLmm-EWASher/demo`, which contains:
   - `input_data.txt` (27k CpGs across 204 subjects)
   - `input_phenotype.txt` (primary phenotype / cancer status)
   - `covariates.txt` (additional covariates)

   The code below loads and lightly preprocesses the files in R, yielding:

- `y1`: β-matrix ($p \times n$)
- `edata`: logit-like transform $\log_2\left(\frac{\beta}{1-\beta}\right)$ for modeling

- `x1,...,x6`: covariates

```r
# Core packages used by multiple sections
library(limma)

# Read methylation beta-values (rows = CpGs, cols = samples).
# Assumes first column holds CpG IDs.
y  <-read.table(".../input_data.txt", sep = "\t", header = TRUE, check.names = FALSE)
y1 <-y[, -1]
rownames(y1) <-y[, 1]

# Remove any CpGs with missingness (simple handling for tutorial purposes)
miss_rows <-unique(which(is.na(as.matrix(y1)), arr.ind = TRUE)[, 1])
if (length(miss_rows) > 0) {
  y1 <-y1[-miss_rows, ]
}

# M-value like transform for modeling
edata <-as.matrix(log2(y1 / (1 - y1)))

# Phenotype and covariates (adapt indices to your files as needed)
covar1 <-read.table(".../input_phenotype.txt", header = FALSE)
covar2 <-read.table(".../covariates.txt",  header = FALSE)

x1 <-as.numeric(covar1[, 2]) # primary covariate of interest (e.g., case/control)
x2 <-as.numeric(covar2[, 3])
x3 <-as.numeric(covar2[, 4])
x4 <-as.numeric(covar2[, 5])
x5 <-as.numeric(covar2[, 6])
x6 <-as.numeric(covar2[, 7])

# Convenience: number of samples (n) and CpGs (p)
n <-ncol(y1); p <-nrow(y1)
```

Listing 1: Load and prepare example data.

2 Reference-free Methods

2.1 RefFreeEWAS [2]

RefFreeEWAS models latent structure to account for cell mixture effects without external reference data and uses permutation-based inference. The workflow below estimates the latent dimension, fits the model, performs bootstrap resampling, and tests association with the primary covariate.

```r
library(RefFreeEWAS)

# Regress out primary covariate to help estimate latent dimension
tmp_lm <-lm(t(edata) ~x1)
dim_hat <-EstDimRMT(cbind(t(coef(tmp_lm)), t(resid(tmp_lm))), FALSE)$dim

# Design = intercept + primary + additional covariates
X <-cbind(1, x1, x2, x3, x4, x5, x6)

# Fit RefFreeEWAS and bootstrap
```

```r
rf_model <-RefFreeEwasModel(y1, X, dim_hat)
rf_boot  <-BootRefFreeEwasModel(rf_model, nBoot = 50)

rf_sum   <-summary(rf_boot)
est      <-rf_sum[, 2, 1, "mean"] # effect for primary covariate
est_sd   <-rf_sum[, 2, 1, "sd"]

# Wald-style test
Q2   <-(est / est_sd)^2
pval <-2 * pt(-sqrt(Q2), df = (n - (dim_hat + 6))) # df heuristic

fdr  <-p.adjust(pval, method = "fdr")
sig  <-which(fdr <= 0.05)

res_refree <-data.frame(
  CpG = rownames(y1)[sig],
  estimate = est[sig],
  se = est_sd[sig],
  pval = pval[sig],
  fdr = fdr[sig],
  row.names = NULL
)

write.table(res_refree, file = ".../RefFreeEWAS_results.csv",
            sep = ",", row.names = FALSE)
```

Listing 2: RefFreeEWAS analysis with bootstrap inference.

2.2 Surrogate Variable Analysis (SVA) [3]

SVA estimates hidden factors from residual structure and augments the design matrix with these factors to control confounding.

```r
# install.packages("BiocManager")
# BiocManager::install(c("sva","limma"))
# install.packages("MASS")
```

Listing 3: Install SVA and dependencies (one-time).

```r
library(sva)
library(MASS)

mod_full <-model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6)
mod_null <-model.matrix(~    x2 + x3 + x4 + x5 + x6)

svobj  <-sva(edata, mod_full, mod_null, n.sv = NULL, method = "two-step")
mod_sva <-cbind(mod_full, svobj$sv)

fit  <-lmFit(edata, mod_sva, method = "robust")
fit  <-eBayes(fit)

tab_sva <-topTable(fit, coef = "x1", number = nrow(edata), p.value = 0.05, adjust = "fdr")
write.table(tab_sva, file = ".../SVA_results.csv", sep = ",", row.names = TRUE)
```

Listing 4: SVA-based adjustment and testing with limma.

