A geometrical approach based on PCA to benchmark the algorithms of batch effect correction applied to the integration of RNA-Seq data

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

Abstract

Batch effect is a bias phenomenon appearing during the integration of sequencing data coming from different sources, due to experimental artifacts but also because of environmental factors influencing the biological samples. Many different algorithms exist to correct this bias, which are based on statistical considerations. As these algorithms follow different approaches, and as some of them depend on a tunable parameter, a quantitative comparison of their results is required. I introduce here both a qualitative classification and a quantitative benchmarking of some correction algorithms, following myself three different approaches. I reproduced the two existing approaches of guided PCA and entropy of batch mixing. For some reasons that make these two approaches not fully adapted to the case of heterogeneous datasets, I introduce as well a new geometrical approach, which I called Eigengenes Angles, based on Principal Component Analysis (PCA). This new approach allows to make a trade-off between the conservation of the original datasets and their good integration in their merger, in terms of biological information.

Contents

[*]Engineering Student in Applied Mathematics, Statistics and Data Science at École Centrale de Nantes, France.
[†]Expression Atlas Data Production Project Leader in the Gene Expression Team, EMBL-EBI, Cambridgeshire, UK.
[‡]Leader of the Gene Expression Team, EMBL-EBI, Cambridgeshire, UK.

1 Introduction and vocabulary

Batch effect correction is the fundamental step of the integration of RNA-Seq experiments, which is itself an essential part of the attempt of increasing knowledge and power of comparison between several tissues, cells or biological conditions.

The data to gather are RNA counts in pieces of tissue or single cells. One can distinguish bulk data and single cell data, depending on whether measures are made on pieces of tissue or single cells respectively. One can also distinguish baseline data and differential data, depending on whether the studied biological covariate is a standard property of tissues or cells for healthy organisms (organism part, strain, ...) or a illness or effect of a mutation. The work I introduce in this article deals with bulk baseline RNA-Seq, although it can be easily reproduced on any type of quantitative data dealing with genetic expression.

The following vocabulary will be used in this article :

- A *RNA-Seq dataset* refers to any collection of data coming from any RNA sequencing technique, containing a count matrix whose items represent the measured expression levels (RNA counts) of the genes in the different samples. The rows of this matrix stand for genes and its columns stand for samples. A dataset can gather samples from a sole source as well as samples from various sources. Thus, to avoid any confusion, the word *dataset* shall not be used and shall be replaced by the words *batch* and *merger*.
- A *batch* is a dataset coming from a sole source, i.e. a dataset whose samples are considered to share a common bias of measurement. Although there is always a bias between two measurements, especially in Biology, samples from the same batch shall be considered to share an intern coherence (same experimentator, same device, same experimental subject, ...), so that this bias does not fall under the scope of batch effect. In this article, the individual experiments datasets available on Expression Atlas [5] will be considered as batches.
- A *merger* is a dataset gathering the samples of several batches. One shall name it an *uncorrected merger* or a *corrected merger* depending on the use of an algorithm of batch effect correction.
- An *algorithm of batch effect correction* (which I will simply name an *algorithm* in the following) is a transformation based on statistical condsiderations applied on the data of an uncorrected merger in order to normalise the bias between its batches. Strictly speaking, such an algorithm is supervised by the prior knowledge of the batches. However, some algorithms do not take any batch argument into account, such as the algorithms of the RUV family. Thus, a distinction will be done between *supervised* and *unsupervised algorithms*.
- A very important distinction between the existing algorithms is whether they are supervised by the knowledge of the biological properties of the samples. I introduce the notion of *wisdom* of an algorithm. A *wise algorithm* is an algorithm that can take into account an argument sumarrising biological groups (and that should be used this way), whereas an *unwise algorithm* does not take this into account and correct the batch effect as if the batches followed the same distribution, which is most of time unfair.

- A *parametric algorithm* is an algorithm that takes a numerical argument into account. The choice of this argument by the user is not so easy, as there is no clear indications about it, in general. A *non-parametric algorithm* is an algorithm without such arguments.
- Algorithms of batch effect correction shall not be confused with the different *benchmarking approaches* that I suggest in this paper. A *benchmarking approach* (which I will simply call an *approach*) is a statistical method applied on the corrected mergers as well as the uncorrected one, in order to compare their performances in batch effect correction. A good approach should be applicable to all the benchmarked algorithms.

2 Preliminary considerations

2.1 General workflow for the integration of datasets[1]

Prior to using any algorithm for batch effect correction, a general workflow must be applied to merge roughly the batches to integrate. In this article, the following preliminary framework was followed :

2.1.1 Check of the biological intersections of the batches

In parallel with their batch split, the samples of the uncorrected merger are also split into biological groups, associated to one or several biological covariates described in the experimental datasets. This biological covariate can be organism part, strain, the presence of a disease, a treatment...

