BRIEFINGS IN BIOINFORMATICS. VOL 14. NO 4. 469–490
Advance Access published on 31 July 2012
doi:10.1093/bib/bbs037

Batch effect removal methods for microarray gene expression data integration: a survey

Cosmin Lazar, Stijn Meganck*, Jonatan Taminau*, David Steenhoff, Alain Coletta, Colin Molter, David Y. Weiss-Solís, Robin Duque, Hugues Bersini and Ann Nowé*

Submitted: 23rd March 2012; Received (in revised form): 31st May 2012

Abstract

Genomic data integration is a key goal to be achieved towards large-scale genomic data analysis. This process is very challenging due to the diverse sources of information resulting from genomics experiments. In this work, we review methods designed to combine genomic data recorded from microarray gene expression (MAGE) experiments. It has been acknowledged that the main source of variation between different MAGE datasets is due to the

Corresponding author. Cosmin Lazar, Como, Vrije Universiteit Brussel, Pleinlaanz, 1050 Brussels, Belgium. E-mail: vlazar@vub.ac.be.
*These authors contribute equally.

Cosmin Lazar is currently working with CoMo Lab at Vrije Universiteit Brussel (VUB) as a postdoctoral researcher. He received his PhD in Informatics, Automatics and Signal Processing from the University of Reims Champagne Ardenne, France (2008). His research interests include data mining, supervised/unsupervised learning, feature selection/extraction, blind source separation and their application in the analysis of microarray gene expression data, multi/hyperspectral images and time series.

Stijn Meganck got his PhD at the Vrije Universiteit Brussels (VUB), Belgium, in 2008. Since then he has been working as a postdoctoral researcher at two research groups at the VUB: AI and ETRO. His main research interests are Bioinformatics, Probabilistic Graphical Models and Causality.

Jonatan Taminau After obtaining his advanced master in Bioinformatics, Jonatan Taminau is currently finishing his PhD at the Vrije Universiteit Brussel on the topic of large-scale analysis of microarray data.

David Steenhoff obtained his master degree in Sciences of Industrial Engineering in Electronics and ICT. This was facilitated by the Erasmushogeschool Brussel, Vrije Universiteit Brussel and the Hanoi University of Technology. His research interests are machine learning and data mining applied in microarray gene expression analysis and hyperspectral imaging.

Alain Coletta PhD graduated from the Université Libre de Bruxelles and obtained his PhD from Manchester University, Faculty of Engineering ad Physical Sciences, Advanced Interfaces Group.

Colin Molter received his PhD in artificial intelligence at the Université Libre de Bruxelles (ULB), Belgium, in 2005. After reception of his PhD, he started a postdoc in computational neuroscience at the RIKEN-Brain Science Institute first and the Ecole Polytechnique Fédérale de Lausanne next. Recently, he got interest in genetics and started working on the inSilico project in Bruxelles.

David Y. Weiss-Solís studied bioengineering (2003) and then pursued a PhD in cancer bioinformatics at the faculty of medicine of the ULB (2009), in collaboration with IRIDIA. His PhD research consisted of the analysis of microarray gene expression profiles of thyroid tumors and of their *in vitro* models, with a special focus on oncogenic signaling pathways. He is currently at IRIDIA within the *In Silico* project. David is involved in building software infrastructure for large-scale public-domain genomic datasets retrieval, meta-data standardisation and preprocessing for data mining applications.

Robin Duque graduated in 2008 as Master in Sciences of Industrial Engineering in Electromechanics followed by a Master in Management in 2009, both from Vrije Universiteit Brussel (VUB). Currently, he is working as programming engineer/developer at IRIDIA laboratory, Universit Libre de Bruxelles (ULB).

Hugues Bersini has a MS degree (1983) and a PhD in engineering (1989) both from Université Libre de Bruxelles (ULB). He is now heading IRIDIA laboratory (the AI laboratory of ULB) with Marco Dorigo. Since 1992, he has an assistant professor position at ULB and has now become full professor, teaching computer science, programming and AI. Over the last 20 years, he has published about 250 articles on his research work which covers the domains of cognitive sciences, AI for process control, connectionism, fuzzy control, lazy learning for modelling and control, reinforcement learning, biological networks, the use of neural nets for medical applications, frustration in complex systems, chaos, computational chemistry, object-oriented technologies, immune engineering and epistemology.

Ann Nowé received the MS degree from Universiteit Gent, Belgium, in 1987, where she studied mathematics with a minor in computer science, and the PhD degree from Vrije Universiteit Brussels (VUB), Belgium, in collaboration with Queen Mary and Westfield College, University of London, UK, in 1994. Currently, she is a Full-Professor at the VUB and co-head of the Computational Modeling Lab. Her major area of interest is machine learning, including multi-agent reinforcement learning and bioinformatics.

so-called 'batch effects'. The methods reviewed here perform data integration by removing (or more precisely attempting to remove) the unwanted variation associated with batch effects. They are presented in a unified framework together with a wide range of evaluation tools, which are mandatory in assessing the efficiency and the quality of the data integration process. We provide a systematic description of the MAGE data integration methodology together with some basic recommendation to help the users in choosing the appropriate tools to integrate MAGE data for large-scale analysis; and also how to evaluate them from different perspectives in order to quantify their efficiency. All genomic data used in this study for illustration purposes were retrieved from InSilicoDB http://insilico.ulb.ac.be.

Keywords: *Microarray gene expression data; batch effect removal; large-scale genomic data analysis; combining microarray datasets; microarray gene expression data merging; data integration*

INTRODUCTION

The integrative analysis of multiple microarray gene expression (MAGE) datasets has been acknowledged to be a crucial approach for extracting the maximum relevant biological information from genomic datasets [1]. Its importance resides mainly in the potentially huge biological insights that could be discovered by analysing at the same time large numbers of genomic datasets available through public repositories such as Gene Expression Omnibus [2], ArrayExpress [3] or InSilicoDB (http://insilico.ulb.ac.be/) [4]. Classical analytical tools show their limits in making useful discoveries that can be generalized especially because robust statistical inference can only be achieved by analysing a high enough number of samples sharing the same characteristics. A clear application, where this limitation has been pointed out, is the prediction of disease outcome where thousands of samples are needed to generate robust gene/protein signatures [5, 6]. Another important beneficial aspect which naturally derives from developing and using integrative analysis tools is related to the cost of MAGE experiments. Recycling and reusing public available data would also considerably reduce the overall costs of experiments.

Roughly speaking, integrative analysis can be performed according to two different strategies: 'meta-analysis' and 'integrative analysis via data merging or pooling' [7]. The first approach consists in analysing each dataset independently and finally the results are combined in a so-called 'meta-analysis'. It is assumed that if a result is found as being significant for a big number of individual studies, it will be significant for the particular problem the studies have been designed for. Moreover, if a finding is not significant in some studies, it could still be significant after meta-analysis if it appears as being significant in a big enough number of other individual studies, as the evidence will cumulate for this particular finding. Several large-scale meta-analysis of MAGE datasets have already been performed based on the above mentioned assumption and reported in the literature [1, 8, 9]. However, the reader has to be aware that when datasets containing few samples are studied, it is hard to derive rigorous inference upon the results issued from their analysis. A direct consequence of combining the results issued from the analysis of datasets containing few samples is the fact that the statistical hypothesis tests used to make decisions using MAGE data are prone to high false-negative rates.

The second approach for integrative analysis differs from the meta-analysis by the fact that in a first instance the samples from the different datasets are combined or merged in a bigger dataset, the subsequent analysis being performed on the new integrated dataset. Its main advantage over the meta-analysis approach consists in the higher statistical relevance of the results obtained by analysing datasets of hundreds or thousands of samples which naturally leads to more robust inference. This has been the main motivation for the development of a wide range of methods for integrating (or combining, or merging) MAGE data originating from different studies over the past years [10–14]. Nevertheless, combining or merging data from different MAGE experiments for integrative analysis suffers from the so-called 'batch effects' and it still is a challenging and difficult problem to be solved in computational biology. It is the very scope of this article and we will discuss it in detail in the upcoming sections.

The problem of combining (or merging) datasets from different MAGE experiments is a difficult task due to three generic sources of unwanted variation: 'noise' or 'expression heterogeneity', 'batch effects' and 'other sources of bias'. We will make the

distinction between these terms in the next Section. The batch effect is one of the main sources of unwanted variation which hinders the combination of different datasets, and it originates from the limitation in the number of samples that can be processed at one time in MAGE experiments [15]. As a consequence, batch effects are 'almost inevitable', but there are only few researchers who address this problem in their analysis flow (see [16] for some concluding figures on the number of articles published from 1 January 2010 to 1 July 2010, which addressed this problem). Large-scale analysis of genomic data based on data integration (or merging) has already been performed; in [17] the authors constructed a global gene expression map based on principal component analysis (PCA), by integrating microarray data from 5372 human samples representing 369 different cell and tissue types, disease states and cell lines. However, the authors did not explicitly address the batch effect problem, which turns to be a critical obstacle for genomic data integration.

The problems raised by the batch specific unwanted variation as well as the potential sources leading to batch effects have already been revealed and widely discussed in a number of publications [18–25]; we will only summarize them briefly in the next Section.

We stress on the fact that without efficient methods to reliably combine the MAGE datasets from different experiments, the analysis can only be performed 'per dataset' using meta-analysis tools which is supposed to derive reliable inference from the data. Only biological findings that are found as being statistical significant in a high enough number of individual studies can be used for generalization. Performed in this way, the overall statistical analysis will suffer from the insufficient number of samples (due to different practical limitations in the design of the experiments) processed in each individual experiment, which leads to high false-negative rates. Robust statistical parameters that could lead to robust inference can only be estimated from a large population of samples. In this context, combining datasets from different sources by efficiently removing the unwanted variation (e.g. the batch effect), will result in larger datasets which will provide more statistical power for the subsequent analysis by a more robust estimation of the different statistical parameters required. Nevertheless, the reader should also be aware that combining datasets using inefficient methods could also result in misleading findings [12, 26].

The literature dedicated to this topic is constantly increasing, but the different methods are not always described or validated in a uniform manner. As a consequence, the reader has often difficulties in understanding or in identifying the differences and similarities between various methods, and so the choice of the most appropriate method is not straightforward. A comparative study between these methods is out of the scope of this article, but several studies exclusively dedicated to this purpose already exist in the literature [16, 26, 27]. Moreover, each new proposed method is systematically compared with a number of existing ones [10, 28]. Our aim in this survey is to provide a complete picture of batch effect removal methods for integrative analysis of MAGE data together with the available evaluation tools in a unified and complete framework in order to reveal their similarities, their strong points and their weaknesses.

The roadmap of this article is as follows: in the next section we reveal the main characteristics of the so-called batch effect by reviewing several proposed definitions found in the literature; we also provide a list of the main sources of variations susceptible to introduce batch effects. Section 'Integrating microarrays by removing batch effects' is a review of existing methods proposed for batch effect removal. In section 'On the evaluation of batch removal methods', we review and describe several validation tools used in comparative studies to evaluate the performances of different methods: visualization tools and quantitative evaluation measures. In section 'Final comments and recommendation', we aim to resume the information synthesized in the previous two sections and also to provide some basic recommendations for users, while section 'Conclusion' is dedicated to authors' concluding remarks.

THE BATCH EFFECT: SOME DEFINITIONS AND POTENTIAL SOURCES

Providing a complete and unambiguous definition of the so-called batch effect is a challenging task, especially because its origins and the way it manifests in the data are not completely known or not recorded. This is the reason why here we enumerate several definitions as found in the literature. According to

these definitions, the batch effect is defined as one of the following:

Definition 1:
the uncontrollable errors unrelated to the biological variation [28].

