Cross-Platform Normalization Methods Implemented in CONOR

Terminology and Notation

Before describing the different cross-platform normalization methods currently available, it is necessary to define some terminology and notation that will be used throughout the remainder of this work. The terminology required is that associated with the levels of variation present in a microarray experiment:

Target A nucleic acid species of interest.

Gene Another commonly used term for target, especially when the target is an mRNA species associated with a particular gene of interest.

Probe The component of a microarray designed to detect a particular target. Usually, a probe is made up of identical oligonucleotides attached to a small region of substrate.

Treatment The experimental protocols or natural conditions to which a biological specimen of interest has been subjected. For example, the control and experimental sets in a biological experiment are treatment groups.

Sample A homogeneous solution of nucleic acids on which assays can be performed. Generally a sample is extracted from a member of a treatment group. If multiple samples are extracted from different members of the same treatment group, those samples are considered biological replicates.

Assay A particular sample measured by a particular microarray. If multiple assays are performed on the same sample, those assays are considered technical replicates.

Platform A particular type of microarray. To be considered the same platform, two microarrays must be the same model, produced by the same manufacturer, and used and analyzed in the same manner on samples originating from the same species.

Each cross-platform normalization method is designed to use data from assays conducted on two platforms and produce data equivalent to the data that might be produced by performing the same assays on a single platform. That is, the

goal of cross-platform normalization is to remove all platform effects from a data set while retaining all treatment and sample effects. For convenience, I will use the same notation in describing all the methods. Let x_{gap} be the expression measurement produced by a particular gene, g; assay, a; and platform, p. The matrix of all data for a particular platform and the vector of all data for a particular platform and assay will be denoted $x_{\cdot\cdot p}$ and $x_{\cdot ap}$, respectively. The number of genes will be denoted m and the number of assays for each platform will be denoted n_p. The cross-platform normalized expression corresponding to x_{gap} will be denoted y_{gap}^{method} for the cross-platform normalization procedure *method*, or simply y_{gap} when the normalization method is unambiguous. The gene ranking function will be denoted $R(x_{gap})$, and will give the rank of x_{gap} in $x_{\cdot ap}$. The n^{th} order statistic for a particular assay and platform will be denoted $O_n(x_{\cdot ap})$. Note that the gene rank and order functions are inverses, such that $x_{gap} = O_{R(x_{gap})}(x_{\cdot ap})$.

XPN

XPN [10] is the most recently developed cross-platform normalization technique and the most complex. XPN is based on a *block linear* model of microarray data, where the blocks are based on estimated gene and assay clusters. The number of gene clusters, K, and assay clusters, L, must be determined by the user. Clustering of genes and assays is performed using a modified k-means algorithm, with the usual distance metric replaced with the value $1 - \text{cor}([x_{g_1 \cdot p_1}; x_{g_1 \cdot p_2}], [x_{g_2 \cdot p_1}; x_{g_2 \cdot p_2}])$ for gene clustering and $1 - \text{cor}\left(x_{\cdot a_1 p(a_1)}, x_{\cdot a_2 p(a_2)}\right)$ for assay clustering, where $[W; Z]$ represents the concatenation of the vectors W and Z and $p(a)$ is the platform associated with assay a. Let $\alpha(g)$ be the gene cluster assigned to gene g and $\beta(a)$ be the assay cluster assigned to assay a. The XPN model is

$$x_{gap} = A_{\alpha(g)\beta(a)p} b_{gp} + c_{gp} + \sigma_{gp}\epsilon_{gap}, \tag{1}$$

where $\epsilon_{gap} \sim \text{Normal}(0, 1)$ is a random variable. The parameters $A_{\alpha(g)\beta(a)p}$, b_{gp}, c_{gp}, and σ_{gp} are estimated by maximum likelihood under the constraints

$$\sum_{a=1}^{L} A_{\alpha(g)\beta(a)p} = 0, \tag{2}$$

$$\sum_{a=1}^{L} \left(A_{\alpha(g)\beta(a)p}\right)^2 = L, \tag{3}$$

$$\sum_{\{i:\alpha(g)=i\}} > 0, \tag{4}$$

to ensure identifiability. Once the parameters have been fitted, the normalized data are generated according to

$$y_{gap} = \hat{A}_{\alpha(g)\beta(a)} \hat{b}_g + \hat{c}_g + \hat{\sigma}_g \left(\frac{x_{gap} - \hat{A}_{\alpha(g)\beta(a)p} \hat{b}_{gp} - \hat{c}_{gp}}{\hat{\sigma}_{gp}} \right) \tag{5}$$

where $\hat{A}_{\alpha(g)\beta(a)}$, $\hat{b}_g$, $\hat{c}_g$, and $\hat{\sigma}_g$ are weighted averages of the maximum likelihood estimates $\hat{A}_{\alpha(g)\beta(a)p}$, $\hat{b}_{gp}$, $\hat{c}_{gp}$, and $\hat{\sigma}_{gp}$, respectively.

