BIOINFORMATICS ORIGINAL PAPER Vol. 24 no. 9 2008, pages 1154–1160
doi:10.1093/bioinformatics/btn083

Gene expression

Merging two gene-expression studies via cross-platform normalization

Andrey A. Shabalin[1,*], Håkon Tjelmeland[2], Cheng Fan[3], Charles M. Perou[3,4,5] and Andrew B. Nobel[1]

[1]Department of Statistics and Operations Research, University of North Carolina at Chapel Hill, Chapel Hill, USA, [2]Department of Mathematical Sciences, Norwegian University of Science and Technology, Trondheim, Norway, [3]Lineberger Comprehensive Cancer Center, University of North Carolina at Chapel Hill, [4]Department of Pathology and Laboratory Medicine, University of North Carolina at Chapel Hill and [5]Department of Genetics, University of North Carolina at Chapel Hill, Chapel Hill, USA

Received on November 28, 2007; revised on February 7, 2008; accepted on March 1, 2008

Advance Access publication March 5, 2008

Associate Editor: David Rocke

ABSTRACT

Motivation: Gene-expression microarrays are currently being applied in a variety of biomedical applications. This article considers the problem of how to merge datasets arising from different gene-expression studies of a common organism and phenotype. Of particular interest is how to merge data from different technological platforms.

Results: The article makes two contributions to the problem. The first is a simple cross-study normalization method, which is based on linked gene/sample clustering of the given datasets. The second is the introduction and description of several general validation measures that can be used to assess and compare cross-study normalization methods. The proposed normalization method is applied to three existing breast cancer datasets, and is compared to several competing normalization methods using the proposed validation measures.

Availability: The supplementary materials and XPN Matlab code are publicly available at website: https://genome.unc.edu/xpn

Contact: shabalin@email.unc.edu

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

1 INTRODUCTION

High-throughput gene-expression microarrays are currently being applied in a wide variety of biomedical problems. There are now several widely used, commercially available, microarray platforms that measure gene expression in related, but different, ways. No matter which technology is used, the evaluation of gene-expression experiments usually begins with statistical analyses that take a variety of forms, including exploratory analysis (such as clustering), classification and assessments of differential expression.

The increasing number and availability of large-scale gene-expression studies of human and other organisms provides strong motivation for cross-study analyses that combine existing and/or new datasets. In a cross-study analysis, the data, relevant test statistics or conclusions of several studies are combined. The simultaneous analysis of different studies of a common organism and phenotype has the potential to strengthen and extend the results obtained from the individual studies. Cross-study analyses can be carried out using existing datasets, so their results hold out the promise of comparatively inexpensive, scientific 'value-added'.

On the other hand, combining data from different expression studies poses a number of statistical difficulties. These difficulties arise from the fact that the constituent datasets have often been produced using different gene-expression platforms and different processing facilities. As a consequence, measurements from different platforms cannot be directly combined. Identifying and removing such systematic effects is the primary statistical challenge in cross-study analysis. We note that technological differences between studies may be confounded with biological differences arising from the choice of patient cohorts (e.g. age, gender or ethnicity). In many cases, technological artifacts are dominant, though care should be taken to verify this, and one can hope to remove them while leaving biological information intact.

There are several potential approaches to cross-study analysis, depending on what information is being synthesized. At the highest level, one may wish to combine, through meta-analysis or other techniques, the broad conclusions of different studies. Most existing work on multi-study gene-expression analysis is focused on an intermediate level, where the goal is to combine information from primary statistics (such as *t*-statistics or *P*-values) or secondary statistics (such as gene lists) that are derived from the individual studies (Choi *et al.*, 2003; Garrett-Mayer *et al.*, 2004; Ghosh *et al.*, 2003). Other approaches to meta-analysis of gene-expression data are considered by (Garrett-Mayer *et al.*, 2007; Parmigiani *et al.*, 2004; Rhodes *et al.*, 2002, 2004; Shen *et al.*, 2004). This article deals with the problem of cross-study normalization: how to combine two available datasets in order to produce a single, unified dataset to which standard statistical procedures (such as clustering, classification and measures of differential expression) can be applied.

There has been a great deal of work on the normalization of gene-expression data within a single study (Bolstad *et al.*, 2003; Irizarry *et al.*, 2003a, b; Yang *et al.*, 2002). Much of that work can be applied, with little modification, to normalizing

*To whom correspondence should be addressed.

data from multiple studies that are based on the same technological platform. The emphasis here is on the problem of combining data from different array platforms. We will use the term cross-platform normalization when this distinction is important.

2 CROSS-PLATFORM NORMALIZATION (XPN) METHOD

Here we describe the basic idea behind the XPN (cross-platform normalization) method. We restrict our attention to merging two studies; the model and fitting procedure can be extended in a natural way to handle three or more studies.

XPN takes as input the gene-expression measurements from two studies, after appropriate preprocessing and imputation. One may work with the set of common genes in the studies, or on a selected subset of these genes. Once an appropriate set G of genes has been identified, the available data can be represented as two matrices

$$X_p = \{x_{gsp} : g \in G, s = 1, \ldots, n_p\} \quad p = 1, 2. \tag{1}$$

Here X_p denotes the available data from study p, and x_{gsp} is the expression of gene g in sample s of study p. Let n_1 and n_2 denote the number of samples in studies 1 and 2, respectively, m denote the number of genes in G. The normalized data can be represented similarly, as two matrices $\tilde{X}_p = \{\tilde{x}_{gsp} : g \in G, s = 1, \ldots, n_p\}$ with the same dimensions as X_1 and X_2.