Definition 2:
the cumulative errors introduced by time and place-dependent experimental variations [16].

Definition 3:
sub-groups of measurements that have qualitatively different behaviour across conditions and are unrelated to the biological or scientific variables in a study [12].

Definition 4:
systematic differences between the measurements of different batches of experiments [26].

Definition 5:
systematic technical differences when samples are processed and measured in different batches [29].

We identify two complementary characteristics of the batch effects: first which makes the distinction between the batch effects and the biological information (Definitions 1 and 3) and second which generically reveals the sources of batch effects (Definitions 2, 4 and 5). We provide a more general definition of the batch effects by combining the two main ideas that derive from the definitions mentioned above, as follows:

Definition 6:
the batch effect represents the systematic technical differences when samples are processed and measured in different batches and which are unrelated to any biological variation recorded during the MAGE experiment.

Here, the term batch denotes a collection of microarrays (or samples) processed at the same site over a short period of time using the same platform and under approximatively identical conditions, as mentioned in [16].

Noise, batch effect and bias

It is important here to bring into the light the distinction between these terms that are sometimes not properly used and which could cause confusion for the inexperienced reader. We will synthetically resume the main differences as described in [29]. The term 'noise or expression heterogeneity' denotes the effect of 'technical components which are not part of the system under investigation but which, if they enter the system, lead to variability in the experimental outcomes'. The main difference between 'noise' and 'batch effect' is the systematic nature of the latest, as shown in Definition 5. The term 'bias' has a wider meaning which includes not only technical but also other confounding factors (confounding factors – also known as distorting factors – represent variables or factors that distort the observed association between the biological variation of interest and the conducted study.) It is defined as 'unintentional, systematic association of some characteristic with a group in a way that distorts a comparison with another group'. For more in-depth readings on this matter, we invite the reader to consult [29].

Potential sources

The batch effect originates from various sources. Basically, at each step of a MAGE experiment, a number of potential sources are susceptible to generate batch effects. There are several works in which the authors focused their efforts in identifying and explaining the potential sources of batch effects [25, 26, 30]. A systematic and very comprehensive description of the origins and the meaning of the various batch effects can be found in [25]. Here we will only list the potential sources of batch effects and the stage where they appear during the MAGE experiments. As a general accepted rule, MAGE experiments can be summarized in five stages: growing the organism, tissue sampling, RNA processing, hybridization and data extraction, and different sources of batch effects can affect the outcome of the MAGE experiments as shown in Figure 1.

For more details and complete explanations on the above mentioned sources of batch effects, we invite the reader to consult [25], where this subject has been addressed in detail. Besides the sources presented in Figure 1, the following are also mentioned in the literature: the different sites or laboratories where the MAGE experiment has been performed [26] or the blemishes due to dust, glass flaws, uneven distribution of fluids or surface coatings [30].

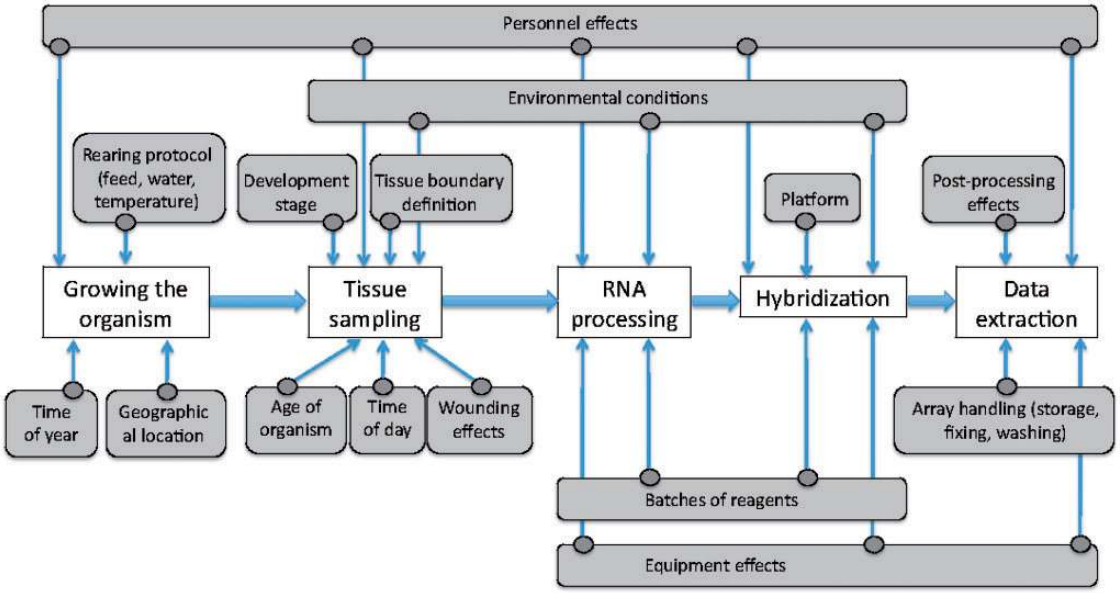

Figure 1: A visualization of batch effect sources at each stage of a MAGE experiment; the grey boxes represent the potential sources of batch effects affecting the different steps of a MAGE experiment, illustrated by the white boxes.

INTEGRATING MICROARRAYS BY REMOVING BATCH EFFECTS

The problem of integrating MAGE data from different experiments consists in combining a number (two or more) of different gene expression datasets in a single dataset which can be used as such for downstream analysis (differential expression analysis, disease prediction or disease discovery). Within this section, we review the methodology for integrating MAGE datasets by removing batch effects.

We start by introducing the notations that will be used in the remainder of this article. All gene expression values from all samples belonging to the same batch X (Y), as defined above, can be represented by a matrix $X^{m\times n}$ ($Y^{m'\times n'}$) where each column represents a sample and each row represents a gene, and x_{ij} (y_{ij}) represents the expression value of gene i in sample j of batch X (Y). We assume that the gene expression data have been log-transformed and preprocessed (using either MAS5 [31], RMA [32], fRMA [33] for Affymetrix platforms or the preprocessing tools provided by 'lumi' Bioconductor package for Illumina platforms [34]) for background correction, normalization and summarization. We stress on the fact that, even though initially designed to remove any technical source of variation from the data, the normalization step performed by either of the preprocessing methods mentioned earlier (excluding fRMA) are ineffective in removing the batch effects, especially when combining data from different platforms[12, 35]. The reason why is because the normalization steps take into account only few sources of batch effects unlike the more specialized methods for batch effect removal. When these sources of variation become important, the normalization steps used by the different preprocessing tools show their limitations. In Table 1, we give an overview of the notation used throughout this article.

General assumptions

In general, it is assumed that the batch effect comes in either (or both) multiplicative or additive form. In the log transformed data, these effects are both represented as (an) additive term(s). All batch effect removal methods assume that measured expression values of gene i in sample j of batch X can be expressed in a general form as follows:

$$x_{ij} = x'_{ij} + b_{ij}^X + \varepsilon_{ij}^X \quad (1)$$

with x'_{ij} the actual gene expression, b_{ij}^X the batch effect term and ε_{ij}^X represents noise. The term x'_{ij} is the value of interest as this represents the abundance of mRNA of that gene in a particular sample. Different batch effect identification and removal methods refine this general description by splitting b_{ij}^X in different terms, or by adding terms that are

Table 1: Overview of unified annotations used in the remainder of the manuscript

Annotation	Explanation
$X^{m\times n}$, $Y^{m'\times n'}$	MAGE dataset (batch) with m (m') genes and n (n') samples
$\hat{X}^{m\times n}$, $\hat{Y}^{m'\times n'}$	MAGE dataset (batch) with m (m') genes and n (n') samples after batch effect removal
x_{ij}, y_{ij}	Expression of gene i in sample j in corresponding batch
$\hat{x}_{ij}$, $\hat{y}_{ij}$	Expression of gene i in sample j in corresponding batch after batch effect removal
$\overline{x}_i$, $\overline{y}_i$	Mean expression of gene i in corresponding batch
σ_{x_i}, σ_{y_i}	Standard deviation of gene i in corresponding batch
x^r_{ij}, y^r_{ij}	Expression of gene i in j'-th reference sample in corresponding batch
b^X_{ij}, b^Y_{ij}	Bias for gene i in sample j of corresponding batch
ε^X_{ij}, ε^Y_{ij}	Noise in gene i of sample j of corresponding batch
γ^X_i, γ^Y_i	Additive gene i specific bias for corresponding batch
δ^X_i, δ^Y_i	Multiplicative gene i specific bias for corresponding batch

specific for known covariates (a covariate is a variable that is possibly predictive of the outcome under study, such as the condition of the experiments or biological information such as male/female.). Within this general description, the term b^X_{ij} can indicate the batch effect related to any of the sources mentioned in 'The batch effect' Section.

When combining the expression values from multiple studies, it is assumed that all studies have the same distribution of samples for each biological variable of interest. Therefore, it is impossible to remove batch effects between two studies for which one study only contains control samples and another only diseased samples, when the disease is a biological variable of interest, unless other prior information is recorded. Moreover, it is impossible to combine datasets from studies containing samples from different organisms.

Assume two MAGE datasets $X^{m\times n}$ and $Y^{m'\times n'}$ containing data with the same distribution relative to the biological variable of interest, then following Equation (1) their expression values can be described as:

$$x_{ij} = x'_{ij} + b^X_{ij} + \varepsilon^X_{ij}, \tag{2}$$

$$y_{ij} = y'_{ij} + b^Y_{ij} + \varepsilon^Y_{ij} \quad . \tag{3}$$

In order to combine the samples from both $X^{m\times n}$ and $Y^{m'\times n'}$ in such a way they can be used together for downstream analysis, the influence of the batch specific terms b^X_{ij} and b^Y_{ij} needs to be removed. This can be done by either attempting to remove the study specific batch effect b^X_{ij} and b^Y_{ij} from the corresponding batches or by adjusting x_{ij} and y_{ij} in such a way that the two datasets become comparable.

Methods

There are two main approaches for removing the batch effects: location-scale (LS) methods and matrix-factorization (MF) methods (Figure 2). LS methods assume a model for the location (mean) and/or scale (variance) of the data within the batches and proceeds to adjust the batches in order to agree with these models. MF techniques assume that the variation in the data corresponding to batch effects is independent on the variation corresponding to the biological variable of interest and it can be captured in a small set of factors which can be estimated through some matrix factorization methods. The strategy adopted by these methods is to identify and remove the influence of these factors. A smaller group of valuable methods for batch effect removal is based on data discretization. According to these methods, the values for each gene are mapped on a certain level of expression, for instance the so-called barcodes assign 1 for expressed and 0 for unexpressed genes. In the following section, we describe the batch effect removal methods from Figure 2, individually.

Location-scale methods

The main idea behind LS methods is to transform the data from each batch to have similar (equal) mean and/or variance for each gene. It is assumed that these transformations, while trivially making data more comparable, do not remove any biological signal of interest.

