Two criterions to benchmark batch effect correction algorithms with an application to the integration of Baseline RNASeqdatasets from Expression Atlas

Guillaume Heger

11/06/2019

I present here two different methods to be used in the context of the integration of datasets from different sources, subject to batch effect. I extracted one of them from an existing paper and I introduce a new one in this paper. Although both of them are based on Principal Components Analysis, the first one has what I would call a "sample-based" approach while the second one focuses more on the information given about genes.

The final aim of this work is the creation of several Expression Atlases by merging the RNASeq datasets available on the existing Expression Atlas. The building of such atlases shall be done using heterogenous datasets which can definitely not be supposed to follow the same distribution. Therefore many batch effect correction algorithms can already be considered as irrelevant for this problem (batch-mean centering to name but one, as well as every method which doesn't take biological factors into account).

A sample-based detection method

The idea of gPCA comes from an article by Sarah Reese [A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis.] I kept this fundamental idea and added some other features to it, among which is the notion of variance rank. In a nutshell, gPCA performs PCA on batchwise aggregated data (merged from all experiments) (aggregation can be sum or mean). The number of samples in the new dataset is the number of batches. Then we can project the non-aggregated data on the new "guided" principal components (gPCs). To see the importance of the "batch variable", one shall compare the parts of variance of the gPCs with the ones of the original principal components (PCs or eigengenes, i.e. from PCA on the merged dataset without batchwise aggregation). We can compute a δ statistic and its associated p-value (see [reference] for more information). This δ statistic represents somehow the part of variance of batch effect. We would like it to be small, or anyway with non-significant p-value, which would mean that batch effect has as much effect as normal random noise.

```
#loading experiments
geod74747<-get(load(
  'Mouse Expression Atlas/batch data/E-GEOD-74747-atlasExperimentSummary.Rdata'
))$rnaseq
mtab4644<-get(load(
  'Mouse Expression Atlas/batch data/E-MTAB-4644-atlasExperimentSummary.Rdata'
))$rnaseq
#extraction of counts matrices
geod74747 %>% assays %$% counts -> geod
mtab4644 %>% assays %$% counts -> mtab
#merging the datasets without correction
cbind(geod,mtab)->whole
#batch and tissue factors
batch<-c('geod' %>% rep(ncol(geod)),'mtab' %>% rep(ncol(mtab)))
tissue<-c(geod74747$organism_part,mtab4644$organism_part)
#extracting samples from tissues which are common to both experiments
```

```r
common.tissues<-intersect(tissue[batch=='geod'],tissue[batch=='mtab'])
whole%<>%extract(,tissue%in%common.tissues)
batch%<>%extract(tissue%in%common.tissues)
tissue%<>%extract(tissue%in%common.tissues)
#pre-filtering genes with only zero counts on one of the experiments
filter<-rowSums(whole[,batch=='geod']!=0)>0 & rowSums(whole[,batch=='mtab']!=0)>0
filtered<-whole[filter,] %>% log1p
```

```r
filtered %>% gPCA(batch) -> gfiltered
```

```
## Computing PCA
## Computing gPCA
## part of variance from gPC1 : 0.756350096719459
## delta statistic : 0.999874798191682
## cumulative delta statistics :
## delta_1=0.999874798191682
##  delta_2=0.925193326441326
## variance ranks : 1
##  variance ranks : 13
```

The first figure shown is the absolute part of variance of the first gPC, analogous to the parts of variance typically computed for classical PCA. Then we can see the δ statistic which is the ratio of the previous part of variance from gPC_1 by the part of variance from PC_1 :

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

As PC1 is the one-dimensional axis which has most variance in the space of eigengenes, this ratio shall be less than 1. If δ is close to 1, it means that batch effect is responsible of a big part of variance in the dataset.

The following figures are cumulative δ statistics. Their number is the number of batches considered. δ_1 is exactly δ while δ_2 is the ratio between the cumulative part of variance of gPC_1 and gPC_2 and the cumulative part of variance of PC_1 and PC_2. In a general way :

$$\delta_k = \frac{\mathbb{V}gPC_1 + ... + \mathbb{V}gPC_k}{\mathbb{V}PC_1 + ... + \mathbb{V}PC_k}$$

Finally, the variance ranks are displayed. They position the parts of variance of the gPCs among the parts of variance of the PCs. The variance rank of gPC1 is the minimal number n such that PC_{n+1} has a smaller part of variance. Here the variance rank of gPC_1 is 1, which means that the batch effect creates variance in a comparable order of magnitude to PC_1.