2.3 ReFACTor [4]

ReFACTor selects informative CpGs and derives principal components that capture cell type variability; these components are then included as covariates in differential testing. The simplified pipeline below performs variance filtering, low-rank ranking, PC extraction on top features, and testing with `limma`.

```r
library(stats)

# 1) Filter CpGs by variance (exclude bottom 5% by SD)
sd_vec <-apply(y1, 1, sd, na.rm = FALSE)
keep   <-which(sd_vec >= quantile(sd_vec, 0.05, na.rm = TRUE))
O      <-as.matrix(y1[keep, ]) # beta-values for selection stage
edata_k <-edata[keep, ]

# 2) Adjust each CpG for covariates (to emphasize cell-related residual patterns)
O_adj <-matrix(NA_real_, nrow = nrow(O), ncol = ncol(O),
               dimnames = list(rownames(O), colnames(O)))
for (i in seq_len(nrow(O))) {
  m <-lm(O[i, ] ~x1 + x2 + x3 + x4 + x5 + x6)
  O_adj[i, ] <-residuals(m)
}
O <-O_adj

# 3) Rank sites using low-rank approximation error
pc0    <-prcomp(scale(t(O)))
coeff  <-pc0$rotation
score0 <-pc0$x
Xlow   <-score0[, 1:7, drop = FALSE] %*% t(coeff[, 1:7, drop = FALSE])

A <-scale(t(O), center = TRUE, scale = FALSE)
B <-scale(Xlow, center = TRUE, scale = FALSE)

A <-t(t(A) * (1 / sqrt(pmax(colSums(A^2), .Machine$double.eps))))
B <-t(t(B) * (1 / sqrt(pmax(colSums(B^2), .Machine$double.eps))))

distances  <-sqrt(colSums((A - B)^2))
ranked_idx <-order(distances, decreasing = FALSE)
top_sites  <-ranked_idx[1:500]

# 4) PCs from top informative sites (ReFACTor components)
pc_ref    <-prcomp(scale(t(O[top_sites, ])))
refactorK <-7
ref_scores <-pc_ref$x[, 1:refactorK, drop = FALSE]
colnames(ref_scores) <-paste0("REF", seq_len(refactorK))

# 5) Include components in the design and test
design_ref <-model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6 + ref_scores)
fit_ref   <-lmFit(edata_k, design_ref, method = "ls")
fit_ref   <-eBayes(fit_ref)

tab_ref <-topTable(fit_ref, coef = "x1", number = nrow(edata_k), p.value = 0.05, adjust = "fdr")
write.table(tab_ref, file = ".../ReFACTor_results.csv", sep = ",", row.names = TRUE)
```

Listing 5: ReFACTor-style components and limma testing.

2.4 RefFreeCellMix (from `RefFreeEWAS`) [2]

RefFreeCellMix performs a matrix factorization to infer K putative cell-type profiles and sample-specific mixture weights (Ω). The inferred Ω can be included as covariates in association testing.

```r
library(RefFreeEWAS)

# Choose a plausible K (e.g., 6-7 for whole blood). Iterate in practice.
K <-7
cellmix <-RefFreeCellMix(Y = as.matrix(y1), mu0 = NULL, K = K, iters = 5,
                         Yfinal = NULL, verbose = TRUE)

Omega <-cellmix$Omega # n x K mixture weights
colnames(Omega) <-paste0("Cell", seq_len(ncol(Omega)))

design_mix <-model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6 + Omega)
fit_mix   <-lmFit(edata, design_mix, method = "robust")
fit_mix   <-eBayes(fit_mix)

tab_mix <-topTable(fit_mix, coef = "x1", number = nrow(edata), p.value = 0.05, adjust = "fdr")
write.table(tab_mix, file = ".../RefFreeCellMix_results.csv", sep = ",", row.names = TRUE)
```

Listing 6: RefFreeCellMix for mixture weights and limma testing.