Batch mean-centering. Assuming the prevalence of multiplicative systematic batch effects, batch mean-centering (BMC) was introduced in [13]. This simple method transforms the data by

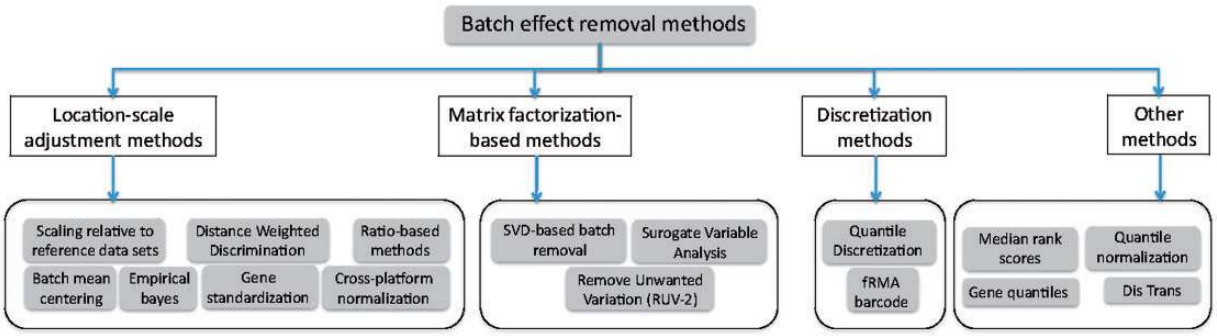

Figure 2: A basic taxonomy of batch effect removal methods.

subtracting the mean of each gene over all samples (per batch) from its observed expression value, such that the mean for each gene becomes zero. BMC assumes that in the general expression in Equation (1), b_{ij}^X represents the multiplicative gene-specific batch effect.

$$\hat{x}_{ij} = x_{ij} - \overline{x}_i, \tag{4}$$

$$\hat{y}_{ij} = y_{ij} - \overline{y}_i. \tag{5}$$

Gene standardization. Gene-wise standardization [36] transforms all genes to have 0 mean and standard deviation (SD) 1 by subtracting the mean and dividing by the SD of each gene over all samples within a batch. A Z-score standardization is used for this purpose. Similar to BMC, it is assumed that in the general expression in Equation (1), the b_{ij}^X represents the multiplicative gene-specific batch effect. The pure (batch effect free) gene expression values are obtained as follows:

$$\hat{x}_{ij} = \frac{x_{ij} - \overline{x}_i}{\sigma_{x_i}}, \tag{6}$$

$$\hat{y}_{ij} = \frac{y_{ij} - \overline{y}_i}{\sigma_{y_i}}. \tag{7}$$

Ratio-based methods. Ratio-based methods [26] scale the expression value of each gene in each sample based on a (set of) reference sample(s) in each batch. If there is more than one reference sample, the arithmetic or geometric mean value of the expression values in the reference samples can be used (it is also possible to use a universal set of reference samples [37].). We denote by x_{il}^r (y_{il}^r) the value of the i*th* gene in the l*th* reference sample in batch X (Y). It is assumed that the genes in the reference samples are subjected to the same batch effect as in the rest of the samples and therefore the term b_{ij}^X in Equation (1) will be removed by subtracting the mean of each gene of the reference samples of the corresponding batch. Assuming k (k') reference samples in batch X (Y), the following two methods are proposed:

Arithmetic mean ratio-based method (Ratio-A):

$$\hat{x}_{ij} = x_{ij} - \frac{1}{k}\sum_{l=1}^{k} x_{il}^r, \tag{8}$$

$$\hat{y}_{ij} = y_{ij} - \frac{1}{k'}\sum_{l=1}^{k'} y_{il}^r. \tag{9}$$

Geometric mean ratio-based method (Ratio-G):

$$\hat{x}_{ij} = x_{ij} - k\sqrt{\prod_{l=1}^{k} x_{il}^r}, \tag{10}$$

$$\hat{y}_{ij} = y_{ij} - k'\sqrt{\prod_{l=1}^{k'} y_{il}^r}. \tag{11}$$

The geometric mean has the benefit that it is less sensitive to outliers. Instead of using the mean, the median could also be used.

Scaling relative to reference dataset. In [38], the authors propose to change the distribution of a gene based on the distribution of that same gene in a reference dataset. Samples are grouped by their biological variable of interest. Assume without loss of generality that $X^{m\times n}$ is the data to be adjusted and $Y^{m'\times n'}$ the reference data. Furthermore, assuming that the samples of each dataset are divided into k categories such that x_{ij}^c is the expression value of gene i in the j*th*

sample belonging to category c in batch X, the batch effect adjusted data are derived as follows:

$$\hat{x}_{ij}^{c} = x_{ij}^{c}\frac{\sigma_{\gamma_i^c}}{\sigma_{x_i^c}} - \left(\overline{x_i^c}\frac{\sigma_{\gamma_i^c}}{\sigma_{x_i^c}} - \overline{\gamma_i^c}\right) \tag{12}$$

$$\hat{\gamma}_{ij}^{c} = \gamma_{ij}^{c}, \tag{13}$$

with c representing the category, while $\overline{x_i^c}$ ($\overline{\gamma_i^c}$) and $\sigma_{x_i^c}$ ($\sigma_{\gamma_i^c}$) are the means and the SDs of gene i in all X (Y) samples belonging to category c, respectively.

Empirical Bayes method. Empirical Bayes (EB) [14] (also known as Extended Johnson-Li-Rabinovich or COMBAT) is a method using estimations for the LS parameters (mean and variance) for each gene. The parameters are estimated by pooling information from multiple genes with similar expression characteristics in each batch. There exist both a parametric and non-parametric approach; we give a concise explanation, and details can be found in the original publication.

It is assumed that measured gene expression values of gene i in sample j in each batch can be expressed as a specialization of Equation (1) as:

$$x_{ij} = \alpha_i + \mathbf{C}\beta_i + \gamma_i^X + \delta_i^X\varepsilon_{ij}^X, \tag{14}$$

where α_i is the gene expression not related to any known covariates, $\mathbf{C}$ is a design matrix for sample conditions (known covariates), β_i is the vector of regression coefficients corresponding to $\mathbf{C}$, γ_i^X and δ_i^X are the additive and multiplicative batch effects for gene i, respectively, and ε_{ij}^X are noise terms. ε_{ij}^X are assumed to follow a normal distribution with mean zero and variance σ_i^2. The first step in EB is to standardize the data using estimates $\tilde{\alpha}_i$, $\tilde{\beta}_i$, $\tilde{\delta}_i^X$ and $\tilde{\sigma}_i^2$ for the corresponding variables. The standardized gene expression z_{ij} is assumed to be normally distributed according to $N(\gamma_i^X, (\delta_i^X)^2)$ and is given by

$$z_{ij} = \frac{x_{ij} - \tilde{\alpha}_i - \mathbf{C}\tilde{\beta}_i}{\tilde{\sigma}_i^X}. \tag{15}$$

The batch effect adjusted data are given by

$$\hat{x}_{ij} = \frac{\tilde{\sigma}_i}{\hat{\delta}_i^{X*}}(z_{ij} - \hat{\gamma}_i^{X*}) + \tilde{\alpha}_i + \mathbf{C}\tilde{\beta}_i, \tag{16}$$

$$\hat{y}_{ij} = \frac{\tilde{\sigma}_i}{\hat{\delta}_i^{X*}}(z\prime_{ij} - \hat{\gamma}_i^{X*}) + \tilde{\alpha}_i + \mathbf{C}\tilde{\beta}_i, \tag{17}$$

where $\hat{\gamma}_i^{X*}$ and $\hat{\delta}_i^{X*}$ are estimates of batch effect parameters in Equation (14) estimated using parametric or non-parametric empirical priors. In case of parametric priors, it is assumed that $\gamma_i^X \sim N(\gamma^X, (\tau^X)^2)$ and $(\delta_i^X)^2 \sim \mathit{InverseGamma}(\lambda^X, \theta^X)$, where γ^X, $(\tau^X)^2$, λ^X and θ^X are estimated empirically. Equivalent properties hold for the terms of $\hat{\gamma}_{ij}$.

Cross-platform normalization (XPN). The basic idea behind the cross-platform normalization [10] approach is to identify homogeneous blocks (clusters) of gene and samples in both studies that have similar expression characteristics. In XPN, a gene measurement within one such block can be considered as a scaled and shifted block mean, where both scaling and shifting are dependent on the gene i and sample j. For a MAGE dataset, gene i and sample j, the recorded gene expression is expressed as a specialization of Equation (1) by

$$x_{ij} = A^X_{\alpha^*(i),\beta^*(j)}b_i^X + c_i^X + \sigma_i^X\varepsilon_{ij}^X, \tag{18}$$

where $A^X_{\alpha^*(i),\beta^*(j)}$ is a block mean and b_i^X and c_i^X represent gene and platform specific sensitivity and offset parameters, respectively. The functions $\alpha^*()$ and $\beta^*()$ map a specific gene measurement in a sample to their corresponding multi-platform cluster. The noise variables ε_{ij}^X are assumed to be independent and normally distributed. Using maximum likelihood methods estimates for the parameters in Equation (18) ($\tilde{A}^X_{ij}$, $\tilde{b}_i^X$, $\tilde{c}_i^X$ and $\tilde{\sigma}_i^X$) are obtained for each batch. Common model parameters ($\hat{A}_{ij}$, $\hat{b}_i$, $\hat{c}_i$ and $\hat{\sigma}_i$) were calculated as weighted averages of these batch-specific estimates. Subsequently, the batch effect adjusted data are given by

$$\hat{x}_{ij} = \hat{A}_{\alpha^*(i),\beta^*(j)}\hat{b}_i + \hat{c}_i + \hat{\sigma}_i\left(\frac{x_{ij} - \tilde{A}^X_{\alpha^*(i),\beta^*(j)}\tilde{b}^X_{x_i} - \tilde{c}_i^X}{\tilde{\sigma}_i^X}\right), \tag{19}$$

$$\hat{y}_{ij} = \hat{A}_{\alpha^*(i),\beta^*(j)}\hat{b}_i + \hat{c}_i + \hat{\sigma}_i\left(\frac{y_{ij} - \tilde{A}^Y_{\alpha^*(i),\beta^*(j)}\tilde{b}^Y_{x_i} - \tilde{c}_i^Y}{\tilde{\sigma}_i^Y}\right). \tag{20}$$

Distance-weighted discrimination. Distance-weighted discrimination (DWD) [11], an adaptation of Support Vector Machines (SVM) [39], can be used for batch effect removal as follows. As a starting point, samples from a single batch are regarded as belonging to a specific class and DWD is used as a classification algorithm by finding the optimal hyperplane $w \times x + b = 0$ separating samples from the different classes (batches), with w the normal vector of

the hyperplane. Next, the samples in each batch are projected in the direction of the normal vector to this hyperplane by calculating the mean distance from all samples in each batch to the hyperplane ($\overline{d^X}$ and $\overline{d^Y}$) and then subtracting the normal vector to this plane multiplied by the corresponding mean distance.

$$\hat{x}_{ij} = x_{ij} - \overline{d^X} w_i, \tag{21}$$

$$\hat{y}_{ij} = y_{ij} - \overline{d^Y} w_i. \tag{22}$$

Matrix factorization-based methods

The idea behind these methods resides in the observation that 'the most important source of differentially expression is nearly always across batches rather than across biological groups' [12]. As a consequence, the most important source of variation also is 'nearly always' associated with batches. Based on this fact, these methods rely on the following strategy:

(i) Perform matrix factorization of the input data matrix (which is in general obtained by sample-wise concatenating of the datasets to be combined); the matrix factorization is usually performed using either singular value decomposition (SVD) [40] or PCA [41], such that the first factor has the highest possible variance (which is associated with batch effects).
(ii) Remove the factors associate with batch effects and reconstruct back the batch effect adjusted dataset.

In the discussion below, we assume that we wish to combine data from two batches $X^{m\times n}$ and $Y^{m'\times n'}$ and denote by $C^{m''\times n''} = [X^{m''\times n} \quad Y^{m''\times n'}]$ the sample-wise concatenation over common genes of the studies, with m'' the number common genes between X and Y and $n'' = n + n'$.

