A geometrical approach based on PCA for the benchmarking of batch effect correction algorithms in the context of Baseline Bulk RNASeq experiments

Trainee internship realised in Gene Expression Team, EMBL-EBI, UK

Guillaume Heger[*]

under the supervision of Irene Papatheodorou[†] *and Pablo Moreno*[‡]

1st April - 26th July 2019

Introduction

General workflow for the integration of datasets

Several tools to benchmark the different correction algorithms

I introduced three tools that I used to benchmark three batch effect correction algorithms : Empirical Bayes method (also known as ComBat), Removal of Unwanted Variation (RUV) and Mutual Nearest Neighbours (MNN). The last two algorithms have an integer parameter, denoted as k, that I have made vary for the comparison.

- Guided PCA comes from an article in *Bioinformatics* by Sarah Reese et al. I wrote a new implementation of this tool in R, where I added some features : the notion of rank of variance, as well as an extrapolation of the δ statistic to the higher dimensions of PCA.
- Entropy of mixing comes from the exchange that I had during my internship with Ruben Chazarra[1] who was working on a benchmarking of batch effect correction methods within the context of Single Cell experiments.
- I developed myself the approach of Eigengenes Angles in regards to the heterogeneity of the datasets I had to integrate. Contrarily to the two previous approaches, this one doesn't assume any identity of distribution (in terms of biological characteristics) between the datasets, that may lead to confuse batch effect and biological differences in the case of heterogeneous datasets.

I have implemented Guided PCA and Eigengenes Angles methods in a R package called `eigenangles`[2].

Guided PCA

The idea of guided PCA is to perform PCA on a batch-aggregated dataset. All the samples from the same batch are averaged (or summed) to form one sample and PCA is performed on these new samples (one sample per batch). This PCA yields some geometrical axes, that we will called guided principal components, different from the ones that standard PCA would give. The original samples (not aggregated) are then projected on these new directions so that the parts of variance of these axes can be estimated.

[*] Engineering Student in Applied Mathematics at École Centrale de Nantes

[†] Leader of Gene Expression Team, EMBL-EBI

[‡] Expression Atlas Data Production Project Leader in Gene Expression Team, EMBL-EBI

[1] Visiting Scientist at Wellcome Sanger Institute

[2] Released on my Github account : https://github.com/gheager/eigenangles

The guided principal components are found in such a way that they represent somehow the directions of batch effect. If these directions are important compared to standard principal components, i.e. if their variance is comparable to the ones of low rank principal components, it would mean that batch effect is important.

The original article about gPCA introduces a statistic called δ defined as

$$\delta = \frac{\mathbb{V}gPC_1}{\mathbb{V}PC_1}$$

where gPC_1 denotes the projection of the data (considered as a random variable) on the first guided principal component and PC_1 denotes its projection on the first principal component. δ is actually the variance of the first axis of gPCA, normalised by the variance of the first axis of PCA. As the first axis of PCA is the 1-dimensional subspace which maximises the variance of the projected data, one necessarily has $\delta \leq 1$.

I introduce an extrapolation of this statistic to higher dimensions by considering for $1 \leq k \leq n_{batches}$ by considering :

$$\delta_k := \frac{\mathbb{V}(gPC_1, ..., gPC_k)}{\mathbb{V}(PC_1, ..., PC_k)}$$

where gPC_k (resp. PC_k) denotes the projection of data on the k^{th} axis of gPCA (resp. PCA) and $(gPC_1, ..., gPC_k)$ denotes the vector formed by the first k guided principal components, which is actually the projection of data on the k-dimensional subspace generated by them.

Since the guided principal components are mutually orthogonal, as well as the principal components, we can calculate this new statistic as :

$$\delta_k = \frac{\sum_{i=1}^{k} \mathbb{V}gPC_i}{\sum_{i=1}^{k} \mathbb{V}PC_i} = \frac{\mathbb{V}gPC_1 + ... + \mathbb{V}gPC_k}{\mathbb{V}PC_1 + ... + \mathbb{V}PC_k}$$

I also introduce the notion of ranks of variance, defined for a guided principal component as the greatest rank such that the standard principal component of this rank has greater variance than the considered guided principal component.

Entropy of mixing

Eigengenes Angles

I introduce here a new method to evaluate the importance of batch effect within an integrated dataset. This method is based on some geometric consideration on the principal components of the single datasets compared to those of their merger (with or without correction). As principal components represents geometrical directions, a way to compare them is to estimate the angle between them.

In the simple case where datasets are supposed to have the same distribution, we expect actually the single datasets to have their principal components similar between them and to those of the merged dataset.

As the coefficients of principal components are akin to weights on the genes and represent somehow their involvement in the variance of a dataset, they are a good summary of the information provided by a dataset on the genes. Thus low angles between the respective principal components of each dataset and the ones of their merger means somehow that the information provided by the single datasets has been conserved through their integration.

Advantages of Eigengenes Angles approach for heterogeneous batches

The advantage of the Eigengenes Angles approach is that it is applicable for the integration of heterogeneous datasets, i.e. datasets whose samples have unshared biological characteristic (for example, a dataset with samples from brain and lung and another one whose samples come from lung and liver) or whose proportions of biological group are different. In such a situation, the datasets cannot be supposed to have the same distribution and both gPCA and Entropy of mixing indices may fail to catch the batch effect with precision. Indeed :

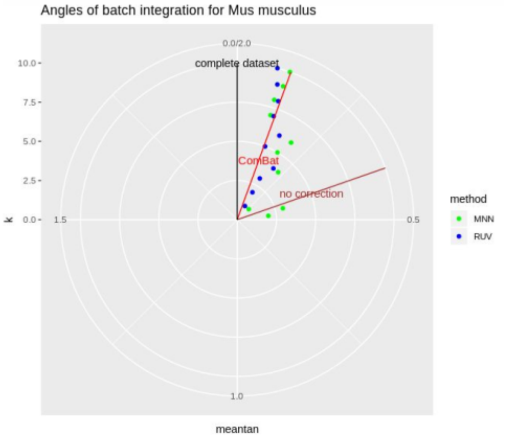

Figure 1: Eigengenes Angles for *Mus musculus*

- gPCA may detect important batch effect in an integrated dataset (corrected or not) only because its batches have different compositions.
- In the same way, Entropy of mixing may be irrelevant as its principle is to check whether the batches are well mixed among the nearest neighbours of every sample. However if a sample from a batch doesn't have any biological replicate in the other batches, there is no reason to expect to find samples from the other batches among its nearest neighbours, but rather its biological replicates from the same batch.

Benchmarking for *Mus musculus* and *Homo sapiens* baseline datasets with organism part as biological covariate

Guided PCA

Entropy of mixing

Eigengenes Angles

Conclusion