Distance Weighted Discrimination

DWD [1, 7] is based on the construction of a linear classifier between the assays of one platform and assays of the other. A linear classifier is defined by a hyperplane separating $\mathbb{R}^n$ into two regions. The orientation and position of the separating hyperplane can be specified by a unit normal vector, w, and a scalar, β. In DWD, the hyperplane is chosen as the solution to the optimization problem

$$\min_{w,\beta,\xi} \sum_{g,a,p} \left(\frac{1}{r_{ap}} + c\xi_{ap} \right),$$

$$s.t. \quad \|w\|^2 \leq 1, \tag{6}$$

$$\xi \geq 0,$$

$$r_{ap} \geq 0 \tag{7}$$

where the residuals r_{ap} are given by

$$r_{ap} = (-1)^p \left(\langle x_{\cdot ap}, w \rangle + \beta \right) + \xi_{ap}, \tag{8}$$

the operation $\langle \cdot, \cdot \rangle$ is the vector inner product, and c is a scalar penalty parameter for the error factor ξ_{ap}. The error factor is included to allow for the possibility that the data sets for the two platforms are not linearly separable. In such a case, the penalty factor c represents a sort of weight given to misclassification. It is assumed that platform indicators have been chosen such that one is odd and the other even. For example, Affymetrix data may be assigned $p = 1$ while Illumina data are assigned $p = 2$. The penalty parameter is determined based on the inverse of the median pairwise distance between the assays of the two platforms. Specifically, the penalty is given by

$$c = \frac{100}{\text{median} \left\{ \|x_{\cdot ap_1} - x_{\cdot ap_2}\| : p_1 \neq p_2 \right\}}. \tag{9}$$

Once w and β have been determined, the normalized data are produced by shifting the data from each platform toward the separating hyperplane. The magnitude of the location shift is determined by the average projection of the assay vectors of each platform into the normal vector w. The normalized data are

$$y_{gap} = x_{gap} + w \frac{1}{n_p} \sum_{\{j : \exists x_{\cdot jp}\}} \langle x_{\cdot jp}, w \rangle . \tag{10}$$

DWD can be thought of as defining an implicit model of platform effects as location parameters, with

$$x_{gap} = \Gamma_{ga} + \eta_p, \tag{11}$$

where η_p and Γ_{ga} are platform effects and all other effects, respectively.

Empirical Bayes

The EB method is based on a straight-forward model of microarray data given by

$$x_{gap} = \alpha_g + D\beta_g + \gamma_{gp} + \delta_{gp}\epsilon_{gap}, \tag{12}$$

where $\epsilon_{gap} \sim \text{Normal}\left(0, \sigma_g^2\right)$, D is a design matrix, and β_g is a vector of regression coefficients. The method makes use of distributional assumptions on the parameters to borrow information across genes when estimating platform effects, making the parameter estimation procedure much less straight-forward than the model (12) would suggest. The method was not originally designed for cross-platform normalization, but has been applied to cross-platform normalization and is available for that purpose as part of the ArrayMining service. My description here is based on re-interpreting the batch effects proposed by the method's original authors as platform effects. Because this work is concerned with cross-platform normalization in the absence of treatment and sample information, the design matrix term is not used. The model used here is

$$x_{gap} = \alpha_g + \gamma_{gp} + \delta_{gp}\epsilon_{gap}, \tag{13}$$

which is identical to (12) except that the design term has been eliminated.

Estimation of the model parameters is a multi-step process. Initial estimates $\hat{\alpha}_g$ and $\hat{\gamma}_{gp}$ are calculated by a constrained least squares approach with $\sum_{p=1}^{2} n_p \hat{\gamma}_{gp} = 0$. The initial estimates are then used to produce a standardized data set

$$z_{gap} = \frac{x_{gap} - \hat{\alpha}_g}{\hat{\sigma}_g^2}, \tag{14}$$

where $\hat{\sigma}_g^2$ is a pooled variance estimate

$$\hat{\sigma}_g^2 = \frac{1}{N} \sum_{p=1}^{2} \sum_{a=1}^{n_p} \left(x_{gap} - \hat{\alpha}_g - \hat{\gamma}_{gp}\right)^2, \tag{15}$$

where N is the total number of assays from both platforms.

