BMC Bioinformatics

Proceedings

Open Access

Network-based de-noising improves prediction from microarray data

Tsuyoshi Kato[1,2], Yukio Murata[3], Koh Miura[3], Kiyoshi Asai[1,2], Paul B Horton[2], Koji Tsuda[2] and Wataru Fujibuchi*[2]

Address: [1]Graduate School of Frontier Sciences, University of Tokyo, 5-1-5, Kashiwanoha, Kashiwa, 277 – 8562, Japan, [2]AIST Computational Biology Research Center, 2-42, Aomi, Koto-ku, Tokyo, 135-0064, Japan and [3]Division of Biological Regulation and Oncology, Department of Surgery, Tohoku University Graduate School of Medicine, 1-1, Seiryo-machi, Aoba-ku, Sendai 980-8574, Japan

Email: Tsuyoshi Kato - kato-tsuyoshi@aist.go.jp; Yukio Murata - yukio-m@surg1.med.tohoku.ac.jp; Koh Miura - k-miura@surg1.med.tohoku.ac.jp; Kiyoshi Asai - asai@cbrc.jp; Paul B Horton - horton-p@aist.go.jp; Koji Tsuda - tsuda@cbrc.jp; Wataru Fujibuchi* - fujibuchi-wataru@aist.go.jp

* Corresponding author

from NIPS workshop on New Problems and Methods in Computational Biology
Whistler, Canada. 18 December 2004

Published: 20 March 2006

BMC Bioinformatics 2006, **7**(Suppl 1):S4 doi:10.1186/1471-2105-7-S1-S4

Abstract

Background: Prediction of human cell response to anti-cancer drugs (compounds) from microarray data is a challenging problem, due to the noise properties of microarrays as well as the high variance of living cell responses to drugs. Hence there is a strong need for more practical and robust methods than standard methods for real-value prediction.

Results: We devised an extended version of the off-subspace noise-reduction (de-noising) method [1] to incorporate heterogeneous network data such as sequence similarity or protein-protein interactions into a single framework. Using that method, we first de-noise the gene expression data for training and test data and also the drug-response data for training data. Then we predict the unknown responses of each drug from the de-noised input data. For ascertaining whether de-noising improves prediction or not, we carry out 12-fold cross-validation for assessment of the prediction performance. We use the Pearson's correlation coefficient between the true and predicted response values as the prediction performance. De-noising improves the prediction performance for 65% of drugs. Furthermore, we found that this noise reduction method is robust and effective even when a large amount of artificial noise is added to the input data.

Conclusion: We found that our extended off-subspace noise-reduction method combining heterogeneous biological data is successful and quite useful to improve prediction of human cell cancer dru responses from microarray data.

Introduction

Cancer diagnosis based on gene expression data has been widely and extensively explored in the clinical research field since the earlier papers on gene expression arrays were published. Early studies mainly focused on the classification of cancer types, for example, discrimination of leukemia classes, a field in which powerful classifiers such as support vector machines are applied and the predictions are largely successful [2].

Recent cancer phenotype analysis is shifting from predicting a class to predicting a real-valued response. For example, predicting the effects of anti-cancer drugs (In this paper, compounds are referred to as drugs.) is an important problem in cancer therapy, since a careful choice of proper not only drug but also dosage is required for different cancer cells and patients to maximize effectiveness and minimize deleterious side-effects. Although the drug response itself is a continuous quantity, this problem has often been simplified to the binary classification problem of drug sensitive vs. drug resistant [3-5]. Among the drug sensitivity classification studies, Staunton et al. [3] selected 232 out of 5,084 compounds or drugs to classify 60 cells (To be exact, the term 'cell' should be called a cell line.) by a sum of vote type classifier. According to their results, the rate of correct classification is significantly better than random classification.

In constrast to the simplified problem of classification, direct prediction of real-valued responses of a cancer drug from microarray data is not an easy task due to the noisy properties of both microarray technology and living cell experiments. Despite the limitation of available data, Mariadason et al [6] attempted to predict the cell apoptosis response against a chemotherapeutic agent (5-FU) by principal component regression (PCR). The leave-one-out test for 30 different cells in their analysis gives correlation coefficients of predicted and observed responses as low as 0.46. Gruvberger-Saal et al [7] also tried to predict the real-valued response of an estrogen receptor from gene expressions using artificial neural networks used in their earlier study.