The last two methods discussed in this section do not straightforward return an adjusted data matrix, but do identify factors associated with batch effects by using matrix factorization techniques. These methods can be used for batch effect removal in two ways: (i) combining them with another batch effect removal method, for instance, COMBAT including the identified batch effects as covariate and (ii) reconstructing the data after removing factors identified as being associated with batch effects. Similar to SVD-based batch effect removal, both these approaches assume that the input data are obtained as a sample-wise concatenation of the different MAGE datasets.

Singular value decomposition-based batch effect removal. Singular value decomposition [40] can be used to adjust for batch effects by factorizing the input gene expression data matrix and then reconstructing it while filtering out those factors that are associated with the batch effect. In a first instance, the matrix $C^{m''\times n''}$ (sample-wise concatenation over common genes) is factorized using SVD as follows:

$$C^{m''\times n''} = U^{m''\times n''} \Sigma^{n''\times n''} (V^{n''\times n''})^T, \tag{23}$$

where $C^{m''\times n''} = [X^{m''\times n} Y^{m''\times n'}]$, m'' is the number of common genes between X and Y, $n'' = n + n'$ is the total number of samples in X and Y, while the columns of $U^{m''\times n''}$ and the rows of $(V^{n''\times n''})^T$ form orthonormal basis for the samples (eigensamples)/genes (eigengenes), respectively. The matrix $\Sigma^{n''\times n''}$ is a diagonal matrix containing the singular values ($s_1 \geq \ldots \geq s_{n''} \geq 0$). The reconstruction of the data, with the batch effect removed, can be done by removing those components in the corresponding matrices that are believed to map to the batch effect:

$$\hat{C}^{m''\times n''} = U^{m''\times l} \Sigma^{l\times l} (V^{n''\times l})^T, \tag{24}$$

with $l \leq n''$ and $U^{m''\times l}$, $\Sigma^{l\times l}$ and $(V^{n''\times l})^T$ representing the same matrices with the rows (columns) corresponding to the components mapping to the batch effect removed. As an alternative matrix factorization method, PCA [41] can be also used.

Surrogate variable analysis. In [42], the assumption made is that it is possible to identify the signal in $C^{n''\times n''}$ due to the biological variance of interest and obtain the residuals $R^{m''\times n''}$ after the removal of this signal. These residuals are assumed to contain the unwanted variation caused by batch effects. In order to remove this unwanted variation, a matrix factorization method (e.g. singular value decomposition (SVD)) is then applied on the residuals. The main variation in the residuals is used as factors to be adjusted for in downstream analysis. This is done by estimating surrogate variables representing the unknown confounding effects by iteratively weighting a subset of the factors identified in the decomposition. For details, the reader is referred to [42].

Remove unwanted variation, 2-step (RUV-2). In [28], a similar method for batch effect removal is proposed which makes use of a set of control genes to identify

the factors associated with the batch effect. It is assumed that the control genes are a priori known to be uncorrelated with the biological factor of interest. They suggested two default sets of genes as control genes: spike-in controls and housekeeping genes. Assume that there are p control genes, then RUV-2 proposes to apply a matrix factorization such as SVD on these genes to identify the components corresponding to the batch effects. So instead of performing SVD on $C^{n'' \times n''}$, it is done on a submatrix $C_c^{p \times n''}$ where the subscript c indicates that only the p control genes are considered as input in this step. Similarly, $U_c^{p \times l}$ and $U_c^{p \times n}$ are the submatrices concerning the control genes of the corresponding matrices in Equation (23).

Based on visual inspection or some variation criteria, the first l components $U_c^{p \times l}$ of the eigensamples $U_c^{p \times n}$ are deemed relevant and are then added as covariates to any type of downstream analysis. When this information is passed on to COMBAT, this can be used to adjust the data for batch effects; another option is to reconstruct the data using the obtained decomposition by removing the first l components.

Discretization methods

Discretization methods aim to transform the expression data into consistently defined categories based on their level of expression. This way merging can be done trivially by concatenating the discretized matrices. A loss of information when discretizing is inevitable, but it has been shown that these methods can sometimes even lead to similar or improved accuracy depending on the type of downstream analysis [43].

Quantile discretization. In [44], they propose a discretization method based on equal frequency binning. The expression values of all arrays are discretized into a fixed number of bins. Equal frequency is imposed by using the quantiles as cut points for the bins. The two central bins with the median value as cut point are merged into one bin yielding one central interval. Next, every expression value is replaced by an integer value corresponding to the bin it falls into; zero is assigned to central bin, all other bins are numbered consecutively beginning with the bins next to the central one, using positive integers for bins containing values above the median and negative integer values for the others.

fRMA barcode. Frozen robust multi-array analysis (fRMA) is an algorithm introduced in [33]. fRMA allows to preprocess individual microarray samples and combine them consistently for analysis. This is done by estimating a reference distribution, to be used for quantile normalization of new individuals, based on a training set of publicly available samples from a diverse population. Estimates of probe-specific effects and variances are also obtained on the same sample set and all information is 'frozen'. For each new array to be preprocessed, background correction is performed similar to the training set and then it is quantile normalized based on the reference distribution. During summarization, batch effects are removed and variances of the gene expressions are estimated by taking into account these probe-specific effects. This way fRMA can be seen as a batch effect removal technique; however, it needs to be noticed that the necessary estimates are platform specific and thus data originating from different microarray platforms can only be consistently combined by using an additional batch effect removal algorithm such as EB [43].

Based on fRMA, a novel algorithm for generating barcodes extending work proposed in [45, 46] was introduced in [47]. Huge sets of samples were collected and normalized using fRMA for several platforms. The distribution of the (non-)expressed observed intensities for each gene is estimated using these normalized sets. Genes are deemed expressed (and their value coded to 1) or unexpressed (and their value coded to 0) according to the following equation:

$$\hat{x}_{ij} = \begin{cases} 1 & \text{if } x_{ij} \geq \mu^{ne} + C \times \sigma^{ne}, \\ 0 & \text{otherwise} \end{cases} \tag{25}$$

where x_{ij} is the normalized intensity of gene i in sample j, C is a user-defined parameter, σ^{ne} is the standard deviation of the non-expressed distribution and μ^{ne} is the mean of the non-expressed distribution. The barcode representation of a sample is a vector of ones and zeros denoting which genes are estimated to be expressed (ones) and unexpressed (zeros).

Other methods

For completeness, we only mention several other less popular techniques that have been used for batch effect removal. Quantile normalization (QN) is more frequently used for normalization at the probe level (see RMA preprocessing algorithm

[32]), but it has been also used explicitly for batch effect removal [48, 49]. Two ideas similar to QN are median rank scores (MRS)[44] and gene quantiles (GQ)[50]. MRS considers one batch reference and all genes are ranked based on their median expression. Genes in each sample in the non-reference batch are also ranked and their value replaced by the corresponding ranked median from the reference. GQ is an extension of MRS that enforces an extra transformation of gene expression values such that the median values for each gene are equal in all batches.

In ref. [51], Distribution Transformation (DisTran) is proposed, where a reference sample is constructed based on a combination of the mean expression of samples having the same biological value of interest; consequently, all samples are transformed to have the same distribution as this constructed reference sample. Afterwards, the samples having one specific value for the biological variable of interest are used as reference samples as in the ratio-based methods. Comparative studies [27] show that methods presented in this section are less efficient that the LS, matrix factorization-based or discretization techniques.

Availability of methods

The different batch effect removal methods are mostly implemented in R and they are available at the locations shown in Table 2. We also developed a specialized R/Bioconductor package for batch effect removal called `inSilicoMerging` (http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html) which gathers several state of the art methods for batch effect removal. Examples of R code showing how to use the different methods implemented in the `InSilicoMerging` R/Bioconductor package can be found in the Supplementary information and further usage details can be found in the dedicated vignette.

ON THE EVALUATION OF BATCH REMOVAL METHODS

Evaluating and validating the results of batch effect removal methods is perhaps as important and difficult as the batch effect removal process itself. Without good and reliable evaluation tools, these methods could result in an even increased distortion of the data, introducing serious errors in the results of any downstream analysis performed. As a general rule, the removal of batch effect should be observed and/or quantified before and after applying a particular method in order to evaluate whether that particular method is effective or not. The difficulty of the batch effect removal evaluation process is in the little amount of information recorded during the MAGE experiments that refer to the sources of the batch effects. Most researchers only mention that as a result of efficiently removing the batch effects, the datasets to be integrated must be comparable. And here, the researchers can use their freedom in choosing the criteria to evaluate how two datasets are comparable.

In this section, we describe the different tools used for the evaluation of the batch effect removal methods as found after surveying the selected literature. We divided the validation tools in two main groups,

Table 2: Availability of implementations: links to the corresponding R packages where available or to software made available by the authors

Method	Availability
`BMC`	http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html
`Gene Standardization`	http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html
`Ratio based methods`	Only basic functionality required
`Scaling relative to reference`	Only basic functionality required
`Empirical Bayes`	http://www.bu.edu/jlab/wp-assets/ComBat/ http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html
`XPN`	https://genome.unc.edu/xpn/ http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html
`DWD`	http://cran.r-project.org/web/packages/DWD/ http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html
`SVD-BR`	http://cran.r-project.org/web/packages/svd/
`SVA`	http://www.bioconductor.org/packages/release/bioc/html/sva.html
`RUV-2`	Example code in ref. [28]

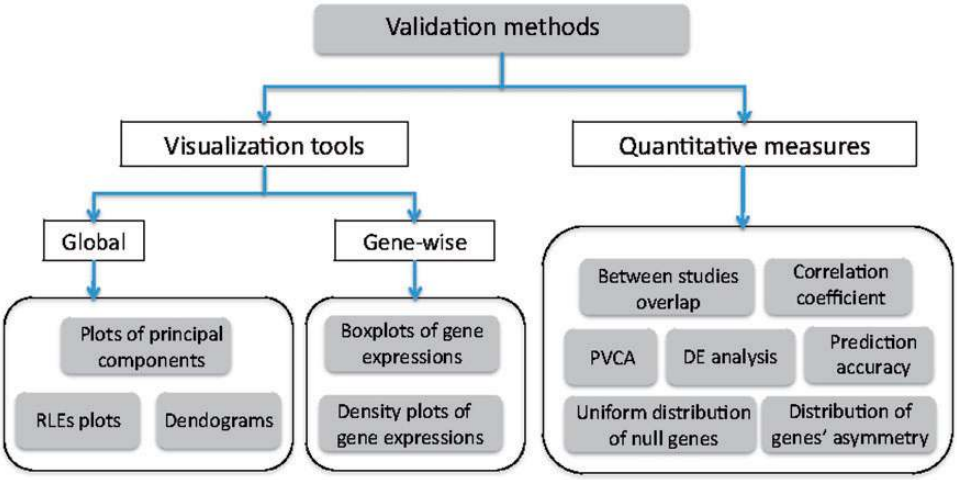

Figure 3: A taxonomy of validation tools to evaluate batch effect removal methods.

visualization tools and quantitative measures. For each group, we identified several methods as shown in Figure 3, which will be further described in this section.

Qualitative evaluation through visualization techniques

The most common and straightforward way to evaluate the effectiveness of batch effect removal methods is by visualization means. The visualization tools provide only a crude approximation of the efficiency of batch effect removal, but they can be used as a first and rapid inspection of the results provided by a method. However, for a more rigorous evaluation, quantitative measures should also be computed to accurately assess the quality of the batch effect removal process.

