Libertas Academica
FREEDOM TO RESEARCH

Open Access: Full open access to this and thousands of other papers at http://www.la-press.com.

Cancer Informatics

Supplementary Issue: Array Platform Modeling and Analysis (B)

Statistical Issues in the Design and Analysis of nCounter Projects

Sin-Ho Jung[1,2] and Insuk Sohn[2]

[1]Department of Biostatistics and Bioinformatics, Duke University, Durham, NC, USA. [2]Biostatistics and Clinical Epidemiology Center, Samsung Medical Center, Seoul, Korea.

ABSTRACT: Numerous statistical methods have been published for designing and analyzing microarray projects. Traditional genome-wide microarray platforms (such as Affymetrix, Illumina, and DASL) measure the expression level of tens of thousands genes. Since the sets of genes included in these array chips are selected by the manufacturers, the number of genes associated with a specific disease outcome is limited and a large portion of the genes are not associated. nCounter is a new technology by NanoString to measure the expression of a selected number (up to 800) of genes. The list of genes for nCounter chips can be selected by customers. Due to the limited number of genes and the price increase in the number of selected genes, the genes for nCounter chips are carefully selected among those discovered from previous studies, usually using traditional high-throughput platforms, and only a small number of definitely unassociated genes, called control genes, are included to standardize the overall expression level across different chips. Furthermore, nCounter chips measure the expression level of each gene using a counting observation while the traditional high-throughput platforms produce continuous observations. Due to these differences, some statistical methods developed for the design and analysis of high-throughput projects may need modification or may be inappropriate for nCounter projects. In this paper, we discuss statistical methods that can be used for designing and analyzing nCounter projects.

KEYWORDS: censoring, false discovery rate, gradient lasso, permutation, proportional hazards model

SUPPLEMENT: Array Platform Modeling and Analysis (B)

CITATION: Jung and Sohn. Statistical Issues in the Design and Analysis of nCounter Projects. *Cancer Informatics* 2014:13(S7) 35–43 doi: 10.4137/CIN.S16343.

RECEIVED: August 12, 2014. **RESUBMITTED:** September 29, 2014. **ACCEPTED FOR PUBLICATION:** October 4, 2014.

ACADEMIC EDITOR: JT Efird, Editor in Chief

TYPE: Review

FUNDING: This research was supported by a grant from the National Cancer Institute (CA142538). The authors confirm that the funder had no influence over the study design, content of the article, or selection of this journal.

COMPETING INTERESTS: Authors disclose no potential conflicts of interest.

COPYRIGHT: © the authors, publisher and licensee Libertas Academica Limited. This is an open-access article distributed under the terms of the Creative Commons CC-BY-NC 3.0 License.

CORRESPONDENCE: sinho.jung@duke.edu

Paper subject to independent expert blind peer review by minimum of two reviewers. All editorial decisions made by independent academic editor. Upon submission manuscript was subject to anti-plagiarism scanning. Prior to publication all authors have given signed confirmation of agreement to article publication and compliance with all applicable ethical and legal requirements, including the accuracy of author and contributor information, disclosure of competing interests and funding sources, compliance with ethical requirements relating to human and animal study participants, and compliance with any copyright requirements of third parties. This journal is a member of the Committee on Publication Ethics (COPE).

Introduction

Genome-wide microarray is a technology to measure the expression level of a large number (20,000–50,000) of genes and to discover those that are differentially expressed and associated with clinical outcomes. From an analysis of microarray data, we may identify a specific set of gene signatures whose expression levels are associated with a particular clinical outcome of interest. These signatures can comprise tens to hundreds of genes, a range that is appropriate for technical validation using NanoString nCounter Gene Expression Assay.

nCounter Gene Expression Assay (Nanostring Technologies, Seattle, WA, USA) is a robust and highly reproducible method for detecting the expression of up to 800 genes in a single reaction with high sensitivity and linearity across a broad range of expression levels. The nCounter assay is based on direct digital detection of mRNA molecules of interest using target-specific, color-coded probe pairs, so that the expression level of each gene is measured by counts. Due to its high reproducibility, nCounter assay is chosen as a good technical validation platform for the findings made from genome-wide microarray assays.

Numerous statistical methods have been proposed for genome-wide microarray projects. These methods, mostly developed for microarray data comprising many thousands of genes, may be specific to the platform or the number of genes so that they may not be appropriate for the analysis of nCounter data. The important analysis processes for microarray data are (i) data preprocess, (ii) discovery of the genes that are associated with different types of clinical outcomes, and (iii) prediction model fitting and validation. One important aspect in the analysis of microarray data is to adjust the false positivity for multiplicity of the genes. Furthermore, some of these methods utilize the fact that a large portion of the genes are not associated with the clinical outcome under investigation.

In this article, we review these statistical methods developed for high-throughput microarray projects and discuss the issues that can be raised when applying them to nCounter projects. We take a real nCounter data set to demonstrate these issues. The clinical outcome can be any type of variable, such as binary (eg, benign versus malignant and response versus nonresponse), continuous (eg, blood pressure), and time to event (eg, time to progression and overall survival) variables. Different types of outcome variables require different statistical methods. In this article, we focus on time-to-event endpoint, which is popular in cancer research.