In this paper, we focus on the noise and errors in microarray data that potentially degrade prediction performance. De-noising is similar to missing value estimation. Both infer the true values. Typical methods for missing value imputation (e.g. [8,9]) capture the important dimensions by principal component analysis (PCA). However, they do not exploit the *side information* about genes, such as sequence similarity, GO classification, or protein-protein interactions, though those heterogeneous data sources are expected to be useful for identifying related genes, and furthermore to effectively correct noisy data.

We devise a new de-noising method using the side information represented as a *network*, where the nodes correspond to the genes, and the edges represent relations among the genes. In de-noising the expression data of a certain gene, we only look at its neighborhood genes in the network. A principal subspace is made only from the neighborhood expression data, and the target vector is denoised by robust projection (Figure 1). Here, we use a so-called "off-subspace" projection method [1] to prevent over-de-noising. This projection algorithm is formulated as a linear program, which can be efficiently solved even for a large number of neighborhood genes. The basic idea of our method is similar to local PCA approaches [10], where de-noising is done by projection to local subspaces. However, the novelty of our method is that the neighborhood relation is determined by the network.

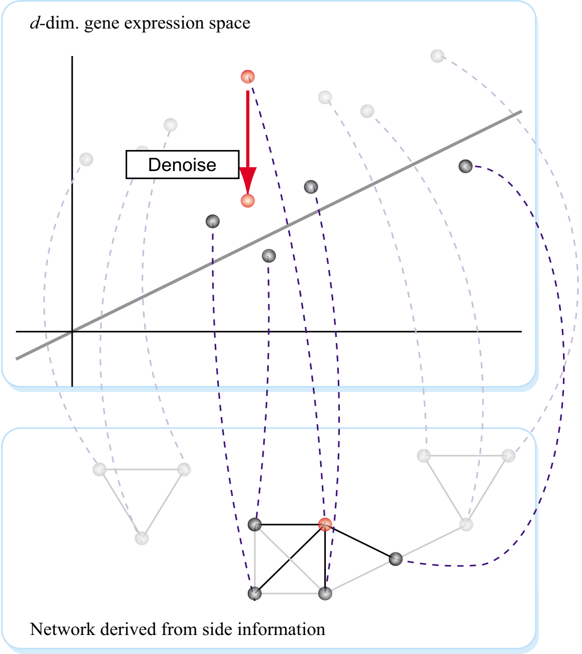

d-dim. gene expression space
Denoise
Network derived from side information

Figure 1
De-noising based on a network. In the top figure, the target expression vector to be de-noised is depicted as a red point. Black points are the neighbors in the network (below) derived from side information; gray points are vectors which are not directly connected (i.e. related) to the target vector in the network. The edge of the network is depicted by a solid line. A dashed curve indicates the correspondence between data in a network and the expression vector. De-noising is done by robust projection onto the principal subspace made from only the neighbors. In this case, the subspace (gray line) is obtained by PCA of the target and the four neighbors.

Typically we have multiple data sources as the side information, which are represented as *multiple networks*. When a principal subspace is derived from each network, we have a set of subspaces for each expression vector. A simple way is to take the sum of all subspaces, and project each target expression vector onto the combined subspace. However, since some of the networks might not be useful for de-noising, it is preferable to select important sub-spaces automatically, and then take the sum of those subspaces. To this end, we extend the off-subspace projection method to deal with multiple networks. Network selection is implemented by giving a non-negative weight parameter to each network, optimizing the weight vector, and removing the networks with zero weights. The de-noising problem for multiple networks is also formulated as a linear program.

In predicting drug responses, it is often the case that the responses for many drugs are predicted from a microarray gene expression dataset. In this case, our problem is to learn a vector-to-vector mapping, where the output vector is composed of drug responses. Since the output vector provided for training is also noisy, our network-based de-noising method can also be applied to the output vector. As we have no side information about drugs, the correlation coefficients among the output vectors are used to construct a network.

