Bioinformatics
doi.10.1093/bioinformatics/xxxxxx
Advance Access Publication Date: Day Month Year
Manuscript Category

Gene expression

Comparison of batch effect removal methods in the presence of correlation between outcome and batch

Emilie Renard [1,*] and P.-A. Absil [1]

[1] ICTEAM Institute, Université catholique de Louvain, Louvain-la-Neuve, 1348, Belgium

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Motivation: Merging gene expression datasets is a simple way to increase the number of samples in an analysis. Experimental and data processing conditions, which are proper to each dataset or batch, generally influence the expression values and can hide the biological effect of interest. It is then important to normalize the bigger merged dataset, as failing to adjust for those batch effects may adversely impact statistical inference. However, over-adjusting can be more damaging than not normalizing at all, especially in the context of prediction tasks where the phenotype to predict is unequally distributed among the batches.
Results: We compare the two main types of batch effects removal approaches: location-scale (ComBat and centering-scaling) and matrix factorization (based on SVD or ICA) normalization methods. We investigate on breast cancer data how those normalization methods improve (or possibly degrade) the performance of a simple classifier in the presence of two kinds of difficulties: (i) strong batch effects due to differences in summarization methods, (ii) strong correlation between outcome and batch. Our results indicate that if the best method varies depending on the two difficulties, a method based on Independent Component Analysis gives the most consistent results across all cases.
Availability: R code available on `http://sites.uclouvain.be/absil/2017.01`
Contact: emilie.renard@uclouvain.be

1 Introduction

Nowadays, the development of sequencing technologies allows to measure gene expression levels at a reasonable cost. The analysis of the resulting data helps to better understand how genes are working, with the goal of developing better cures for genetic diseases such as cancer. Due to different constraints such as the limited number of samples that can be processed at the same time in an experiment, the size of such datasets is often limited in samples. However, statistical inferences need a high number of samples to be robust enough and generalizable to other data. As more and more of those datasets are available on public repositories such as GEO `http://www.ncbi.nlm.nih.gov/geo/`, merging and combining different datasets appears as a simple solution to increase the number of samples analyzed and potentially improve the relevance of the biological information extracted.

Expression levels of genes are the result of interactions between different biological processes. When measuring those expression levels, noise may also be added at each step of data acquisition due to imprecisions. In particular, different biases can be introduced depending on experimental conditions. Such confounding factors, or batch effects, that complicate the analysis of genomic data can be for example due to difference in chip type or platform, procedures that can differ from one laboratory to another, storage conditions, ambient conditions during preparation,... A carefully designed experimental process can limit the impact of such effects, but some are often unavoidable, especially when a large number of samples is necessary. Those batch effects can be quite large and hide the effects related to the biological process of interest. Not including those effects in the analysis process may adversely affect the validity of biological conclusions drawn from the datasets (Leek and Storey, 2007; Leek *et al.*, 2010; Teschendorff *et al.*, 2011). It is then important to be able to combine data from different sources while removing the batch effects. The difficulty is that the precise effects of those technical artefacts on gene expression levels is often unknown. However some partial information is usually available, such as the batch number or the date of experiment, and can be used as a proxy for those effects.

From here, we will refer to the smaller datasets to merge as *batches*, and to the bigger dataset resulting from the concatenation of the smaller datasets as *dataset*. We assume that the batch effect removal methods, termed here *(cross-batch) normalization methods*, have only access to the gene-by-samples matrix of expression levels ans to the batch number of each sample.

Available normalization can be classified in two main approaches: location-scale methods and matrix factorization methods. The location-scale methods assume a model for the data distribution within batches, and adjust the data within each batch to fit this model. This approach is the most straight-forward one and many methods have already been proposed: XPN (Shabalin *et al.*, 2008), DWD (Benito *et al.*, 2004), ratio-based methods (Luo *et al.*, 2010), ComBat (Johnson *et al.*, 2007), quantile based methods (Warnat *et al.*, 2005), mean or median centering (Sims *et al.*, 2008), ... The matrix factorization based methods assume that the gene-by-sample expression matrix can be represented by a small set of rank-one components which can be estimated by means of matrix factorization. The components that correlate with the batch number are then removed to obtain the normalized dataset (Alter *et al.*, 2000; Renard *et al.*, 2016). In Leek *et al.* (2010); Teschendorff *et al.* (2011), matrix factorization is used to model covariates in a differentially expressed gene (DEG) detection process. Matrix factorization based methods are less used, probably because of the indirect approach to the problem: if no batch effects is recovered in the rank-one components, then the data matrix is left unchanged.

