BIOINFORMATICS ORIGINAL PAPER

Vol. 29 no. 22 2013, pages 2877–2883
doi:10.1093/bioinformatics/btt480

Gene expression

Advance Access publication August 19, 2013

A new statistic for identifying batch effects in high-throughput genomic data that uses guided principal component analysis

Sarah E. Reese[1,*], Kellie J. Archer[1,2], Terry M. Therneau[3], Elizabeth J. Atkinson[3], Celine M. Vachon[4], Mariza de Andrade[3], Jean-Pierre A. Kocher[3] and Jeanette E. Eckel-Passow[3,*]

[1]Department of Biostatistics, [2]Biostatistics Shared Resource Core, VCU Massey Cancer Center, Virginia Commonwealth University, Richmond, VA 23284, USA, [3]Division of Biomedical Statistics and Informatics and [4]Division of Epidemiology, Department of Health Sciences Research, Mayo Clinic, Rochester, MN 55905, USA

Associate Editor: Janet Kelso

ABSTRACT

Motivation: Batch effects are due to probe-specific systematic variation between groups of samples (batches) resulting from experimental features that are not of biological interest. Principal component analysis (PCA) is commonly used as a visual tool to determine whether batch effects exist after applying a global normalization method. However, PCA yields linear combinations of the variables that contribute maximum variance and thus will not necessarily detect batch effects if they are not the largest source of variability in the data.

Results: We present an extension of PCA to quantify the existence of batch effects, called guided PCA (gPCA). We describe a test statistic that uses gPCA to test whether a batch effect exists. We apply our proposed test statistic derived using gPCA to simulated data and to two copy number variation case studies: the first study consisted of 614 samples from a breast cancer family study using Illumina Human 660 bead-chip arrays, whereas the second case study consisted of 703 samples from a family blood pressure study that used Affymetrix SNP Array 6.0. We demonstrate that our statistic has good statistical properties and is able to identify significant batch effects in two copy number variation case studies.

Conclusion: We developed a new statistic that uses gPCA to identify whether batch effects exist in high-throughput genomic data. Although our examples pertain to copy number data, gPCA is general and can be used on other data types as well.

Availability and implementation: The gPCA *R* package (Available via CRAN) provides functionality and data to perform the methods in this article.

Contact: reesese@vcu.edu or eckel@mayo.edu

Supplementary information: Supplementary data are available at *Bioinformatics* online.

Received on November 16, 2012; revised on July 3, 2013; accepted on August 14, 2013

1 INTRODUCTION

1.1 Batch effects

Batch effects are defined to be systematic non-biological variation between groups of samples (or batches) due to experimental artifacts (Benito *et al*., 2004; Johnson *et al*., 2007; Luo *et al*., 2010). Many factors contribute to the generation of batch effects. Some of these include chip type, platform, laboratory, technician, storage and shipment conditions, protocols (sample, amplification, labeling and hybridization), cRNA/cDNA synthesis, wash conditions, etc (Luo *et al*., 2010).

Few methods have been developed to detect batch effects. For expression data, existing methods include principal component analysis (PCA) (Holmes *et al*., 2011; Yang *et al*., 2008) and unsupervised hierarchical clustering (Chow *et al*., 2012; Johnson *et al*., 2007; Konstantinopoulos *et al*., 2011). However, neither of these methods provides a statistical test for detecting whether batch effects are present.

A common method for visualizing the existence of batch effects is PCA. The first two principal components are plotted with each sample colored by the suspected batch, and separation of colors is taken as evidence of a batch effect. However, as pointed out by Benito *et al*. (2004), if the batch effect is not the greatest source of variation then PCA methods do not work well, as they look for the directions of greatest variation. Also, visual inspection of the first and second principal components is subjective. Methods that can detect batch effects are needed, as ignoring the potential for batch effects can have a serious effect on downstream analysis results. In this article, we propose a test statistic derived using both the traditional PCA method and guided PCA (gPCA) for detecting batch effects. We evaluate the performance of our test in extensive simulation studies. We also demonstrate the difference between PCA and gPCA using two copy number variation datasets; however, the methods are appropriate for any type of high-throughput genomic data.

2 METHODS

2.1 Statistical methods

2.1.1 Principal component analysis PCA is used for data reduction and interpretation. It is used to explain the variance–covariance structure of a set of variables through linear combinations of the variables (Johnson and Wichern, 2002). PCA is a form of unsupervised learning that seeks to find the ‘combination of conditions that explain the greatest variation in the data’ (Yang *et al*., 2008). It is used in many types of analyses including neuroscience and computer graphics (Shlens, 2005), in addition to microarray data analyses (Holmes *et al*., 2011; Yang *et al*.,

*To whom correspondence should be addressed.

2008). The numerical workhorse of PCA is singular-value decomposition (SVD).