In numerical prediction experiments using the drug response data by Staunton et al. [3], our method with multiple networks outperformed standard prediction methods, PCR and the *k*-nearest neighbor method, significantly. Note that the output de-noising was also effective to enhance the accuracy of prediction. The improvement of correlation between true responses and predictions was observed for 930 drugs out of 1,427 drugs. The number (930/1,427 = 0.65) is statistically significant ($p < 10^{-2}$) in a cumulative binomial distribution model under the null hypothesis that half of the drugs (714) are chosen by chance.

De-Noising with a Single Network

The dataset we analyzed contains one microarray hybridization experiment [11] for each cell sample. Let d denote the number of hybridizations and N denote the number genes used in the analysis. We consider the d dimensional space of hybridizations populated by N points representing individual genes. In our drug response case, which has expressions of 60 cell samples for each gene, $d = 60$. Unfortunately in our application, the N points are generally quite noisy. Thus our goal is to "correct" the N points in a way which effectively reduces noise while maintaining signal.

Derivation of Subspaces

Let us define $\boldsymbol{x}_i$ as the d-dimensional gene expression vector of the i-th gene. Our task is to de-noise $\boldsymbol{x}_i$ using other vectors and a network which is represented by the $N \times N$ symmetric matrix W. The (i, j) element w_{ij} represents the strength of the edge between two nodes i and j. If there is no edge, $w_{ij} = 0$. The principal subspace for the i-th gene is computed using the neighborhood nodes only, i.e., the nodes with $w_{ij} \neq 0$. The basis vectors of the subspace are obtained as the principal eigenvectors z_{is}, $s = 1,..., n_i$, of the following covariance matrix,

$$S_i = \frac{\sum_{j=1}^{N} w_{ij} \boldsymbol{x}_i \boldsymbol{x}_j^T}{\sum_{j=1}^{N} w_{ij}} \qquad (1)$$

The weighting covariance matrix represents the distribution of neighbors, and thereby yields the sub-space by taking major eigenvectors as the basis vectors. We determine the number of basis vectors, n_i according to the Kaiser-Guttman rule [12]: the number of basis vectors is set as the number of eigenvalues greater than one in the normalized covariance matrix $\tilde{S}_i$ in which $\left[\tilde{S}_i\right]_{kl} = \left[S_i\right]_{kl} / \sqrt{\left[S_i\right]_{kk}\left[S_i\right]_{ll}}$.

Off-subspace Projection

Most simply, de-noising is done by projecting $\boldsymbol{x}_i$ to the subspace in terms of the least squares error. However, when the number of neighborhood nodes is small, or the non-zero weights w_{ij} are concentrated in only a few neighbors, the dimensionality of the local subspace can be too small. In that case, simple projection may result in an unacceptably large loss of signal, called *over-de-noising*.

Tsuda and Rätsch [1] addressed this by devising an off-subspace projection method (Figure 2) which corrects the data to a point generally closer to, but not necessarily in the subspace. Let $\bar{x}$ denote the de-noised result of $\boldsymbol{x}_i$, which is obtained by solving the following optimization problem,

$$\min_{\bar{\boldsymbol{x}}, v_s} d \left\| \bar{\boldsymbol{x}} - \sum_{s=1}^{n_i} v_s z_{is} \right\|_\infty + \beta_0 \left\| \boldsymbol{x}_i - \bar{\boldsymbol{x}} \right\|_1, \qquad (2)$$

where β_0 is a non-negative constant parameter. Multiplying the first term by d is a convenient normalization which reduces the dependence of the numerical value of β_0 on d. With a suitable transformation in Eq. 2 can be efficiently minimized with standard linear programming.

As can be seen, the objective function in Eq. 2 is the sum of two distances; the distance between the subspace and the off-subspace point $\bar{x}$, and the distance between $\bar{x}$ and the input vector $\boldsymbol{x}_i$. The two distances are measured with different norms; the former with ℓ_∞, the later with ℓ_1.