2.1 Block linear model

The XPN procedure is based on a simple block-linear model. In this model, the observed value x_{gsp} is a scaled and shifted block mean plus noise. The block mean is constant over a range of gene and sample values, and is the same in each platform. The slope and offset of the linear transformation, as well as the variance of the noise, depend on the gene g and the platform p. More precisely, we assume that

$$x_{gsp} = A_{\alpha^*(g),\beta_p^*(s),p} \cdot b_{gp} + c_{gp} + \sigma_{gp}\varepsilon_{gsp}. \tag{2}$$

The functions $\alpha^* : \{1, \ldots, m\} \mapsto \{1, \ldots, K\}$ and $\beta_p^* : \{1, \ldots, n_p\} \mapsto \{1, \ldots, L\}$, $p = 1, 2$, define linked groups of genes and samples, respectively. The numbers A_{ijp} are block means, while b_{gp} and c_{gp} represent sensitivity and offset parameters, respectively, that are specific to each gene and platform. The noise variables ε_{gsp} are independent standard normals, so the final term in (2) has variance σ_{gp}^2. The model reflects the assumption that the samples of each available study fall roughly into one of L statistically homogenous groups, and that each group is defined by an associated gene profile that is constant within each of K groups of similar genes. The block means $\{A_{i,j} : i = 1, \ldots, K\}$ represent the profile of the jth group. Figure 1 illustrates the underlying block structure. Note that the basic studies may be of different sizes. A heatmap illustrating the same idea on real data is provided in the Supplementary Materials.

2.2 Description of XPN

Initially, the data from the available studies are sample standardized and gene median centered, in order to remove gross systematic differences, and then combined. Following the model (2), clustering is then used to identify homogenous groups of genes and samples in the combined data matrix. Specifically, k-means clustering is applied independently to the rows and columns of the combined data matrix, using $k = K$ gene clusters and $k = L$ sample clusters, respectively. Application of k-means begins with a random choice of centroids for the clusters. In clustering rows, we select K rows of the data matrix at random, and use these as the initial centroids. Cluster assignments and centroids are then updated iteratively until convergence to a local minimum of the sum of squared Euclidean distances. A similar procedure is used for clustering of the columns.

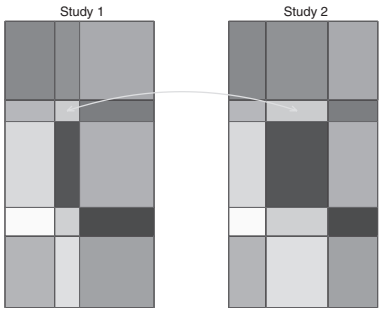

Fig. 1. Studies 1 and 2 after row and column clustering of their combined data, with $K = 5$ gene groups and $L = 3$ sample groups. Shading indicates linked gene-sample blocks.

The gene clusters in the combined data matrix are summarized by the assignment function $\alpha : G \to \{1, \ldots, K\}$. Gene clusters are naturally linked across studies, as we work with the same genes in each study. The column clusters in the combined data matrix are summarized by assignment functions $\beta_p : \{1, \ldots, n_p\} \mapsto \{1, \ldots, L\}$ for $p = 1, 2$. Specifically, $\beta_p(s)$ is the index of the combined sample cluster containing sample s from Study p. The ℓth combined cluster splits into linked clusters $\{s : \beta_1(s) = \ell\}$ in Study 1 and $\{s : \beta_2(s) = \ell\}$ in Study 2.

From the mappings $\alpha(g)$ and $\beta_p(s)$, estimates of the model parameters $\hat{A}_{ijp}$, $\hat{b}_{gp}$, $\hat{c}_{gp}$ and $\hat{\sigma}_{gp}$ are obtained using standard maximum likelihood methods. Details are given in the Appendix. Common model parameters $\hat{\theta}_g = (\hat{b}_g, \hat{c}_g, \hat{\sigma}_g^2)$ and $\hat{A}_{ij}$ are then calculated as weighted averages of the parameters in Study 1 and Study 2:

$$\hat{\theta}_g = \frac{n_1\hat{\theta}_{g,1} + n_2\hat{\theta}_{g,2}}{n_1 + n_2} \qquad \hat{A}_{ij} = \frac{n_{j,1}\hat{A}_{i,j,1} + n_{j,2}\hat{A}_{i,j,2}}{n_{j,1} + n_{j,2}}$$

where $n_{j,p}$ is the number of samples in the jth sample group of platform p. The expression values of each platform are then modified in accordance with the estimated model parameters to produce normalized values

$$x_{gsp}^* = \hat{A}_{\alpha(g),\beta_p(s)}\,\hat{b}_g + \hat{c}_g + \hat{\sigma}_g\left(\frac{x_{gsp} - \hat{A}_{\alpha(g),\beta_p(s),p}\,\hat{b}_{gp} - \hat{c}_{gp}}{\hat{\sigma}_{gp}}\right).$$

The output of the XPN algorithm is based on multiple clusterings of the data. The procedure described above is applied 30 times, with different randomly chosen initial centroids for the row and column clusters. The output of the algorithm is the average of the normalized values obtained over the repeated runs.