An Example nCounter Data

Lee et al.[1] designed an nCounter probe set (Nanostring Technologies) consisting of 800 candidate prognostic genes and 48 internal reference genes identified from a WG-DASL microarray study, as well as some known cancer genes, kinase genes, and G protein–coupled receptor genes, and profiled 428 patients with stage II gastric cancer. This study was undertaken to identify high-risk gastric cancer patients for tumor recurrence after surgery. The primary endpoint was disease-free survival (DFS), defined as time from surgery to the date of documented tumor recurrence or death.

Study Design

Usually, microarray experiments are conducted by multiple batches, and the expression data of genes are different across different batches. There are various publications on statistical methods to remove the batch effect, but most of them just try to make the distribution of gene expression data equal or similar among different batches, eg, Lee et al.[2] and the references therein. The batch effects have so complicated impact among different genes in the chips, so that these attempts are not very successful in removing batch effects. Furthermore, the effort to make the gene expression profile similar among different batches just removes the real difference between patient groups we want to detect, so that we often fail to discover prognostic or predictive genes with a batch effect adjustment. Owzar et al.[3] showed that known data normalization methods do not appropriately remove batch effect either.

If we cannot avoid batch effect and it cannot be removed in data analysis, it is critical that the batch allocation do not compound with any known predictors or the clinical outcome under investigation. Suppose that we want to discover the genes that are differentially expressed between two disease types. In this case, if the two disease types are assigned to different batches, then the main effect (the difference between two disease types) and batch effect are completely compounded and any effort to remove the batch effect will remove the main effect. To avoid this issue, we propose to randomly assign the patients among different batches while stratifying for the predictors (and clinical outcome also if possible). In the example study, we stratified the batch allocation with respect to tumor size and year of surgery.

Due to various reasons, the overall gene expression level among different chips may vary even within each batch. So, we need some control genes to normalize the overall expression levels across different chips. Some control genes are provided by the manufacturer, but we can also add some more together with candidate prognostic genes. In our example project, NanoString provided 26 prognostic genes and the project team identified 48 control genes from a DASL study conducted prior to the nCounter project. The criteria used when selecting the 48 genes from the DASL study were (i) the variance of the expression level is small, (ii) the mean expression level is similar to those of the 26 prognostic genes selected for this NanoString project, and (iii) the expression level is not associated with the clinical outcome, DFS. Microarray data from genome-wide platforms are normalized using the whole data set. This is one of the reasons why these high-throughput platforms are not appropriate for clinical use. However, nCounter chips are very reproducible, so that control genes within each chip provide very good normalization.

Data Analysis

Data preprocessing. In the example project, we excluded six samples with positive control normalization factor outside a range of 0.3–3. Positive control normalization factor was calculated as a ratio of the sum of expression levels of six positive controls in each sample to the average of sum of the six positive controls across all samples. We considered a sample having low quality if too many endogenous genes were expressed lower than the eight negative control genes. More specifically, we excluded 20 samples because, for each of them, the number of genes with expression levels larger than the maximum of the eight negative controls was smaller than 360 ($= 0.45 \times 800$ endogenous genes). Consequently, 402 samples, of the 428, were used for further statistical analysis.

By most high-throughput microarray platforms, gene expression level is measured as a positive continuous variable, so that the logarithm transformation has been popularly used to convert the distribution of data into a normal distribution. For nCounter chips, however, it is measured as counts, so that a Poisson distribution may be more appropriate. It is well known that the square root transformation converts data with a Poisson distribution to those with a normal distribution. Supplementary Figure 1 reports histograms of raw data, log-transformed (with base 2) data, and square root–transformed data for the 48 control genes. We also checked the distribution of endogenous genes, but decided not to report the results from them because their expression levels might depend on the clinical