We can show all these results graphically. Figure 1 shows the first plan of gPCA (gPC_1 and $gPC2$). The grey lines link the samples extracted from the same tissue in the different experiments.

```r
gfiltered %>% viz_gpca + geom_line(aes(group=tissue),colour='grey')
```

The function `gPCA` computes both gPCA and PCA, in order to compare them. Figure 2 is the first plan of PCA (PC_1 and PC_2).

```r
gfiltered %>% viz_gpca(guided=FALSE) + geom_line(aes(group=tissue),colour='grey')
```

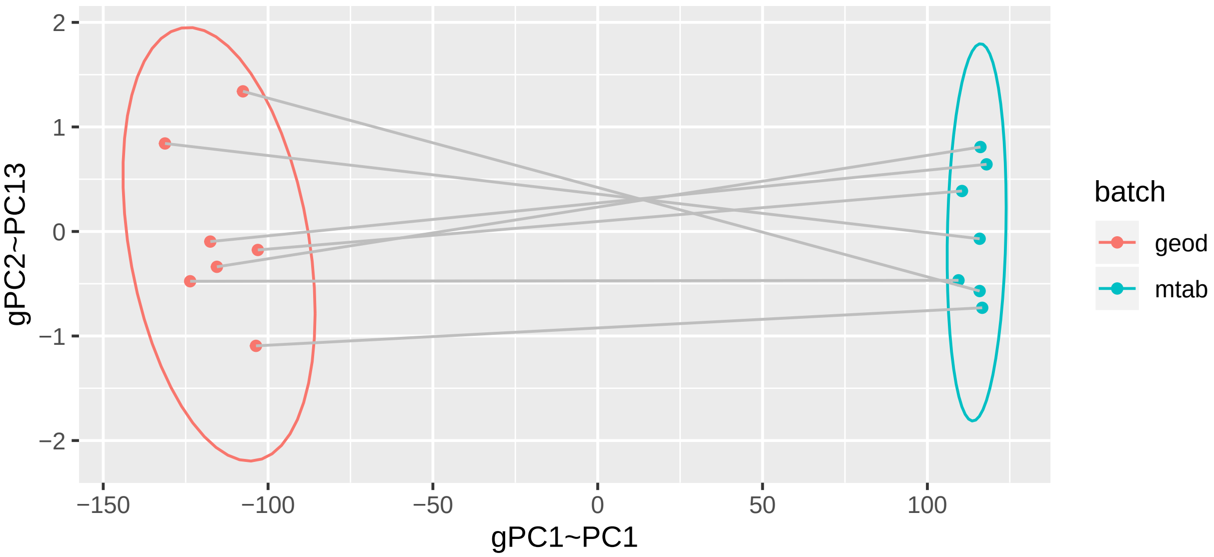

Figure 1: First plan of variance of the merged dataset for guided PCA

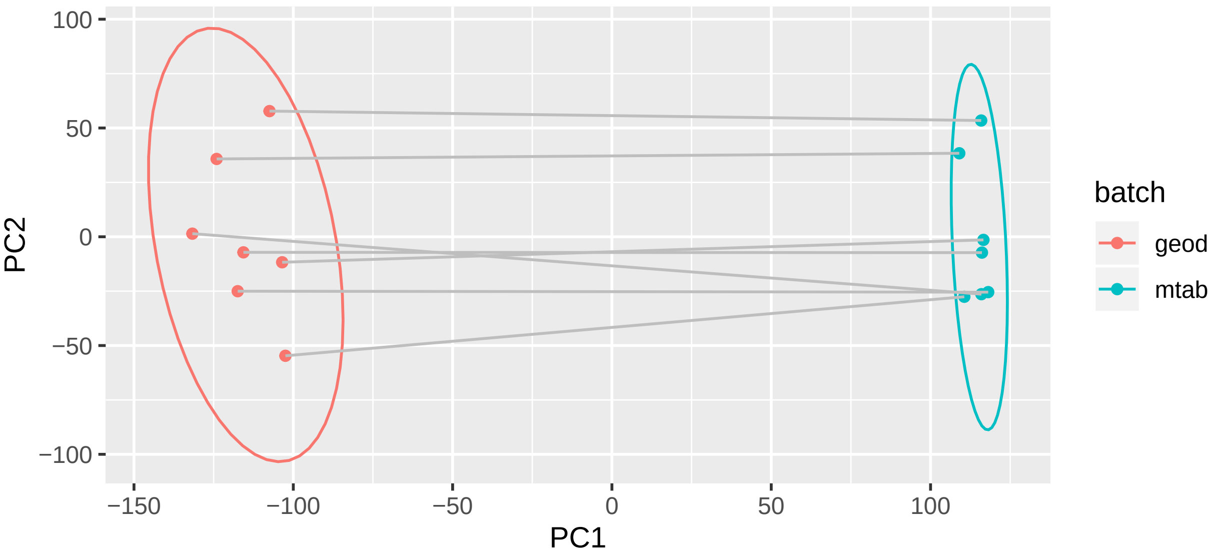

Figure 2: First plan of variance of the merged dataset for PCA

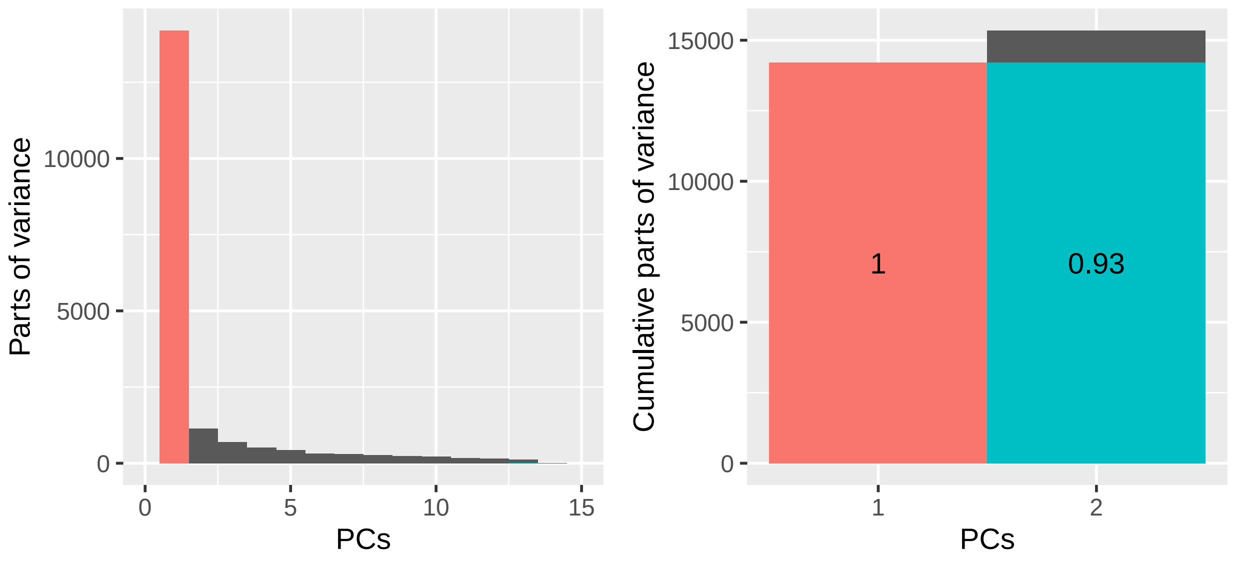

Figure 3: Visualisation of the parts and ranks of variance summarising the importance of batch effect within the merged dataset

We can also visualise the cumulative parts of variance and cumulative δ statistics with their variance ranks shown on the variance profile of the dataset (figure 3). The plot on the left shows the variances of the successive principal components in grey and the variances of the successive guided principal components in colours. The bars for guided principal components are positioned on x-axis according to their variance ranks.

```
gfiltered %>% viz_gpca_contrib
```

A new gene-based detection method

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.

Mathematical consideration for the calculation of angles