Singular-value decomposition Let $\mathbf{X}$ be a centered $n \times p$ matrix of real numbers where n denotes sample and p denotes genomic feature (e.g. probe). Then there exists an $n \times n$ orthogonal matrix $\mathbf{U}$ and a $p \times p$ orthogonal matrix $\mathbf{V}$ such that

$$\mathbf{X} = \mathbf{UDV}'$$

where the $n \times p$ matrix $\mathbf{D}$ has diagonal (q, q) entry $\lambda_q \geq 0$ for $q = 1, \ldots, \min(n,p)$ where, by convention, $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min(n,p)}$ and the other entries are 0. The positive constants λ_q are called the *singular values* of $\mathbf{D}$ (Johnson and Wichern, 2002).

Principal components are the length n column vectors $(P_1, P_2, \ldots, P_p)$ of

$$\mathbf{P} = \mathbf{XV}$$

where $\mathbf{X}$ is an $n \times p$ matrix, $\mathbf{V}$ is the matrix of right singular vectors, $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_p$, from the singular value decomposition and $\mathbf{P}$ is the $n \times p$ principal component matrix.

The first principal component has the highest variance, and the second principal component has the next highest variance under the constraint that it is uncorrelated with the proceeding component. Typically, PCA is performed on $\mathbf{X}$ alone. Herein, we refer to this as 'unguided' PCA. As discussed in Section 1, unguided PCA is not effective for identifying batch effects if they are not the largest source of variation. In this case, it does not mean that batch effects do not exist in the data, but that alternate methods must be used to find them.

2.1.2 Guided PCA

For detecting batch effects, a more informative version of PCA is on $\mathbf{Y}'\mathbf{X}$, where $\mathbf{Y}$ is an $n \times b$ indicator matrix where b denotes batch and n denotes sample.

$$\mathbf{Y} = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & \mathbf{1} \end{bmatrix}$$

where $\mathbf{1}$ and $\mathbf{0}$ are block matrices with

$$y_{ik} = \begin{cases} 1 & \text{if sample } i \text{ is in batch } k \\ 0 & \text{otherwise} \end{cases}$$

for $i = 1, \ldots, n_k(\sum_k n_k = n)$ and $k = 1, \ldots, b$. Performing SVD on $\mathbf{Y}'\mathbf{X}$ results in a $b \times b$ matrix $\mathbf{U}$ that denotes the batch loadings and the $p \times p$ matrix $\mathbf{V}$ that denotes the probe loadings. Large singular values imply that the batch is important for the corresponding principal component. gPCA guides the SVD to look for batch effects in the data based on the batch indicator matrix $\mathbf{Y}$, which can be defined to indicate any type of potential batch effect.

Another commonly used method in this situation is Canonical Correlation Analysis, which finds the linear combination with maximum correlation; however, we are interested in variance, not correlation.

2.1.3 Proposed method: test statistic for testing whether batch effects exist

Our test statistic, δ, quantifies the proportion of variance owing to batch effects in experimental genomic data. The proportion of total variance owing to batch is the ratio of the variance of the first principal component from gPCA to the variance of the first principal component from unguided PCA.

$$\delta = \frac{var(\mathbf{XV}_{g_1})}{var(\mathbf{XV}_{u_1})}$$

where g indicates gPCA and u indicates unguided PCA. $\mathbf{V}$ is the matrix of probe loadings resulting from gPCA or PCA, respectively. Large values of δ (values near 1) imply that the batch effect is large.

To determine whether δ is significantly larger than would be obtained by chance, a P-value is estimated using a permutation distribution created by permuting the batch vector $M = 1000$ times so that δ_{p_m} is computed for $m = 1, \ldots, M$ where p indicates permutation. Here, δ_{p_m} is the proportion of the total variance due to the first principal component from the m^{th} permutation from gPCA to the total variance due to the first principal component from the m^{th} permutation from unguided PCA. A one-sided P-value is estimated as the proportion of times the observed δ was in the extreme tail of the permutation distribution.

$$P\text{-value} = \frac{\sum_{m=1}^{M} \left(\hat{\delta} < \hat{\delta}_{p_m}\right)}{M}.$$

Estimating percentage of total variation explained by batch. The percentage of total variation explained by batch is then calculated as

$$\frac{\widehat{PC}_g - \widehat{PC}_u}{\widehat{PC}_g} \times 100$$

where

$$\widehat{PC}_u = \frac{var(\mathbf{XV}_{\mathbf{u_1}})}{\sum_{i=1}^{n} var(\mathbf{XV}_{\mathbf{u_i}})} \quad \text{and} \quad \widehat{PC}_g = \frac{var(\mathbf{XV}_{\mathbf{g_1}})}{\sum_{k=1}^{b} var(\mathbf{XV}_{\mathbf{g_k}})}$$

where u and g represent unguided PCA and gPCA, respectively.