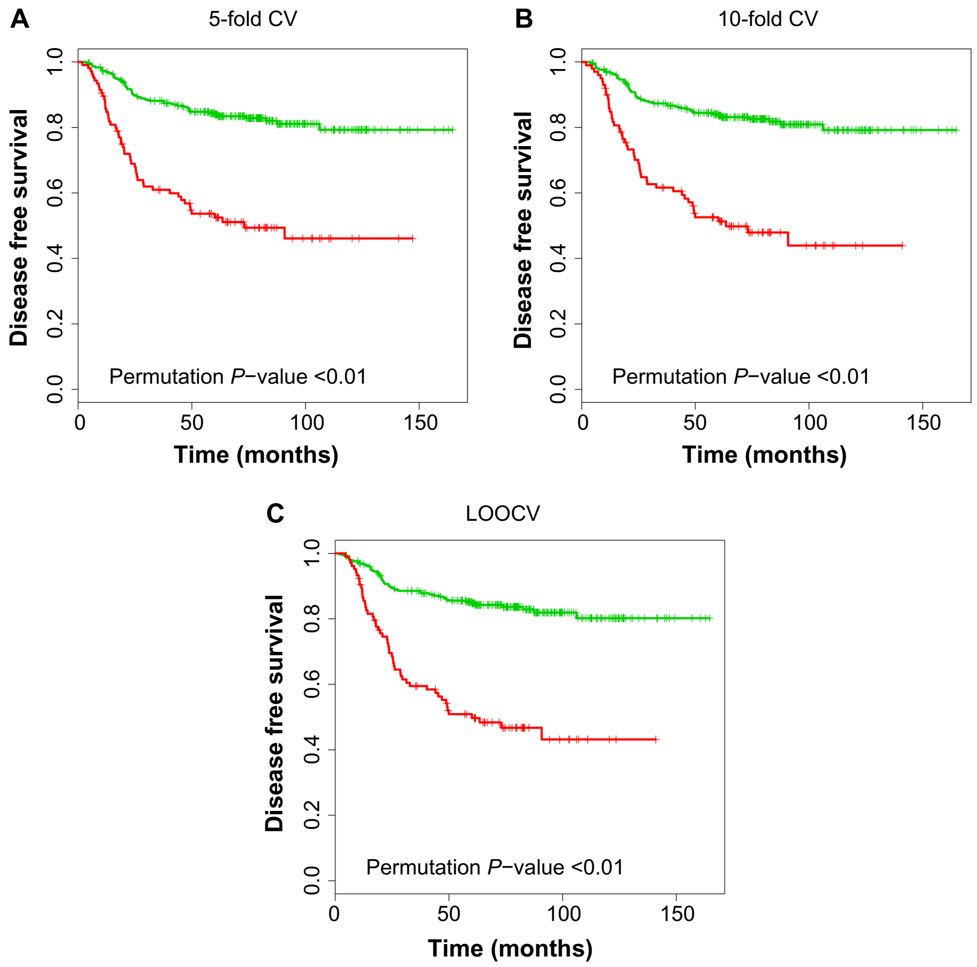

Figure 1. Kaplan–Meier plot for high-risk (red) and low-risk (green) groups classified by different CV methods. The *P*-value is calculated from 100 permutations.

outcome of the patients and violate the independent and identically distributed assumption. From the histograms, we do not observe that square root transformation works better than the log transformation but that the latter seems even to outperform, slightly, the former. So, we decided to use the log transformation for further analysis of this data set. Definitely, the raw data have skewed distributions.

Variations in the technology and experimental conditions, rather than from biological differences between subjects, result in different amount of expression among different chips. Normalization is to adjust microarray data for these spurious effects. One of most popular normalization methods for genome-wide microarray data is the quantile normalization,[4] which makes the quantiles of the gene expression data among m genes identical across n subjects. As in the quantile normalization, most normalization methods for microarray data require the whole microarray data to normalize the expression data of each chip. This prohibits the high-throughput microarray chips from being used as a commercial platform. On the contrary, nCounter data are normalized using the control genes within each chip, so that we do not need the expression data from other chips for normalization.

Gene discovery. We want to identify the genes whose expression levels are associated with a survival outcome under investigation. With m candidate genes, we will conduct m univariate tests to associate each gene with the clinical outcome. For gene $j(= 1,\ldots, m)$, the null hypothesis H_j means that its expression level is not associated with the clinical outcome and the alternative hypothesis $\bar{H}$ means that they have some association. This discovery procedure involves m hypothesis tests. Suppose that, among the m tests, null hypotheses are true for m_0 genes and alternative hypotheses are true for $m_1(= m - m_0)$ genes. Furthermore, suppose that, of the m_0 null hypotheses of unassociated genes, A_0 are accepted (true negative) and R_0 are rejected (false rejection, false discovery, or false positive). Among the m_1 alternative hypotheses of associated genes, A_1 are rejected (false negative) and R_1 are accepted (true rejection, true discovery, or true positive).

Family-wise error rate. The family-wise error rate (FWER) is defined as $P(A_0 > 0|m_0 = m)$, ie, the probability to select any genes when none of them are associated with the clinical outcome.

In order to test the association between the expression level of gene $j(= 1,\ldots,m)$ with the survival outcome, we propose to use the partial score test statistic of a univariate Cox[5] proportional hazards model. Let U_j denote the corresponding P-value, which is also called a marginal P-value. A single-step procedure (SSP) uses a common critical value c for all P-values to select each gene if $U_j < c$. For an exact control of the FWER at α, critical value c should satisfy

$$\begin{aligned}\alpha &= P(U_1 < c \text{ or } U_2 < c \text{ or, } \ldots, \text{or } U_m < c \mid H_0)\\ &= P(\min_{j=1,\ldots,m} U_j < c \mid H_0) \qquad (1)\end{aligned}$$

where

H_0: genes are not associated with the survival outcome or equivalently $H_0 = \bigcap_{j=1}^{m} H_j$, is the complete null hypothesis and the relevant alternative hypothesis is $H_a = \bigcup_{j=1}^{m} \bar{H}_j$. In order to control the FWER below a prespecified level α, Bonferroni test uses $c = \alpha/m$. The Bonferroni procedure is always conservative, especially with correlated test statistics. Different genes coexpress, so that the expression data among different genes are expected to be correlated and the m test statistics use the same survival data, so that the m test statistics tend to be correlated. In order to accurately control the FWER, Westfall and Young[6] propose to approximate the probability (1) by generating the distribution of $U = \min_{j=1,\ldots,m} U_j$ under H_0 using permutations.

Since it is impossible to analytically derive the null distribution of the m P-values while maintaining the dependence structure and distributional characteristics of the gene expression measures, we approximate it by using permutations with the subjects, not genes, as sampling units. In each permutation, we randomly match the random vectors of m gene expression levels with the survival outcomes. This type of resampling has been widely used in multiple testing to avoid the specification of the true distribution for the gene expression data.[7–10] Note that the number of possible permutations B can be very large even with a small number of patients, $B = n!$ if there are no ties among survival data points.