The standardized data are assumed to be distributed $z_{gap} \sim \text{Normal}\left(\hat{\gamma}_{gp}, \hat{\delta}_{gp}^2\right)$. Based on this assumption, a second estimate, $\tilde{\gamma}_{gp}$, is produced from the standardized data by

$$\tilde{\gamma}_{gp} = \frac{1}{m} \sum_{a=1}^{n_p} z_{gap}, \tag{16}$$

and an estimate $\tilde{\delta}_{gp}$ is produced by

$$\tilde{\delta}_{gp} = \frac{1}{n_p - 1} \sum_{a=1}^{n_p} \left(z_{gap} - \tilde{\gamma}_{gp}\right)^2. \tag{17}$$

Prior distributions are assumed for the estimates thus produced:

$$\tilde{\gamma}_{gp} \sim \text{Normal}\left(Y_p, \tau_i^2\right), \tag{18}$$

$$\tilde{\delta}_{gp}^2 \sim \text{InverseGamma}\left(\lambda_p, \theta_p\right). \tag{19}$$

The hyper-parameters Y_p, τ_p^2, λ_p, and θ_p are estimated using the method of moments by

$$\tilde{Y}_p = \frac{1}{m}\sum_{g=1}^{m}\tilde{\gamma}_{gp}, \tag{20}$$

$$\tilde{\tau}_p^2 = \frac{1}{m-1}\sum_{g=1}^{m}\left(\tilde{Y}_p - \tilde{\gamma}_{gp}\right)^2, \tag{21}$$

$$\tilde{\lambda}_p = \frac{\frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2 + \frac{2}{m-1}\sum_{g=1}^{m}\left(\tilde{\delta}_{gp}^2 - \frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)^2}{\frac{1}{m-1}\sum_{g=1}^{m}\left(\tilde{\delta}_{gp}^2 - \frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)^2}, \tag{22}$$

$$\tilde{\theta}_p = \frac{\left(\frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)^3 + \left(\frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)\left(\frac{1}{m-1}\sum_{g=1}^{m}\left(\tilde{\delta}_{gp}^2 - \frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)^2\right)}{\frac{1}{m-1}\sum_{g=1}^{m}\left(\tilde{\delta}_{gp}^2 - \frac{1}{m}\sum_{g=1}^{m}\tilde{\delta}_{gp}^2\right)^2} \tag{23}$$

Final estimates γ_{gp}^* and δ_{gp}^{2*} are obtained by applying Bayes' Theorem to the assumed normal distribution of z_{gap} to obtain a posterior distribution for γ_{gp} conditioned on the standardized data and denoted by

$$\gamma_{gp} \sim \pi\left(\gamma_{gp} \mid z_{gp}, \delta_{gp}^2\right), \tag{24}$$

which is shown to be normally distributed. That distribution is used to derive the system defining the final estimators as

$$\begin{cases} \gamma_{gp}^* = \frac{n_p\tilde{\tau}^2\tilde{\gamma}_{gp}^2 + \delta_{gp}^{2*}\tilde{Y}_p}{n_p\tilde{\tau}_p^2 + \delta_{gp}^{2*}}, \\ \delta_{gp}^{2*} = \frac{\tilde{\theta}_p + \frac{1}{2}\sum_{a=1}^{n_p}\left(z_{gap} - \gamma_{gp}^*\right)^2}{\frac{n_p}{2} + \tilde{\lambda}_p - 1}. \end{cases} \tag{25}$$

Finally, the cross-platform normalized data is obtained by

$$y_{gap} = \frac{\hat{\sigma}_g^2}{\delta_{gp}^*}\left(z_{gap} - \gamma_{gp}^*\right) + \hat{\alpha}_g, \tag{26}$$

after solution of the system (25).

Median Rank Scores

MRS [11] is a comparatively simple approach to cross-platform normalization and is extremely similar to QN and DisTran. In MRS, the normalized data is given by

$$\begin{cases} y_{ga1} = x_{ga1} \\ y_{ga2} = O_{R(x_{ga2})}\left(\text{GeneMedians}\left(x_{\cdot\cdot 1}\right)\right) \end{cases} \tag{27}$$

where GeneMedians $(x_{..1})$ is the vector of median expression values the genes in the data matrix $x_{..1}$. The outcome of MRS is that the distribution of each assay for platform 2 is identical to the distribution of the gene medians of platform 1. Note that the distributions of expression values within each assay may still differ within the platform 2 data, unless they have already been fixed by some other pre-processing step as they have been in this study.