If $\beta_0 \to \infty$ and if ℓ_∞-norm is replaced with ℓ_2-norm, this becomes a least square method and the resulting off-subspace solution $\bar{x}$ is equal to x_i. As β_0 decreases, $\bar{x}$ becomes close to the subspace.

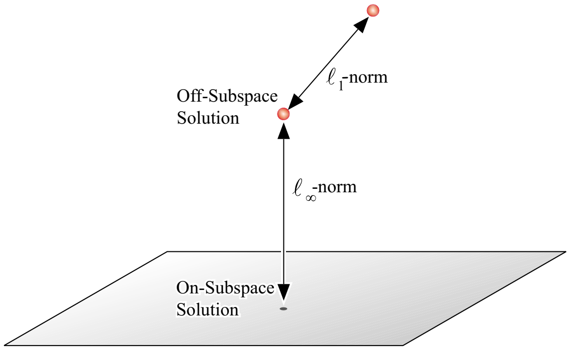


Figure 2
Off-subspace de-noising method. This figure illustrates how to de-noise an expression vector using a principal subspace. Instead of simple projection, we simultaneously find two points (off-subspace solution and on-subspace solution) which minimize the sum of ℓ_1-norm distance and ℓ_∞-norm distance given in Eq. 2.

Now let us describe the ℓ_1-norm and ℓ_∞ norm. Since the ℓ_1-norm regularizer yields a sparse vector as the optimal solution, this algorithm almost corrects only the contaminated elements in $\boldsymbol{x}_i$ without change of the non-contaminated elements. The ℓ_∞ norm has similar behavior to the ℓ_2-norm [13]. Indeed, experiments in [1] show that the ℓ_∞ and ℓ_2 achieve similar de-noising performance. If ℓ_2 is used, the optimization problem can be transformed into a quadratic programming problem. If ℓ_∞ is used, the problem is a linear program, which can be solved more quickly and more stably than quadratic programs.

De-Noising with Multiple Networks

We further introduce a way to incorporate heterogeneous data sources into the noise reduction process by weighting the multiple networks W_k, $k = 1,..., m$. The advantage of this method is its ability to incorporate various kinds of biological knowledge into a single framework.

Denote the s-th basis vector obtained from the k-th network by $z_{is}^{(k)}$. To put it more precisely, each network yields the weighted covariance matrix $S_i^{(k)}$ and we take $n_i^{(k)}$ basis vectors $z_{is}^{(k)}$, ($s = 1,..., n_i^{(k)}$) from $S_i^{(k)}$. To use all the subspaces gained from the networks, one can take the sum of the subspaces and apply the off-subspace projection method. Then, the optimization problem can be described as

$$\min_{\bar{\boldsymbol{x}}, v_{is}^{(k)}} d\|\bar{\boldsymbol{x}} - \sum_{k=1}^{m}\sum_{s=1}^{n_i^{(k)}} v_{is}^{(k)} \boldsymbol{z}_{is}^{(k)}\|_\infty + \beta_0 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_1. \qquad (3)$$

Note that any point in the sum of the subspaces can be represented as $\sum_{k=1}^{m}\sum_{s=1}^{n_i^{(k)}} v_{is}^{(k)} z_{is}^{(k)}$. To automatically select important subspaces, we need to introduce a regularization term to the above optimization problem so that all the coefficients $v_{is}^{(k)}$ of unnecessary networks degenerate to zero.

For that, we introduce the upper bound of the absolute values of the coefficients of the k-th subspace as

$$t_k = \max_{1 \le s \le n_i} \left| v_{is}^{(k)} \right| = \left\| \left[v_{i,1}^{(k)}, \cdots, v_{i,n_i}^{(k)} \right] \right\|_\infty$$

and penalize the ℓ_1-norm of the vector of upper bounds $\boldsymbol{t}$ = $(t_1,..., t_m)^\top$ as follows,

$$\min_{\bar{\boldsymbol{x}}, v_{is}^{(k)}} d\|\bar{\boldsymbol{x}} - \sum_{k=1}^{m}\sum_{s=1}^{n_i} v_{is}^{(k)} \boldsymbol{z}_{is}^{(k)}\|_\infty + \beta_0 \|\boldsymbol{x}_i - \bar{\boldsymbol{x}}\|_1 + \beta_1 \|\boldsymbol{t}\|_1 \qquad (4)$$

Due to the regularizer, some elements of $\boldsymbol{t}$ are exactly zero at the optimal solution. Moreover one can control the number of nonzero elements with the constant parameter β_1. If $t_k = 0$, all the coefficients of the k-th network are zero, implying that that network is not used at all in deriving the de-noised result $\bar{x}$. This optimization problem can also be transformed into a linear program which can be solved efficiently (details not shown).