For a P-value p_j observed from the original data, we define an FWER-adjusted P-value for gene j as the minimum FWER for which H_j will be rejected, ie, $\hat{p}_j = P(\min_{j'=1,\ldots,m} U_{j'} < p_j \mid H_0)$. In what follows, this probability is estimated from the permutation distribution:

Algorithm 1: Single-step procedure

A. Compute the P-values $p_1,\ldots,p_m$ from the original data.
B. For the b-th permutation of the original data ($b = 1,\ldots, B$), compute the P-values $u_1^{(b)},\ldots,u_m^{(b)}$ and $u_b = \min_{j=1,\ldots,m} u_m^{(b)}$.
C. Estimate the adjusted P-values by
$$\hat{p}_j = B^{-1}\sum_{b=1}^{B} I(u_b \ge p_j) \text{ for } j = 1,\ldots,m.$$
D. Reject all hypotheses H_j ($j = 1,\ldots, m$) such that $\hat{p}_j < \alpha$.

Alternatively, the cutoff value c_α can be determined with steps (C) and (D) replaced as follows:

Algorithm 1′

C. Sort $u_1,\ldots,u_B$ to obtain the order statistics $u_{(1)} \le\cdots\le u_{(B)}$ and compute the critical value $c_\alpha = u_{([B(1-\alpha)+1])}$, where $[a]$ is the largest integer no greater than a. If there exist ties, $c_\alpha = w_{(k)}$, where k is the smallest integer such that $u_{(k)} \ge u_{([B(1-\alpha)+1])}$.
D. Reject all hypotheses H_j ($j = 1,\ldots, m$) for which $p_j < c_\alpha$.

Below is a step-down analog suggested by Dudoit et al.[7,8] originally proposed by Westfall and Young[6,11] (see Algorithms 2.8 and 4.1 in their book).

Algorithm 2: Step-down procedure

A. Compute the P-values $p_1,\ldots, p_m$ from the original data.
A1. Sort $p_1,\ldots, p_m$ to obtain the ordered statistics $p_{r_1} \ge\cdots\ge p_{r_m}$, where $H_{r_1}, \ldots, H_{r_m}$ are the corresponding hypotheses.
B. For the b-th permutation of the original data ($b = 1,\ldots,B$), compute the P-values $u_{r_1}^{(b)},\ldots,u_{r_m}^{(b)}$, and $u_{b,j} = \min_{j'=j,\ldots,m} u_{r_{j'}}^{(b)}$ for $j = 1,\ldots, m$.
C. Estimate the adjusted P-values by $\tilde{p}_{r_j} = B^{-1}\sum_{b=1}^{B} I(u_{b,j} \ge p_j)$ for $j = 1,\ldots, m$.
C1. Enforce monotonicity by setting $\tilde{p}_{r_j} \leftarrow \max(\tilde{p}_{r_{j-1}}, \tilde{p}_{r_j})$ for $j = 2,\ldots, m$.
D. Reject all hypotheses H_{r_j} ($j = 1, \ldots, m$) for which $\tilde{p}_{r_j} < \alpha$.

It can be shown that an SSP, controlling the FWER weakly as in equation (1), also controls the FWER strongly under the condition of subset pivotality (see p. 42 in Westfall and Young[11]).

The SSP and step-down procedure can be useful for gene discovery using nCounter data while controlling the FWER accurately. Jung et al.[12] proposed an FWER control procedure to associate a survival outcome with gene expression data using a rank test statistic. We apply this procedure to the example study. Table 1 lists the genes that are selected by the SSP.

Table 1. Genes with a FWER-adjusted P-value smaller than 0.05 by the SSP using B = 10,000 permutations.

GENE ID	*P*-VALUE	
	MARGINAL	ADJUSTED
gene54	0.0000	0.0486
gene65	0.0000	0.0002
gene115	0.0000	0.0052
gene119	0.0001	0.0370
gene120	0.0000	0.0038
gene129	0.0000	0.0214
gene480	0.0000	0.0002
gene492	0.0001	0.0484
gene526	0.0000	0.0010

We just use serial numbers for gene ID to limit the scope of this paper to statistical issues. We use a slightly different data set from that of Lee et al.[1] so that we decided not to give the real gene names here. Readers may refer to Lee et al.[1] for the biological findings from this project. We obtained a very similar result from step-down procedure.

Jung et al.[13] and Jung and Young[14] proposed sample size formulas to discover genes that are differentially expressed between two patient groups and Jung[15] proposed one to discover genes that are associated with a survival endpoint. These methods can be used to estimate the number of patients required for nCounter projects for gene discovery while controlling the FWER.

False discovery rate. Let $R = R_0 + R_1$ denote the total number of rejections (or discoveries). Then R_0/R denotes the proportion of false discoveries among the total discoveries. Benjamini and Hochberg[16] define the false discovery rate (FDR) as

$$\text{FDR} = E\left(\frac{R_0}{R}\right).$$

This expression is undefined if $\Pr(R = 0) > 0$. Storey[17] claims that $\Pr(R > 0) \approx 1$, with a large m as in high-throughput microarray project cases. In nCounter data, however, m is not that large. But this condition may still hold because most of the genes in the chips are prognostic.

Benjamini and Hochberg[16] propose a multi-step procedure to control the FDR at a specified level. However, this is known to be conservative, and the conservativeness increases in m_0 (see, eg, Storey et al.[18]).