2.2 Simulation study

Most often investigators are interested in modeling their data in the presence of a known phenotype. Therefore, we simulated data to represent copy number data under three scenarios: (i) feature data (here, feature denotes probe) with no phenotypic effect; (ii) feature data with a phenotypic effect with high variance; and (iii) feature data with a phenotypic effect with low variance. The feature data were generated independently from a multivariate normal distribution with 1000 features and 90 observations. To study type I and II errors, for all three scenarios, the data were simulated in two ways: to include a true batch effect and without a true batch effect. When a batch effect was present, there were two batches with batch mean vectors of $\mathbf{0}$ and $\mathbf{1}$. The variance associated with batch was $\sigma_b^2\mathbf{I}$, where σ_b^2 was allowed to be 0.5 or 1. In the true phenotype scenarios, 10% of the features were affected by phenotype using mean vectors $\mathbf{0}$ and $\mathbf{1}$ and variance matrix $\sigma_p^2\mathbf{I}$ where $\sigma_p^2 = 2$ for the high phenotypic variance scenario and $\sigma_p^2 = 0.2$ for the low phenotypic variance scenario. The proportion of features affected by the phenotype was `pprop` = 0.1 or 0.05. In all scenarios with a phenotypic effect, the phenotype was generated independent from any batch effect. Each simulation scenario was repeated 500 times.

For the scenarios with no true batch effect, the resulting proportion of P-values <0.05 formed our estimate of the type I error. The proportion of P-values <0.05 for the scenarios with a true batch effect formed our estimate of the power. Here, phenotype can be thought of as any variable of interest, whether categorical (e.g. case versus control) or continuous (e.g. mammographic density).

2.3 Case studies

Our method was applied to two case studies. The $\mathbf{U}$ and $\mathbf{V}$ matrices are assumed to be orthogonal $n \times n$ (or $b \times b$ for gPCA) and $p \times p$ matrices, respectively. To adjust for missing values, mean value imputation was performed on the centered data $\mathbf{X}$ before PCA.

2.3.1 Filtering

For unsupervised learning problems, non-informative features contribute random noise to distance calculations. The resulting effect is that non-informative features mask useful information provided by informative features. Therefore, non-informative features should be assigned a zero weight in the clustering algorithm (Kohane *et al.*, 2003). The simplest implementation for assigning a non-zero weight in a cluster analysis is to exclude identified non-informative features. This filtering step is applied to genomic data to remove sources of obscuring variation before applying a clustering algorithm. In our simulation studies, we observed higher power when the proportion of features affected by batch increased; therefore, we filtered our data stringently to keep the most variable or informative features. A variance filter was applied to the data to remove noise and reduce the number of features. The standard deviation of each feature was calculated and the 1000 most variable features were retained (Causton *et al.*, 2003; Dudoit *et al.*, 2002; Inza *et al.*, 2004). A sensitivity analysis was performed allowing the number of features retained by the variance filter to range between 10 and the full GENEMAM dataset. Further analysis implementing an analysis of variance filter was also investigated.

2.3.2 GENEMAM The GENetic Epidemiology of MAMmogr-aphic Density (GENEMAM) study data included 614 samples from the Minnesota Breast Cancer family study (Sellers *et al.*, 1995). These samples were genotyped using the Illumina Human 660 bead-chip array. Samples were processed over three time periods on eight plates. Forty-two samples failed quality-control checks from plates 1–4 because of an Illumina reagent problem, and these samples were replated on plate 5, along with six other samples. Samples on plates 6–8 were genotyped at a later date. This effectively yielded three batches corresponding to the three different runs. Data for all chromosomes were used. Illumina's GenomeStudio software was used to obtain the Log_2 *R* ratio (LRR) values. LRR is a measure of relative intensity where *R* is the sum of the normalized allelic probe intensities produced by SNP assays and the ratio is of observed *R* divided by the expected value (Laurie *et al.*, 2010).

2.3.3 GENOA The Genetic Epidemiology Network of Arteriopathy (GENOA) data included 1418 of the non-Hispanic white adults enrolled in the GENOA study of the Family Blood Pressure Program, a study designed to identify germline genetic determinants of hypertension in multiple ethnic groups. These samples were genotyped on Affymetrix SNP Array 6.0 chips, and all samples had contrast QC values >0.4. The PennCNV-Affy Protocol (http://www.openbioinformatics.org/penncnv/penncnv_tutorial_affy_gw6.html) was followed to obtain the LRR values. The analysis focused on chromosome 22 data using the first 10 plates consisting of 703 samples.

Table 1. Estimated type I error

	$\sigma = 0.5$	$\sigma = 1$
(a) No phenotype	0.034	0.034
(b) High phenotype (pprop = 0.1)	0.014	0.014
(c) Low phenotype (pprop = 0.1)	0.000	0.002
(d) Low phenotype (pprop = 0.05)	0.010	0.046

Note: For all scenarios, there is no true batch effect. Scenario (a) has no phenotypic effect in the data; however, scenario (b) has a phenotypic effect with high variance included and scenarios (c and d) have phenotypic effects with low variance included in the analysis with phenotypic effect at pprop = 0.1 or 0.05, respectively.