The presence of a biological group within a batch, as well as its proportion, makes the distribution of this batch generally different to the distributions of the other batches. This property is at the origin of the notion of wisdom of correction algorithms, that I introduced above.

Whereas these differences of distribution have to be taken into account during batch effect correction, the bias within a set of batches can be corrected only if there is an intersection of biological groups between them. Otherwise the batch covariate is completely confounded with the biological covariate, and correction is not possible. More precisely, the graph of biological intersections must be a connected graph in order to perform batch effect correction, i.e. every two batches must be linked (directly or through intermediary batches) within the graph (cf. figure 1).

If some experiments are isolated from the graph, they have to be removed. If several separated connected subgraphs appear in the graph, one has to choose a subset of batches corresponding to a connected subgraph.

2.1.2 Rough integration and normalisation

First, the batches are merged roughly. Naturally, the batches have to deal with the same genes. Some genes shall possibly be removed if they are not shared across all the batches.

Also genes with 0 counts across a whole batch shall be removed since this is apparently disturbing the implementation of ComBat.

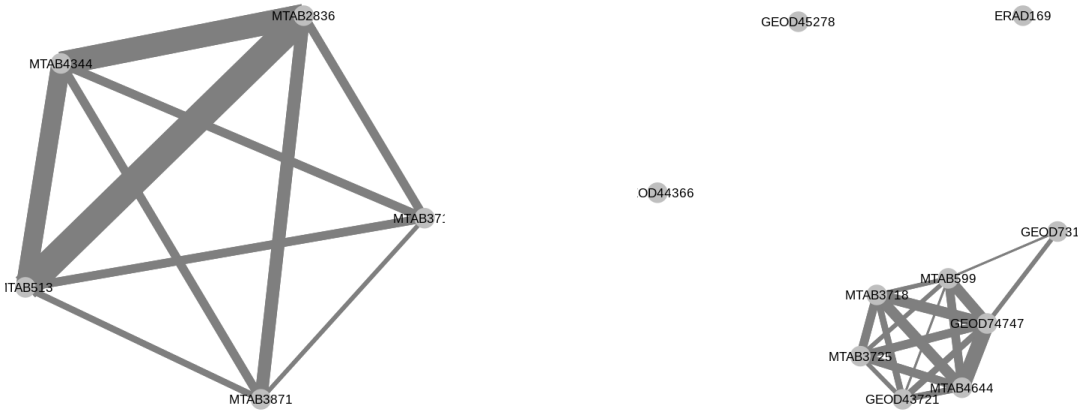
Finally the count data shall be log-transformed (traditionally after the addition of 1 in order to avoid the singularity in 0).

The data were processed this way for the benchmarking described hereafter.

[1] A detailed user guide is available on my GitHub account : https://github.com/gheager/Pipeline-for-batch-effect-correction-in-Baseline-experiments

Table 1: Primary classification of some algorithms of batch effect correction.

	Batch Supervision	Wisdom	Autonomy
Batch-mean centering	supervised	unwise	non-parametric
Ratio-based methods	supervised	unwise	non-parametric
RUVr	unsupervised	unwise	parametric
RUVg	unsupervised	unwise	parametric
RUVs	unsupervised	wise	parametric
MNN	supervised	unwise	parametric
Empirical Bayes framework (ComBat)	supervised	wise	non-parametric



(a) For Human experiments.

(b) For Mouse experiments.

Figure 1: Graphs of intersections. The thickness of an edge represents the number of organism parts in common between the two experiments it links. For Human datasets, all the experiments are linked together in a connected graph, whereas for Mouse experiments, three experiments have to be removed to be able to correct batch effect.

2.2 Classification of the algorithms

Primarily, I propose the classification given in table 1 for the algorithms of batch effect correction (the list is not exhaustive). This classification is done according to three criteria, which I defined in the introduction.

The batch supervision criterion is important : most algorithms are actually supervised and actually, the term of batch effect correction should be dedicated to supervised algorithms. However, although the three algorithms proposed in the package RUVSeq [7] are unsupervised, I have inserted them in this classification, as they are broadly used.

The wisdom criterion is particularly important in the case of heterogeneous batches, such as baseline datasets. In a general way, wise algorithms should be prefered to unwise ones, which doesn't take into account the biological information available about the samples, and may lead to confusion.

Finally, the criterion of autonomy deals with the presence of tunable parameters in the algorithm. Non-parametric algorithms provides the user with a slight advantage since their usage doesn't need an optimisation of parameters by the latter. However, this advantage is not significative and shall not be decisive in a benchmarking. The algorithms of the RUV family have a parameter k denoting the number of components of unwanted variation to remove [7], whereas MNN has a parameter also denoted k, which is the number of nearest neighbours to consider for each sample [1].