Suppose that, in the j-th testing, we reject the null hypothesis H_j if the P-value p_j is smaller than or equal to $\alpha \in (0, 1)$. Assuming independence of the m P-values, we have

$$R_0 = \sum_{j=1}^{m} I(H_j \text{ true}, H_j \text{ rejected})$$
$$= \sum_{j=1}^{m} \Pr(H_j \text{ true})\Pr(H_j \text{ rejected}|H_j) + o_p(m),$$

which equals $m_0\alpha$, where $m^{-1}o_p(m) \to 0$ in probability as $m \to \infty$.[17]

Ignoring the error term, we have

$$\text{FDR}(\alpha) = \frac{m_0\alpha}{R(\alpha)}, \tag{2}$$

where $R(\alpha) = \sum_{j=1}^{m} I(p_j \le \alpha)$. Given α, estimation of FDR by equation (2) requires estimation of m_0. This approximation is valid only when m is large and the expression data among m genes are independent or weakly correlated.

For the estimation of m_0, Storey[17] assumes that the histogram of m P-values is a mixture of m_0 P-values that are corresponding to the true null hypotheses and following $U(0, 1)$ distribution, and m_1 P-values that are corresponding to the alternative hypotheses and expected to be close to 0. Consequently, for a chosen constant $\lambda \in (0, 1)$, which has a value that is not near 0, none (or only few, if any) of the latter m_1 P-values will fall above λ, so that the number of P-values above λ, $\sum_{j=1}^{m} I(p_j > \lambda)$, can be approximated by the expected frequency among the m_0 P-values above λ from $U(0, 1)$ distribution, ie, $m_0/(1 - \lambda)$. Hence, given λ, m_0 is estimated by

$$\hat{m}_0(\lambda) = \frac{\sum_{j=1}^{m} I(p_j > \lambda)}{1-\lambda}.$$

By combining this m_0 estimator with equation (2), Storey[17] obtains

$$\widehat{\text{FDR}}(\alpha) = \frac{\alpha \times \hat{m}_0(\lambda)}{R(\alpha)} = \frac{\alpha \sum_{j=1}^{m} I(p_j > \lambda)}{(1-\lambda)\sum_{j=1}^{m} I(p_j > \alpha)}.$$

For an observed P-value p_j, Storey[17] defines the q-value, the minimum FDR level at which we reject H_j, as

$$q_j = \inf_{a \ge pj} \widehat{\text{FDR}}(\alpha).$$

This formula is reduced to

$$q_j = \widehat{\text{FDR}}(p_j)$$

if FDR(α) is strictly increasing in α, see Theorem 2 of Storey.[19] We reject H_j (or, discover gene j) if q_j is smaller than or equal to the prespecified FDR level.

The independence assumption among m test statistics is loosened to independence only among m_0 test statistics corresponding to the null hypotheses by Storey and Tibshirani,[20] and to weak independence among all m test statistics by Storey[19] and Storey et al.[18] These approaches are implemented in the statistical package called SAM.[21]

It is questionable if we can use this FDR procedure for gene discovery using nCounter data. This procedure is valid only when the number of genes is very large and a large portion of them are null genes. This is not the case for most nCounter data, so that the FDR control methods by Storey and his colleagues do not seem be appropriate for nCounter data analysis.

Jung[22] proposes a sample size calculation method for microarray projects to discover genes by controlling the FDR. Jung and Jang[23] evaluate the performance of the FDR control methods by Benjamini and Hochberg[16] and Storey and Tibshirani[21] for gene discovery using microarray data and show that neither maintains the FDR accurately.

Prediction and Validation

In this section, we discuss how to develop a statistical model to predict the survival time using gene expression data and validate the developed prediction model. Suppose that we have n subjects. For subject $i(= 1,..., n)$, we have a corresponding measure of time to event T_i, such as tumor recurrence or death. The event time may be censored due to loss to follow-up or study termination. Therefore, we observe $X_i = \min(T_i, C_i)$ with an event indicator $\delta_i = I(T \leq C_i)$, where C_i is the censoring time that is independent of T_i given the gene expression data. Let Z_{ij} denote the expression measurement for gene $j(= 1,\ldots, m)$ of subject i.

The goal in survival prediction is to build a model with input from $Z = (Z_1,..., Z_m)$ to predict T or its distribution. Using this model, the survival distributions of future subjects can be predicted from their gene expression measures. These models can be built via proportional hazards regression model by Cox.[5] The hazard function at time t for a subject i with gene expression values $Z_i = (Z_{i1},..., Z_{ip})^T$ is given by

$$\lambda_i(t) = \lambda_0(t)exp(\beta^T Z_i) \tag{3}$$

where $\lambda_0(t)$ is an unspecified baseline hazard function and $\beta = (\beta_1,..., \beta_p)^T$ is a set of unknown regression parameters. Usually in high-throughput genome-wide microarray studies, the number of genes under study (or candidate genes), m, is much larger than the number of prognostic genes, p, and the sample size (or number of subjects) n. So, a challenge of prediction problem using high-dimensional genomic data is to fit the regression model (3) while identifying a small number (p) of genes that are associated with subject's survival trait under consideration. In nCounter projects, the number of genes is much smaller than those in genome-wide microarray studies but is still large compared to the number of subjects or too large to fit a full regression model including all m genes. Therefore, the standard regression analysis method does not work for the selection of prognostic genes. Penalized methods have been widely investigated to overcome the large-m-small-n problem, including ridge regression,[24] lasso,[25] and elastic net.[26] These methods require intensive computations. A fitted prediction model should be validated using an independent test set or a cross-validation (CV) method using the original data. Pang and Jung[27] proposed a sample size procedure to design a microarray project for prediction and validation. This method can be applied to the design of an nCounter project with minor modifications.

