Combating batch effects: surrogate variable analysis and ComBat

What are batch effects?

Batch effects arise from differences between samples that are not rooted in the design of the experiment and can have different sources. This could be a matter of people handling the experiments or even the location where the experiment is conducted. It can also be different batches of reagents or even biological artifacts (e.g. growth location).

Why we want to avoid batch effects?

Batch effects are undesirable because their careless correction can lead to the loss of the biological signal contained in the data. In addition, they can introduce genes that are expressed differently between groups - they are not biologically significant and are only detected between batches. It follows, therefore, that this is an important element of research, and proper handling of data is crucial to obtain repeatable and successful research.

Minimizing batch effects

Technical factors that potentially lead to batch effects can be avoided with mitigation strategies in the lab and during sequencing:

- Some laboratory strategies:
    - same-day cell sampling
    - use of the same handling staff
    - protocols
    - reagent lots
    - reducing PCR amplification error
    - using the same equipment
- Some sequencing strategies may include multiplexing the libraries in flow cells.

Although one can try to minimize batch effects by implementing the above-mentioned laboratory strategies, one must be aware of their occurrence.

Computational batch correction

Computational batch correction seeks to remove technical variability from the data in an effort to prevent this variability from interfering with subsequent analysis. There are several batch correction methods and tools that implement them, but in general there are currently two classes of batch correction methods:

1. those that trying to identify and control potential batch effects -> **SVA** - may remove biological variation of interest.

2. those that using linear modeling when batches are known or assumed -> **ComBat** - may miss artifacts due to biology, and unannotated technical variation.

SVA - surrogate variable analysis

The main concept behind SVA is modelling the potential confounding factors, which may or may not be known, as singular vectors (so called 'surrogate variables') derived from a singular value decomposition (SVD).

We have a data matrix, X_{ij}, with i(i=1,…,p) labeling the features (genes,CpGs,…) and j(j=1,…,n) labeling the samples, with p>>n. Furthermore, we assume that each row of X has been mean centered, and that we have a phenotype of interest (POI) encoded by a vector $\vec{y}=\{y_1,\ldots,y_n\}$. The starting model for SVA takes the form

$$X_{ij}=f_i(y_j)+\epsilon_{ij}$$

SVA proceeds by performing a SVD of the residual matrix

$$R=UDV^T$$

where the residual matrix is defined by $R_{ij}\equiv X_{ij}-\hat{f}_i(y_j)$. Thus, the singular vectors of the SVD capture variation which is orthogonal to the variation associated with the POI. This residual variation is likely to be associated with other biological factors not of direct interest, or experimental factors, all of which constitute potential confounders. SVA provides a prescription for the construction of surrogate variables in terms of the singular vectors of this SVD.

More information and the entire derivation of the SVA algorithm can be found in this paper: ,,*Capturing heterogeneity in gene expression studies by surrogate variable analysis"* - https://pubmed.ncbi.nlm.nih.gov/17907809/

ComBat - Combating batch effects when combining batches

It is a linear model which adjusts for mean shift and variance scaling in batch data. The most important features in this model are the scanner error terms γ_{iv} and δ_{iv}, both of which we want to “remove” to bring our data into a common (batch-free) space.

$$y_{ijv} = \alpha_v + \beta_v X_j + \gamma_{iv} + \delta_{iv}\epsilon_{ijv}$$

i - the batch identifier,
j - the subject,
v - the feature
$y_{i\square v}$ - the measured feature v value for subject j in batch i,
α_v - the feature intercept
X - the covariates for subject j,
β_v - the coefficients for covariates
γ_{iv} - the batch additive (mean shift) term,
δ_{iv} - the batch variance scaling term
ε - random, normally distributed noise with mean 0 and standard deviation σ_v.

ComBat assumes either parametric or nonparametric hierarchical Bayesian priors in the batch effect parameters (γig and δig) and uses an empirical Bayes procedure to estimate these parameters. This procedure pools information across genes in each batch to shrink the batch effect parameter estimates toward the overall mean of the batch effect empirical estimates. These are used to adjust the data for batch effects. This approach provides a robust and often more accurate adjustment for the batch effect on each gene.

More information and the entire derivation of the SVA algorithm can be found in this paper: *„COMBAT: A Combined Association Test for Genes Using Summary Statistics”* - https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5676236/

Conclusion

ComBat is used when we know how the batch effect is structured. We know what may cause the batch effect - for example microarrays prepared and run at different dates, where we expect "date" to cause a change in and of itself.

SVA is used when we suspect that our data has underlying variation that's not being caused by the biology you're interested in or factors that you can easily identify. This often happens when there are a combination of background effects affecting our data, but we don't know much about them.

Literature

- https://en.wikipedia.org/wiki/Batch_effect

- Wang M, Huang J, Liu Y, Ma L, Potash JB, Han S. COMBAT: A Combined Association Test for Genes Using Summary Statistics. Genetics. 2017 Nov;207(3):883-891. doi: 10.1534/genetics.117.300257. Epub 2017 Sep 6. PMID: 28878002; PMCID: PMC5676236.

- Yuqing Zhang, Giovanni Parmigiani, W Evan Johnson, *ComBat-seq*: batch effect adjustment for RNA-seq count data, *NAR Genomics and Bioinformatics*, Volume 2, Issue 3, September 2020, lqaa078, https://doi.org/10.1093/nargab/lqaa078

- Sprang, M., Andrade-Navarro, M.A. & Fontaine, JF. Batch effect detection and correction in RNA-seq data using machine-learning-based automated assessment of quality. *BMC Bioinformatics* 23 (Suppl 6), 279 (2022). https://doi.org/10.1186/s12859-022-04775-y

- Zhang, Y., Jenkins, D.F., Manimaran, S. *et al.* Alternative empirical Bayes models for adjusting for batch effects in genomic studies. *BMC Bioinformatics* 19, 262 (2018). https://doi.org/10.1186/s12859-018-2263-6

- Jaffe, A.E., Hyde, T., Kleinman, J. *et al.* Practical impacts of genomic data “cleaning” on biological discovery using surrogate variable analysis. *BMC Bioinformatics* 16, 372 (2015). https://doi.org/10.1186/s12859-015-0808-5