The presence or absence of batch effects in a dataset — and thus the effectiveness of normalization methods — can be evaluated by different methods, which can be classified in three main groups: local approaches, global approaches and, in supervised cases, performance based approaches. Global methods aim to illustrate the global behavior of the dataset: the evaluation of batch effect presence uses many features at once. For example, the global behavior of all genes can be summarized by a clustering dendogram or a plot of the first principal components. A clustering of samples by batch shows presence of batch effects; it means that the predominant trend present in the data is linked to batches. However it does not imply that all genes are affected to the same extend, or even affected at all. On the contrary, local methods examine the behavior of one gene at a time: expression levels of a gene should have the same behavior (typically similar probability distributions) for all batches. When evaluating a normalization method, the clustering by batch and/or the differences in behavior across batches should disappear (or at least be weaker). Those evaluation methods are unsupervised in the sense that there is no ground truth to refer to: maybe the observed differences are due to a biological difference in the batches, and not to a technical artefact only. If we have access to some (part of) ground truth, then the adequacy of the batch effect removal method can be evaluated quantitatively. If some genes are known to be truly (not) differentially expressed, the proportion of those genes in the DEG list after normalization should ideally be higher (lower). P-values corresponding to null genes or negative control genes should be uniformly distributed across $[0, 1]$. The list of DEG should also be more stable after normalization. In prediction tasks where the final objective is for example to determine to which class a new sample belongs, performances should be improved after normalization. See for example Lazar *et al.* (2013) for a list of techniques to evaluate batch effects presence.

In sample classification tasks, if the class labels (i.e., the phenotype to predict) are quite balanced across batches, then batch effects are less likely to adversely impact the outcome (Taminau *et al.*, 2014; Parker and Leek, 2012). In the extreme case of perfect confounding of batches with the phenotype, it is much more difficult to be sure that the differentially expressed genes are linked to the batch or the phenotype. Such a case can occur when, for example, a laboratory first analyses the disease samples and matches them later with the controls. When combining data from already existing studies, the reality is often somewhere between those two extremes: group repartition of the phenotype to predict is often unbalanced, as for example in the breast cancer batches we use later in this paper (see Table 1). In cases where the repartition of the phenotype of interest is highly unbalanced among batches, being able to separate batch effects from phenotype is more challenging (Soneson *et al.*, 2014). Using datasets only preprocessed with either RMA or fMRA summarization, Parker and Leek (2012) tried to predict estrogen receptor status in a breast cancer dataset when batch and status are perfectly confounded. Their results suggest that the algorithm tends to predict the batch more than the status, and that a careful feature selection can improve those results. Nygaard *et al.* (2016) propose a sanity check using random numbers: replacing real data with normal random numbers shows that the F-statistic is inflated in presence of unbalanced repartition. Rudy and Valafar (2011) compare the list of DEG obtained from two batches separately to the DEG list obtained from the normalized merge of the two batches. When introducing unbalance in the repartition of group treatment among batches, most location-scale normalization methods they tested tend to detect less DEG. So if batch effects should be taken into account to avoid to predict batch instead of phenotype, we should also check that what is removed during the normalization step is really only the batch effects and does not contain potential useful information about the phenotype to predict. This more challenging configuration is at the heart of this paper, that aims to understand which normalization methods can handle this situation.

In this paper, we investigate the impact of cross-batch normalization methods on sample classification tasks. We compare the two approach families (location-scale vs matrix factorization) to understand their advantages and weaknesses, and to find out which methods perform best in which scenario. More specifically, we examine three configurations of batch effects. The first one includes an 'outlier' batch where all gene expressions are really different from the other batches, which implies a strong correlation (in the broad meaning of correlation) between gene expression and batch. The second one does not include any 'outlier', but presents a correlation between batch and phenotype to predict. The last one combines both difficulties: the 'outlier' batch, and unbalanced phenotype repartition among batches.

The paper is organized as follows. Section 2 details the methods examined, which are experimented and analyzed in Section 3, and conclusions are drawn in Section 4.

2 Normalization methods

Among the many normalization methods mentioned above, we choose to investigate more in depth factorization based methods using ICA and SVD and to compare them to the location-scale method ComBat. ComBat is widely used in the literature, often appears among the best normalization methods (Chen *et al.*, 2011; Rudy and Valafar, 2011; Taminau *et al.*, 2014; Shabalin *et al.*, 2008; Luo *et al.*, 2010) and can easily be applied to more than two batches at a time. Another widely used method, based on SVD, is SVA Leek *et al.* (2010). However unlike ComBat this method necessitates more information than only batch and requires the phenotype to predict to be available. This is why SVA is not included in the comparison. Centering-scaling was also investigated, but as ComBat is an improved version of it, results for centering-scaling are not systematically described.

2.1 Location-scale methods

Location-scale methods are maybe the most intuitive way to handle the data. Choosing a reasonable model representing the probability distribution of gene expression, and assuming that genes behave in the same way in each batch, expression values are adapted to fit this model within each batch. The goal is to let each gene have a similar mean and/or variance in each batch. A main hypothesis in such methods is that by adjusting the gene distributions no biological information is removed.

The simplest way to normalize a dataset in order to remove batch effects is to standardize each batch separately. That is, for each gene in each batch, the expression values are centered and divided by their standard deviation.

A widely used and more complex location-scale method is ComBat (Johnson *et al.*, 2007). The expression value of gene i for sample j in batch b is modeled as $X_{bij} = \alpha_i + \beta_i C_j + \gamma_{bi} + \delta_{bi}\epsilon_{bij}$ where α_i is the overall gene expression, and C_j is the vector of known covariates representing the sample conditions. The error term ϵ_{bij} is assumed to follow a normal distribution $N(0, \sigma_i^2)$. Additive and multiplicative batch effects are represented by parameters γ_{bi} and δ_{bi}. ComBat uses a Bayesian approach to model the different parameters, and then removes the batch effects from the data to obtain the clean data $X^*_{bij} = \hat{\epsilon}_{bij} + \hat{\alpha}_i + \hat{\beta}_i C_j$. By pooling information across genes, this approach is more robust to outliers in small sample sizes.