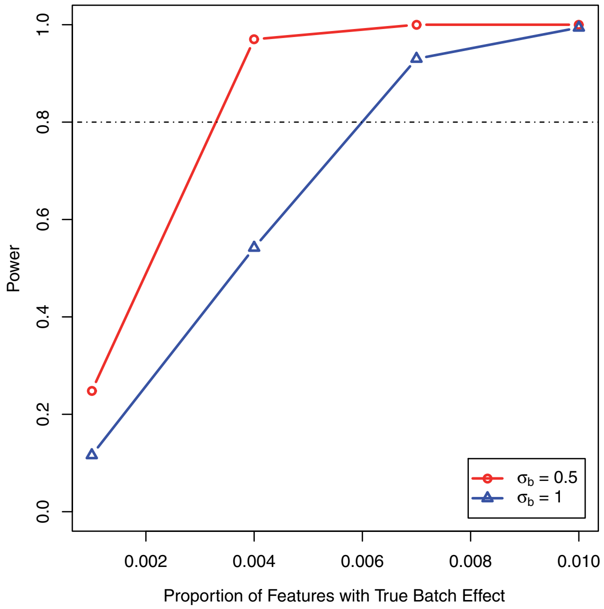

Fig. 1. Power for detecting batch effect as a function of the proportion of features that are affected by batch when no true phenotype was included with batch proportion ranging from 0.1 to 1%

3 RESULTS

3.1 Simulation study

The estimates for type I error for all scenarios are reported in Table 1. The proportion of features with a phenotypic effect is pprop = 0.1 for scenarios (b–c) and 0.05 for scenario (d). In all scenarios, the type I error is at or below the nominal 0.05 level. Figure 1 shows power of our test statistic as a function of the proportion of features with a true batch effect if there is no true phenotypic effect. If $\sigma_b^2 = 0.5$, then our test statistic has 80% power if ~0.3% of the features are affected by batch. If $\sigma_b^2 = 1$, then ~0.6% of features need to have a batch effect to achieve 80% power. If a phenotypic effect exists with high phenotypic variance, then ~1.5 or 2% of the features need to have a batch effect to achieve 80% power for $\sigma_b^2 = 0.5$ and $\sigma_b^2 = 1$, respectively (Fig. 2a). Similarly, if a phenotype exists with low phenotypic variance and 10% of features are affected by phenotype, then ~1.5 or 1.2% of the features need to have a batch effect to achieve 80% power for $\sigma_b^2 = 0.5$ and $\sigma_b^2 = 1$, respectively, and if 5% of features are affected by phenotype, then ~0.75% of the features need to have a batch effect to achieve 80% power for both $\sigma_b^2 = 0.5$ and $\sigma_b^2 = 1$ (Fig. 2b).

Power is also higher when the batch variance is smaller. Further simulations varying the batch variance, with the difference between batch means smaller than the difference between the phenotype means, and with high proportions of features affected by batch can be found in Supplementary Section 4. In the scenario where batch variance is varied and the batch mean difference is smaller than the phenotype mean difference, we found that as batch variance increased, so did the estimated power. The smaller the difference in the phenotypic means, the higher the power. In the no phenotype scenario, we found that power decreased as the batch variance increased. This is attributable to the first principal component from unguided PCA and gPCA being similar when no phenotype is affecting the feature data, which is unlikely in application datasets. In the scenario where a high proportion (between 50 and 90%) of features are