There are several reasons for averaging the results of multiple clusterings of the combined data matrix. To start, there is unlikely to be a single, 'biologically correct' clustering of the available genes and samples: disease subtypes and gene pathways are not always uniquely defined, and they may exhibit moderate overlap. Multiple clusterings better capture the structure present in this situation. By combining normalization results from multiple clusterings (each of which yields a local minimum of the sum of squares cost function) the XPN algorithm performs a simple form of model averaging. Averaging also controls (minor) instability that may arise from use of the k-means clustering procedure, whose output is dependent on the initial choice of cluster centroids. In this latter respect, XPN is similar in spirit to resampling-based approaches to cluster stability such as those in (Dudoit and Fridlyand, 2002; Tibshirani *et al.*, 2001; Tseng, 2007; Tseng and Wong, 2005).

In principle, the XPN method procedure can be used with any clustering method that produces a pre-specified number of clusters from a given set of vectors, or with resampling, based improvements of such methods. We chose to use k-means clustering because of its simplicity and computational efficiency. The validation study below indicates that

the XPN method performs well, and generally outperforms competing normalization methods, when it is used with basic k-means clustering. The validation results leave open the possibility of further improvements with alternative clustering methods, but a number of experiments with other clustering methods have not produced better results.

In the current implementation of XPN, the number of row and column clusters, $K \geq 1$ and $L \geq 1$, respectively, are fixed in advance, and will depend on the type and dimension of the data under study. In general, L should be large enough to capture principal sample groups or subtypes, and L should be large enough to capture large, homogenous groups of genes. In the numerical experiments below we chose $K = 5$ and $L = 25$. (In practice, XPN is not sensitive to the choice of K and L, see Section 6.1 below). As a general rule we suggest letting the number L of sample clusters be in range of 5–8, and the number K of row clusters to be on the order of 10–30, depending on the number of genes. As an alternative, one may employ a method such as the GAP statistic (Tibshirani *et al*., 2001), implemented as an R function `kmeansGap` in library '`SLmisc`', to assess the number of row and column clusters in the data. Applied to the dataset used in this article, the GAP statistic suggested 4–8 sample clusters and 8–9 gene clusters.

3 OTHER METHODS

We compare XPN with several other normalization methods in the literature. The other methods have previously been applied to batch correction on single platforms, but are well adapted to more general cross-study situations. As a baseline, we standardized each available column (sample) (CS). Beginning with CS data, we median centered each gene in each study and then combined studies. The resulting procedure is denoted by (MC). The MC method is currently used in practice, and in spite of its simplicity, performs relatively well in our validation experiments. We also consider the Empirical Bayes (EB) method (Johnson *et al*., 2007). EB is based on the model

$$x_{gsp} = a_g + \gamma_{gp} + \delta_{gp}\sigma_g\varepsilon_{gsp}, \quad \varepsilon_{gsp} \sim N(0,1)$$

The platform specific parameters γ_{gp} and δ_{gp} are estimated using an EB approach, and are essentially equal to least squares estimates shrunken towards their respective cross-platform means. Other parameters are estimated by gene-wise OLS. The data is then transformed to remove the effects of different γ_{gp} and δ_{gp} across platforms. Finally, we considered the Distance Weighted Discrimination (DWD) method for batch correction (Benito *et al*., 2004), which is based on the DWD method (Marron and Todd, 2004). DWD normalization finds a direction in which the sample-vectors from the two studies are well-separated, and then translates the samples from each study along that direction until their respective families of vectors have significant overlap.

The Probability of Expression (POE) method (Parmigiani *et al*., 2002; Shen *et al*., 2004), transforms each data value into a signed probability in the range $[-1, 1]$. While this transformation is useful for identifying meta-signatures, the resulting data is difficult to compare with normalized values produced by other methods, and we do not include its analysis here.

We note that each of the alternative normalization methods described above is gene-wise affine, that is, for each gene g there exist constants a_g and b_g, with $a_g > 0$, such that $\tilde{x}_{s,g} = a_g x_{s,g} + b_g$. As a result, the correlation between $x_{s,g}$ and $\tilde{x}_{s,g}$ across samples s is 1 for every g. In contrast, XPN seeks to simultaneously borrow strength across genes and samples via linked row and column clusters, and as a result, XPN is not gene-wise affine.

4 DATASETS AND PREPROCESSING

We applied XPN and the methods described above to three existing breast cancer datasets. The first dataset, from (Huang *et al*., 2003), has 89 samples and 8948 genes. Their experiments were performed with Affymetrix GeneChip U95Av2 arrays. The 89 samples were obtained at the Koo Foundation Sun Yat-Sen Cancer Centre (KF-SYSCC), Taipei. The second dataset, which will be referred to as Nederlands Kanker Instituut [Netherlands Cancer Institute (NKI)], comes from (van't Veer *et al*., 2002). It contains 97 samples and 16 360 genes, and was obtained from Netherlands Cancer Institute and Rosetta Inpharmatics-Merck custom designed 25K Agilent oligonucleotide arrays. Most of the NKI patients had stage I or II breast cancer. The third dataset, referred to as University of North Carolina (UNC), is from (Hu *et al*., 2006). It contains 114 samples representing 104 patients and 12 065 genes, and was obtained using 22K Agilent oligonucleotide arrays. The UNC sample set represents an ethnically and geographically diverse cohort.