Quantile Discretization

QD [11] is similar to quantile normalization in that it results in an identical distribution for each assay. Instead of generating a distribution for the assays from the data, QD uses an equal frequency binning procedure to create a modified discrete uniform distribution. That modified distribution is given by the probability mass function (pmf)

$$P_Z(z \mid b) = \begin{cases} \frac{1}{b} I_{\{1..b\}}\left(z + \frac{b+1}{2}\right), & b \text{ odd} \\ \frac{1}{b} I_{\{1..b\}}\left(z + \frac{b-1}{2}\right) + \frac{1}{b} I_{\{1\}}(|z|), & b \text{ even,} \end{cases} \tag{28}$$

where b is the number of bins and $I_{\{\alpha..\beta\}}(z)$ is the identity function for the set of integers from α to β, inclusive. The distribution is identical to a discrete uniform distribution except that the middle two bins are merged when the number of bins, b, is even. Let $G(z \mid b)$ be the cumulative distribution function (cdf) associated with P_Z. The QD transformation is given by

$$y_{gap} = G^{-1}\left(\hat{F}_a(x_{gap})\right) \tag{29}$$

where $\hat{F}_a$ is the empirical cdf for assay a. The result of QD is that data for each assay are distributed into b bins, each containing the same number of genes, except the middle bin in the case of b being even which contains twice as many. The values of the bins are shifted so that the middle bin has the expression value of 0 and the resulting expression values range from $-\left(\lceil \frac{b}{2} \rceil - 1\right)$ to $\lceil \frac{b}{2} \rceil - 1$, where $\lceil \cdot \rceil$ is the ceiling function.

Normalized Discretization

NorDi [8] was developed as part of the GenMiner program [9] and is available as part of ArrayMining as a cross-study normalization technique (called cross-platform normalization here). NorDi is based on a process of removing outliers from each assay under the assumption of normality, determining the mean and variance for each assay from the trimmed samples, and then discretizing all expression values into $\{-1, 0, 1\}$ based on a z-score cut-off derived from the estimated mean and variance.

The presence of outliers is determined by the Grubb test [4] and confirmed by the Jarque-Bera statistic [2]. Specifically, a trimmed data set is produced for each assay by the following algorithm:

Let $n := 0$

Let $d_0 := x_{\cdot ap}$
Let JB_0 be the Jarque-Bera statistic for d_n
loop
  Set $n := n + 1$
  Let p be the p-value of the Grubbs' statistic for d_{n-1}
  if $p < pvalue$ **then**
    Let d_n be a data set equivalent to d_{n-1} with the most outlying data point removed
  else
    Let $d_n := d_{n-1}$
  end if
  Let JB_n be the Jarque-Bera statistic for d_n
  if $d_n = d_{n-1}$ and $JB_n > JB_{n-1}$ **then**
    return d_n
  else if $JB_n < JB_{n-1}$ **then**
    Set $d_n := d_{n-1}$
  end if
end loop

where *pvalue* is a pre-determined cut-off value, which for all my experiments was set to 0.01, and the most outlying data point is defined as the point with the greatest absolute distance from the sample mean.

The normalized data is then given by

$$y_{gap} = \begin{cases} -1, & \frac{x_{gap} - \hat{\mu}_{ap}}{\hat{\sigma}_{ap}} \leq -Z_{\alpha/2} \\ 1, & \frac{x_{gap} - \hat{\mu}_{ap}}{\hat{\sigma}_{ap}} \geq Z_{\alpha/2} \\ 0, & \text{otherwise} \end{cases} \tag{30}$$

where $\hat{\mu}_{ap}$ and $\hat{\sigma}_{ap}$ are the sample mean and standard deviation from the trimmed data from assay a and platform p, α is a pre-determined cut-off set at 0.05 for all of my work here, and $Z_{\alpha/2}$ is the z-score associated with the one-sided p-value $\alpha/2$. Note that the outlying values removed from the trimmed set are still included in the transformation, and that they are scored based on statistics estimated from the trimmed data set from which they were removed.

Distribution Transformation

DisTran [5] is quite similar to the MRS method. For DisTran the normalized data are given by

$$\begin{cases} y_{ga1} & = x_{ga1}, \\ y_{ga2} & = O_{R(x_{ga2})}(z), \end{cases} \tag{31}$$

where

$$z = \frac{1}{L} \sum \text{GeneMeans}(x_{iT_i1}), \tag{32}$$

where L is the total number of treatment groups and T_i is the set of all assays belonging to treatment group $i$for platform 1. Because I am studying normalization in the absence of treatment information, treatment groups are estimated by

k-means clustering as in the XPN method. For all of my experiments, the number of treatment groups to estimate, L, matched the true number of treatment groups.