2.2 Matrix factorization based methods

The factorization based normalization process, adapted from Renard *et al.* (2016), is detailed in Algorithm 1. It is more general than in Renard *et al.* (2016) because there are still two main choices to do: first the matrix factorization method to use (line 1), second the measure used to evaluate the correlation between batches and components (line 2). Here correlation should be understood in is broad sense, that is a relationship, whether causal or not, between two variables. Let us now discuss Algorithm 1 line by line.

The first step is to find the best approximation of the gene-by-sample matrix X using a low-rank factorization (line 1):

$$X \approx AB^T = \sum_{k=1}^{K} A_{:,k} B^T_{:,k}. \tag{1}$$

$A_{:,k}$ can be interpreted as the gene activation pattern of component k and $B_{:,k}$ as the weights of this pattern in the samples. Each $A_{:,k}$ can be interpreted as a meta-gene, i.e. a group of genes working together in a specific condition. $B_{:,k}$ should then be related to the activation pattern of this condition among samples. Many methods exist to factorize a matrix, depending on the properties the factorization components have to fulfill. Imposing orthogonality among components leads to a Singular Value Decomposition (SVD). Normalization of gene expression data using SVD was first proposed in Alter *et al.* (2000). Minimizing statistical dependence across component leads to Independent Component Analysis (ICA) methods. Different variants exist for such methods depending on how statistical dependence is evaluated, and if we want to impose independence among genes or samples dimension, or even using a trade-off between both options. ICA was shown to better model the different sources of variation than SVD (Teschendorff *et al.*, 2011). A normalization method using an implementation based on JADE approach and offering the possibility to choose the trade-off between samples and genes independence was proposed in Renard *et al.* (2016). Other factorization methods exist in the literature, such as Non-negative Matrix Factorization (Lee and Seung, 1999) which can be used when dealing with non-negatives matrix values to obtain components with non-negatives values only. Matrix of binary values have also their corresponding factorization techniques (Zhang *et al.*, 2010).

Once the factorization computed, we select the $B_{:,k}$'s that correlate with the batch. If a component presents enough correlation with the batch (line 3), then this component is selected. As batch is a categorical information and the $B_{:,k}$'s are continuous, the usual linear correlation formula (Pearson or Spearman) cannot be used. To estimate which components are related to batch, as in Renard *et al.* (2016) we use the R^2 value that measures how well a variable x (here, c) can predict a variable y (here, $B_{:k}$) in a linear model:

$$R^2(x, y) \equiv 1 - \frac{SS_{res}}{SS_{tot}}.$$

$SS_{tot} = \sum_i (y_i - \bar{y})^2$ is the sum of squares of the prediction errors if we take the mean $\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ as predictor of y. $SS_{res} = \sum_i (y_i - \hat{y}_i)^2$ is the sum of squares of the prediction errors if we use a linear model $\hat{y}_i = f(x_i)$ as predictor: if x is continuous the prediction model is a linear regression, if x is categorical we use a class mean. The R^2 value indicates the proportion of the variance in y that can be predicted from x, and has the advantage to be usable with categorical or continuous variables. So the higher the R^2 value, the better the association between both variables. As the batch information is categorical, $R^2(c, B_{:k})$ compares the prediction of B_{ik} by a general mean $\sum_j \frac{B_{jk}}{n}$ or by a batch mean $\sum_{j \in C_i} \frac{B_{jk}}{\#C_i}$ (where C_i represents all samples in the same batch as sample j).

An additional step can be added in the process to check if the selected components do not correlate with some information of interest (lines 4-6, optional). The selected components are then removed from the matrix X to obtain a cleaned dataset (line 7).

Algorithm 1 Matrix factorization based normalization

Require: X ($p \times n$) the aggregated dataset to be normalized, c (n) a categorical variable indicating the batch number, $matfact$ the matrix factorization method, $t \in [0, 1]$ the threshold to consider a component associated to c, [optional] c_2 (n) categorical/continuous information that we want to preserve

```
1: A, B ← matfact(X)
2: R ← cor(c, B)
3: ix ← which(R ≥ t)
4: R2 ← cor(c2, B)                ▷ optional
5: ix2 ← which(R2 ≥ R)            ▷ optional
6: ix ← ix \ ix2                  ▷ optional
7: Xn ← X − A[:, ix] ∗ B[:, ix]^T
```

3 Results and discussion

We tested the normalization methods on breast cancer expression. We combined different batches which can be accessed under GEO numbers GSE2034 (Wang *et al.*, 2005) and GSE5327 (Minn *et al.*, 2007), GSE7390 (Desmedt *et al.*, 2007), GSE2990 (Sotiriou *et al.*, 2006), GSE3494 (Miller *et al.*, 2005), GSE6532 (Loi *et al.*, 2007) and GSE21653 (Sabatier *et al.*, 2011). All batches were summarized with MAS5 and represented in log2 scale, except GSE6532 which was already summarized with RMA. We took as phenotype of interest to predict the estrogen-receptor status (ER). We removed the samples and features with missing information which gives an aggregated dataset of 22276 genes.