Initially, locally weighted regression (LOWESS) normalization was applied to the NKI and UNC datasets; robust multi-array analysis (RMA) was used to obtain expression values for the Huang dataset. The raw expression values in each study were then log-2 transformed, and missing values were imputed with 1-nearest neighbor imputation (Troyanskaya *et al*., 2001). Duplicated genes in each datasets were collapsed by median using Entrez Gene ID. There were 6092 common genes among the three platforms. Cross-study normalization methods were applied to this set of common genes, and subsequently to a smaller set of 'intrinsic genes' (Perou *et al*., 2000) identified as playing an active role in the biology of breast cancer.

The next section presents validation results for the set of common genes. The same analysis for the set of intrinsic genes is presented in the supplementary materials. In our experiments, all cross-platform normalization methods worked better on the set of intrinsic genes, and more generally, on smaller gene sets selected using integrative correlation filtering. Prior to cross-study normalization, the log-2 transformed expression values in each platform were column standardized.

5 VALIDATION

Broadly speaking, cross-study normalization methods can be assessed in terms of two competing criteria. Ideally, a normalization method should produce a single unified dataset, in which samples originating in Study 1 are not distinguishable from those originating in Study 2 on the basis of non-biological features. A method that fails to remove systematic differences between studies under-corrects the data. On the other hand, excessive homogenization of the studies (over-correction) can result in a loss of biological information, and the combined dataset may be less useful than its constituents.

The validation results presented below are intended to assess the performance of the methods under study, and their tendency towards over- and under-correction. We begin with the column-standardized datasets X_1, X_2 and X_3. Every method is applied to each pair X_i, X_j with $1 \leq i < j \leq 3$ to produce normalized data $\tilde{X}_{i,j} = [\tilde{X}_i, \tilde{X}_j]$. Validation measures are applied to each pair, and the average value of the measure over the three pairs is reported. For before and after comparisons, we take as a reference the initial data $[X_i, X_j]$ produced by column-standardization (denoted CS in what follows).

In order to better understand the baseline behavior and biases of the normalization methods under consideration, we also apply them to artificial studies obtained by randomly dividing the arrays in a given platform into two pseudo-studies, similar to the procedure in (Gentleman *et al*., 2006). To be more

precise, from a single column-standardized dataset X_i, we produce a pair X_i^1, X_i^2 of pseudo studies by randomly assigning each sample to one of two groups. Different normalization methods are then applied to $[X_i^1, X_i^2]$, yielding a normalized datasets $\tilde{X}_i = [\tilde{X}_i^1, \tilde{X}_i^2]$. Validation measures are applied to compare the pseudo-study and its normalized version. Each of the three available datasets is randomly split 10 times, and the average measure (over splits and studies) is reported.

By design, the data in each pair of pseudo studies come from a common platform and study. Thus we anticipate that a cross-study normalization method should have relatively little effect, beyond its attempt to correct the unavoidable differences that result from splitting the studies in half. While these differences are not negligible, they are typically smaller than the differences between platforms.

5.1 Measures of center and spread

For a given array, the difference between the mean and the median of its values provides a rough measure of its asymmetry in regards to location. After normalization, it is desirable to see a similar distribution of asymmetry across both studies. Figure 2 shows the area between the cumulative distribution function (CDFs) of mean minus median in the two available studies. Graphs for both standard and split-study validation are shown.

A similar comparison for scale can be carried out by considering the SD (σ) and median absolute deviation from median (MAD). For the standard normal distribution with CDF Φ, we have $\sigma = \mathrm{MAD}/\Phi(0.75)$. Figure 3 shows the area between CDFs of $\sigma - \mathrm{MAD}/\Phi(0.75)$ in each of the two available studies. XPN reduces both measures more than the other methods; the split study results show little bias for all methods.

5.2 Average distance to nearest array in another platform

The set of arrays in given platform can be viewed as a set of points in m-dimensional Euclidean space. After normalization it is reasonable to expect that the point 'clouds' associated with distinct platforms will have substantial overlap. (This is one of the motivations behind the DWD normalization method.) To measure overlap in a pair of normalized studies, we measure the Euclidean distance from each array in the first study to the nearest array in the second study, then repeat, swapping the roles of the studies, and finally average the results. The results are presented in Figure 4, with smaller values indicating greater overlap. XPN and EB reduce the average distance more than other methods. The split study results show little bias for all the methods.

5.3 Correlation with column standardized data

The previous validation measures assess the similarity of two datasets after normalization. A natural way to see how much the normalization methods affect the data is to calculate correlation between the data matrices before and after normalization, where 'before' is represented by CS. This measure does not by itself support a given normalization method, but in choosing between methods that perform similarly across other validation measures, the method that has less effect on the data should clearly be preferred. The average correlation of arrays before and after normalization for the different methods under study is shown in Figure 5. Median centering has the least effect on the data; the other three methods yield average correlations close to 0.8, with XPN lying between DWD and EB. Table 1 shows the average

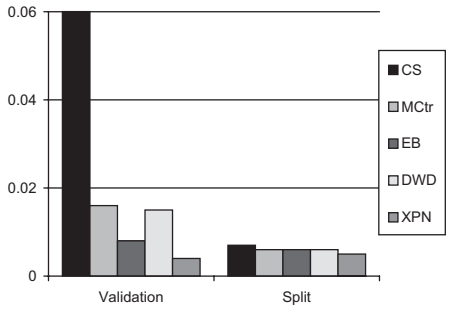

Fig. 2. Area between the CDFs of array mean minus array median across platforms. Lower values indicate greater similarity of datasets after normalization.