To calculate an angle between two vectors $\vec{u}$ and $\vec{v}$ in a space of any dimension, the most commonly used definition is :

$$\widehat{(\vec{u},\vec{v})} = \arccos \frac{\langle \vec{u},\vec{v} \rangle}{\|\vec{u}\|\|\vec{v}\|}$$

where $\langle \vec{u}, \vec{v} \rangle$ denotes the euclidian inner product between vectors $\vec{u}$ and $\vec{v}$, and $\|.\|$ denotes the euclidian norm.

To calculate angles between the first principal component of each dataset and the integrated one, this definition can be used easily as $PC1$ are 1-dimensional direction. $\vec{u}$ shall be chosen as an orientation vector of $PC1$ of the considered individual dataset and $\vec{v}$ as an orientation vector of $PC1$ of the integrated dataset. In R, orientation vectors of principal components are given by the columns of the `$rotation` element in the output of a call to `prcomp` function.

Although this definition doesn't allow to extrapolate this idea to principal components with higher ranks. Indeed, estimating angles between principal components of higher rank doesn't make sense contrarily to angles between $PC1$, as only $PC1$ maximises the variance of the dataset projected on a 1-dimensional axis. Principal components of higher rank don't have such properties by themselves. However the plane generated by $PC1$ and $PC2$ maximises the variance of the dataset projected on a 2-dimensional subspace, in just the same way as the n-dimensional subspace generated by $PC1, ..., PCn$ is such that the variance of the dataset projected on such a subspace is maximised.