In order to compare the effect of normalization in various cases, we considered three different aggregated datasets. The first one keeps all samples, the main difficulty being that batch 5 is summarized using a different method. In the second we removed batch 5, but introduced deliberately a correlation between ER status and the batch number by subsampling

Table 1. Repartition of ER status in the batches: number N of samples by batch, and proportion p of positive estrogen receptors

Batch	Dataset 1		Dataset 2		Dataset 3		GEO number
	N	p	N	p	N	p	
1	344	0.61	169	0.20	-	-	GSE2034, GSE5327
2	198	0.68	-	-	-	-	GSE7390
3	183	0.81	183	0.81	183	0.81	GSE2990
4	247	0.86	247	0.86	247	0.86	GSE3494
5	126	0.68	-	-	50	0.20	GSE6532
6	263	0.58	138	0.20	138	0.20	GSE21653

among 4 batches. The third dataset combines both difficulties (different summarizations and correlation). The repartition of the ER status in these three cases is described in Table 1.

To evaluate the presence of batch effects, we plotted the two first principal components of the datasets (see Figure 1). Principal components represent linear combinations of features (here, genes values) giving the largest possible variance, such that components are uncorrelated: the first components capture most of the variability in the data. On Figure 1, we can see that the first two principal components of the datasets show a global clustering by batch. Batch 5 is the most clearly separated, probably due to its summarization using RMA and not MAS5 as the others.

In the rest of this section, three normalization methods (ComBat, SVD and ICA based) and the case without normalization are compared regarding possible batch effects. First we investigate the effect of factorization in Section 3.1. In a second time, we compare in Section 3.2 the results obtained in the context of a classification task.

3.1 Factorization interpretation

We computed different factorizations of the aggregated dataset, yielding the factors A and B as in Equation 1. We compared the SVD factorization to three different ICA factorization: independence among genes, independence among samples, and a trade-off between both options. Those three ICA factorizations correspond to set the α parameter in Renard *et al.* (2014) to 0, 1 and 0.5 respectively.

Ideally, the components removed for normalization should be associated to the batch, an not to ER. We compared the association between factorization components and ER or batch using the R^2 value on Figure 2. For all datasets, all factorizations recovered components associated to batch number. Associations with ER status are weaker, but non negligible nevertheless. Due to the presence of correlation between batch and ER status in datasets 2 and 3, it is more difficult to obtain components linked to the batch but not to the ER, and conversely. To better evaluate the R^2 values when components have significant R^2 values for both batch and ER status, we sub-sampled the data to re-balance batch and ER status. Comparison of mean R^2 values obtained from 100 sub-samplings in the ICA$_{\alpha=0}$ and SVD in dataset 2 are given in Table 2. For ICA components, the re-evaluations of R^2 suggest that each component is associated to only one effect, the one with the initial higher association. In the SVD case, there is still one component associated to both effects.

On Figure 3 page 7 we represented for different ICA decompositions the R^2 values between all components and ER/batches, and the Pearson correlation between components themselves. In the ICA case, $\alpha = 0$ allows to recover a higher number of components linked to the batch. For example on dataset 2, components 5, 6, 11 and 14 have a R^2 higher than 0.5. However, we can see on Table 3 that all components are highly correlated. As expected, using an α greater than 0 enables to decreases the correlation among B components. Imposing independence among the

Table 2. Various estimations of R^2 values, on all batches or only the training batches (see Section 3.1), using all samples or a mean obtained from 100 sub-samplings to rebalance the POI among batches in training (for dataset 2, ICA$_{\alpha=0}$ and SVD).

Cmpt	R^2 on all samples		$R^2(batch)$ on samples in batches nb						$R^2(ER)$ on samples in batches nb						
	$batch$	ER	1, 3	1, 4	1, 6	3, 4	3, 6	4, 6	1, 3	1, 4	1, 6	3, 4	3, 6	4, 6	
ICA 5	0.79	0.25	0.24	0.92	0.73	0.80	0.43	0.48	*0.00*	0.37	0.24	0.30	0.11	*0.04*	All samples
13	0.19	0.40	*0.00*	0.18	0.18	0.20	0.21	*0.00*	0.24	0.37	0.41	0.39	0.42	0.27	
14	0.58	0.25	*0.02*	0.80	0.80	0.31	0.33	0.18	*0.01*	0.32	0.27	0.23	0.21	*0.03*	
18	0.26	0.53	*0.01*	0.25	0.30	0.23	0.28	*0.01*	0.31	0.57	0.54	0.53	0.51	0.45	
ICA 5	0.82	*0.00*	0.26	0.92	0.79	0.83	0.55	0.50	*0.00*	*0.00*	*0.00*	*0.01*	*0.01*	0.07	Balanced sub-sampling
13	*0.01*	0.35	*0.00*	*0.00*	*0.01*	*0.01*	*0.01*	*0.00*	0.34	0.36	0.37	0.34	0.36	0.38	
14	0.60	*0.01*	*0.02*	0.82	0.82	0.23	0.27	0.32	*0.01*	*0.00*	*0.00*	*0.05*	*0.04*	*0.05*	
18	*0.02*	0.47	*0.03*	*0.00*	*0.02*	*0.00*	*0.02*	*0.00*	0.43	0.57	0.43	0.54	0.41	0.54	
SVD 1	0.87	0.27	0.06	0.93	0.91	0.59	0.61	0.82	*0.00*	0.38	0.37	0.26	0.30	*0.00*	All samples
2	0.89	0.14	0.17	0.96	0.88	*0.03*	0.08	0.95	*0.01*	0.46	0.36	0.08	*0.01*	*0.00*	
3	0.42	0.42	*0.00*	*0.05*	*0.04*	0.55	0.54	0.34	0.27	0.31	0.29	0.60	0.55	0.22	
SVD 1	0.88	*0.00*	*0.02*	0.94	0.93	0.60	0.60	0.83	*0.00*	*0.00*	*0.00*	*0.00*	*0.01*	*0.00*	Balanced sub-sampling
2	0.91	*0.00*	0.15	0.96	0.91	*0.00*	0.13	0.95	*0.01*	*0.01*	*0.00*	0.09	*0.01*	*0.01*	
3	0.24	0.31	*0.00*	*0.05*	*0.03*	0.23	0.27	0.33	0.36	0.43	0.36	0.35	0.28	0.30	