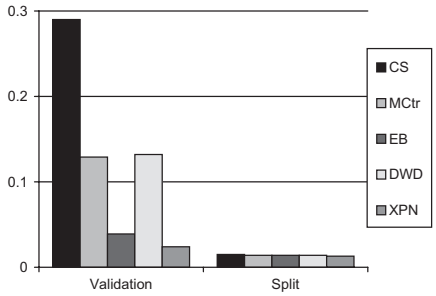

Fig. 3. Area between the CDFs of $\sigma - \mathrm{MAD}/\Phi(0.75)$ for arrays of different platforms. Lower values indicate greater similarity of datasets after normalization.

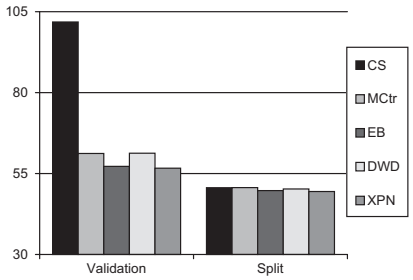

Fig. 4. Average L_2 distance from the samples of one study to the nearest sample from the other study. Lower values indicate greater similarity of the study point 'clouds' after normalization.

correlation of genes before and after normalization, averaged over both studies. As discussed above, all methods but XPN perform normalization by transforming each gene in an affine fashion; thus the gene correlation for these methods is equal to 1. Similar remarks apply to the integrative correlation and t-statistic measures described below. The gene correlation for XPN is 0.99, with a split-study value of 0.996.

5.4 Global integrative correlation

Integrative correlation (Cope *et al.*, 2007) is a means of identifying genes with concordant expression in different studies. Let $\boldsymbol{r}_1(g)$, $\boldsymbol{r}_2(g)$ be the gth row of $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$, respectively. The global integrative correlation (GIC) between X_1 and X_2 is the correlation between

$$(\mathrm{Corr}(r_1(g), r_1(g')) : g, g' \in G) \quad (\mathrm{Corr}(r_2(g), r_2(g')) : g, g' \in G)$$

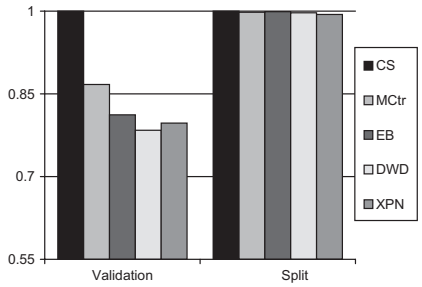

Fig. 5. Average correlation of arrays with their values before normalization (CS). Larger values indicate less modification of the data by the normalization procedure.

Table 1. Gene-based correlation measures

	CS, MCtr, EB, DWD	XPN	Change	Change (%)
Avg gene corr w/CS				
Validation	1.000	0.990	−0.010	−1.0%
Split	1.000	0.996	−0.04	−0.4%
GIC				
Valid'n	0.255	0.338	0.083	33%
Split	0.556	0.597	0.041	7%
ER *t*-stat correlation				
Valid'n	0.312	0.451	0.139	45%
Split	0.446	0.543	0.096	22%

The first row shows the average correlation of genes with their value before normalization (CS). The second row shows global integrative correlation (GIC) between platform pairs after normalization, with larger values indicating better concordance between platforms. The third row shows the average correlation of ER *t*-statistics across platforms, with larger values indicating better concordance.

here regarded as vectors with $|G|^2$ components. High values of $IC(g)$ indicate good concordance between the values in Studies 1 and 2. GICs for different normalization methods are shown in Table 1. The results for CS shows that the average GIC between halves of the same platform (0.556) is much higher than average GIC between different studies (0.255). XPN is the only method among those considered that affects GIC. It increases GIC by 33% to 0.338 in cross-study validation, well below the split-study level (0.556). XPN increases GIC between pseudo studies by a relatively small 7%.

Each tumor sample in the datasets under consideration has an associated, clinically based ER status (ER+ or ER−). We next consider several validation measures based on this biological information. The Huang dataset has only 15 ER negative samples out of 89, making its split-study results unstable, and is therefore excluded from the split study analysis of the ER-based validation measures.

5.5 Correlation of *t*-statistics

For each platform, *t*-statistics measuring the association of gene-expression values with the ER status are calculated. Ideally, the vectors of *t*-statistics for different platforms should become more concordant after platform normalization. Table 1 shows the Pearson correlation between the *t*-statistics for ER status for different normalization methods. (Results for rank correlation are similar.) As expected, the average correlation of *t*-statistics is higher in split study (0.446) than between platforms (0.312). XPN increases the correlation of *t*-statistics between platforms by 45% to 0.451. In split-study validation it increased correlation by roughly 22%. Overall, XPN has greater effect than the other methods considered. The correlation measurements above show that, on average, XPN does not make dramatic changes in the rows of the data matrices, and we believe that much of the split study increase in *t*-statistic correlation is due to inherent differences between the randomly selected pseudo studies.