Therefore the good generalisation of this idea is to compute the angle between the n-dimensional subspaces $\text{span}(PC_1^i, ..., PC_n^i)$ and $\text{span}(\dot{PC}_1, ..., \dot{PC}_n)$. This requires to be able to calculate angles between subspaces, whereas the previous definition only gives a way to calculate angles between vectors and therefore only between 1-dimensional subspaces.

Thus we give the following definition for the angle between two subspaces $U = \text{span}(\vec{u}_1, ..., \vec{u}_n)$ and $V = \text{span}(\vec{v}_1, ..., \vec{v}_m)$, parts of a space of dimension $p = m + n$ and where $(\vec{u}_1, ..., \vec{u}_n)$ and $(\vec{v}_1, ..., \vec{v}_m)$ are orthonormal bases of those subspaces respectively :

$$\widehat{(U,V)} = \arcsin \det(\vec{u}_1, ..., \vec{u}_n, \vec{v}_1, ..., \vec{v}_m) = \arcsin \begin{vmatrix} u_1^1 & . & . & . & u_n^1 & v_1^1 & . & . & . & v_m^1 \\ . & . & & & . & . & . & & & . \\ . & & . & & . & . & & . & & . \\ . & & & . & . & . & & & . & . \\ u_1^p & . & . & . & u_n^p & v_1^p & . & . & . & v_m^p \end{vmatrix}$$

where the coordinates of the vectors are given in an orthonormal basis of the space. If one disposes of non-orthonormal bases for U and V, one can use any orthogonalisation process, such as any QR-factorisation method, in order to apply the previous formula legitimately.

Here there is a constraint on the dimension of data, given above by $p = m + n$, due to application of determinant operator, only defined for a square matrix. In our problem, the dimension p of data is the number of genes considered. Although, still in our problem, one wants to calculate angles between two n-dimensional subspaces for any value of n, so that the condition $p = m + n = 2n$ shall not be satisfied in general.

However, this is not a real issue as the $2n$ base vectors $\vec{u}_1, ..., \vec{u}_n, \vec{v}_1, ..., \vec{v}_n$ are themselves situated in a $2n$-dimensional subspace, where determinant can be applied as well as in the original p-dimensional space. The issue is then to rewrite the problem in this particular subspace i.e. $U + V = \text{span}(U \cup V)$, using an orthonormal basis of this subspace. One can easily find such a basis by performing a QR-factorisation to the family of vectors $(\vec{u}_1, ..., \vec{u}_n, \vec{v}_1, ..., \vec{v}_n)$ (ordered as columns in a matrix) where the R matrix contains the coordinates of the original vectors in this new basis.

Thus we adopt the following framework to calculate angles between the subspaces $\text{span}(PC_1^i, ..., PC_n^i)$ and $\text{span}(\dot{PC}_1, ..., \dot{PC}_n)$ for any rank n and any batch i :

- Apply QR-factorisation to the family of p-dimensional vectors $(PC_1^i, ..., PC_n^i, \dot{PC}_1, ..., \dot{PC}_n)$ to find their coordinates (given in the R matrix) in an orthonormal basis of their $2n$-dimensional subspace. Thus we get $2n$ new vectors of coordinates (although they represent geometrically the same vectors) whose dimension is also $2n$, that is to say a square matrix to which determinant is applicable. Let's denote $(\widetilde{PC}_1^i, ..., \widetilde{PC}_n^i, \widetilde{\dot{PC}}_1, ..., \widetilde{\dot{PC}}_n)$ these new vectors of coordinates.

- Apply QR-factorisation to the family $(\widetilde{PC}_1^i, ..., \widetilde{PC}_n^i)$ to get an orthonormal basis of its span (given in the Q matrix) : $(\widetilde{PC}_1^{i\perp}, ..., \widetilde{PC}_n^{i\perp})$. Do the same with the family $(\dot{P}C_1, ..., \dot{P}C_n)$ to obtain an orthonormal basis of its span : $(\widetilde{\dot{P}C}_1^{\perp}, ..., \widetilde{\dot{P}C}_n^{\perp})$
- Hence the angle between the subspaces $\text{span}(PC_1^i, ..., PC_n^i)$ and $\text{span}(\dot{P}C_1, ..., \dot{P}C_n)$ is given by :

$$\alpha_n^i = \arcsin \det(\widetilde{PC}_1^{i\perp}, ..., \widetilde{PC}_n^{i\perp}, \widetilde{\dot{P}C}_1^{\perp}, ..., \widetilde{\dot{P}C}_n^{\perp})$$

Discussion on the scaling step before PCA