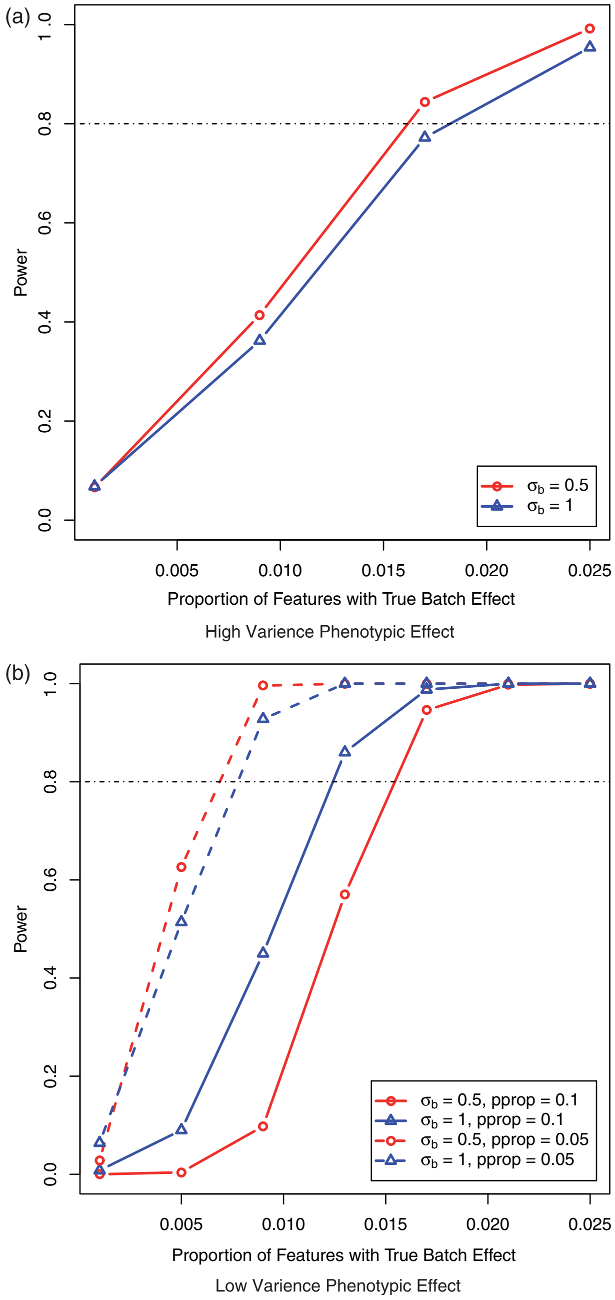

Fig. 2. Power for detecting batch effect as a function of the proportion of features that are affected by batch when (**a**) phenotypic data with high variance were included in gPCA with batch proportion ranging from 0.1 to 2.5% and (**b**) phenotypic data with low variance were included in gPCA with batch proportion ranging from 0.1 to 2.5%

affected by batch, we found that the estimated power was 100% (see Supplementary Table S5).

3.2 GENEMAM

The standard use of PCA is to look at the plot of the first principal component of the data ($n \times p$ matrix **X,** where n denotes sample and p denotes probe) versus the second principal component (Fig. 3a). The GENEMAM data have an obvious batch effect, and the PCA plot of the first two principal components shows that this batch effect is due to the plate when colored by plate with three batches consisting of plates 1–4, 5 and 6–8. As is common with batch effects, this batch effect is due to the plates being run at different times.

Next, we performed a gPCA with plate as the batch indicator. The gPCA plot of the first two principal components (Fig. 3b) shows greater separation in the batches, especially of plate 3 from plates 1, 2 and 4, than the unguided principal component plot (Fig. 3a). After filtering out all but the $p = 1000$ most variable features, our permutation test confirms that there is a significant batch effect separating the plates ($\delta = 0.5987$; P-value < 0.001). Of the variance due to features in these data, 87.3% of the total variation is explained by batch.

We also performed a sensitivity analysis allowing the number of features retained by the variance filter to range between 10 and the full GENEMAM dataset. We found that our test statistic was not sensitive to filtering (for the application datasets and when no phenotypic effect was present in the simulation scenario). The test statistic applied to the simulated data was not affected by filtering provided that the number of features retained was 5% when there was a phenotype with high variance (a somewhat weak phenotypic effect) and ~50% when there was a phenotype with low variance (i.e. a strong phenotypic effect), and thus filtering can be used as a method to reduce the analysis time required provided it is judiciously applied (Supplementary Table S1). We also implemented an analysis of variance filter to identify probes with a significant batch effect and found that even with stringent multiple comparison methods, the filtered datasets were still very large. A detailed discussion can be found in the Supplementary Section 1.

This case study is an example with an obvious batch effect and thus did not require specialized methods to detect, as batch was the largest source of variability.

3.3 GENOA

In this case study, batch is not so easily detected using unguided PCA. Unguided PCA was performed and Figure 4a shows the PCA plot of the first two principal components. Figure 4a shows that plates 7 and 8 might be slightly separated from the rest of the plates. A gPCA with batch defined by plate (Fig. 4b) shows that plates 7 and 8, along with plate 4, separate slightly from the other plates. It is not obvious from the unguided PCA that plate 4 is separate from the rest of the plates. However, gPCA shows a separation between plate 4 and the rest of the plates. After filtering out all but the $p = 1000$ most variable features, our permutation test shows that there is a significant batch effect separating the plates ($\delta = 0.9219$; P-value < 0.001). Of the variance due to features in these data, 71% of the total variation is explained by batch. gPCA identifies a batch (plate 4) that does not otherwise stand out in an unguided principal component plot.