5.6 Cross platform prediction of ER status

If we regard ER status as a binary phenotype, we may explore misclassification rates associated with its prediction. Ideally, combining labeled studies via cross-platform normalization should lead to lower misclassification rates on test datasets. To test the compatibility of different studies after normalization in regards to classification, we treated the data from one study as a training set, and the data from the other study as a test set, and vice versa. Lower error rates indicate better concordance. Classification was performed using two methods: nearest shrunken centroids prediction analysis for microarrays (PAM) (Tibshirani *et al*., 2002) and support vector machines (SVM) (Boser *et al*., 1992; Cortes and Vapnik, 1995). The results are presented in Figures 6 and 7. As can be seen, all of the normalization methods greatly reduce cross-platform prediction error, with the minimum error achieved by XPN. In the split-study test, none of the methods produces significant reductions in classification error, as expected.

One might also be interested in the 5- or 10-fold cross-validation prediction error rate on the combined studies. However, none of the normalization methods has a significant effect on the cross-validated classification error. This appears to arise from the fact that, in cross validation, the classification methods are trained on elements of both platforms, and the distinguishing features of ER status are strong enough to enable the methods to perform well without prior normalization.

5.7 Preservation of significant genes

Lastly, we consider gene lists produced using ER-based *t*-statistics at a nominal 0.1% significance threshold. Let L_i be the list of genes in Study $i=1,2$, and let $L_{1,2}$ be the list produced at the same nominal 0.1% level from the combined data $\tilde{X}$. Ideally, genes that are in both L_1 and L_2 should appear in $L_{1,2}$, and most genes that appear in at least one of the single study lists will be in the joint list. We assess these two types of overlap by measures $V_1=|(L_1\cap L_2)\cap L_{1,2}|/|L_1\cap L_2|$ and $V_2=|(L_1\cup L_2)\cap L_{1,2}|/|L_1\cup L_2|$, respectively. The results are presented in Table 2. The value of V_1 is 1 for all normalization methods except CS, showing the importance of platform normalization. The V_2 measure is increased by all methods, with the greatest increase achieved by MC and DWD.

6 FURTHER DISCUSSION OF XPN

6.1 Stability with respect to *K* and *L* parameters

To test stability of XPN with respect to the numbers K and L of row and column clusters, we applied XPN with a range of parameters. For $L=5$ we tried $K=2$, 10, 20, 25, 30, 50, 100, 500,

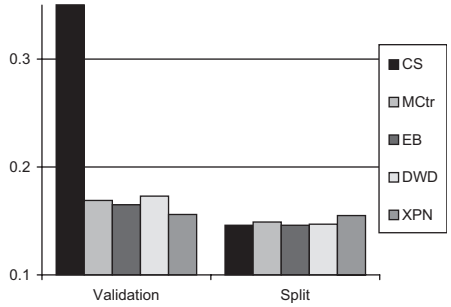

Fig. 6. Cross-platform prediction error of the PAM (nearest shrunken centroids) classifier. Smaller values indicate better concordance between platforms.

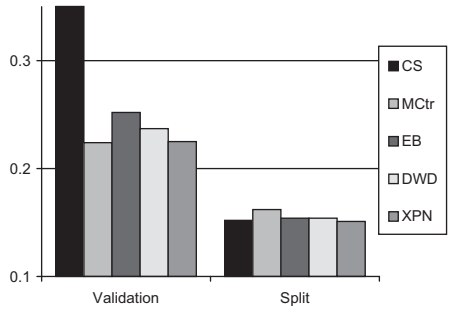

Fig. 7. Cross-platform prediction error of the SVM (Support Vector Machine) classifier. Smaller values indicate better concordance between platforms.

and for $K = 25$ we tried $L = 2, 4, 5, 6, 7, 8, 10$. The results (presented in the Supplementary Materials) indicate that XPN is generally insensitive to the choice of the K and L. However, we do see (expected) degradation of performance in situations where K or L is below four, in which case the clustering is too coarse to adequately capture homogenous blocks of samples or genes. At the other extreme, when L is large, one finds column clusters containing samples from a single platform. For such clusters the algorithm cannot combine information across platforms, and its results will be degraded accordingly. (In its current implementation, XPN excludes such clusterings from the average that forms its output.) Values of K larger than 25 make the algorithm slower and do not substantially improve its performance.

6.2 Stability of XPN output

The XPN algorithm averages the normalization results from B row/column clusterings. To assess the stability of XPN, we calculated the SD of each element in the normalized matrix over the $B = 100$ runs of the basic procedure. The average SD (over all elements and platform pairs) was 0.004. In contrast, the average SD of the entries of the normalized matrices was 0.79. Thus, the variability of the normalized entries due to random clusterings was, on average, two orders of magnitude less than the variability between the final normalized entries.

7 CONCLUSION

The increasing number and public availability of large-scale gene-expression studies provides impetus for cross-study analyses that combine existing, and potentially new, datasets. Properly combined datasets give researchers more power for biological and statistical analysis. In this article we propose a new, block model-based method, called XPN, for cross-platform normalization. The block model distinguishes XPN from other platform normalization methods such as DWD and EB, which are gene-wise linear.