In this article, I will show the application of my benchmarking approaches to the following algorithms :

- BMC (Batch-Mean Centering) implemented in the package pamr [2];
- RUVs (Remove Unwanted Variation using replicate/negative control sample) from the package RUVSeq [7];
- MNN (Mutual Nearest Neighbours) from the package batchelor [1];
- ComBat (Empirical Bayes framework) from the package sva [3].

3 Approaches for the benchmarking of the correction algorithms

I introduce hereafter three tools that I used to benchmark the correction algorithms.

- Guided PCA was originally developed for batch effect detection and can be used for the benchmarking of correction algorithms [6]. I wrote a new implementation of this tool in R which I completed with an multidimensiona extrapolation of the δ statistic for the principal components of rank higher than 1.
- Entropy of batch mixing comes from the exchange that I had during my internship with Ruben Chazarra[2] who was working on a benchmarking of batch effect correction methods within the context of Single Cell experiments. This approach has already been used for the development of some correction algorithms, such as MNN[1] and BBkNN[4].
- 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 batches, that may lead to confuse batch effect and biological differences between the batches. I implemented this tool in a R package called `eigenangles`[3] as well as visualisation tools to benchmark the algorithms.

3.1 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 in [6]) 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.

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 δ statistic is defined [6] 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 $\delta \leq 1$.

[2]Visiting Scientist at Wellcome Sanger Institute
[3]Released on my GitHub account : https://github.com/gheager/eigenangles

I introduce an extrapolation of this statistic to higher dimensions by considering for $1 \leq i \leq B$ by considering:

$$\delta_i := \frac{\mathbb{V}(gPC_1, ..., gPC_i)}{\mathbb{V}(PC_1, ..., PC_i)}$$

where gPC_i (resp. PC_i) denotes the projection of data on the i^{th} axis of gPCA (resp. PCA) and $(gPC_1, ..., gPC_i)$ denotes the vector formed by the first i guided principal components, which is actually the projection of data on the i-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 :

$$\delta_i = \frac{\sum_{j=1}^{i} \mathbb{V}gPC_j}{\sum_{j=1}^{i} \mathbb{V}PC_j} = \frac{\mathbb{V}gPC_1 + ... + \mathbb{V}gPC_i}{\mathbb{V}PC_1 + ... + \mathbb{V}PC_i}$$

3.2 Entropy of batch mixing

For a dataset of point $(x_i)_{1 \leq i \leq n}$ and for $1 \leq k \leq n$, the regional entropy of batch mixing using k nearest neighbours is defined [1] for each point x_i as :

$$E_k(x_i) = -\sum_{b=1}^{B} p_b^k \log(p_b^k)$$

where B is the number of batches and for $1 \leq b \leq B$, p_b^k is the proportion of points from batch b among the k nearest neighbours of x_i according to a specified metric (we will choose euclidean distance here).

The total mixing entropy is then given by :

$$E_k = \sum_{i=1}^{n} E_k(x_i)$$

Since this entropy of batch mixing depends of a chosen parameter k, one can compute it for every value of k and plot the discrete curve of $(E_k)_{1 \leq k \leq n}$.

As the experimental batches are not supposed to share any biological information, the entropy of batch mixing of a corrected merger should be as high as possible, showing that the batches are well mixed.

3.3 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 batches compared to those of their merger (corrected or uncorrected). As principal components represents geometrical directions, a way to compare them is to estimate the angle between them.

In the simple case where batches are supposed to have the same distribution, we actually expect them to have their principal components similar to those of the merged dataset. Out of this case, one can consider the biological intersection of the batch and the merger. This is to say that each batch should contain the same biological information than the global dataset on the biological domain carried by this batch.

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 given on the genes by a dataset. Thus low angles between the respective principal components of each dataset and the ones of their merger mean somehow that the information provided by the single datasets has been conserved through their integration.

Moreover, since this idea is based on a metric comparing datasets, it can be reused to quantify the transformation performed by a correction on a batch. Thus it allows us to make a trade-off between the good integration of batches within the merger and the conservation of these batches through the correction.

3.3.1 Precise framework for the calculation of Eigengenes Angles

The following steps are followed by my implementation of the Eigengenes Angles method :

For each batch b :

- Reduce the merger (including the considered batch) by keeping only the samples which have biological replicates in the considered batch (this is what I called above the biological intersection of the batch and the merger);
- Within the considered batch b, average each group of biological replicates so that there remain as many points as biological groups in b;
- Within the merger (including the considered batch), average the biological replicates in the way described above (including those of the considered batch);
- Compute PCA for the considered averaged batch and for the averaged merger : thus we get two lists of principal components of the same size k_b (this size is the number of biological groups within the considered batch);
- For each rank $1 \leq k \leq k_b$, calculate the angle between the principal subspaces of rank k from the two lists. For rank 1, this simply corresponds to the angle between the first principal component of the first PCA and the first principal component of the second PCA. For higher ranks, the calculation is explained thoroughly below.