The visualization tools described below are grouped in two main categories: gene-wise and global visualization tools. Both categories serve to visualize the batch effect between two MAGE datasets but in a different way. The gene-wise tools (e.g. boxplots and density plots of gene expression data) provide a local visualization of the batch effect at the gene level. It is expected that the expression levels of the same gene across two different studies have similar distributions if no batch effect is present, under the assumption that the two studies have the same distribution of samples relative to the biological variable of interest. This assumption is critical for the data integration process because the estimation of the different statistical parameters of the gene expressions for each dataset is highly dependent on the number of samples from each category in the biological variable of interest. If, for example, the biological variable of interest is 'disease' and the available categories are 'normal tissue' and 'tumour', the different datasets to be merged should contain similar proportions of 'normal tissue' and 'tumour' samples. Otherwise the different estimates of the statistical parameters will be different even when there is no batch effect affecting the data; since the estimated statistical parameters are not comparable anymore, the overall analysis is prone to misleading results. The global visualization tools (e.g. dendrograms, plots of the principal components or relative log expression plots) provide a 'big picture' of the presence of the batch effect at study/sample level. According to these plots, it is expected that the samples corresponding to the same category in the biological variable of interest (e.g. all males or all females) will group together regardless of the MAGE experiment they originate from, if no or little batch effect is present, and if the assumption that the two studies have the same distribution of samples relative to the biological variable of interest holds. Even more important, if the samples group by batch, that indicates the presence of batch effects. Hence, the two groups of visualization tools provide complementary information about the batch effects and it is advisable to be jointly used for evaluation. In the following sections, we will discuss several visualization tools and we will also illustrate how they can be used to assess the quality of a batch effect removal method. To illustrate each visualization tool, we use two MAGE lung cancer studies (`GSE19804` and `GSE10072`) which were retrieved through the `inSilicoDb` R/Bioconductor package [4]. All plots have been obtained using the `inSilicoMerging` R/Bioconductor package

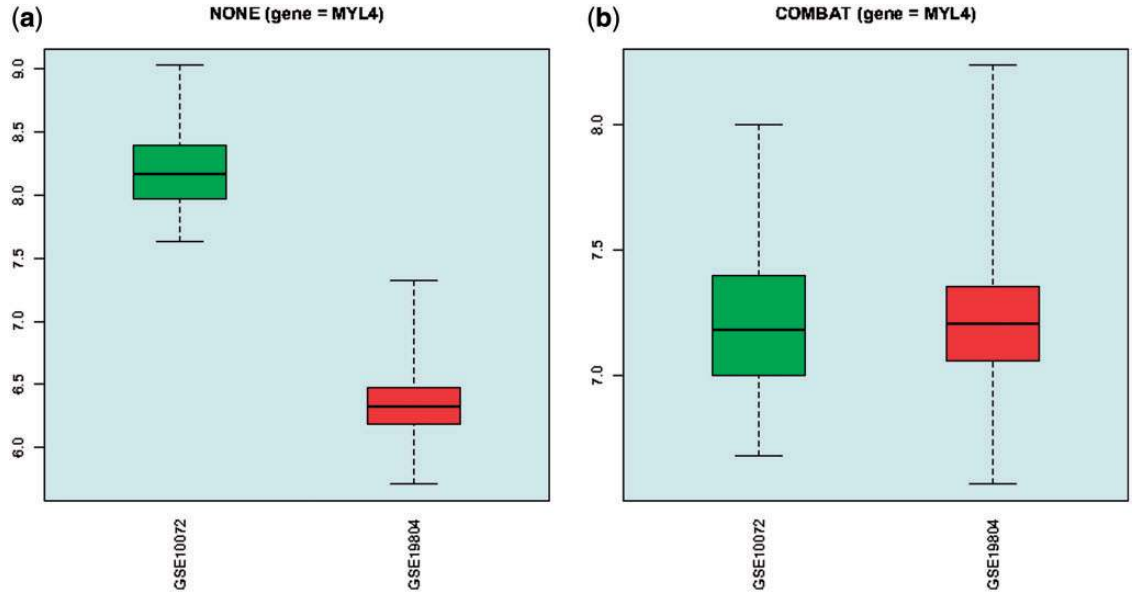

Figure 4: Illustration of boxplots as validation tools for batch effect removal: (a) before and (b) after batch effect removal (using EB method).

(http://www.bioconductor.org/packages/release/bioc/html/inSilicoMerging.html). The examples have been performed using EB (COMBAT) as method for batch effect removal and the code used to generate each figure is available in the Supplementary Information.

Boxplots of gene expression data

In statistics, boxplots are used to graphically summarize the distribution of a population of samples through five parameters: the extreme (minimum and maximum) values as well as the lower, upper and median quartile of the population. In the batch effect removal context, the boxplots are used to compare the gene-wise distribution of two different datasets (originating from two different MAGE experiments) as suggested in [12, 38]. A method is considered as being efficient if the boxplots are located around the same value. An illustration on how boxplots are used to visually inspect batch effect at gene level is provided in Figure 4, where the boxplots of the same gene randomly chosen from the two MAGE studies mentioned above are shown before and after batch effect removal.

Density plot for the gene expression distribution

A different way to visualize the batch effect between two different studies is to visually inspect the distribution of expression values of genes. In [38], this is performed by plotting the distribution of several genes selected randomly from the available pool of genes. The densities are estimated through the Parzen–Rosenblat method [52]. A method is considered as being efficient if the two pdfs are fully overlapped. An illustration on how gene-wise density plots are used to visually inspect batch effect at gene level is provided in Figure 5, where the probability density function (pdf) of the same gene randomly chosen from the two MAGE studies mentioned above are shown before and after batch effect removal.

Dendrograms

In cluster analysis, a dendrogram is a tree representation of the clustering solution obtained through hierarchical clustering. In MAGE analysis, dendrograms are commonly used to cluster either genes or samples in homogeneous groups. In the context of batch effect removal, the dendrodrams are used to visualise how well the samples exhibiting the same biological characteristics, originating from two different studies, cluster together [12, 38]. A common intuition is that samples exhibiting the same biological characteristics should cluster together regardless of the experiment they originate from. If this is not true, it is very likely that there is a bias between the two datasets. Another and probably more important interpretation of dendrograms as validation tools for batch effect removal is that if the samples cluster by study, that indicates the batch effects presence.

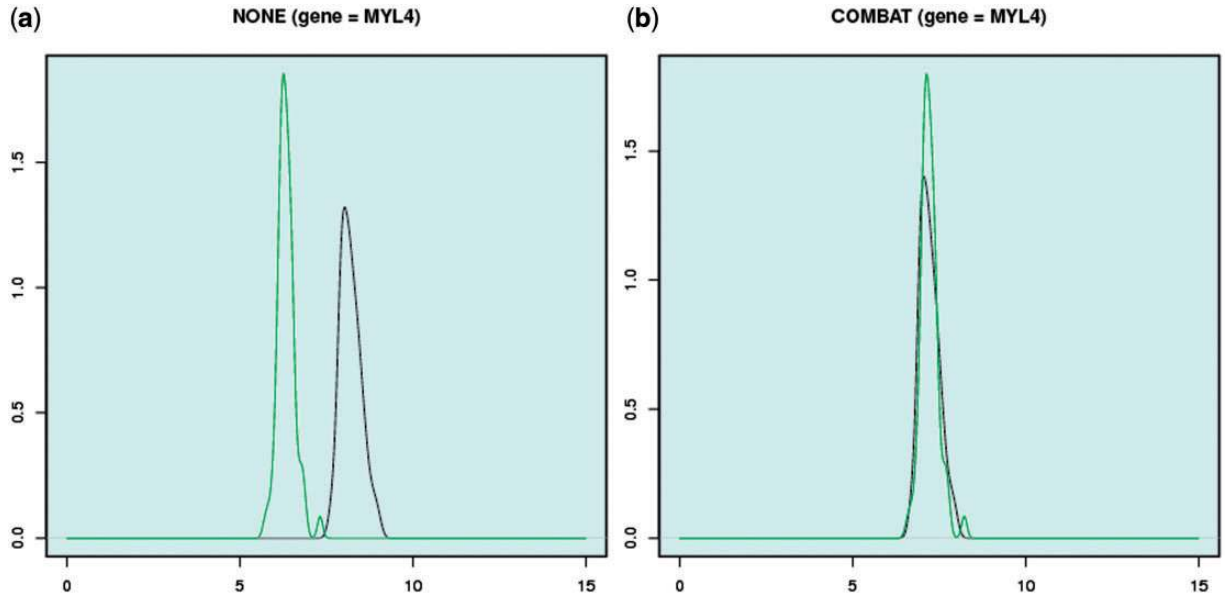

Figure 5: Illustration of gene-wise density plots as validation tools for batch effect removal: (a) before and (b) after batch effect removal (using EB method).

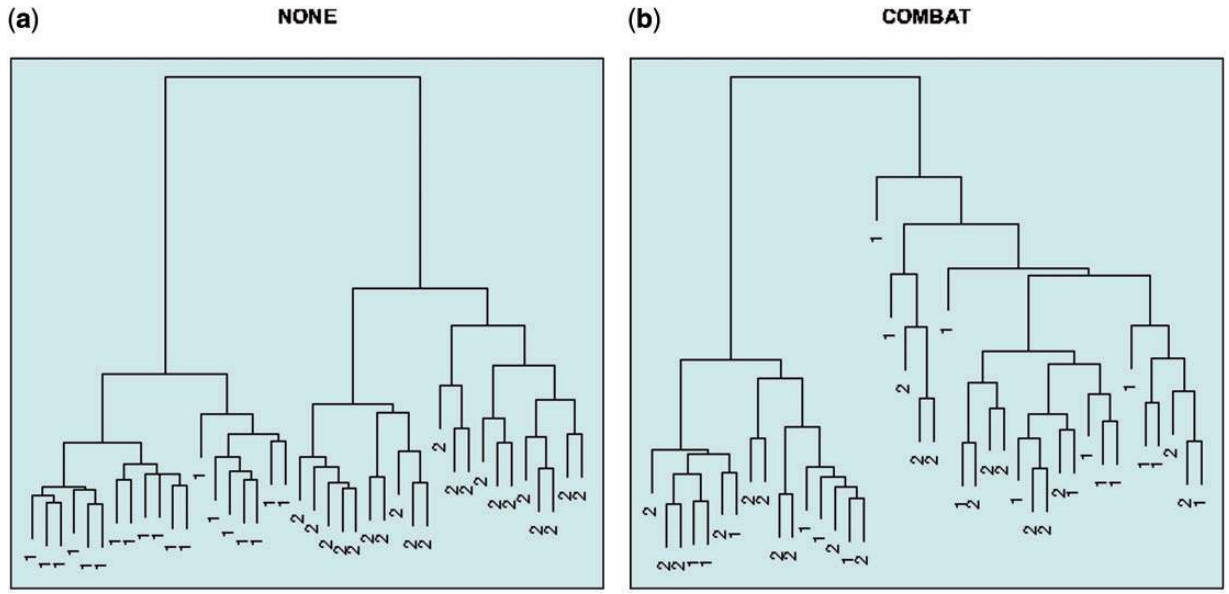

Figure 6: Illustration of using dendrograms as validation tools for batch effect removal: (a) before and (b) after batch effect removal (using EB method). The samples denoted by the same number originate from the same MAGE study.

This can be helpful in situations where the annotations corresponding to the biological characteristics of samples might not be available or might not have a strong influence on gene expression. For visualization reasons, in Figure 6 we only display the dendrogram of 40 samples randomly selected from the two above mentioned MAGE datasets. One can easily notice that the samples cluster by study before batch effect removal (suggesting the presence of batch effect) which is not the case after applying a batch effect removal method (suggesting an overlapping of the two studies and hence the batch effect removal).