We propose a set of validation measures for comparison of different normalization methods. The validation measures can be roughly split in two groups. One group assesses the ability of normalization methods to remove systematic differences across platforms, while the other measures how much the data is transformed by normalization procedures. Based on the proposed validation measures, XPN successfully combined three existing breast cancer datasets without incurring substantial overfitting. In particular, cross-platform ER prediction error rates indicate that XPN successfully preserved biological information while removing systematic differences between platforms.

The XPN method has three parameters: the number of row and column clusters (K and L) and the number of basic iterations B. Our experiments indicate that the results of XPN are robust to the choice of K and L (see Section 6.1). The analysis in Section 6.2 suggests setting $B = 30$ is sufficient for stable output.

Table 2. Measures of gene-list preservation

	CS	MCtr	EB	DWD	XPN
V_1					
Valid'n	0.826	1	1	1	1
Split	1	1	1	1	1
V_2					
Valid'n	0.646	0.895	0.774	0.887	0.759
Split	0.876	0.867	0.875	0.878	0.870

V_1 (V_2) is the fraction of genes from the intersection (union) of platform-specific gene lists present in the list produced from the combined data $\tilde{X}$ at 0.1% level.

ACKNOWLEDGEMENTS

Funding for this work was provided by National Science Foundation Grant (DMS 0406361) to A.B.N. and A.A.S.; National Cancer Institute Breast SPORE program to University of North Carolina at Chapel Hill (P50-CA58223-09A1) to C.M.P. and C.F.; National Cancer Institute (RO1-CA-101227-01) to C.M.P. and C.F.; by the Breast Cancer Research Foundation. The authors would like to thank J.S. Marron for helpful conversations and suggestions regarding the validation procedures discussed in the article.

Conflict of Interest: none declared.

REFERENCES

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

Bolstad,B. *et al.* (2003) A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, **19**, 185–193.

Boser,B. *et al.* (1992) A training algorithm for optimal margin classifiers. In *Proceedings of the Fifth Annual Workshop on Computational Learning theory*. Berkeley Electronic Press, Berkeley, USA, pp. 144–152.

Choi,J. *et al.* (2003) Combining multiple microarray studies and modeling interstudy variation. *Bioinformatics*, **19**, 84–90.
Cope,L. *et al.* (2007) The Integrative Correlation Coefficient: A Measure of Cross-study Reproducibility for Gene Expressionea Array Data. *Working Papers, Deptartment of Biostatistics, Johns Hopkins University*, Berkeley Electronic Press, Berkeley, USA, p. 152.
Cortes,C. and Vapnik,V. (1995) Support-vector networks. *Mach. Learn.*, **20**, 273–297.
Dudoit,S. and Fridlyand,J. (2002) A prediction-based resampling method for estimating the number of clusters in a dataset. *Genome Biol.*, **3**, 1–21.
Garrett-Mayer,E. *et al.* (2004) Cross-study Validation and Combined Analysis of Gene Expression Microarray Data. *Working Papers*, *Deptartment of Biostatistics*, *Johns Hopkins University*. Berkeley Electronic Press, Berkeley, US, p. 65.
Garrett-Mayer,E. *et al.* (2007) Cross-study validation and combined analysis of gene expression microarray data. *Biostatistics*, kxm033.
Gentleman,R. *et al.* (2006) Meta-analysis for microarray experiments. *Bioconductor*. http://www.bioconductor.org/packages/bioc/vignettes/GeneMeta/inst/doc/GeneMeta.pdf.
Ghosh,D. *et al.* (2003) Statistical issues and methods for meta-analysis of microarray data: a case study in prostate cancer. *Funct. Integr. Genomics*, **3**, 180–188.
Huang,E. *et al.* (2003) Gene expression predictors of breast cancer outcomes. *The Lancet*, **361**, 1590–1596.
Hu,Z. *et al.* (2006) The molecular portraits of breast tumors are conserved across microarray platforms. *BMC Genomics*, **7**, 96.
Irizarry,R. *et al.* (2003a) Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.*, **31**, e15.
Irizarry,R. *et al.* (2003b) Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, **4**, 249–264.
Johnson,W.E. *et al.* (2007) Adjusting batch effects in microarray expression data using empirical Bayes methods. *Biostatistics*, **8**, 118–127.
Marron,J.S. *et al.* (2007) Distance weighted discrimination. *Journal of the American Statistical Association*.
Parmigiani,G. *et al.* (2002) A statistical framework for expression-based molecular classification in cancer. *J. R. Stat. Soc. Series B (Stat. Method.)*, **64**, 717–736.
Parmigiani,G. *et al.* (2004) A cross-study comparison of gene expression studies for the molecular classification of lung cancer. *Clin. Cancer. Res.*, **10**, 2922–2927.
Perou,C. *et al.* (2000) Molecular portraits of human breast tumours. *Nature*, **406**, 747–752.
Rhodes,D. *et al.* (2002) Meta-analysis of microarrays: interstudy validation of gene expression profiles reveals pathway dysregulation in prostate cancer. *Cancer Res.*, **62**, 4427–4433.
Rhodes,D. *et al.* (2004) Large-scale meta-analysis of cancer microarray data identifies common transcriptional profiles of neoplastic transformation and progression. *Proc. Nat. Acad. Sci.*, **101**, 9309–9314.
Shen,R. *et al.* (2004) Prognostic meta-signature of breast cancer developed by two-stage mixture modeling of microarray data. *BMC Genomics*, **5**, 94.
Tibshirani,R. *et al.* (2001) Estimating the number of clusters in a data set via the gap statistic. *J. R. Stat. Soc. Ser. B (Stat. Method.)*, **63**, 411–423.
Tibshirani,R. *et al.* (2002) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Nat. Acad. Sci.*, **99**, 6567.
Troyanskaya,O. *et al.* (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*, **17**, 520–525.
Tseng,G. and Wong,W. (2005) Tight clustering: a resampling-based approach for identifying stable and tight patterns in data. *Biometrics*, **61**, 10–16.
Tseng,G. (2007) Penalized and weighted K-means for clustering with scattered objects and prior information in high-throughput biological data. *Bioinformatics*, **23**, 2247.
van't Veer,L. *et al.* (2002) Gene-expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.
Yang,Y. *et al.* (2002) Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Res.*, **30**, e15.