Hence we get for each batch b, a list of angles $(\alpha_k^b)_{1 \leq k \leq k_b}$ between 0 and $\frac{\pi}{2}$. This B lists have different sizes k_b that are the numbers of biological groups within the respective batches.

In order to summarise this information across the batches, I suggest the following quasi-arithmetic tangent-mean angle :

$$\mathrm{A}_k = \arctan\left(\frac{1}{B}\sum_{b=1}^{B} \tan \alpha_k^b\right)$$

This quasi-arithmetic mean has two advantages compared to a simple arithmetic mean of the angles across the batches :

- the arithmetic mean of a list of angles has no geometrical meaning, whereas the mean of their tangents is slightly more meaningful, since they represent geometrical distances (although one could choose to compute their directional mean : $\arctan\left(\frac{\frac{1}{B}\sum_{b=1}^{B} \sin \alpha_k^b}{\frac{1}{B}\sum_{b=1}^{B} \cos \alpha_k^b}\right)$);
- the other advantage is that the use of the tangent function penalises more the high angles, i.e. the badly integrated batches, since $\lim_{x \to \frac{\pi}{2}} \tan x = +\infty$.

3.3.2 A trade-off between batch integration and batch conservation

The way to quantify the conservation of batches by the algorithms is very similar to the process described above. However, contrarily to it and to the previous approaches, it considers the uncorrected data as well as the corrected ones.

The following steps are followed for each batch :

- Within the considered batch taken from the corrected merger, average the biological groups;
- Within the considered batch taken from the uncorrected merger, average the biological groups;

- Compute PCA for those two averaged batches : one gets two lists of principal components;
- For each rank $1 \leq k \leq k_b$, calculate the angle between the principal subspaces of rank k from the two lists.

Hence we get for each batch b, a list of angles $(\beta_k^b)_{1 \leq k \leq k_b}$ between 0 and $\frac{\pi}{2}$. This B lists have different sizes k_b that are the numbers of biological groups within the respective batches.

One can again summarise these angles across the batches using a quasi-arithmetic tangent-mean :

$$\mathrm{B}_k = \arctan\left(\frac{1}{B}\sum_{b=1}^{B} \tan \beta_k^b\right)$$

3.3.3 Mathematical consideration for the calculation of angles in higher dimension

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 euclidean inner product between vectors $\vec{u}$ and $\vec{v}$, and $\|.\|$ denotes the euclidean 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.

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 the calculation of angle between the n-dimensional subspaces $\mathrm{span}(PC_1^b, ..., PC_n^b)$ and $\mathrm{span}(\dot{PC}_1, ..., \dot{PC}_n)$. 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 $U = \mathrm{span}(\vec{u}_1, ..., \vec{u}_n)$ and $V = \mathrm{span}(\vec{v}_1, ..., \vec{v}_m)$, 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 $\text{span}(PC_1^b, ..., PC_n^b)$ 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 $(PC_1^b, ..., PC_n^b, \dot{PC}_1, ..., \dot{PC}_n)$ 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 $(\widetilde{PC}_1^b, ..., \widetilde{PC}_n^b, \widetilde{\dot{PC}}_1, ..., \widetilde{\dot{PC}}_n)$ these new vectors of coordinates.
- Apply QR-factorisation to the family $(\widetilde{PC}_1^b, ..., \widetilde{PC}_n^b)$ 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 $(\dot{PC}_1, ..., \dot{PC}_n)$ to obtain an orthonormal basis of its span : $(\widetilde{\dot{PC}}_1^{\perp}, ..., \widetilde{\dot{PC}}_n^{\perp})$
- Hence the angle between the subspaces $\text{span}(PC_1^b, ..., PC_n^b)$ and $\text{span}(\dot{PC}_1, ..., \dot{PC}_n)$ is given by :

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

3.3.4 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 batches, i.e. batches whose samples have unshared biological characteristic (for example, a batch with samples from brain and lung and another one whose samples come from lung and liver) or whose proportions of biological groups are different. In such a situation, the batches cannot be supposed to follow the same distribution and both gPCA and entropy of batch mixing indices may fail to catch the batch effect with precision, since the batch effect is partially confused with the biological covariates. Indeed :

- Guided PCA may detect important batch effect in the uncorrected merger only because its batches have different biological compositions. Also it could lead to evaluate unfairly well an unwise algorithm correcting batch effect while ignoring this bias.
- In the same way, entropy of batch 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.