Plots of principal components

PCA [41] makes use of a linear orthogonal transformation to map a set of observations of possibly correlated variables into a new set of observations

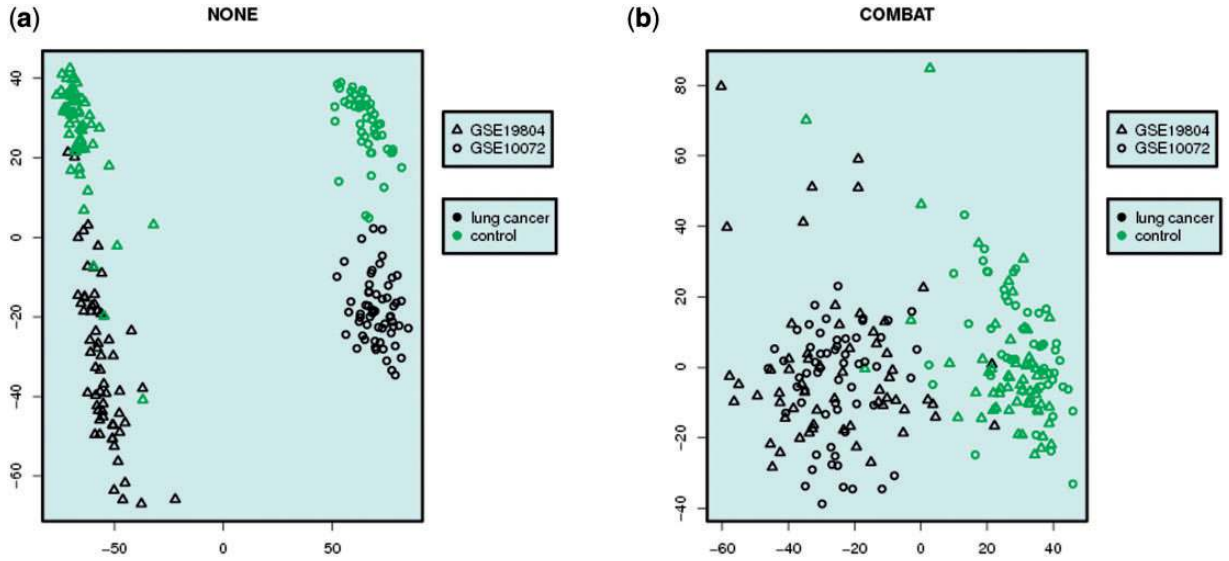

Figure 7: Illustration of PCA plots as validation tools for batch effect removal. Plot of first two principal components: (a) before batch effect removal and (b) after batch effect removal (using EB method).

of uncorrelated variables called 'principal components'. The transformation is defined such that the first principal component captures as much as possible of the variance under the constraint that it is orthogonal on the other principal components. The rest of the principal components are ordered according to the amount of variance captured.

According to [12], 'in gene expression studies, the greatest source of differentially expression is nearly always across batches rather than across biological groups'. This statement is based on the observations made in a number of various studies, e.g. [46]. This is the reason why the plots of the first two principal components are commonly used to visualize the batch effect between two studies [38]. According to these plots, batch effects are present if the samples originating from two different MAGE studies are separated. A method is considered as being efficient if it results in a consistent overlap between the samples in the two studies to be combined. An illustration on how plots of principal components are used to visually inspect batch effect at sample/study level is provided in Figure 7.

Relative log expression plots

The relative log expression (RLE) plots [53] were initially proposed to measure the overall quality of a dataset aiming to identify bad chips [28]. We will briefly explain how the RLE plots are used to visualize the removal of the unwanted variation introduced by batch effects, based on the description in [28]: let us consider a set of n samples, each with m genes and denote with X the log-transformed gene expression matrix of the n samples, where x_{ij} denotes the log expression level of the ith gene on the jth sample. For each gene, the median (over all samples) log expression level is computed; consequently, for each gene on each sample, the deviation from the median log expression level is computed by: x_{ij}—*median* (x_i). Then, for each sample, a boxplot for its m deviations can be displayed, as illustrated in Figure 8. For visualization reasons, we only display the RLE plots of 50 samples randomly selected from the two above mentioned MAGE datasets. For an efficient batch effect removal method, the individual boxplots will be all distributed around 0.

Other visualization techniques

Only for completeness, we will report here two other visualization techniques mentioned in [26], which can be used by the users to visualize the batch effects: correlation heat maps or variance components pie charts.

Quantitative evaluation measures

Quantitative measures provide with a more accurate evaluation of the batch effect removal and they are very effective tools for comparing the results of different methods. Here we briefly describe the most commonly used methods as found in the literature. Excepting the overlap score and the mixture score which provide the same conceptual information

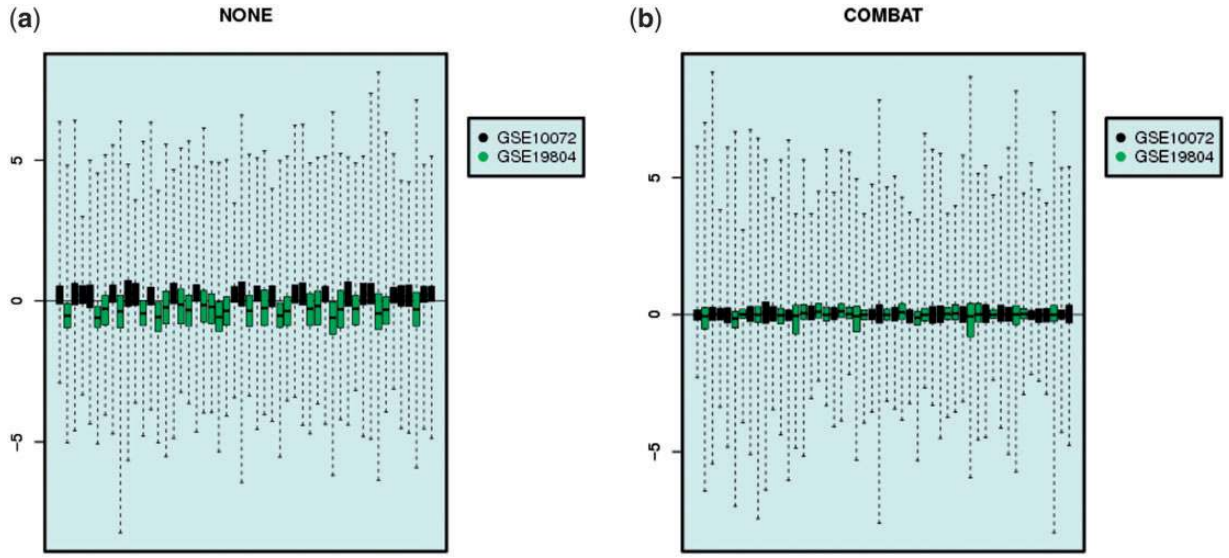

Figure 8: Illustration of using RLE plots as validation tools for batch effect removal: (a) before and (b) after batch effect removal (using EB method).

about the quality of the batch effect removal, all the other methods bring complementary information to assess the effectiveness of a method.

Measuring the overlap between samples from independent studies

This validation strategy is proposed to measure the expected overlap between two independent studies before and after applying the batch effect removal methods [10]. Considering the samples as points in a n dimensional dataset, a method is considered as being effective if it results in a substantial overlap between the samples originating from the two independent studies. The overlap is quantified as follows:

(i) Compute the distance between each sample in the first study and its nearest neighbour in the second study.
(ii) Repeat Step 1 by changing the roles of the studies.
(iii) Average the results in Steps 1 and 2.

The overlap between datasets can be easily visualized in the scatter plot of the first two principal components. Obviously, the higher the overlap, the better the data integration process.

Mixture score

A similar validation approach is proposed in [38, 54]. The 'mixture score' is proposed to evaluate the efficiency of batch effect removal methods by using the idea of k-nearest neighbours ($kNNs$). Assume that C is the sample-wise concatenation over common genes of two datasets X and Y. For each sample in C belonging to X, a ratio is defined between how many of its $kNNs$ belong to X and Y, respectively. The mixture score is defined as follows:

$$MS = \frac{\sum_{x \in X} \#\{y \in Y : y \in kNN(x) \text{ in } C\}}{k \times |X|}. \quad (26)$$

The mixture score is bounded in (0, 1). If both studies have an almost equal amount of samples, values close to extremities mean that the two studies are well separated while values close to 0.5 suggest that the studies are highly overlapped suggesting the removal or the absence of batch effects. If there is an unbalance in the number of samples then the mixture score indicates removal of batch effects if it mimics the ratio between the number of samples.

Comparing the distribution of genes' asymmetry across studies

A simple and efficient way to quantify the results of batch effect removal methods is to compare the distribution of samples' asymmetry before and after batch removal, as proposed in [10]. For a given random variable, a raw approximation of its asymmetry is given by the difference between its mean and median values. However, the 'skewness' is another efficient statistical measure that quantifies the asymmetry of a distribution, and it can be used instead. This index is computed as being the area

between the cumulative density functions (CDFs) of samples' asymmetry, estimated before and after batch effect removal. The CDFs of samples' asymmetry have the same support which approximates the range between the minimum and the maximum values of samples' asymmetry before and after batch effect removal. This index can be defined as follows:

$$bias_{X,\hat{X}} = \sum_{i=a}^{b} (CDF_X(i) - CDF_{\hat{X}}(i)), \quad (27)$$

where X and $\hat{X}$ are gene expression data matrices of samples originating from one or multiple experiments before and after batch effect removal, CDF_X and $CDF_{\hat{X}}$ represent the CPF of samples's asymmetry before or after batch effect removal, while a and b are the minimum, respectively, the maximum values of samples' asymmetry before and after batch effect removal. A method is considered to be efficient if, after batch effect removal, the two CDFs are as similar as possible, and so the index should have a value close to 0.

Principal variance component analysis

A different approach to evaluate the batch effect removal methods is presented in [55, 56]. Principal variance component analysis (PVCA) combines the advantages of two well-known statistical methods for data analysis: PCA [41] and variance component analysis (VCA) [57]. The authors describe it 'as a screening tool to determine the sources of variability' in a dataset 'and to quantify the magnitude of each source of variability, including each batch effect'. The method is summarized in four steps:

- Perform PCA on the gene expression dataset and select the first principal components that retain most of the variability of the data. Since the principal components are all ordered in the decreasing order of their eigenvalues, one will select the first principal components that contain an amount of variation higher than a predefined threshold (e.g. 60–90%)
- Fit a mixed model separately to each selected principal component with all factors of interest as random effects and any undesired factor as fixed effect. In this step, one can use a mixed linear model having the following generic form:

$$A = B\beta + Zu + e, \quad (28)$$

where A denotes a vector of observations, in this case one selected principal component, B is the gene expression data matrix of samples before or after batch effect removal, β is the known fixed-effect parameter vector (or the undesired factor), Z is the design matrix of random effects (or the factors of interest), u is the vector of unknown random effect parameters and e is the unobserved vector of random Gaussian errors. It is assumed that u and e are normally distributed. The goal in this step is to estimate the variance of u and e (σ_u, respectively, σ_e) for all selected principal components. We encourage the reader to consult [56] for more information on the estimation of σ_u and σ_e.

- For each principal component, average the estimated variance components from the previous step (σ_u, respectively, σ_e) with their corresponding eigenvalues as weights.
- Standardize the weighted variance components estimates by dividing them by their sum; in this way, the magnitude of each effect can be represented as a proportion of the total variance.