Experimental Settings

In the following experiments, our task is to predict the drug resistance levels of cells based on their gene expression data. By combining our novel de-noising methods and a standard prediction method, we wish to predict the resistance levels of the *test cells* accurately by learning from the input-output relations of the *training cells*.

The drug resistance dataset by Staunton et al. [3] contains the expression level of 6,817 genes from 60 human cancer cells. Among them, we pre-selected 2,067 highly variant genes (details not shown). For each cell, the drug resistance levels for 5,084 drugs are available as well. Those resistance levels are measured on a continuous scale by growth inhibition score (GI50). Prediction would be too easy, when the drug resistance levels are almost constant among cells. So we chose 1,427 drugs whose gap between the maximum and minimum levels was more than 0.5 after log-normalization.

We applied our network-based de-noising method in two ways. First, the expression profiles, i.e., the input of prediction, are de-noised with various networks. In the following experiments, we built three networks based on the correlation coefficients of the profiles, the gene ontology (GO), and the protein-protein interactions. Here, the vector being the de-noised $\boldsymbol{x}_i$ is the 60-dimensional expression vector of the *i*-th gene. Second, we also de-noised the vector of drug resistance levels, i.e., the output of prediction. Here, the vector $\boldsymbol{x}_i$ is composed of resistance levels of training cells for the *i*-th drug, and apply Eq. 4 to de-noise $\boldsymbol{x}_i$. In this case, we used only one network based on the correlation coefficients of the resistance level vectors. Namely, $m = 1$.

A schematic representation of the entire process is shown in Figure 3.

For predicting drug responses from the de-noised expression data, we tested two kinds of standard prediction algorithms. One is principal component regression (PCR), the other is *k*-nearest-neighbors (kNN). Both algorithms have one constant parameter to be tuned manually, that is, the number of principal components and the number of nearest neighbors, respectively.

Principal Component Regression

PCR is also used in Mariadason et al. [6]. For training cells, the Pearson correlation between the de-noised expression data of each of the 3,725 genes and (de-noised) responses of a drug of interest were calculated, and the 50 highest absolute value correlations (i.e., corresponding to 50 genes) were selected. To reduce the number of genes to a smaller set of variables, PCA was performed. From the PCA, the principal components (PCs) having the d_{pca} larg-