In the case of Eigengenes Angles, it is possible to avoid this confounding factor since the angle is calculated on batches considered individually. Thus it is possible to subset the merger for every comparison by keeping only its biological intersection with the compared batch. After this subsetting step, all the considered samples have at least a replicate in the considered batch. However, there is still no warranty that their quantitative compositions are the same. To counter this issue, a solution is to aggregate the replicates (by averaging them) within the considered batch and within the merger. An extra advantage of this way to do is that it removes the component of variance that is due to replicating experiments. Only the variance between the different biological groups is conserved.

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

4.1 The data

The batches used for this benchmarking all comes from Expression Atlas [5], internet resource provided by EMBL-EBI Gene Expression Team.

For *Homo sapiens*, 5 experiments have been used and all of them were conserved for integration. Their references are : E-MTAB-513, E-MTAB-2836, E-MTAB-3716, E-MTAB-3871, E-MTAB-4344.

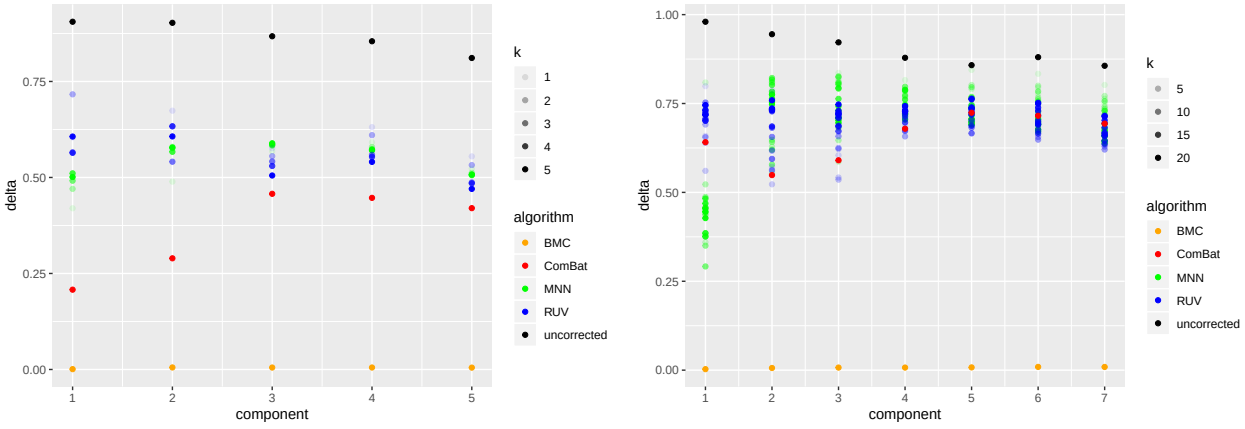
For *Mus musculus*, 10 experiments were used but only 7 of them were kept due to the filtering step considering biological inersections of the batches (cf. figure 1). Their references are : E-GEOD-43721, E-GEOD-73175, E-GEOD-74747, E-MTAB-599, E-MTAB-3718, E-MTAB-3725, E-MTAB-4644. The non-selected ones are E-ERAD-169, E-GEOD-44366, E-GEOD-45278.

4.2 Benchmarking using guided PCA

The figure 2 shows the multidimensional application of guided PCA on the uncorrected and corrected mergers. For the parametric algorithms, several levels of transparency show the different values of their parameters, so that we can see a general tendency of an algorithm. The tool of guided PCA seems to be sensitive to correction since all the delta statistics are always maximised by the uncorrected merger.

One must consider of BMC algorithm that have zero values for all its δ statistics. This is explained simply by the fact that guided PCA aspires to catch batch effect by averaging the batches between them and by evaluating the parts of variance of the principal components of the averaged dataset so obtained. However, the principal components of this averaged dataset are actually not defined, since BMC translates the means of every batches to the zero point. So the averaged dataset considered in guided PCA consists in a unique point. Thus, guided PCA is not applicable to BMC algorithm.

Knowing this, ComBat seems to win on every guided principal component for the Human dataset. For both datasets, MNN seems to minimise efficiently the δ_1 statistic compared to RUV, while the opposite becomes true on higher rank components.



(a) For Human datasets (k=1,...,5).

(b) For Mouse datasets (k=1,...,20).

Figure 2: Delta statistics from gPCA in every dimension.

