RNA-seq batch effect removal

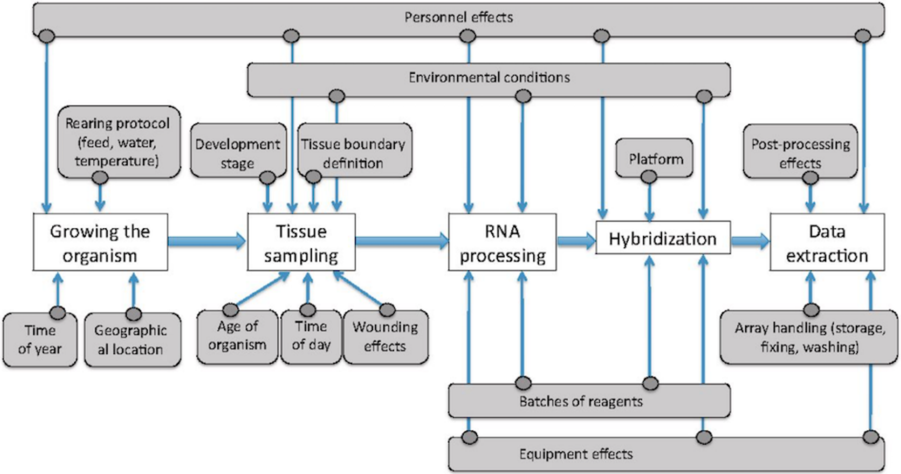
Mikhail Dozmorov

Spring 2018

Batch effects

- Batch effects are widespread in high-throughput biology. They are artifacts not related to the biological variation of scientific interests.
- For instance, two experiments on the same technical replicates processed on two different days might present different results due to factors such as room temperature or the two technicians who did the two experiments.
- Batch effects can substantially confound the downstream analysis, especially meta-analysis across studies.

Batch sources



ComBat: Removing known batch effects

ComBat - Location-scale method

The core idea of ComBat was that the observed measurement Y_{ijg} for the expression value of gene g for sample j from batch i can be expressed as

$$Y_{ijg} = \alpha_g + X\beta_g + \gamma_{ig} + \delta_{ig}\epsilon_{ijg}$$

where X consists of covariates of scientific interests, while γ_{ig} and δ_{ig} characterize the additive and multiplicative batch effects of batch i for gene g.

https://www.bu.edu/jlab/wp-assets/ComBat/Abstract.html

https://github.com/brentp/combat.py

ComBat: Removing known batch effects

After obtaining the estimators from the above linear regression, the raw data Y_{ijg} can be adjusted to Y^*_{ijg}:

$$Y^*_{ijg} = \frac{Y_{ijg} - \hat{\alpha_g} - X\hat{\beta_g} - \hat{\gamma_{ig}}}{\hat{\delta_{ig}}} + \hat{\alpha_g} + X\hat{\beta_g}$$

For real application, an empirical Bayes method was applied for parameter estimation.

https://www.bu.edu/jlab/wp-assets/ComBat/Abstract.html

SVA: Removing unknown batch effects

- When batches were unknown, the surrogate variable analysis (SVA) was developed.
- The main idea was to separate the effects caused by covariates of our primary interests from the artifacts not modeled.
- Now the raw expression value Y_{jg} of gene g in sample j can be formulated as:

$$Y_{jg} = \alpha_g + X\beta_g + \sum_{k=1}^{K} \lambda_{kg}\eta_{kj} + \epsilon_{jg}$$

where η_{kj}s represent the unmodeled factors and are called as "surrogate variables".