Prediction. A prediction model is fitted from the training set. Before applying a prediction method to microarray data, we standardize the expression data of each gene by subtracting the sample mean and dividing by sample standard deviation. The prediction methods discussed above select a covariate for prediction model by the size of its regression coefficient (β_j) rather than by its significance (or P-value). The size of each regression coefficient, however, depends on the scale of the corresponding covariate (individual gene's expression data in this case). Hence, we need to standardize the expression data across the genes before applying a prediction procedure.

In order to lower the computational burden of prediction in genome-wide microarray data case, we decrease the number of candidate genes by selecting a feasible number (usually between hundreds to a few thousands) of genes using univariate Cox regression method with $\lambda(t|Z_j) = \lambda_{0j}(t)\exp(\beta_j Z_j)$ for gene $j(= 1,..., m)$. For nCounter data case, however, we do not need this selection process before conducting a prediction algorithm as the number of genes is much smaller in this case and most genes (excluding some control genes) are strong candidates known to be prognostic for different types of cancer diseases.

For genome-wide microarray data with a large m, Sohn et al.[28] propose a gradient lasso procedure which maximizes the penalized partial likelihood[5] to fit prediction models for time-to-event endpoint. They show that the procedure is guaranteed to converge to the optimum under mild regularity conditions with efficient computations. In this section, we focus on this prediction method.

Suppose that a multivariate Cox regression model is fitted from the training data using the chosen $p(<<m)$ genes as covariates to predict the survival outcome of subjects in the test set. Let $(\hat{\beta}_1,\ldots,\hat{\beta}_p)$ denote the regression estimates of the prediction model fitted from the training set and $(Z_1,..., Z_p)$ the expression data of the corresponding genes. We call $S = \hat{\beta}_1 Z_1 + \ldots + \hat{\beta} Z_p$ a risk score, a large value representing a short survival time.

Validation. The first step of validation is to standardize the gene expression data of the test set (also called validation set) using the sample means and sample standard deviations calculated from the training set. Using the risk score fitted from the training set and its median as a cutoff value, we partition the subjects in the test set into a high-risk group and a low-risk group. We may choose a different cutoff value depending on how large the high-risk patient group we want. We may not even dichotomize the risk score. If we want to use the continuous risk score for clinical applications, then we may validate the prediction model by regressing the survival time on the raw (continuous) risk score as a single covariate using the test set.

Assessing the accuracy of a fitted prediction model based on the same data set that was used to develop the model can result in an overly optimistic performance assessment of the model for future samples, which is called overfitting bias.[29] To remove this bias, validation combined with a resampling method, such as bootstrapping, CV, and permutation, can be employed. We describe below some resampling techniques that are popularly used for validation.

- Hold-out or split sample method
  The hold-out method or split sample method is the simplest of all the resampling methods considered in this paper. It randomly partitions the whole data set into a training set of proportion P and a test set of proportion $1 - P$. Since we do not reuse the training set for test, no overfitting bias is involved in this validation method.

- *k*-fold CV

  The *k*-fold CV method randomly divides the data set into *k* partitions that are close to equal in size. At each of the *k*-th iteration, $k - 1$ partitions will be used as the training set and the remaining partition will be used as the test set. The most commonly used methods are 5-fold and 10-fold CVs. A twofold CV is essentially a 50% hold-out method with the role of the training set being switched with that of the test set. By reusing the training set for testing, a twofold CV has a higher validation efficiency than a 50% hold-out method while they have a similar prediction efficiency (as the sample size for prediction is identical between the two methods).
- Leave-one-out-cross-validation

  Leave-one-out-cross-validation (LOOCV) is a special case of *k*-fold CV with $k = n$, the sample size of the whole data set. At each of the *n* iterations, the whole data set is used as the training except the one sample which is left out as test set. Since it requires the largest number of iterations, it is the most computationally expensive resampling method introduced here.

As mentioned above, we partition the subjects in the test set into high- and low-risk groups using the median risk score estimated from the training set as a cutoff value for their risk scores. Since the gene expression data are standardized, we expect about 50–50 allocation between the two risk groups. In order to validate the fitted prediction model (or the risk score) from the training set, we compare the survival distributions of the training set subjects between the two risk groups using the log-rank test.[30] With a two-sided type I error rate α, we conclude the validation of the fitted prediction model by the hold-out resampling method if the absolute value of the standardized log-rank test statistic is larger than the $100(1 - \alpha/2)$-th percentile of the standard normal distribution.

Overfitting bias can be an issue when the data points used for prediction are reused for validation. So, among the above resampling methods to split the whole data of size *n* into training and test sets, only the hold-out method is free of overfitting bias since the resulting training and validation sets are mutually exclusive and the data points used to fit a prediction model are never reused for validation. All other methods (ie, CV methods) use each data point for testing as well as training. For example, in *k*-fold CV method, each data point is used $k - 1$ times for training and once for testing. In order to remove the overfitting bias of the CV methods, we use a permutation method as follows.