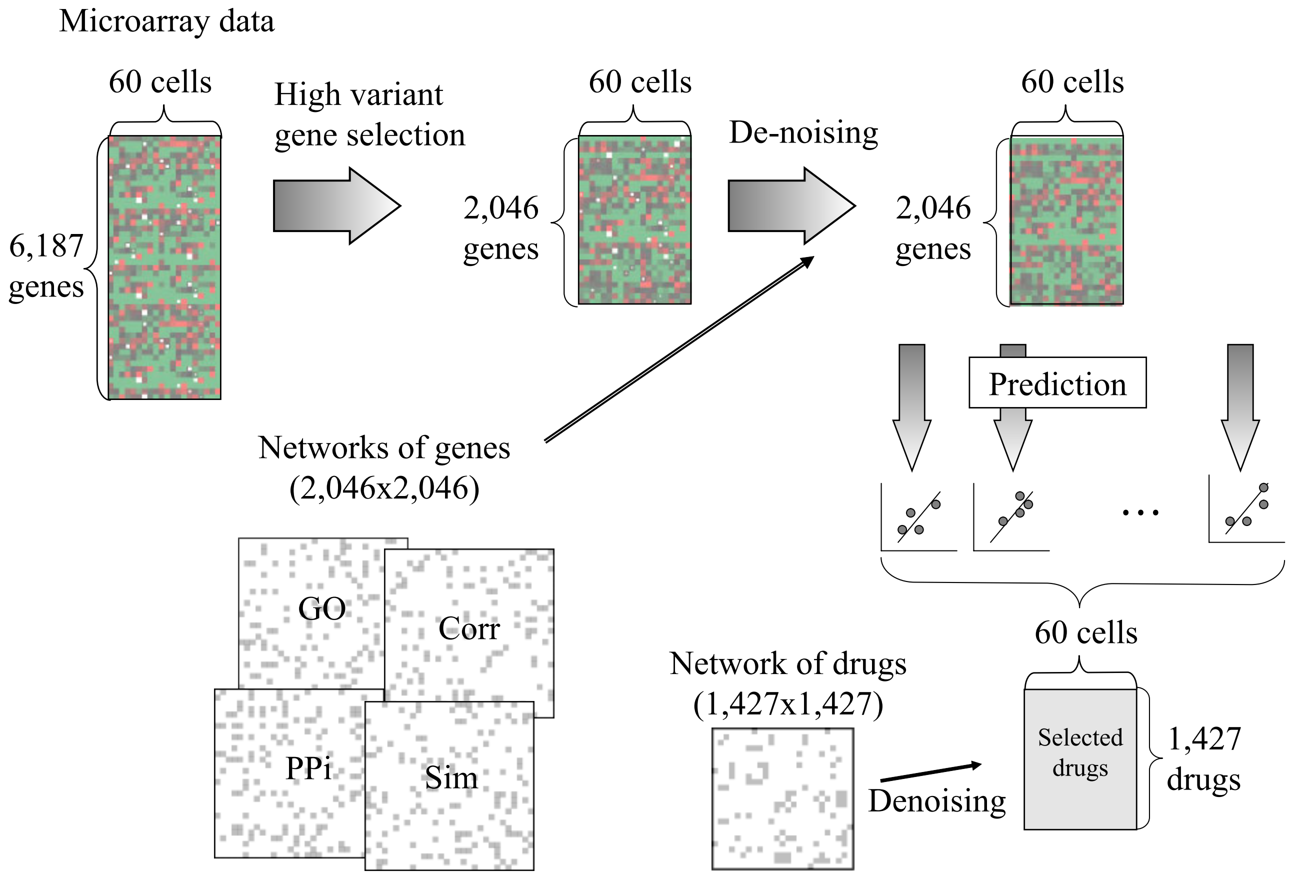


Figure 3
Method overview.

est eigenvalues were selected. Next we get the parameters, $\boldsymbol{a} \in \Re^{d_{\mathrm{pca}}}$ and $b \in \Re$, of a linear regression function $g(\boldsymbol{z}|\boldsymbol{a},b) = \boldsymbol{a}^{\top}\boldsymbol{z} + b$ by least square manner, where $\boldsymbol{z} \in \Re^{d_{\mathrm{pca}}}$ is the d_{pca} PCs of a cell. Namely, we find the values of $\{\boldsymbol{a}, b\}$ which minimize the sum-of-square errors:

$$\sum_i \left(g\left(z_i | \boldsymbol{a}, b \right) - y_i \right)^2$$

where $z_i \in \Re^{d_{\mathrm{pca}}}$ and y_i are the PCs and the drug response of i-th training cell, respectively. Once the regression function $g(\boldsymbol{z}|\boldsymbol{a}, b)$ was derived, the d_{pca} PCs corresponding to the test cell were computed and substituted into the derived regression function to yield a prediction of response of the test cell.

k-Nearest Neighbor Prediction

Using the same manner as the above Mariadason et al.'s method [6], 50 genes are selected. Then each cell has a 50-dimensional vector. For each test cell, we find the k_{nn} closest cells and predict the drug response of the test cell by the average of k_{nn} responses.

Building Networks

In this section, we describe the details of network construction, namely, the computation of matrix W. The first four networks are used for de-noising the input (i.e., expression data), and the last one is for the output (i.e., drug resistance levels).