4.3 Benchmarking using entropy of batch mixing

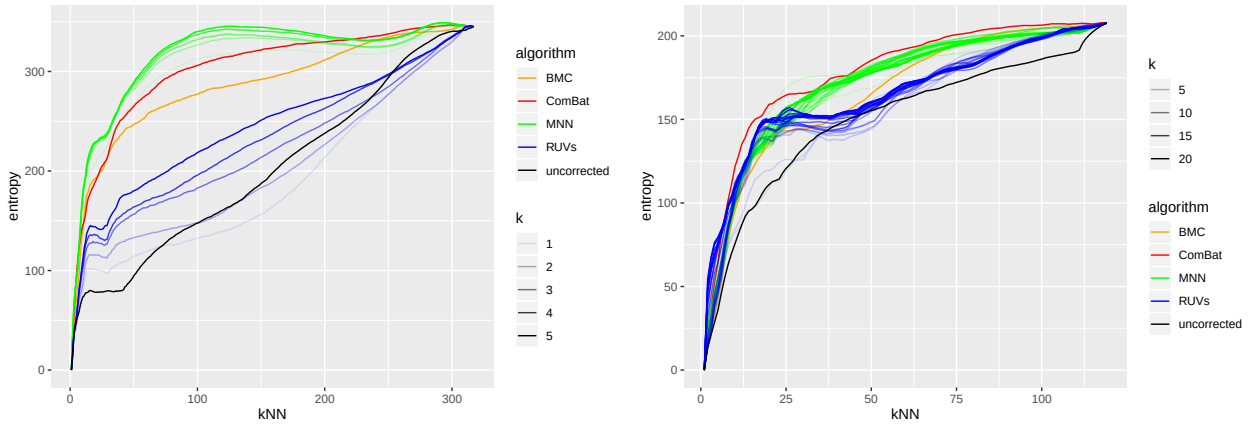
The entropy of batch mixing is a parametric estimator : the number k of nearest neighbours considered to calculate regional entropy of mixing must be chosen quite arbitrarily.

I plot below the discrete curve of entropies of batch mixing for every applicable value of k (from 1 to the total number n of samples).

These plots allow to benchmark the different algorithms of correction, by seeing which one seems to maximise the entropy in general. One shall look at intermediary and high values of kNN, since small values are much affected by the bias created by the heterogeneity of the batches.

For the Human dataset, MNN algorithm seems to win in front of the other algorithms, whatever the value of its k parameter.

For the Mouse dataset, ComBat and MNN seem to do better than the other algorithms. Moreover, ComBat seems to win against most values of k for MNN.



(a) For Human datasets (k=1,...,5).

(b) For Mouse datasets (k=1,...,20).

Figure 3: Entropy of batch mixing for every number of nearest neighbours (from 1 to the total number of samples).

4.4 Benchmarking using Eigengenes Angles

I show here Eigengenes Angles, before and after the quasi-arithmetic tangent-mean summary of angles across batches.

The non-averaged angles shown on figure 4 has several facets representing the different batches and might be used to spot the batches that are badly integrated into the corrected mergers. It can be used as a dashboard to select the datasets to choose for integration and possibly remove some of them. Each batch is represented by a facet of the plot, and each facet is showing two angles for the uncorrected merger and each corrected merger. The angles for each merger are represented by a line or a point, depending on whether the used algorithm is parametric. Either way, the angle is drawn between it and the top vertical black line, in an obvious way for lines, or for points, with a non-drawn line from the center of the chart to the considered point. This is for readability of the chart. On the left of the chart are the angles of batch integration, whereas on the right are the angles of batch conservation.

The averaged angles shown on figures 5 and 6 allow to benchmark the correction algorithms more easily. On the left of the chart are the average angles of batch integration, whereas on the right are the average angles of batch conservation. A good correction should keep these two quantities quite low, that is to say its two angles should be tight around the top vertical axis. The angle of batch conservation of the uncorrected