3.4 Impact of identifying and correcting for batch effects

Although various methods exist for adjusting for batch effects, these methods do not incorporate a procedure for identifying whether a batch effect is truly present (Benito *et al.*, 2004;

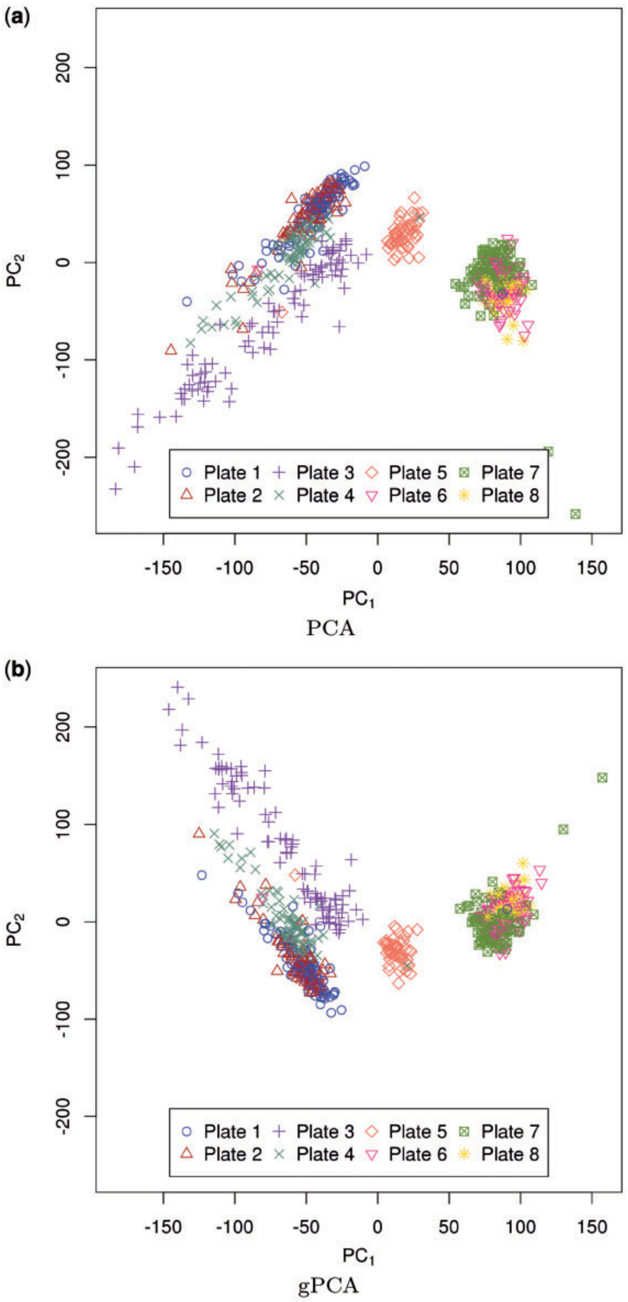

Fig. 3. GENEMAM–(**a**) Unguided PCA of **X** and (**b**) gPCA of **Y′X**. Samples for each plate are denoted by a different color (online version) and/or symbol

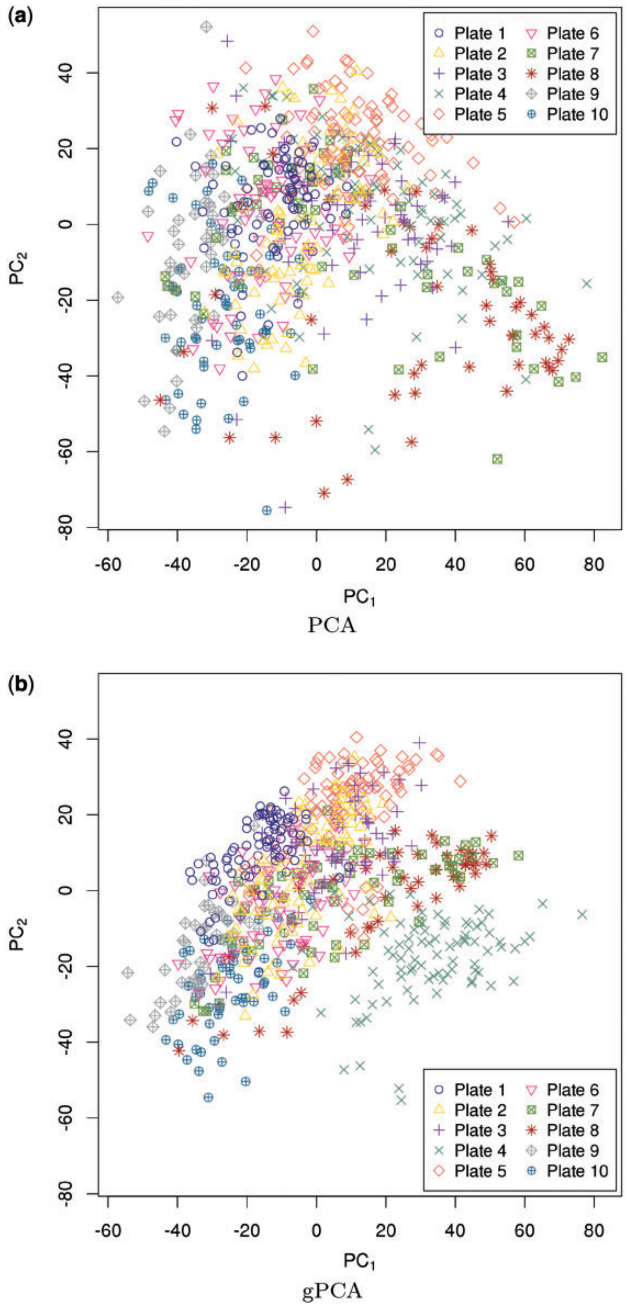

Fig. 4. GENOA–(**a**) Unguided PCA of **X** and (**b**) gPCA of **Y′X**. Samples for each plate are denoted by a different color (online version) and/or symbol