3 Reference-based Methods

3.1 Houseman et al. (27k) [6, 7]

The original Houseman framework identifies regions that capture cell variability in sorted reference data and applies regression calibration to infer cell type proportions in mixed samples. The estimated fractions can then be added to the model matrix in downstream analyses.

```r
beta <-t(as.matrix(read.table(".../input_data.txt", sep = "\t", header = TRUE,
                              check.names = FALSE)))
source("Rcodes_Cell_mixture.R") # produces "Ewasher_data_housecell.csv"
```

Listing 7: Houseman 27k: supply input and source the script.

```r
library(limma); library(MASS)

cell <-read.csv("Ewasher_data_housecell.csv", header = TRUE, check.names = FALSE)

# Example: select specific cell fractions (adapt to your CSV structure)
# Here we include five fractions as covariates
mod_house <-model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6 +
                           cell[, 2] + cell[, 3] + cell[, 4] + cell[, 5] + cell[, 6])

fit_house <-lmFit(edata, mod_house, method = "robust")
fit_house <-eBayes(fit_house)

tab_house <-topTable(fit_house, coef = "x1", number = nrow(edata), p.value = 0.05, adjust = "fdr")
write.table(tab_house, file = ".../Houseman_results.csv", sep = ",", row.names = TRUE)
```

Listing 8: Adjusting with Houseman-estimated cell fractions.

3.2 Extended Houseman via `minfi` (450k) [8, 9]

For 450k arrays, `minfi` implements an extended approach using flow-sorted references and probe selection strategies compatible with 450k (and 27k). The estimated fractions can be incorporated as covariates in a linear modeling framework.

```
# BiocManager::install(c("minfi","quadprog","FlowSorted.Blood.450k",
#                        "IlluminaHumanMethylation450kmanifest",
#                        "IlluminaHumanMethylation450kanno.ilmn12.hg19"))
library(minfi); library(quadprog)
library(FlowSorted.Blood.450k)
library(IlluminaHumanMethylation450kmanifest)
library(IlluminaHumanMethylation450kanno.ilmn12.hg19)
```

Listing 9: Install and load the required Bioconductor packages.

```
# Read beta-values into grSet-like matrix (rows=CpGs, cols=samples)
dat ←read.table(".../input_data.txt", header = TRUE, check.names = FALSE)
gr  ←data.matrix(dat[, -1])
rownames(gr) ←dat[, 1]

reference ←FlowSorted.Blood.450k
ctypes    ←c("CD8T","CD4T","NK","Bcell","Mono","Gran","Eos")

compData ←minfi:::pickCompProbes(reference, cellTypes = ctypes)
coefs    ←compData$coefEsts
coefs    ←coefs[intersect(rownames(gr), rownames(coefs)), , drop = FALSE]

counts ←minfi:::projectCellType(gr[rownames(coefs), ], coefs)
rownames(counts) ←colnames(gr)

write.table(counts, file = ".../minfi_cellprops.csv", sep = ",", row.names = TRUE)

# Use estimated fractions as covariates
cell ←read.csv(".../minfi_cellprops.csv", header = TRUE, check.names = FALSE)
# Example: include six fractions as covariates (drop one to avoid collinearity)
mod_minfi ←model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6 +
                          cell$CD4T + cell$NK + cell$Bcell + cell$Mono + cell$Gran + cell$Eos)

fit_minfi ←lmFit(edata, mod_minfi, method = "robust")
fit_minfi ←eBayes(fit_minfi)

tab_minfi ←topTable(fit_minfi, coef = "x1", number = nrow(edata), p.value = 0.05, adjust = "fdr")
write.table(tab_minfi, file = ".../minfi_results.csv", sep = ",", row.names = TRUE)
```

Listing 10: Estimate cell fractions with minfi and adjust in limma.