In order to evaluate the efficiency of a batch effect removal method, the estimation of variance components should be performed before and after batch effect removal. PVCA has been used as a validation method for comparing six batch effect removal methods (which are also discussed in this article) in ref. [16]. More details about the PVCA can be found in [56].

Evaluation through differential expression analysis

Several studies [28, 42, 58] propose to evaluate the effectiveness of the batch effect removal methods in the context of the differentially expression (DE) analysis. It is generally assumed that DE analysis performed on the adjusted dataset should result in a more reproducible list of genes which are differentially expressed. The authors in [28] propose a quality metric to measure the effectiveness of a batch effect removal method. The metric proposed is proportional to the number of 'positive control genes' (genes that are known *a priori* to be truly differentially expressed) found in the top k-ranked genes according to a particular method for differentially expressed genes (DEGs) discovery, see [59] for a recent survey on this topic. A batch effect removal method should be considered as being effective if the number of positive control genes found in the adjusted dataset increases with respect to those found in the original studies. This evaluation approach would be preferred when a high number of positive control genes are available. In other cases where the positive controls

are only a few, the authors suggest to examine the *P*-values of both positive and negative control genes (genes that are known *a priori* as being uncorrelated with the biological variation of interest). In this case, the rule proposed is that a batch effect removal method is considered to be effective if the *P*-value of the positive controls decreases (increasing their statistical significance) while the *P*-value of the negative controls increases. If the *P*-value of the positive controls and the *P*-value of the negative controls decrease or increase in the same time, the method should be considered ineffective. For more details about this evaluation strategy, we invite the reader to consult [28].

If the positive/negative control genes are unknown, the authors in [60] propose an evaluation strategy based on functional enrichment analysis [61] which assesses whether specific cellular functions are overrepresented within a set of significant genes.

In [13], the authors propose a different way to use DE analysis to assess the efficiency of batch effect removal methods, at gene/probe level. The idea is to first identify lists of the most DE genes/probes in the newly combined dataset and to compare those lists with the most DE genes/probes from other single or differently combined datasets. The efficiency of a method is in this case proportional to the number of overlapping probes in the compared lists.

Uniform P*-value distribution of null genes*

A different way to evaluate batch effect removal methods is to asses their effectiveness through significance analysis, as suggested in [42]. The authors in [42] provide a very comprehensive explanation of this evaluation strategy. According to them, a significance analysis is performed 'correctly' if the null distribution is calculated properly; this means that the *P*-values corresponding to null genes or negative control genes are uniformly distributed across (0, 1). Hence, a batch effect removal method will be considered as being effective if the *P*-value distribution of null genes in the adjusted dataset will be distributed as such. The presence of batch effect will result in non-uniform *P*-value distributions of the negative control genes since the batch effect introduces strong dependencies between genes [28]. More details about this evaluation strategy can be found in [28, 42].

However, there are situations where this evaluation method fails in identifying the batch effect in studies affected by heterogeneity. In [42], the authors reveal that for a particular study affected by batch effect, the *P*-value distribution of negative control genes was almost identical with the *P*-value distribution of negative control genes from a unaffected study. This is why it is advisable to use this method with precaution or jointly with another evaluation strategy.

Remark. Both evaluation methods based on the DE analysis and on the *P*-value distribution are subjective to the choice of positive/negative control genes.

Evaluating the prediction performances with respect to the biological annotation of interest

Another expected beneficial consequence of combining two datasets through batch effect removal methods is an increase of the prediction accuracy on the test datasets [10]. Therefore, it is expected that the error rates of various classifiers should significantly decrease if the batch effect has been effectively removed. In [10], the authors evaluated the performance of different methods for batch effect removal in terms of 'classification accuracy', where after adjustment, the first dataset was used for training and the second one for test. The classification has been performed with respect to the biological variation of interest (ER status) mapped into a binary variable.

As the prediction accuracy is dependent on the class prevalence, it can sometimes provide some misleading results. In order to overcome this insufficiency, the authors in [26] proposed to use instead the Matthews correlation coefficient (MCC), defined as follows:

$$MCC = \frac{\mathrm{TP} \times \mathrm{TN} - \mathrm{FP} \times \mathrm{FN}}{\sqrt{(\mathrm{TP}+\mathrm{FP})(\mathrm{TP}+\mathrm{FN})(\mathrm{TN}+\mathrm{FP})(\mathrm{TN}+\mathrm{FN})}}, \tag{29}$$

where TP, FP denotes the true/false positives and TN, FN denotes the true/false negatives.

Correlation coefficient

In the context of batch effect removal, the correlation coefficient is used to observe and to quantify how much the batch effect removal methods affect the data [10, 38]. Note that this evaluation method does not give any clues on how effective a method is, but it is more a way to choose between two different methods that perform similarly according to other evaluation indices. According to [10], in such

situations, the method that least affects the data should be preferred. This index is computed as being the average correlation coefficient between genes before and after removing the batch effect. Low values will indicate that the batch effect removal distorted the initial data, and hence the method will be considered as inefficient. For completeness we will mention another very similar validation index for batch effect removal methods, 'the global integrative correlation', see [10, 62].

FINAL COMMENTS AND RECOMMENDATION

As many researchers already pointed out [12], integrating data from different MAGE experiments is a complicated procedure that requires careful examination of the data and also rigorous evaluation in order to avoid negative effects that could be further introduced by the different methods. As a good practice, it is always advisable to start by visualizing and quantifying the bias between the different datasets to be combined. The choice of the appropriate method for integrating MAGE datasets is conditioned by the number of samples per dataset as well as by the distribution of samples in the datasets to be combined relative to the biological variation of interest. Table 3 summarizes the batch effect removal methods presented in 'Integrating microarrays by removing batch effects' Section according to six general features, which condition or favour the applicability of each method, as follows:

- **Complexity:** this is proportional to the number of parameters of the method. As a natural choice, simple methods like BMC and Gene Standardization should be preferred to more complicated ones if they provide similar results.
- **Minimum number of samples required:** some methods specify a limit number of samples per study in order to be able to be performed correctly [63]. In general, DWD and most factorization methods require more than 25 samples [14] in order to work correctly. The XPN method even fails to run with less than 30 samples per study (based on a comparative study performed in-house – data not provided.). EB is known to be fairly independent on the sample size.
- **Number of datasets:** most methods are able to combine several datasets at once but XPN and DWD are limited to only two. If multiple datasets need to be integrated, these methods have to be called recursively by considering only two datasets at each iteration which gets complicated when combining a large number of datasets.
- **Flexibility:** we denote by 'flexibility', the ability of each method in coping with a high number of covariates (other than the biological variation of interest and the batch information). Some methods like EB and RUV2 can deal with multiple

Table 3: Summary of batch effect removal methods

Method	Complexity	No. of samples	No. of studies	Flexibility	Additional info required	Computational time
BMC	Low	> 25	> 2	Low	No	Low
Gene standardization	Low	> 25	> 2	Low	No	Low
Ratio based methods	Low	> 25	> 2	Low	Yes	Low
Scaling relative to reference	Low	> 25	> 2	Average	No	Low
Empirical Bayes	High	> 5	> 2	High	No	Low
XPN	High	> 30	2	Low	No	High
DWD	Average	> 25	2[a]	Low	No	Average
SVD-BR	Average	> 25	> 2	Average	No	Average
SVA	Average	> 25	> 2	Average	No	Average
RUV-2	Average	> 25	> 2	High	Yes	Average
Quantile discretization	Low	> 25	> 2	Low	No	Low
fRMA Barcode	Low	≥ 1	> 2	Low	No	Low

Note: The complexity is proportional to the number of parameters in the scoring function; the minimum number of samples required is based on statistical assumptions and in-house performed comparative study; for the number of studies we differentiate between the ability to combine 2 or more studies at once; additional info means the requirement of known covariates; flexibility indicates the possibility to add background information such as covariates; computational time is also based on in-house comparative study (data not provided). [a]Recently an extension for multiple datasets was suggested (kDWD), however internally this is an iterative pairwise combination

covariates and are able to remove different sources of bias at once. If such information is available, these methods can provide a more refined adjustment.

- **Additional prior information required:** in contrast to the flexibility, some methods can only be applied if some prior information, other than the batch information are available. `RUV2`, for example, is based on the fact that it takes control genes (genes which are unrelated to the biological information of interest) as input, while `Ratio-based methods` require reference samples. It is not always trivial to obtain this extra information and therefore this is a limitation for these approaches.
- **Computational time:** a last feature to be considered when selecting an appropriate batch removal method is the computational time that is dependent on the number of samples and studies that should be integrated. In a comparative study (data not provided), we found out that `XPN` requires the most computational time, followed by `DWD` and the factorization methods. The other methods performed relatively fast.

The choice of a method for microarray data integration is application dependent. As a general rule, methods with a low complexity also requiring low computational time should be preferred; however, these methods do not always provide the best results. Another important aspect that should be considered in the choice of the most appropriate method is related to the number of samples per dataset available. When data sets with few samples are available, the choice of the appropriate method should be performed according to the specifications in Table 3. Most of the methods are not able to provide accurate results if the number of samples per dataset is lower than the thresholds specified in Table 3. It should also be preferred to use methods with a high flexibility and also methods allowing to incorporate additional prior information (e.g. control genes) when such information is available. The practitioner should also be aware about the difficulties and dangers of combining datasets from various MAGE experiments, as revealed in [12]. First, the sources of variation from one experiment to another are not completely known or they are not all recorded, and this issue hinders the data integration process. The only way to overcome this limitation is to record all possible sources of batch effects during the MAGE expression experiments. Second, the sources of variation responsible for batch effects are sometimes correlated with the biological variables of interest; this means that by removing the batch effects most of the biological variation of interest will also be removed.

CONCLUSION

Integrating data from different MAGE experiments is without any doubt the solution towards reliable large scale analysis of genomic data. It has the potential of unlocking the current limitations related to the relatively weak statistical and generalization power of the results issued from analysing small datasets, which is currently an unsurpassable obstacle in the design of genomic experiments. Combining such datasets is a complex and difficult problem due to its very diverse origin, and so far, there is no general solution to solve it. Moreover, the problem is hard to be completely solved due to the fact that, during experiments, the potential sources of bias are not fully reported and recorded, as stated in [12]. Many researchers are aware of this problem, but unfortunately it is not always adequately incorporated in their preprocessing or analysis workflow. Here we review a wide range of methods proposed to tackle this problem by removing one of the main source of variation between datasets from different MAGE experiments: the so-called batch effects. Without a solid validation framework, the use of any methods proposed to solve the problem would be irrelevant. This is the reason why we dedicated equal attention to the different evaluation techniques of batch effect removal methods. Nevertheless, we ended our survey with a section where we provide the reader with a generic comparison framework of the different methods which reveals the conceptual differences between the methods as well as their advantages and weaknesses. In the end, we are joining and emphasizing the last concluding remark of Leek *et al.* [12]: 'the need to incorporate adjustment for batch effects as standard step in the analysis of high-throughput data'.

Key points

- Integrating gene expression data from different experiments is a tempting and also a difficult challenge for large-scale analysis of genomics data.
- Data integration is hindered by batch effects and efficient methods for batch effect removal are needed for integrative analysis

of MAGE data. Current methods for batch effect removal proposed to integrate MAGE data have been reviewed and grouped in a taxonomy.
- Batch effect removal implies further transformations applied to the gene expression data and they should be carefully applied since they could further introduce or amplify undesired effects in the data. In this context, the quality assessment of the data integration process is crucial. Validation tools for assessing the quality of MAGE data integration process have been objectively discusses and grouped in a taxonomy.