Carvalho *et al*., 2010; Chow *et al*., 2012; Huang *et al*., 2012; Johnson *et al*., 2007; Konstantinopoulos *et al*., 2011; Leek and Storey, 2007, 2008; Leek *et al*., 2012; Marron and Todd, 2002; McCall *et al*., 2010; Sun *et al*., 2011). Using both simulated and real data (see Supplementary Section S3), we further assessed the effects of correcting for batch on the number of significant features. In our simulated dataset, there were 50 features with a phenotypic effect, 50 features with a batch effect and 100 features with both a phenotypic and a batch effect. After fitting a linear model using the `lmFit()` function with phenotype as the predictor, the number of significant features in simulated data was assessed using the `eBayes()` function in the `limma` package both before batch correction and after batch correction using the batch mean-centering method of Sims *et al*. (2008) and the FDR method of Benjamini and Hochberg (1995) for adjusting for multiple testing, letting $\alpha = 0.1$. Forty-eight of the 150 features had a significant phenotypic effect before batch correction, whereas 148 of the 150 features were significant post-batch

Table 2. Evaluating whether a batch correction method was successful: test statistic δ and corresponding P-values before and after batch correction for the three simulated data scenarios (no phenotypic effect, high-variance phenotype and low-variance phenotype), and the two case study datasets, GENEMAM and GENOA

	Uncorrected		Corrected	
	$\hat{\delta}$	P	$\hat{\delta}$	p
No phenotype	0.902	<0.001	0.060	1.000
High-variance phenotype	0.700	<0.001	0.030	1.000
Low-variance phenotype	0.572	<0.001	0.020	1.000
GENEMAM (run time)	0.583	<0.001	0.044	1.000
GENEMAM (plate)	0.599	<0.001	0.050	1.000
GENOA (plate)	0.922	<0.001	0.017	1.000

Note: Batch mean-centering (Sims *et al.*, 2008) was used for batch effect correction.

correction (Supplementary Table S4). This shows that batch correction allows features with a true phenotypic effect that is masked by batch to be identified as significant after batch correction.

3.5 Evaluating batch correction methods

Luo *et al.* (2010) observed the impact of batch effect removal on cross-batch prediction performance, and Lazar *et al.* (2012) and Chen *et al.* (2011) provided surveys of some of the many methods of batch effect removal. In Table 2, we report our test statistic δ and the corresponding P-values when analyzing the raw uncorrected and batch mean-centering corrected data. Although there is a highly significant batch effect in the uncorrected data, the correction method successfully removed enough batch variation from all datasets. Therefore, our proposed test statistic is useful for identifying whether any batch adjustment methods should be applied before statistical analysis and for assessing the adequacy of the batch adjustment method applied.

4 DISCUSSION

gPCA can be used to identify batch effects in large and messy data, such as expression, CNV, and methylation data, by computing the SVD while taking batch into account. Principal component plots are a standard method of looking for batch effects in high-throughput data. Here, we show how gPCA can be used both to visualize batch effects and to formally test whether batch effects are present in the data. From our simulation studies, the type I error of our statistic is close to nominal 0.05 level and power is reasonably good when an adequate proportion of the features are affected by batch. Additionally, when the proportion of features affected by batch is high (between 50 and 90%), the estimated power is 100% (Supplementary Table S5).

The **Y** matrix in the gPCA analysis can be formed by considering any combination of variables. We note that with the **Y** matrix coding multiple variables, the variance ascribed to the first principal component of the gPCA may incorporate multiple sources, which would be difficult to disentangle. To estimate the variance attributed to multiple sources, gPCA could be used to examine each one by defining **Y** in separate analyses. Note that gPCA is dependent on knowing how to define potential batch effects. If this is not known, this statistic should not be used. If batch is misspecified by the investigator, provided the misspecified batch effect indicator matrix has no relationship to the experimental design, then the test will likely not reject the null hypothesis because type I error was close to the nominal 0.05 level.