- From a resampling of the original data $\{(X_i, \delta_i), (Z_{i1},\ldots, Z_{ip}): i = 1,\ldots, n\}$ calculate the two-sample log-rank *P*-value p_0 comparing the survival distributions between high- and low-risk groups of test set.
- Generate the *b*-th ($b = 1,\ldots, B$) permutation data by shuffling the survival data $\{(X_i, \delta_i): i = 1,\ldots, n\}$ and randomly matching them with the nCounter data $\{(z_{i1},\ldots, z_{im}): i = 1,\ldots, n\}$.
- At the *b*-th permutation, apply the prediction–validation procedures, which are used for the original data, to the permuted data, and calculate the log-rank *P*-value p_b.
- Repeat the permutations *B* times and estimate an unbiased *P*-value for validation by

$$P\text{-value} = B^{-1}\sum_{b=1}^{B} I(p_b \le p_0).$$

For a prespecified type I error rate α, we conclude a positive validation of the prediction if *P*-value $< \alpha$.

With survival data as clinical outcomes, we may not be able to develop a prediction model if there are too few events in the training set and the log-rank test to compare the survival distributions between high- and low-risk groups of test set may not have enough power if there are too few events in the test set. So, it is critical that the whole data set have enough number of events for a reasonable prediction–validation procedure. In order to increase the power of a chosen prediction–validation procedure, we propose to randomly allocate the subjects with events evenly among different partitions in resampling.

We applied the 5-fold CV, 10-fold CV, and LOOCV methods to the example data. Figure 1 depicts the Kaplan–Meier curves of the high- and low-risk groups by these CV methods. The two Kaplan–Meier curves of DFS split more in the order of 5-fold CV, 10-fold CV, and LOOCV. But this does not imply more significant validation of the fitted prediction models as the amount of overfitting bias increases in this order too. For each CV procedure, an unbiased *P*-value was estimated from $B = 100$ permutations to compare the DFS between two risk groups. Note that the two risk groups classified by the fitted prediction models have significantly different DFS distributions by each of these CV methods. However, we could not compare the significance among the three CV methods because of the strong prediction power of the example data and the limited number of permutations in these analyses. We could not increase the number of permutations due to computational burden. Jung and Pang[31] show that the 5-fold and 10-fold CV methods use the whole data more efficiently among different resampling methods.

Since the data set was shown to be valid to predict the DFS of future patients, we fitted a prediction model using the whole data set of $n = 402$. Table 2 lists the 10 genes that were included in the final prediction model from the whole data set, their regression estimates, marginal univariate Cox *P*-values, FWER-adjusted *P*-values approximated from 10,000 permutations, and the number of times to be included in the prediction models during the prediction–validation steps of each CV method. We observe that genes 65, 115, 480, and 526 are highly significant with large regression coefficients in absolute value and always included in the prediction models from the training sets during the 10-fold CV and LOOCV. As expected,

Table 2. The prediction model fitted using the whole 402 samples included 10 genes.

GENE ID	$\hat{\beta}$	MARGINAL *P*-VALUE	FWER-ADJUSTED *P*-VALUE BY SSP	# TIMES INCLUDED		LOOCV
				5-FOLD CV	10-FOLD CV	
gene52	−0.0137	0.0241	0.9815	0	1	35
gene65	0.0774	0.0000	0.0002	4	10	402
gene93	−0.0438	0.0367	0.9949	0	0	0
gene96	−0.0670	0.0101	0.8548	0	0	0
gene115	0.1599	0.0000	0.0052	5	10	402
gene472	−0.2070	0.0011	0.2981	0	0	0
gene480	0.2336	0.0000	0.0002	5	10	402
gene526	0.0133	0.0000	0.0010	3	10	402
gene701	−0.0992	0.0020	0.3173	1	2	2
gene710	0.0176	0.0006	0.1503	0	0	0

Notes: This table reports their regression coefficients, marginal *P*-values, FWER-adjusted *P*-values approximated by SSP, the number of times they were included during the prediction–validation procedure using 5-fold CV, 10-fold CV, or LOOCV.

many genes overlap between Tables 1 and 2. But, a significant gene identified from gene discovery using a univariate analysis may not be included in the prediction model because of co-expression among genes.

Discussion

We have reviewed some useful design and analysis methods developed for high-throughput microarray projects. Unlike the most high throughput microarray platforms, nCounter chips include relatively small number of genes, most of which are potentially prognostic, and the expression level of each gene is a count variable, rather than a continuous variable. We have discussed some issues raised when applying the methods developed for high-throughput microarray projects to nCounter projects and proposed modifications required to address the issues. We also have introduced sample size calculation methods that can be used when designing nCounter projects for gene discovery and prediction of clinical outcomes. Some modifications we may need to make when analyzing nCounter data using the existing analysis methods developed for genome-wide microarray data are:

- We have to check the distribution of expression of the control genes to determine if we need another transformation than logarithm, such as square root transformation which is known to be appropriate counting data.
- For gene discovery, the existing FDR methods are not appropriate for nCounter data and use the FWER method with permutations.
- We do not need a gene screening before prediction model fitting.

We illustrated these procedures using an example nCounter study. These methods can be used for studies with pathway panel arrays which are provided by NanoString and usually containing a few hundred genes too.

Author Contributions

Conceived and designed the experiments: S-HJ. Analyzed the data: IS. Wrote the first draft of the manuscript: S-HJ. Contributed to the writing of the manuscript: S-HJ, IS. Agree with manuscript results and conclusions: S-HJ, IS. Jointly developed the structure and arguments for the paper: S-HJ, IS. Made critical revisions and approved final version: S-HJ, IS. Both authors reviewed and approved of the final manuscript.