Components shown are those exhibiting a significant R^2 value with both batches and ER status. In italic the R^2 values ≤ 0.05. After resampling, ICA components 5 and 14 and SVD components 1 and 2 did not show anymore correlation with ER status, while ICA components 13 and 18 did not show anymore correlation with batches. Unlike ICA, an indetermination remains in SVD components: resampling did not allow to determine if component 3 is linked to only ER status or batch.

Table 3. Correlation between B components linked to batches ($R^2 > 0.5$) for dataset 2, $\alpha = 0$

	$B_{*,5}$	$B_{*,6}$	$B_{*,11}$	$B_{*,14}$
$B_{*,5}$	1.00	0.65	0.57	0.52
$B_{*,6}$	0.65	1.00	0.76	0.35
$B_{*,11}$	0.57	0.76	1.00	0.40
$B_{*,14}$	0.52	0.35	0.40	1.00

samples only ($\alpha = 1$) decreases the number of component linked to the batch. We choose to investigate only the $\alpha = 0$ and $\alpha = 0.5$ cases further.

3.2 Validation by impact on classification

To compare the different normalization methods, we used them in a whole classification process where the ER status is predicted using an SVM classifier. The ER status is thus the phenotype to predict, i.e. the outcome. The whole process is described in Algorithm 2. The first step is to normalize the aggregated dataset X with the chosen method (here, ComBat, SVD or ICA based normalization) (line 2). Optionally, the training labels (separated from test labels in line 1) can be used as extra information to preserve. Here we did not use this option in our experiments (in particular, c_2 is not used in Algorithm 1). In ICA and SVD based normalizations, we computed the $K = 20$ first components and removed the components with an R^2 value higher than $t = 0.5$. Training and testing sets are then separated (line 3). To mimic the case where the model is built based on some studies and then validated on other separate studies, we kept each time two batches out for testing, and trained the SVM classifier on the other batches. This gave a total of $C_6^2 = 15$ experiments for dataset 1, and $C_4^2 = 6$ experiments for datasets 2 and 3. A basic feature selection is performed by selecting the genes with the best association with the ER label based on a t-test (line 4). A standard SVM model is trained based on those genes (lines 5 to 8). To keep the SVM model simple, we used the linear kernel implementation provided in the LiblineaR R package. The cost parameter is fixed using the heuristic implemented in the package, and the classes weights are set to $[1 - p_0, 1 - p_1]$ where p_i gives the proportion of samples in class i. When training the SVM model, train data are first centered and scaled to ensure to treat all features with the same

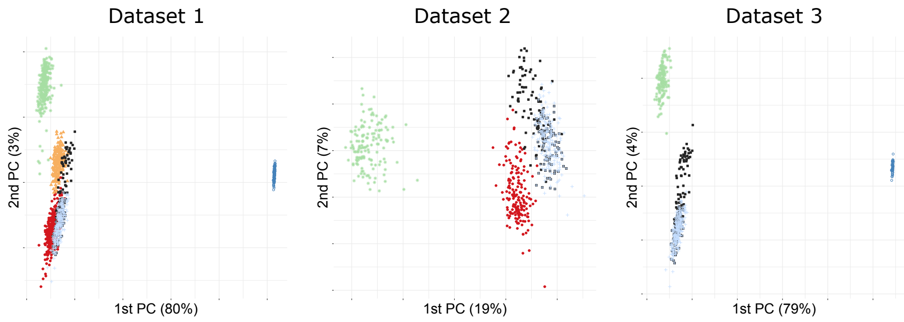

Fig. 1. Global evaluation of batch effects in the 3 datasets: plots of the first two principal components. Colors represent the batch membership.