In the case of microarray data, scaling of the batch identifier matrix **Y** is not in general useful for balanced experiments. However, when some batches have far more samples than others, scaling of **Y** is a useful tool to correct for the imbalance. In the case of the GENEMAM data, while plates 5 and 8 had half as many or fewer samples than the rest of the plates, the effect of scaling **Y** was minimal, although it did have an effect. For microarray data, we do not want to scale the data matrix **X**, as all the variables, probes in our case, are already on the same scale and scaling **X** would only serve to adjust the variance. If the variances are smoothed, then we may miss an important difference between variables or batches.

gPCA can be used on other problems and types of data as well, including B-allele frequency data and expression data. Because pre-processing of microarrays is time-consuming, expensive and with abundant systematic errors, the ability to discover and adjust for these errors is important. Our test statistic that uses gPCA allows one to find the sources of systematic errors, or batch effects, in all types of microarray data and adjust for it during analysis.

In summary, herein we present a novel statistic to test for the presence of batch effects. The test is particularly useful to test whether batch effects exist after applying a global normalization procedure such as quantile or loess normalization. Although these global normalization procedures correct for batch effects that affect all probes similarly, they do not correct for probe-specific batch effects. Furthermore, our test statistic is useful for determining whether a batch-correction method has adequately removed observed batch effects.

Funding: National Institutes of Health research grants (R01 HL87660 to M.d.A.); (R01 CA128931 and CA140286 to C.M.V.); (T32 ES007334 S.E.R. and K.J.A.); Mayo Clinic Center for Individualized Medicine.

Conflict of Interest: none declared.

REFERENCES

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

Benjamini,Y. and Hochberg,Y. (1995) Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. R. Stat. Soc. Series B Methodol.*, **57**, 289–300.

Carvalho,B.S. *et al.* (2010) Quantifying uncertainty in genotype calls. *Bioinformatics*, **26**, 242–249.

Causton,H.C. *et al.* (2003) *Microarray Gene Expression Data Analysis: A Beginners Guide*, chapter 3. Blackwell Publishing Inc, Oxford, UK.

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

Chow,M.L. *et al.* (2012) Preprocessing and quality control strategies for Illumina DASL assay-based brain gene expression studies with semi-degraded samples. *Front. Genet.*, **3**, 11.

Dudoit,S. *et al.* (2002) Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.*, **97**, 77–87.

Holmes,S. *et al.* (2011) Visualization and statistical comparisons of microbial communities using R packages on phylochip data. In: *Bioscomputing 2011: Proceedings of the Pacific Symposium*, Hawaii, USA, pp. 142–153.

Huang,H. *et al.* (2012) R/DWD: distance-weighted discrimination for classification, visualization and batch adjustment. *Bioinformatics*, **28**, 1182–1183.

Inza,I. *et al.* (2004) Filter versus wrapper gene selection approaches in DNA microarray domains. *Artif. Intell. Med.*, **31**, 91–103.

Johnson,R.A. and Wichern,D.W. (2002) *Applied Multivariate Statistical Analysis*. 5th edn. Prentice Hall, Upper Saddle River, New Jersey, USA.

Johnson,W.E. *et al.* (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.

Kohane,I. *et al.* (2003) *Microarrays For An Integrative Genomics*. The MIT Press, Cambridge, Massachusetts, USA.

Konstantinopoulos,P.A. *et al.* (2011) Integrated analysis of multiple microarray datasets identifies a reproducible survival predictor in ovarian cancer. *PLoS One*, **6**, e18202.

Laurie,C.C. *et al.* (2010) Quality control and quality assurance in genotypic data for genome-wide association studies. *Genet. Epidemiol.*, **34**, 591–602.

Lazar,C. *et al.* (2012) Batch effect removal methods for microarray gene expression data integration: a survey. *Brief. Bioinform.*, **14**, 469–490.

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

Leek,J.T. and Storey,J.D. (2008) A general framework for multiple testing dependence. *Proc. Natl Acad. Sci. USA*, **105**, 18718–18723.

Leek,J.T. *et al.* (2012) The sva package for removing batch effects and other unwanted variation in high-throughput experiments. *Bioinformatics*, **28**, 882–883.

Luo,J. *et al.* (2010) A comparison of batch effect removal methods for enhancement of prediction performance using MAQC-II microarray gene expression data. *Pharmacogenomics J.*, **10**, 278–291.

Marron,J.S. *et al.* (2007) Distance-weighted discrimination. *J. Am. Stat. Assoc.*, **102**, 1267–1271.

McCall,M.N. *et al.* (2010) Frozen robust multiarray analysis (fRMA). *Biostatistics*, **11**, 242–253.

Sellers,T.A. *et al.* (1995) Epidemiologic and genetic follow-up study of 544 Minnesota breast cancer families: design and methods. *Genet. Epidemiol.*, **12**, 417–429.

Shlens,J. (2005) *A Tutorial on Principal Component Analysis*. Systems Neurobiology Laboratory, Salk Institute for Biological Studies, La Jolla, California, USA.

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

Sun,Z. *et al.* (2011) Batch effect correction for genome-wide methylation data with Illumina Infinium platform. *BMC Med. Genomics*, **4**, 84.

Yang,H. *et al.* (2008) Randomization in laboratory procedure is key to obtaining reproducible microarray results. *PLoS One*, **3**, e3724.