Supplementary Material

Supplementary Figure 1. This figure reports histograms of raw data, log-transformed (with base 2) data, and square root–transformed data for the 48 control genes.

REFERENCES

1. Lee J, Sohn I, Do I-G, et al. Nanostring-based multigene assay to predict recurrence for gastric cancer patients after surgery. *PLoS One*. 2014;9(3):e90133.
2. Lee JA, Dobbin KK, Ahn J. Covariance adjustment for batch effect in gene expression data. *Stat Med*. 2014;33:2681–95.
3. Owzar K, Barry WT, Jung SH. Statistical considerations for analysis of microarray experiments. *Clin Transl Sci*. 2011;4(6):466–77.
4. Bolstad BM, Irizarry RA, Astrand M, Speed TP. A Comparison of normalization methods for high density oligonucleotide array data based on bias and variance. *Bioinformatics*. 2003;19(2):185–93.
5. Cox DR. Regression models and life tables (with discussion). *J R Stat Soc Series B Stat Methodol*. 1972;34:187–220.
6. Westfall PH, Young SS. *P*-value adjustments for multiple tests in multivariate binomial models. *Journal of the American Statistical Association*. 1989;84:780–6.
7. Dudoit S, Yang YH, Callow MJ, Speed TP. Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Statistica Sinica*. 2002;12:111–39.
8. Dudoit S, Shaffer JP, Boldrick JC. Multiple hypothesis testing in microarray experiments. *Statistical Science*. 2003;18:71–103.
9. Mutter GL, Baak JPA, Fitzgerald JT, et al. Global express changes of constitutive and hormonally regulated genes during endometrial neoplastic transformation. *Gynecologic Oncology*. 2001;83:177–85.
10. Ge Y, Dudoit S, Speed TP. Resampling-based multiple testing for microarray data analysis. *TEST*. 2003;12(1):1–44.
11. Westfall PH, Young SS. (1993). *Resampling-Based Multiple Testing: Examples and Methods for P-value Adjustment*. New York: Wiley.
12. Jung SH, Owzar K, George SL. A multiple testing procedure to associate gene expression levels with survival. *Stat Med*. 2005;24:3077–88.
13. Jung SH, Bang H, Young S. Sample size calculation for multiple testing in microarray data analysis. *Biostatistics*. 2005;6(1):157–69.

14. Jung SH, Young SS. Power and sample size calculation for microarray studies. *J Biopharm Stat*. 2012;22:30–42.
15. Jung SH. Sample size calculation for microarray studies with survival end-points. *J Comput Sci Syst Biol*. 2013;6:3.
16. Benjamini Y, Hochberg Y. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J R Stat Soc Series B Stat Methodol*. 1995;57(1):289–300.
17. Storey JD. A direct approach to false discovery rates. *J R Stat Soc Series B Stat Methodol*. 2002;64(1):479–98.
18. Storey JD, Taylor JE, Siegmund D. Strong control, conservative point estimation and simultaneous conservative consistency of false discovery rates: a unified approach. *J R Stat Soc Series B Stat Methodol*. 2004;66(1):187–205.
19. Storey JD. The positive false discovery rate: a Bayesian interpretation and the q-value. *Ann Stat*. 2003;31(6):2013–35.
20. Storey JD, Tibshirani R. Estimating false discovery rates under dependence, with applications to DNA microarrays. Technical Report 2001–28, Department of Statistics, Stanford University, Stanford; 2001.
21. Storey JD, Tibshirani R. SAM thresholding and false discovery rates for detecting differential gene expression in DNA microarrays. In: Parmigiani G, Garrett ES, Irizarry RA, Zeger SL, eds. *The Analysis of Gene Expression Data: Methods and Software*. New York: Springer; 2003:272–90.
22. Jung SH. Sample size for FDR-control in microarray data analysis. *Bioinformatics*. 2005;21(14):3097–104.
23. Jung SH, Jang W. How accurately can we control the FDR in analyzing microarray data? *Bioinformatics*. 2006;22:1730–6.
24. Hoerl AE, Kennard RW. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*. 1970;12:55–67.
25. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc Series B Stat Methodol*. 1996;58(1):267–88.
26. Zou H, Hastie T. Regularization and variable selection via the elastic net. *J R Stat Soc Series B Stat Methodol*. 2005;67(2):301–20.
27. Pang H, Jung SH. Sample size considerations of prediction-validation methods in high-dimensional data for survival outcomes. *Genet Epidemiol*. 2013; 37:276–82.
28. Sohn I, Kim J, Jung SH, Park C. Gradient lasso for Cox proportional hazards model. *Bioinformatics*. 2009;25:1775–81.
29. Simon RM, Subramanian J, Li MC, Menezes S. Using cross-validation to evaluate predictive accuracy of survival risk classifiers based on high-dimensional data. *Brief Bioinform*. 2011;12(3):203–14.
30. Peto R, Peto J. Asymptotically efficient rank invariant test procedures. *J R Stat Soc Series A Stat Soc*. 1972;135(2):185–206.
31. Pang H and Jung SH. Sample size considerations of prediction-validation methods in high-dimensional data for survival outcomes. *Genetic Epidemiology*. 2013; 37:276–82.