4 Removing Unwanted Variation (RUV)

RUV leverages negative control probes to estimate latent factors that capture technical and biological confounding, including cell mixture effects. Below, principal components derived from blood-related CpGs are used as covariates in a robust linear model.

```
library(data.table)
```

```r
library(psych)
library(pracma)

# Control set of blood-related CpGs (e.g., top ~600)
beta_ctrl <-read.csv(".../Top_600_Ewasher_27k.csv", header = TRUE, check.names = FALSE)
B        <-beta_ctrl[, -1]
rownames(B) <-beta_ctrl[, 1]
B_t      <-t(B) # samples x 600

# PCA on standardized control set
pc_ctrl <-prcomp(B_t, center = TRUE, scale. = TRUE)

# Option: varimax-rotated loadings for interpretability; then compute scores
k <-4
rawLoad <-pc_ctrl$rotation[, 1:k, drop = FALSE] %*% diag(pc_ctrl$sdev[1:k], k, k)
rotLoad <-varimax(rawLoad, normalize = TRUE)$loadings
invLoad <-t(pracma::pinv(rotLoad))
scores  <-scale(B_t) %*% invLoad # samples x k

PC1 <-scores[, 1]; PC2 <-scores[, 2]; PC3 <-scores[, 3]; PC4 <-scores[, 4]

design_ruv <-model.matrix(~ x1 + x2 + x3 + x4 + x5 + x6 + PC1 + PC2 + PC3 + PC4)
fit_ruv   <-lmFit(edata, design_ruv, method = "robust")
fit_ruv   <-eBayes(fit_ruv)

tab_ruv <-topTable(fit_ruv, coef = "x1", number = nrow(edata), p.value = 0.05, adjust = "fdr")
write.table(tab_ruv, file = ".../RUV_like_results.csv", sep = ",", row.names = TRUE)
```

Listing 11: PC/Factor-based removal of unwanted variation using control CpGs.

5 FaST-LMM-EWASher (Python) [5]

FaST-LMM-EWASher is a linear mixed model (LMM) framework that uses spectral decomposition for efficient maximum likelihood estimation and accounts for sample structure, including cell mixture.

```
python ..\..\..\fastlmm\ewasher\src\fastlmm-ewasher.py input_data.txt \
       input_phenotype.txt -covar covariates.txt
```

The output (e.g., `results/out_ewasher.txt`) contains p-values and test statistics for each CpG.

6 Practical Notes and Tips

- **Model design.** Avoid perfect collinearity when adding cell fractions or surrogate factors; drop one column if needed.
- **Choice of K.** For RefFreeCellMix/ReFACTor, evaluate multiple K values and check stability using diagnostic metrics (e.g., test statistic inflation, biological plausibility).
- **Multiple testing.** Use FDR control (e.g., Benjamini–Hochberg) across all tested CpGs.
- **Transformations.** Modeling on M-values (logit of β) often improves distributional assumptions; we use $\log_2(\beta/(1-\beta))$ for illustration.

References

[1] Kaushal A., Zhang H., Karmaus W.J.J., *et al.* (2017). Comparison of different cell type correction methods for genome-scale epigenetics studies. *BMC Bioinformatics* **18**, 216. https://doi.org/10.1186/s12859-017-1611-2.

[2] Houseman E.A., Molitor J., Marsit C.J. (2014). Reference-free cell mixture adjustments in analysis of DNA methylation data. *Bioinformatics* **30**(10):1431–1439.

[3] Leek J.T., Storey J.D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genetics* **3**(9):e161.

[4] Zou J., Lippert C., Heckerman D., Aryee M., Listgarten J. (2014). Epigenome-wide association studies without the need for cell-type composition. *Nature Methods* **11**(3):309–311.

[5] Lippert C., Listgarten J., Liu Y., Kadie C.M., Davidson R.I., Heckerman D. (2011). FaST linear mixed models for genome-wide association studies. *Nature Methods* **8**(10):833–835.

[6] Houseman E.A., Accomando W.P., Koestler D.C., *et al.* (2012). DNA methylation arrays as surrogate measures of cell mixture distribution. *BMC Bioinformatics* **13**:86.

[7] Carroll R.J., Ruppert D., Stefanski L.A., Crainiceanu C.M. (2012). *Measurement Error in Nonlinear Models: A Modern Perspective*. CRC Press.

[8] Jaffe A.E., Irizarry R.A. (2014). Accounting for cellular heterogeneity is critical in epigenome-wide association studies. *Genome Biology* **15**(2):R31.

[9] Reinius L.E., Acevedo N., Joerink M., *et al.* (2012). Differential DNA methylation in purified human blood cells: implications for cell lineage and disease susceptibility. *PLoS One* **7**(7):e41361.