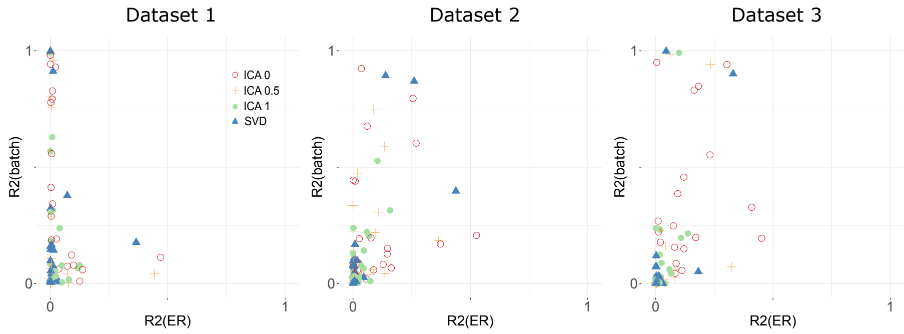

Fig. 2. Association between factorization components and ER or batch, for SVD and ICA with $\alpha = \{0, 0.5, 1\}$.

weight; the same centering-scaling is then applied on the test data. The ER labels of testing set are finally predicted using the SVM model (line 9). We repeated the prediction step (line 5 to 9) using different numbers of selected genes (line 4).

Algorithm 2 Classification process

Require: X $(p \times n)$ aggregated matrix of gene expression, y (n) the label to predict, c (n) the batch information

1: $y_{tr}, y_{te} \leftarrow y$
2: $X_n \leftarrow \texttt{normalizationMethod}(X, c, y_{tr})$
3: $X_{tr}, X_{te} \leftarrow X_n$;
4: $idx_{bestGenes} \leftarrow \texttt{ttest}(y_{tr}, X_{tr})$
5: $X_{SVM} \leftarrow X_{tr}[idx_{bestGenes}, :]$
6: $c \leftarrow \texttt{heuristic}(X_{SVM})$
7: $w \leftarrow 1 - [prop_{y_{tr}=1}\ prop_{y_{tr}=0}]$
8: $model_{SVM} \leftarrow \texttt{SVM}(X_{SVM}, y_{tr}, c, w)$
9: $\hat{y}_{te} \leftarrow \texttt{prediction}(model_{SVM}, X_{te}[idx_{bestGenes}, :])$

The results on the testing sets for the different normalization methods applied on the three datasets were compared to the case without normalization on Figure 4 (top). We choose to use the balanced classification rate (BCR) as performance measure. The balanced classification rate, or balanced accuracy, is computed as $0.5(\frac{TP}{TP+FN} + \frac{TN}{TN+FP})$ and allows to take into account a potential imbalance between classes. Similar results (not shown) were obtained when using area under the ROC curve as performance measure. The better performances are obtained unsurprisingly with dataset 1 that has a higher number of training samples, and the worse with dataset 3 that presents a higher correlation between batch and outcome. We can see with dataset 1 that if all normalization methods behaves similarly, the normalization step is necessary and allows to increase the BCR. The necessity of normalization is mainly due to batch 5, which is highly decentered compared to the others, as shown with the plots of principal components on Figure 1. Dataset 2 presents a correlation between batch and outcome, so applying location-scale normalization method like ComBat tends to remove too much information and decreases the BCR. No normalization or an ICA$_{\alpha=0.5}$ based normalization gave the best results; SVD is in between. The effect of the normalizations on the two first genes selected by a t-test is shown on Figure 5. The worst class separation is obtained using ComBat: by forcing all batches to have a similar mean, both classes are pushed closer to each other. Dataset 3 is the worse dataset in the sense that there is an 'outlier' batch that necessitates some normalization, and correlation between outcome and batch which requires to be careful when normalizing: for a limited number of genes ICA based normalization gives the best performances. In this specific case, the ICA$_{\alpha=0}$ gives significantly better results, nearly as good as dataset 1. We can hypothesize that with so strong batch effects, assuming independence in the sample dimension is less realistic.

To investigate robustness with respect to features, we removed from the dataset (before any normalization) the 100 best genes associated to ER status. For this we ranked the genes using a t-test within each batch and within the un-normalized merged dataset. Then we took the median of those ranks for each genes as a final ranking. As expected, removing those genes decreased performances (on Figure 4, bottom), especially when selecting a limited number of genes. The "none" curve (no normalization) curve is more affected than the others, especially in dataset 3.

4 Conclusion

In the context of merging gene expression datasets, we have conducted a detailed comparison between two types of batch effect removal approaches: location-scale (ComBat and centering-scaling) and matrix factorization (based on SVD or ICA) normalization methods. We have compared those normalization methods, used as classifier preprocessing