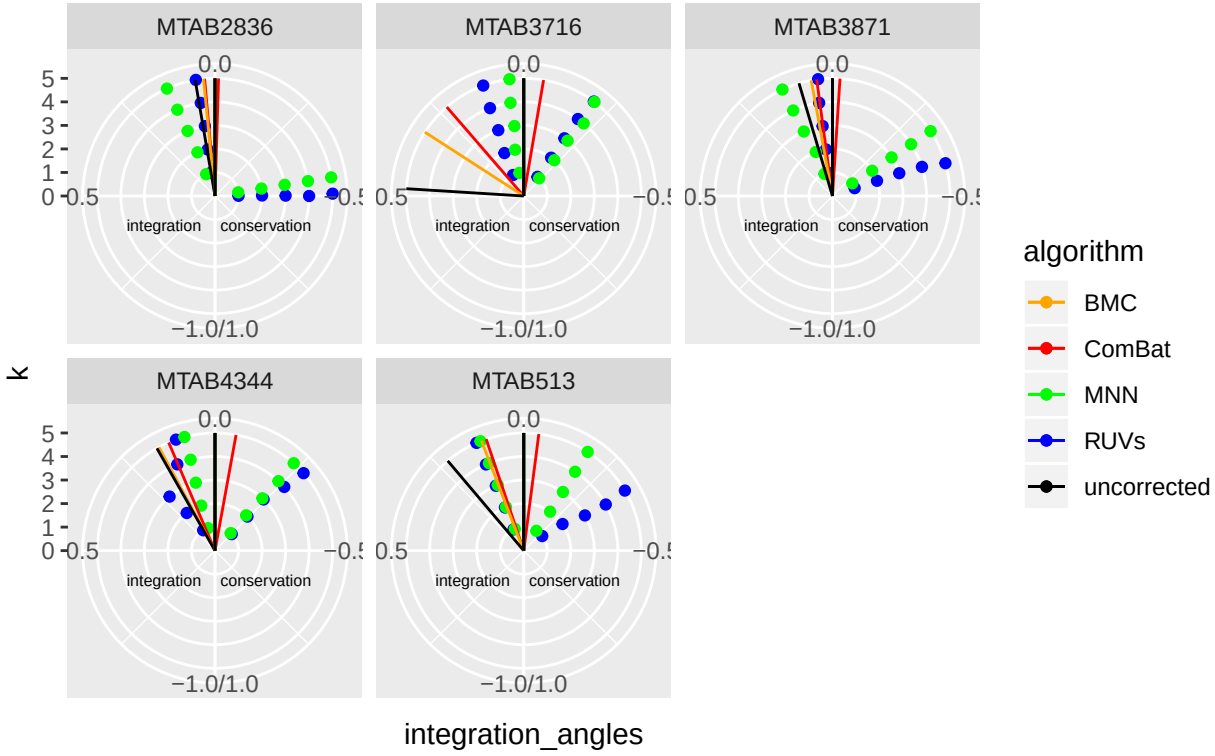
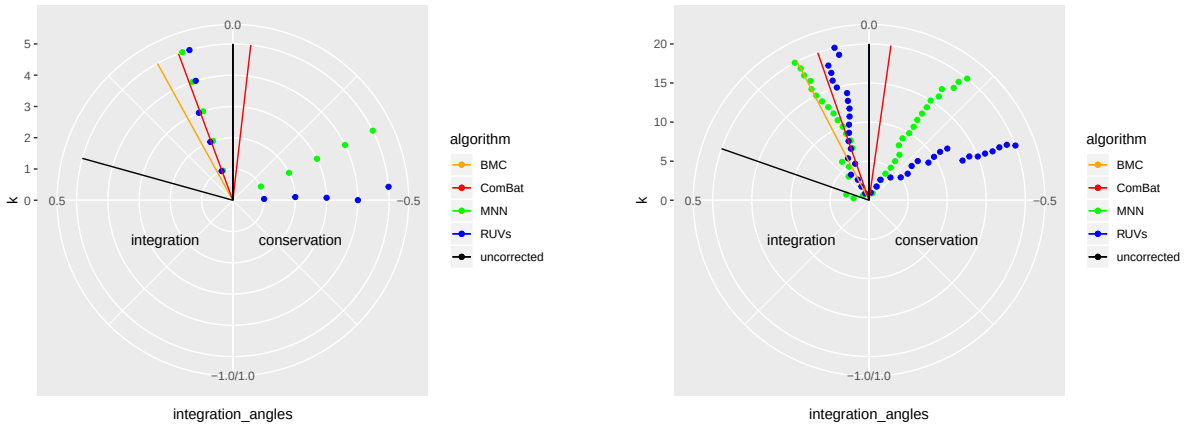


Figure 4: Eigengenes Angles dashboard (angles of rank 1) for the integration of the Human datasets.

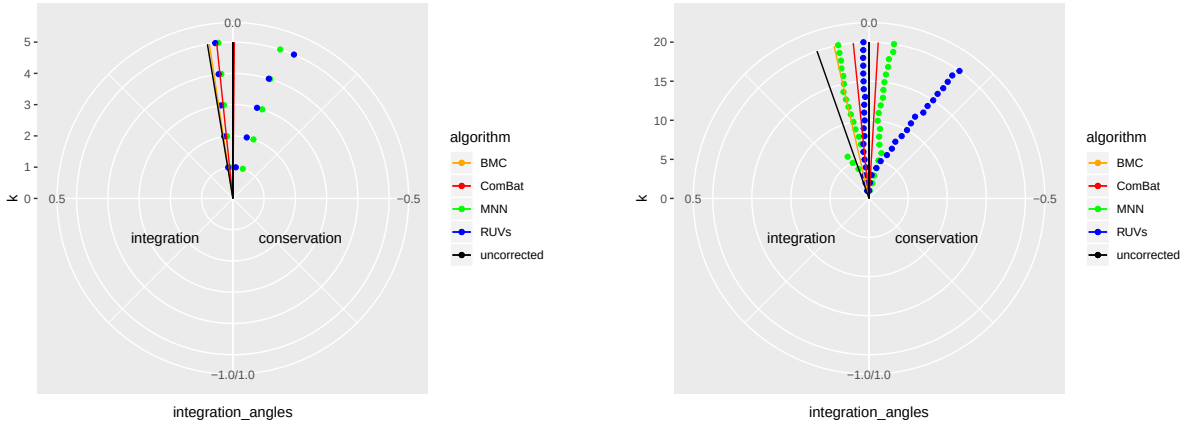
merger is obviously zero, as well as the one of the BMC-corrected merger. Indeed, BMC performs a uniform translation on the data of each batch, which is not detected by the Eigengenes Angles method, due to the centering step in PCA.