APPENDIX: MAXIMUM LIKELIHOOD ESTIMATION OF THE MODEL

The XPN algorithm estimates the parameters of the Model (2) using maximum likelihood approach. The model has distinct sets of parameter for different gene clusters and different platforms. Thus the problem of parameter estimation can be split into $2K$ smaller tasks. Fix $i \in \{1, \ldots, K\}$ and $p \in \{1, 2\}$. The log-likelihood function associated with gene group i and platform p can be expressed as

$$2l_{i,p} = C + \sum_{(s,g):\alpha(g)=i} \ln(\sigma_{gp}^2) + \sum_{(s,g):\alpha(g)=i} (x_{gsp} - A_{i,\beta_p(s),p}\, b_{gp} - c_{gp})^2/\sigma_{gp}^2$$

To ensure identifiability of the coefficients $\{A_{ijp}\}$ and $\{b_{gp}\}$, we set

$$\sum_{j=1}^{L} A_{ijp} = 0, \quad \sum_{j=1}^{L} A_{ijp}^2 = L \text{ and } \sum_{g:\alpha(g)=i} b_{gp} > 0$$

The parameters A_{ijp}, b_{gp}, c_{gp} and σ_{gp}^2 are chosen to maximize the log-likelihood. To find them we take first derivative of the log-likelihood with respect to these parameters and set the result equal to zero:

$$\mathrm{d}l/\mathrm{d}c_{gp} = 0 = \sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})$$

$$\mathrm{d}l/\mathrm{d}b_{gp} = 0 = \sum_s A_{i,\beta_p(s),p}(x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})$$

$$\mathrm{d}l/\mathrm{d}A_{ijp} = 0 = \sum_{(g,s):\beta_p(s)=j} b_g(x_{gsp} - A_{ijp} b_{gp} - c_{gp})/\sigma_g^2$$

$$\mathrm{d}l/\mathrm{d}\sigma_{gp}^2 = 0 = n_p\sigma_{gp}^{-2} - \sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2 \sigma_{gp}^{-4}$$

Here and in what follows, each sum is taken over all the genes in the ith cluster. The above equations simplify to

$$c_{gp} = \bar{x}_{gp} - n^{-1} b_{gp} \sum_j A_{ijp} n_{jp}$$

$$b_{gp} = \left[\sum_s A_{i,\beta_p(s),p}(x_{gsp} - c_{gp})\right] / \left[\sum_j A_{ijp}^2 n_{jp}\right]$$

$$A_{ijp} = \left[\sum_{(g,s):\beta_p(s)=j} (x_{gsp} - c_{gp}) b_{gp}/\sigma_{gp}^2\right] / \left[n_{jp} \sum_g b_{gp}^2/\sigma_{gp}^2\right]$$

$$\sigma_{gp}^2 = n_p^{-1} \sum_s (x_{gsp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2$$

Define the sample mean and variance of the expression values of a gene in sample block j:

$$\bar{x}_{gjp} = n_{jp}^{-1} \sum_{s:\beta_p(s)=j} x_{gsp}, \quad s_{gjp}^2 = n_{jp}^{-1} \sum_{s:\beta_p(s)=j} (x_{gsp} - \bar{x}_{gjp})^2$$

This allows further simplification of the equations

$$c_{gp} = n_p^{-1} \sum_j (\bar{x}_{gjp} - b_{gp} A_{ijp}) n_{jp}$$

$$b_{gp} = \left[\sum_j A_{ijp}(\bar{x}_{gjp} - c_{gp}) n_{jp}\right] / \left[\sum_j A_{ijp}^2 n_{jp}\right]$$

$$A_{ijp} = \left[\sum_g b_{gp}(\bar{x}_{gjp} - c_{gp})/\sigma_{gp}^2\right] / \left[\sum_g b_{gp}^2/\sigma_{gp}^2\right]$$

$$\sigma_{gp}^2 = n_p^{-1} \sum_j \left[(\bar{x}_{gjp} - A_{i,\beta_p(s),p} b_{gp} - c_{gp})^2 + s_{gjp}^2\right] n_{jp}$$

There is no closed form solution for this system of equations. To obtain the estimates, the formulas are applied iteratively until convergence of the parameters. Each iteration increases the log-likelihood and the limit values satisfy all first order conditions.