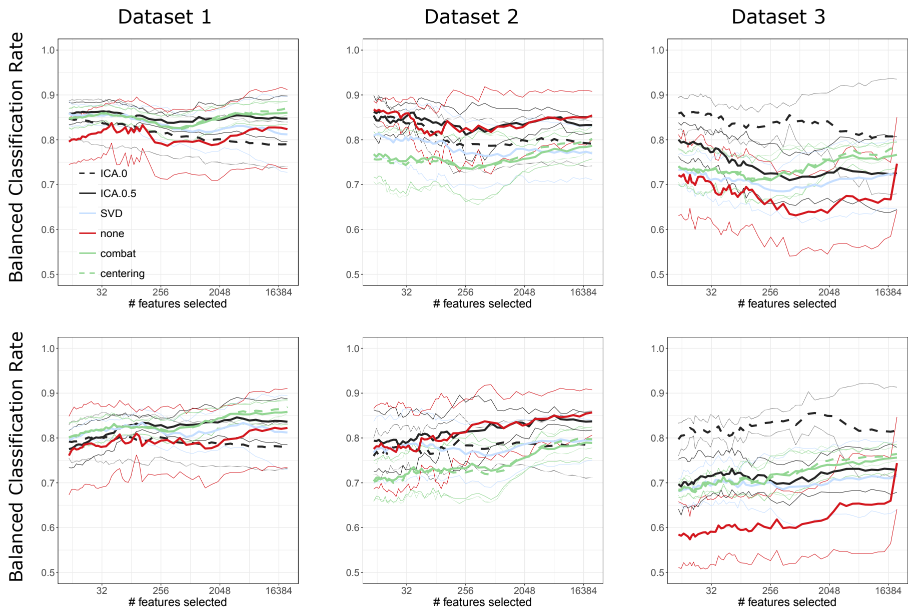

Fig. 4. Influence of the normalization methods on BCR. Thiner lines corresponds to standard deviations. The horizontal axis gives the imposed number of genes to be selected by line 4 of Algorithm 2. Top: results using all features. Bottom: results when removing from the dataset (before any normalization) the 100 first features with the best association with the ER.

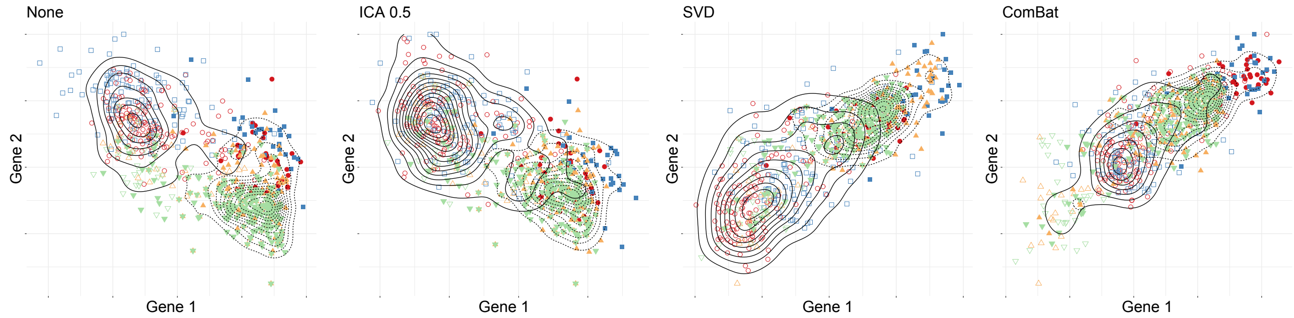

Fig. 5. Effect of the normalization for dataset 2 on the first two genes selected when applying a t-test on the normalized dataset (gene 1 is the same for all, while gene 2 is the same for none and ICA on one side, and for SVD and ComBat on the other side). Each dot corresponds to a sample; shape/color gives the batch and full or empty correspond to ER status. The contour lines represent estimated densities of ER positive (dashed lines) and negatives classes (plain lines).

tools, in three different configurations with stronger or weaker batch effects, and stronger or weaker presence of correlation between batch and the phenotype to predict.

When merging batches preprocessed differently (such as using MAS5 and RMA summarization methods) with a reasonably balanced repartition of the phenotype to predict, applying any of the tested normalization methods (even a simple centering-scaling) allows to improve classification results (see Figure 4 left). In classification tasks, batch effects become a real problem when the batch and the phenotype to predict are partially confounded. The difficulty is then to separate batch effects from real changes due to the phenotype, to avoid the risk of removing meaningful information. As location-scale methods using only batch information tend to remove also phenotype information in such a case, they are not recommended. As shown here (see Figure 4 center), using a matrix factorization method instead of a location-scale one can have a significant positive impact on classification performances when samples in batches are not well balanced. ICA factorization tends to better separate batch influence from the phenotype of interest than SVD factorization. However without major difference between batches (same preprocessing of data), merging summarized batches directly without normalization gives good results. When both difficulties are combined (different summarizations and unbalanced repartition of the phenotype to predict), the ICA based normalization is capable of dealing with the summarization differences while not removing too much information, and gives considerably better results (see Figure 4 right). Finally, we have observed that the ICA_0 normalization method clearly outperforms the other methods in the most challenging setting (Figure 4 right).

In summary, a normalization is necessary when merging batches preprocessed with different summarizations. However, location-scale methods do more harm than good in presence of unbalanced repartition of the phenotype to predict. On all tested normalization methods, the ICA based normalization appears to give the most consistent results in all configurations.

References

Alter, O., Brown, P. O., and Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, **97**(18), pp. 10101–10106.