SVA

- Once again, the basic idea was to estimate η_{kj}s and adjust them accordingly.
- An iterative algorithm based on singular value decomposition (SVD) was derived to iterate between estimating the main effects $\hat{\alpha_g} + X\hat{\beta}_g$ given the estimation of surrogate variables and estimating surrogate variables from the residuals $r_{jg} = Y_{jg} - \hat{\alpha_g} - X\hat{\beta}_g$

sva package in Bioconductor

- Contains ComBat function for removing effects of known batches.
- Assume we have:
  - edata: a matrix for raw expression values
  - batch: a vector named for batch numbers.

```
# Design matrix containing all covariates but not batch
modcombat = model.matrix(~1, data=as.factor(batch))

combat_edata = ComBat(dat=edata, batch=batch, mod=modcombat,
                      par.prior=TRUE, prior.plot=FALSE)
```

https://bioconductor.org/packages/release/bioc/html/sva.html

SVASEQ

- svaseq, the version of SVA algorithm adapted to sequencing data
- Suggests applying a moderated log transformation to the count data or fragments per kilobase of exon per million fragments mapped (FPKM) to account for the nature of discrete distributions

https://academic.oup.com/nar/article-lookup/doi/10.1093/nar/gku864

https://www.bioconductor.org/packages/release/bioc/html/sva.html

SVASEQ

```
dat <- counts(dds, normalized=TRUE)
idx <- rowMeans(dat) > 1 # Indices of rows with non-zero count
dat <- dat[idx,] # Remove rows with low read counts
# Design matrix containing all covariates
mod <- model.matrix(~ dex, colData(dds))
# Design matrix assuming no effect
mod0 <- model.matrix(~ 1, colData(dds))
svseq <- svaseq(dat, mod, mod0, n.sv=2)
# Number of significant surrogate variables is:  2
ddssva <- dds
ddssva$SV1 <- svseq$sv[,1]
ddssva$SV2 <- svseq$sv[,2]
# Add latent variables to the design formula
design(ddssva) <- ~ SV1 + SV2 + dex
```

RUVSeq: Remove Unwanted Variation from RNA-Seq Data

- Instead of a direct transformation on the raw counts or FPKM, remove unwanted variation (RUV) adopted a generalized linear model. For n samples and J genes,

$$log\ E[Y|W,X,O] = W\alpha + X\beta + O$$

- Y is an $n \times J$ matrix of the observed gene-level read counts,
- X is an $n \times p$ matrix of the p covariates of interest (wanted variation),
- β is a $p \times J$ matrix of parameters of interest
- W is an $n \times k$ matrix corresponding to hidden factors (unwanted variation),
- α is a $k \times J$ matrix of nuisance parameters,
- O is an $n \times J$ matrix of offsets
- Goal - estimate the unwanted factors W

RUVSeq: Remove Unwanted Variation from RNA-Seq Data

Three approaches:

- **RUVg** uses negative control genes, assumed not to be differentially expressed with respect to the covariates of interest, estimates $\hat{W}$ from this subset,
- **RUVs** uses negative control samples for which the covariates of interest are constant
- **RUVr** uses residuals from a first-pass GLM regression of the unnormalized counts on the covariates of interest.

https://www.bioconductor.org/packages/devel/bioc/html/RUVSeq.html

Risso, Davide, John Ngai, Terence P Speed, and Sandrine Dudoit. “Normalization of RNA-Seq Data Using Factor Analysis of Control Genes or Samples.” Nature Biotechnology 32, no. 9 (August 24, 2014): 896–902. https://doi.org/10.1038/nbt.2931.
https://www.nature.com/articles/nbt.2931
https://bioconductor.org/packages/release/bioc/html/RUVSeq.html

Peixoto, Lucia, Davide Risso, Shane G. Poplawski, Mathieu E. Wimmer, Terence P. Speed, Marcelo A. Wood, and Ted Abel. “How Data Analysis Affects Power, Reproducibility and Biological Insight of RNA-Seq Studies in Complex Datasets.” Nucleic Acids Research 43, no. 16 (September 18, 2015): 7664–74. https://doi.org/10.1093/nar/gkv736.
https://github.com/drisso/peixoto2015_tutorial

BatchQC - Batch Effects Quality Control

- A Bioconductor package with a GUI (shiny app)
- Allows for interactive visualization of batch effects via clustering, dimensionality reduction
- Applies ComBat or SVA to remove batches and observe the effect

https://github.com/mani2012/BatchQC