Gene Quantiles

GQ is not described in any publication, but is implemented as part of WebArrayDB [12] for cross-platform normalization of microarray data. GQ is identical to MRS but for the inclusion of a platform dependent location shift, which ensures that the median for each gene in platform 1 is equal to the median for the corresponding gene in platform 2. Specifically, the transformation is given by

$$\begin{cases} y_{ga1} & = x_{ga1}, \\ y_{ga2} & = O_{R(x_{ga2})} \left(\text{GeneMedians} \left(x_{\cdot\cdot 1} \right) \right) - \text{GeneMedians} \left(x_{\cdot\cdot 1} \right) + \text{GeneMedians} \left(x_{\cdot\cdot 2} \right), \end{cases} \tag{33}$$

where the GeneMedians function is defined as it is for MRS. The normalized data produced by GQ do not satisfy the same distributional outcome as MRS transformed data.

Quantile Normalization

QN [3] is a method for intra-platform normalization that has been applied to cross-platform normalization [6]. The method is similar to MRS except that the median is replaced with the mean as a measure of center, data are sorted within each assay before gene summarization, and the normalization is performed without regard to which of the two platforms produced each assay. The transformation is

$$y_{gap} = O_{R(x_{gap})} \left(\text{GeneMeans} \left(\text{AssaySort} \left([x_{\cdot\cdot 1}; x_{\cdot\cdot 2}] \right) \right) \right) \tag{34}$$

where GeneMeans is defined similarly to GeneMedians as the vector of mean expression values the genes in the data matrix $x_{\cdot\cdot 1}$ and AssaySort gives the matrix of expression values in which data have been sorted within each assay. When applied to the sorted data, GeneMeans is assumed to calculate means based on the rank of each gene, rather than the actual identity of each. That is, genes are matched by rank before averaging. The outcome of quantile normalization is that the distribution of expression values is identical for all assays.

References

[1] M Benito, J Parker, Q Du, J Wu, D Xiang, CM Perou, and JS Marron. Adjustment of systematic microarray data biases. *Bioinformatics*, 20(1):105, 2004.

[2] A Bera and C Jarque. Efficient tests for normality, homoscedasticity and serial independence of regression residuals: monte carlo evidence. *Economics Letters*, 7(4):313–318, Jan 1981.

[3] BM Bolstad, RA Irizarry, M Astrand, and TP Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2):185, 2003.

[4] FE Grubbs. Procedures for detecting outlying observations in samples. *Technometrics*, 11(1):1–21, 1969.

[5] H Jiang, Y Deng, H Chen, L Tao, and Q Sha. Joint analysis of two microarray gene-expression data sets to select lung adenocarcinoma marker genes. *BMC Bioinformatics*, 5:81, Jan 2004.

[6] R Lacson, E Pitzer, J Kim, P Galante, C Hinske, and L Ohno-Machado. Dsgeo: software tools for cross-platform analysis of gene expression data in geo. *Journal of Biomedical Informatics*, 43:709–715, 2010.

[7] J Marron, M Todd, and J Ahn. Distance-weighted discrimination. *Journal of the American Statistical Association*, 102(480):1267, Jan 2007.

[8] R Martinez, C Pasquier, and N Pasquier. Genminer: mining informative association rules from genomic data. *Proceeding of the IEEE International Conference on Bioinformatics and Biomedicine.*, 1:15–22, Jan 2007.

[9] R Martinez, N Pasquier, and C Pasquier. Genminer: mining non-redundant association rules from integrated gene expression data and annotations. *Bioinformatics*, 24(22):2543–2644, Jan 2008.

[10] AA Shabalin, H Tjelmeland, C Fan, CM Perou, and AB Nobel. Merging two gene-expression studies via cross-platform normalization. *Bioinformatics*, 24(9):1154, 2008.

[11] P Warnat, R Eils, and B Brors. Cross-platform analysis of cancer microarray data improves gene expression based classification of phenotypes. *BMC Bioinformatics*, 6:265, Jan 2005.

[12] X.-Q Xia, M Mcclelland, S Porwollik, W Song, X Cong, and Y Wang. Webarraydb: cross-platform microarray data analysis and public data repository. *Bioinformatics*, 25(18):2425–2429, Sep 2009.