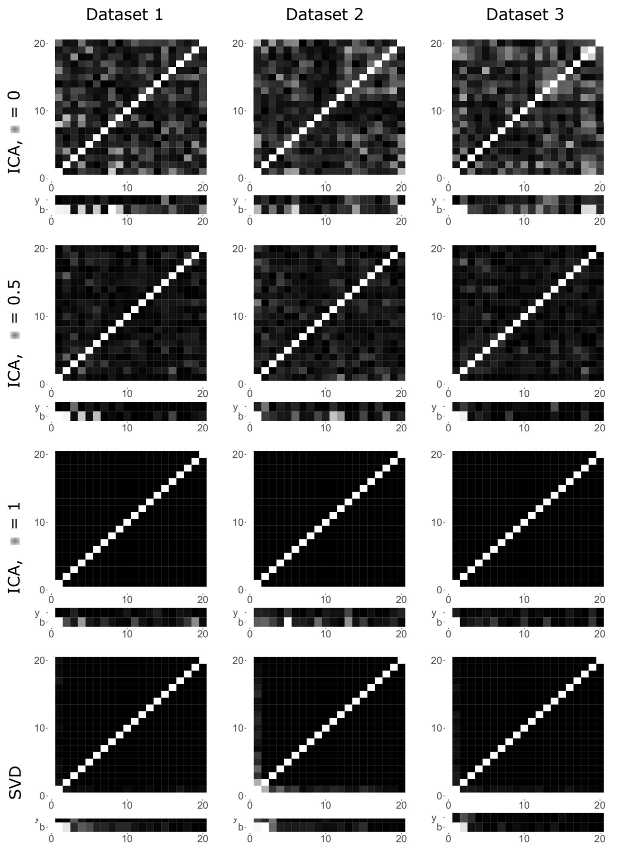

Fig. 3. Pearson correlation between components and R^2 value between components and batches or ER status (respectively b and y in the figure) for different factorization methods. Black corresponds to 0, white to 1.

Benito, M. *et al.* (2004). Adjustment of systematic microarray data biases. *Bioinformatics*, **20**(1), 105–114.

Chen, C. *et al.* (2011). Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PloS one*, **6**(2), e17238.

Desmedt, C. *et al.* (2007). Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series. *Clinical cancer research*, **13**(11), 3207–3214.

Johnson, W. E., Li, C., and Rabinovic, A. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, **8**(1), 118–127.

Lazar, C. *et al.* (2013). Batch effect removal methods for microarray gene expression data integration: a survey. *Briefings in bioinformatics*, **14**(4), 469–490.

Lee, D. D. and Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, **401**(6755), 788–791.

Leek, J. *et al.* (2010). Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet*, **11**(10), 733–739.

Leek, J. T. and Storey, J. D. (2007). Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet*, **3**(9), e161.

Loi, S. *et al.* (2007). Definition of clinically distinct molecular subtypes in estrogen receptor–positive breast carcinomas through genomic grade. *Journal of clinical oncology*, **25**(10), 1239–1246.

Luo, J. *et al.* (2010). A comparison of batch effect removal methods for enhancement of prediction performance using maqc-ii microarray gene expression data. *The pharmacogenomics journal*, **10**(4), 278–291.

Miller, L. D. *et al.* (2005). An expression signature for p53 status in human breast cancer predicts mutation status, transcriptional effects, and patient survival. *Proceedings of the National Academy of Sciences of the United States of America*, **102**(38), 13550–13555.

Minn, A. J. *et al.* (2007). Lung metastasis genes couple breast tumor size and metastatic spread. *Proceedings of the National Academy of Sciences*, **104**(16), 6740–6745.

Nygaard, V., Rødland, E. A., and Hovig, E. (2016). Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, **17**(1), 29–39.

Parker, H. S. and Leek, J. T. (2012). The practical effect of batch on genomic prediction. *Statistical applications in genetics and molecular biology*, **11**(3).

Renard, E., Teschendorff, A. E., and Absil, P. (2014). Capturing confounding sources of variation in dna methylation data by spatiotemporal independent component analysis. In *22nd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN 2014),*.

Renard, E., Branders, S., and Absil, P.-A. (2016). Independent component analysis to remove batch effects from merged microarray datasets. In *International Workshop on Algorithms in Bioinformatic*.

Rudy, J. and Valafar, F. (2011). Empirical comparison of cross-platform normalization methods for gene expression data. *BMC bioinformatics*, **12**(1), 467.

Sabatier, R. *et al.* (2011). A gene expression signature identifies two prognostic subgroups of basal breast cancer. *Breast cancer research and treatment*, **126**(2), 407–420.

Shabalin, A. *et al.* (2008). Merging two gene-expression studies via cross-platform normalization. *Bioinformatics*, **24**(9), 1154–1160.

Sims, A. *et al.* (2008). The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets–improving meta-analysis and prediction of prognosis. *BMC medical genomics*, **1**(1), 42.

Soneson, C., Gerster, S., and Delorenzi, M. (2014). Batch effect confounding leads to strong bias in performance estimates obtained by cross-validation. *PloS one*, **9**(6), e100335.

Sotiriou, C. *et al.* (2006). Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. *Journal of the National Cancer Institute*, **98**(4), 262–272.

Taminau, J. *et al.* (2014). Comparison of merging and meta-analysis as alternative approaches for integrative gene expression analysis. *ISRN bioinformatics*, **2014**.

Teschendorff, A. E., Zhuang, J., and Widschwendter, M. (2011). Independent surrogate variable analysis to deconvolve confounding factors in large-scale microarray profiling studies. *Bioinformatics*, **27**(11), 1496–1505.

Wang, Y. *et al.* (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, **365**(9460), 671–679.

Warnat, P., Eils, R., and Brors, B. (2005). Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC bioinformatics*, **6**(1), 1.

Zhang, Z.-Y. *et al.* (2010). Binary matrix factorization for analyzing gene expression data. *Data Mining and Knowledge Discovery*, **20**(1), 28–52.