By compromising between integration and conservation angles, ComBat seems to be better than the other algorithms. Actually, angles of rank 1 give a similar angle gap to BMC. However, a look at the angles of rank 2 allows to establish the victory of ComBat for both Human and Mouse datasets.



(a) For Human datasets (k=1,...,5). (b) For Mouse datasets (k=1,...,20).

Figure 5: Eigengenes Angles summarised across batches using the quasi-arithmetic tangent-mean (angles of rank 1).



(a) For Human datasets (k=1,...,5). (b) For Mouse datasets (k=1,...,20).

Figure 6: Summarised Eigengenes Angles of rank 2.

5 Conclusion

The three different approaches used in this article to benchmark the algorithms of batch effect correction show slightly different results about the correction methods. First, one can notice that these methods always make the benchmarking dependent on the integrated datasets, which makes the necessity of reproducible benchmarking methods. However, these benchmarkings seem to agree on a good evaluation to the ComBat implementation of Empirical Bayes framework.

Whereas the existing methods of guided PCA and entropy of batch mixing aspire to evaluate the importance of the experimental factor within the data, the new approach of Eigengenes Angles makes a paradigm shift as it aspires to evaluate the good integration of corrected individual experiments within a bigger dataset incorporating all of them. Its individual treatment of the batches allow a visual analysis of them, useful for data selection. Moreover, it is making silmutaneously a trade-off between this notion of integration and the conservation of the original experiments through the correction.

Although the mathematical realisation of this approach could be perfected, through the choice of other metrics to compare linear subspaces (other definition of angles, or distances instead of angles), or also through a technic to summarise the multidimensionality of its results, this type of approach can be used as an efficient visual tool to check the integration and conservation of RNA-Seq datasets, as well as more general type of quantitative data.

This approach must be seen in a more general frame that actually handles the batches as the pieces of a patchwork. The issues of batch integration and batch conservation can then be summarised in the following question : can one arrange these pieces harmoniously within the patchwork without cutting them too much?

Acknowledgements I gratefully thank the Embassy of France in London for the financial support it provided me during the four months of my internship. I want to thank, in particular, Yeliz Demirci, for her precious support during this term and for the pleasant way she gave me the taste for genetics.

References

[1] Laleh Haghverdi et al. "Batch effects in single-cell RNA-sequencing data are corrected by matching mutual nearest neighbors". In: *Nat. Biotechnol.* 36.5 (2018), pp. 421–427. DOI: 10.1038/nbt.4091.

[2] T. Hastie et al. *pamr: Pam: Prediction Analysis for Microarrays*. R package version 1.56.1. 2019. URL: https://CRAN.R-project.org/package=pamr.

[3] Jeffrey T. Leek et al. *sva: Surrogate Variable Analysis*. R package version 3.32.1. 2019.

[4] Jong-Eun Park et al. "Fast Batch Alignment of Single Cell Transcriptomes Unifies Multiple Mouse Cell Atlases into an Integrated Landscape". In: *bioRxiv* (2018). DOI: 10.1101/397042. eprint: https://www.biorxiv.org/content/early/2018/08/22/397042.full.pdf. URL: https://www.biorxiv.org/content/early/2018/08/22/397042.

[5] Robert Petryszak et al. "Expression Atlas update—an integrated database of gene and protein expression in humans, animals and plants". In: *Nucleic Acids Research* 44.D1 (Oct. 2015), pp. D746–D752. ISSN: 0305-1048. DOI: 10.1093/nar/gkv1045. eprint: http://oup.prod.sis.lan/nar/article-pdf/44/D1/D746/16941355/gkv1045.pdf. URL: https://doi.org/10.1093/nar/gkv1045.

[6] Sarah E. Reese et al. "A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis". In: *Bioinformatics* 29.22 (Aug. 2013), pp. 2877–2883. ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btt480. eprint: http://oup.prod.sis.lan/bioinformatics/article-pdf/29/22/2877/17099626/btt480.pdf. URL: https://doi.org/10.1093/bioinformatics/btt480.

[7] Davide Risso et al. "Normalization of RNA-seq data using factor analysis of control genes or samples". In: *Nature Biotechnology* 32.9 (2014). In press, pp. 896–902. URL: http://www.nature.com/nbt/journal/v32/n9/full/nbt.2931.html.