Acknowledgment
The authors thank the Brussels Institute for Research and Innovation (INNOVIRIS) which funded this research and also the referees for their constructive comments.

FUNDING

This work is part of the 'InSilico: The in Silico Wet Lab' project and it has been funded by the Brussels Institute for Research and Innovation (INNOVIRIS).

SUPPLEMENTARY INFORMATION

Here we provide some examples of R code showing how the different methods implemented in the `inSilicoMerging` package can be used in practice. Further information can be found in the associated vignette (http://www.bioconductor.org/packages/release/bioc/vignettes/inSilicoMerging/inst/doc/inSilicoMerging.pdf).

References

1. Rhodes DR, Chinnaiyan AM. Integrative analysis of the cancer transcriptome. *Nat Genet* 2005;**37**:S31–S37.
2. Edgar R, Domrachev M, Lash AE. Gene expression omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res* 2002;**30**(1):207–10.
3. Parkinson H, Sarkans U, Kolesnikov N, *et al*. ArrayExpress update—an archive of microarray and high-throughput sequencing-based functional genomics experiments. *Nucleic Acids Res* 2011;**39**(Database issue):D1002–4.
4. Taminau J, Steenhoff D, Coletta A, *et al*. inSilicoDb: an R/Bioconductor package for accessing human Affymetrix expert-curated datasets from GEO. *Bioinformatics* 2011; **27**(22):3204–5.
5. Ein-Dor L, Zuk O, Domany E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. *Proc Natl Acad Sci USA* 2006;**103**(15):5923–8.
6. Michiels S, Koscielny S, Hill C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. *Lancet* 2005;**365**(9458):488–92.
7. Ma S. *Integrative analysis of cancer genomic data. Durban, South Africa: The 57th Session of the International Statistical Institute*, 2009.
8. Rhodes DR, Yu J, Shanker K, *et al*. Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc Natl Acad Sci USA* 2004;**101**(25):9309–14.
9. Wirapati P, Sotiriou C, Kunkel S, *et al*. Meta-analysis of gene expression profiles in breast cancer: toward a unified understanding of breast cancer subtyping and prognosis signatures. *Breast Cancer Res* 2008;**10**(4):R65.
10. Shabalin AA, Tjelmeland H, Fan C, *et al*. Merging two gene-expression studies via cross-platform normalization. *Bioinformatics* 2008;**24**(9):1154–60.
11. Benito M, Parker J, Du Q, *et al*. Adjustment of systematic microarray data biases. *Bioinformatics* 2004;**20**(1):105–14.
12. Leek JT, Scharpf RB, Bravo HC, *et al*. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nat Rev Genet* 2010;**11**(10):733–9.
13. Sims A, Smethurst G, Hey Y, *et al*. The removal of multiplicative, systematic bias allows integration of breast cancer gene expression datasets—improving meta-analysis and prediction of prognosis. *BMC Medical Genom* 2008;**1**(1):42–56.
14. Johnson WE, Li C, Rabinovic A. Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics* 2007;**8**(1):118–27.
15. Demetrashvili N, Kron K, Pethe V, *et al*. How to deal with batch effect in sequential microarray experiments? *Mol Inform* 2010;**29**(5):387–93.
16. Chen C, Grennan K, Badner J, *et al*. Removing batch effects in analysis of expression microarray data: an evaluation of six batch adjustment methods. *PLoS One* 2011;**6**(2):e17238.
17. Lukk M, Kapushesky M, Nikkila J, *et al*. A global map of human gene expression. *Nat Biotechnol* 2010;**28**(4):322–4. ISSN 1087-0156.
18. Chu TM, Deng S, Wolfinger R, *et al*. Cross-site comparison of gene expression data reveals high similarity. *Environ Health Perspect* 2004;**112**(4):449–455.
19. Dobbin KK, Beer DG, Meyerson M, *et al*. Interlaboratory comparability study of cancer gene expression analysis using oligonucleotide microarrays. *Clin Cancer Res* 2005;**11**(2): 565–72.
20. Zakharkin S, Kim K, Mehta T, *et al*. Sources of variation in Affymetrix microarray experiments. *BMC Bioinformatics* 2005;**6**(1):214–25.
21. Han ES, Wu Y, McCarter R, *et al*. Reproducibility, sources of variability, pooling, and sample size: important considerations for the design of high-density oligonucleotide array experiments. *J Gerontol A Biol Sci Med Sci* 2004; **59**(4):306–15.
22. Breit S, Nees M, Schaefer U, *et al*. Impact of pre-analytical handling on bone marrow mRNA gene expression. *Br J Haematol* 2004;**126**(2):231–43.
23. Bakay M, Chen Y-W, Borup R, *et al*. Sources of variability and effect of experimental approach on expression profiling data interpretation. *BMC Bioinformatics* 2002;**3**(1):4–16.
24. Brown JS, Kuhn D, Wisser R, *et al*. Quantification of sources of variation and accuracy of sequence discrimination in a replicated microarray experiment. *Biotechniques* 2004; **36**(2):324–32.
25. Scherer A. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. John Wiley and Sons, 2009. Chapter 4.
26. Luo J, Schumacher M, Scherer A, *et al*. A comparison of batch effect removal methods for enhancement of

prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics J* 2010;**10**(4):278–91.

27. Rudy J, Valafar F. Empirical comparison of cross-platform normalization methods for gene expression data. *BMC Bioinformatics* 2011;**12**(1):467–89.
28. Gagnon-Bartsch JA, Speed TP. Using control genes to correct for unwanted variation in microarray data. *Biostatistics* 2011.
29. Scherer A. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. John Wiley and Sons, 2009. Chapter 1.
30. Suarez-Farinas M, Pellegrino M, Wittkowski K, *et al*. Harshlight: a 'corrective make-up' program for microarray chips. *BMC Bioinformatics* 2005;**6**(1):294–305.
31. Hubbell E, Liu W-M, Mei R. Robust estimators for expression analysis. *Bioinformatics* 2002;**18**(12):1585–92.
32. Irizarry RA, Hobbs B, Collin F, *et al*. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* 2003;**4**(2):249–64.
33. McCall MN, Bolstad BM, Irizarry RA. Frozen robust multiarray analysis (fRMA). *Biostatistics* 2010;**11**(2):242–53. doi: 10.1093/biostatistics/kxp059.
34. Du P, Kibbe WA, Lin SM. lumi: a pipeline for processing Illumina microarray. *Bioinformatics* 2008;**24**(13):1547–8.
35. Sun Z, Chai H, Wu Y, *et al*. Batch effect correction for genome-wide methylation data with Illumina Infinium platform. *BMC Medical Genom* 2011;**4**(1):84–96.
36. Li C, Wong WH. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc Natl Acad Sci USA* 2001;**98**(1):31–36.
37. Novoradovskaya N, Whitfield M, Basehore L, *et al*. Universal reference RNA as a standard for microarray experi- ments. *BMC Genom* 2004;**5**(1):20–33.
38. Kim K-Y, Kim SH, Ki DH, *et al*. An attempt for combining microarray data sets by adjusting gene expressions. *Cancer Res Treat* 2007;**39**(2):74–81.
39. Cristianini N, Shawe-Taylor J. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. New York, NY, USA: Cambridge University Press, 2000. ISBN 0-521-78019-5.
40. Alter O, Brown PO, Botstein D. Singular value decomposition for genome-wide expression data processing and modeling. *Proc Natl Acad Sci USA* 2000;**97**(18):10101–6.
41. Jolliffe IT. *Principal Component Analysis*. 2nd edn. Springer, 2002.
42. Leek JT, Storey JD. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS Genet* 2007;**3**(9):e161.
43. McCall M, Irizarry R. Thawing frozen robust multi-array analysis (fRMA). *BMC Bioinformatics* 2011;**12**(1):369–76.
44. Warnat P, Eils R, Brors B. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics* 2005;**6**(1):265–80.
45. Katz S, Irizarry RA, Lin X, *et al*. A summarization approach for Affymetrix GeneChip data using a reference training set from a large, biologically diverse database. *BMC Bioinformatics* 2006;**7**:464–75.
46. Zilliox MJ, Irizarry RA. A gene expression barcode for microarray data. *Nat Methods* 2007;**4**(11):911–13.
47. McCall MN, Uppal K, Jaffee HA, *et al*. The gene expression barcode: leveraging public data repositories to begin cataloging the human and murine transcriptomes. *Nucleic Acids Res* 2011;**39**:D1011–15.
48. Bolstad BM, Irizarry RA, Astrand M, *et al*. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 2003; **19**(2):185–93. ISSN 1460-2059.
49. Lacson R, Pitzer E, Kim J, *et al*. DSGeo: software tools for cross-platform analysis of gene expression data in GEO. *J Biomed Inform* 2010;**43**(5):709–15. ISSN 15320464 doi: 10.1016/j.jbi.2010.04.007.
50. Xia X, McClelland M, Porwollik S, *et al*. WebArrayDB: cross-platform microarray data analysis and public data repository. *Bioinformatics* 2009;**25**(18):2425–9.
51. Jiang H, Deng Y, Chen HS, *et al*. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics* 2004;**5**:81–93.
52. Parzen E. On estimation of a probability density function and mode. *Ann Math Stat* 1962;**33**(3):1065–76.
53. Brettschneider J, Collin F, Bolstad BM, *et al*. Quality assessment for short oligonucleotide microarray data. *Technometrics* 2008;**50**(3):241–64.
54. Kim KY, Ki D, Jeong H, *et al*. Novel and simple transformation algorithm for combining microarray data sets. *BMC Bioinformatics* 2007;**8**(1):218–30.
55. Boedigheimer M, Wolfinger R, Bass M, *et al*. Sources of variation in baseline gene expression levels from toxi- cogenomics study control animals across multiple laboratories. *BMC Genom* 2008;**9**(1):285–301.
56. Scherer A. *Batch Effects and Noise in Microarray Experiments: Sources and Solutions*. John Wiley and Sons, 2009. Chapter 12.
57. Khuri AI, Sahai H. Variance components analysis: a selective literature survey. *Int Stat Rev* 1985;**53**(3):279–300.
58. Bylesjo M, Eriksson D, Sjodin A, *et al*. Orthogonal projections to latent structures as a strategy for microarray data normalization. *BMC Bioinformatics* 2007;**8**(1):207–17.
59. Lazar C, Taminau J, Meganck S, *et al*. A survey on filter techniques for feature selection in gene expression microarray analysis. *IEEE/ACM Trans Comput Biol Bioinform* 2012; **99**. (PrePrints) ISSN 1545-5963 doi: http://doi.ieeecomputersociety.org/10.1109/TCBB.2012.33.
60. Nueda MJ, Ferrer A, Conesa A. ARSyN: a method for the identification and removal of systematic noise in multifactorial time course microarray experiments. *Biostatistics* 2011.
61. Al-Shahrour F, Minguez P, Tárraga J, *et al*. FatiGO +: a functional profiling tool for genomic data. Integration of functional annotation, regulatory motifs and interaction data with microarray experiments. *Nucleic Acids Res* 2007; **35**:91–96.
62. Cope LM, Liz GM, Gabrielson E, *et al*. *The Integrative Correlation Coefficient : a Measure of Cross-study Reproducibility for Gene Expressionea Array Data.*. Johns Hopkins University, Dept. of Biostatistics Working Paper Series, 2007. (152).
63. Walker W, Liao I, Gilbert D, *et al*. Empirical Bayes accomodation of batch-effects in microarray data using identical replicate reference samples: application to RNA expression profiling of blood from Duchenne muscular dystrophy patients. *BMC Genom* 2008;**9**(1):494–507.