Expression correlation

Presumably, as seen in missing value estimation, the most informative relations between genes for noise reduction can be obtained from co-expression. Thus, we use the Pearson correlation coefficient of expression as one of the heterogeneous data sources. The correlation is converted to probability under the hypothesis of $H0 : r = 0$ using one sample t-test, and the probability value is assigned to the weight w_{ij}. Notice that the expression data are not only the input of prediction, but also used for de-noising themselves.

Sequence similarity

To build a network based on the sequence similarity, the RNA sequences corresponding to the genes were extracted via GenBank accession numbers described in the annotation fields of the gene expression data. The sequence similarity was computed by FASTA. Only the highest e-value between two sequences was used when there were multiple local alignment candidates. If the e-value was more than 10^{-5}, w_{ij} was set to zero, otherwise w_{ij} was set to the negative logarithm of the e-value. Among the 2,046 highly variant genes, only 550 were found to have edges to other genes.

Gene ontology

Gene ontology data were downloaded from the GO annotation project (GOA) at http://www.ebi.ac.uk/GOA/. The GenBank accessions were translated to protein IDs and checked for any GO-relationships for gene pairs at the protein level. The edge strength w_{ij} was determined as the number of GO categories into which both proteins of a pair are classified. We obtained a total of 141,402 GO relations for 1,371 highly variant genes.

Protein-protein interaction

We obtained the protein-protein interaction data from the Biomolecular Interaction Network Database (BIND) at http://www.blueprint.org/bind/bind.php. This network has binary edges, i.e., w_{ij} is 0 or 1.

Drug response correlation

For de-noising drug responses, we calculated the p-values of the correlation coefficients of compound pairs with respect to their drug response data to generate the matrix W.

Parameter Selection

Our de-noising method has the two parameters, β_0 and β_1. In addition PCR and kNN have one constant parameter, d_{pca} and k_{nn}, respectively. For this purpose, we performed a joint grid search over the following values:

$$\beta_0 = 0.1, 0.2, 0.5, 1.0, 2.0, 3.0, 4.0,$$

$$\beta_1 = 0.00, 0.02, 0.05, 0.1, 0.2, 0.5,$$

$$d_{\mathrm{pca}} = 1,2,3,4,5,\ 7,10,\ 20,30,40,50,$$

$$k_{\mathrm{nn}} = 1,2,3,4,5,6,7,8,9,10,$$

and chose the parameter values yielding the best regression performance.

Results

Prediction performances of PCR with and without off-subspace noise reductions for different combinations of networks are shown in the upper plot in Figure 4. The accuracy of prediction is measured by the mean correlation coefficients in 12-fold cross validation. The leftmost bar 'None' corresponds to the performance without any de-noising. As anticipated, integration of both input and output de-noising yields the highest mean correlation coefficient ('All&Drug': 0.439). Among the other cases where only the input is de-noised, we obtained the best result when all networks are combined with weights ('All': 0.428). The weightless combination ($w_{ij} = 1$ for $\forall i, j$) was significantly poorer ('Unweighted'). The lower plot of Figure 4 shows the performance of kNN. The kNN also achieves the best performance when both input and output are de-noised. In the case of de-noising only the input, noise reduction without using side-information degrades the prediction performance, but the use of side-information improves prediction. We also counted the drugs achieving statistically-significant predictions (Table 1). The lists of Drug IDs are available at our supplemental web page at http://www.cbrc.jp/~kato/GI50_prediction/. De-noising successfully increases the predictable drugs, which allows us to put more drugs into our choices for medical treatment.

Statistically, the PCR of 868 among 1,427 drugs was improved by input de-noising. The number increased to 930 when the output was also de-noised. The drugs that achieve the best improvement after de-noising are listed in Table 3. The correlation coefficient for the top drug (NSC ID: 642049) was boosted from R = 0.42 to 0.71, which corresponds to more than a five order of magnitude increase in significance ($p = 2.7 \times 10^{-3}$ to 8.4×10^{-9}). The second top drug (NSC ID: 644945) improved from R = 0.51 to 0.69, which is also a large improvement from $p = 3.9 \times 10^{-5}$ to 1.38×10^{-9}. Regressions for these two drugs are plotted in Figure 5. The top 100 drugs with compound

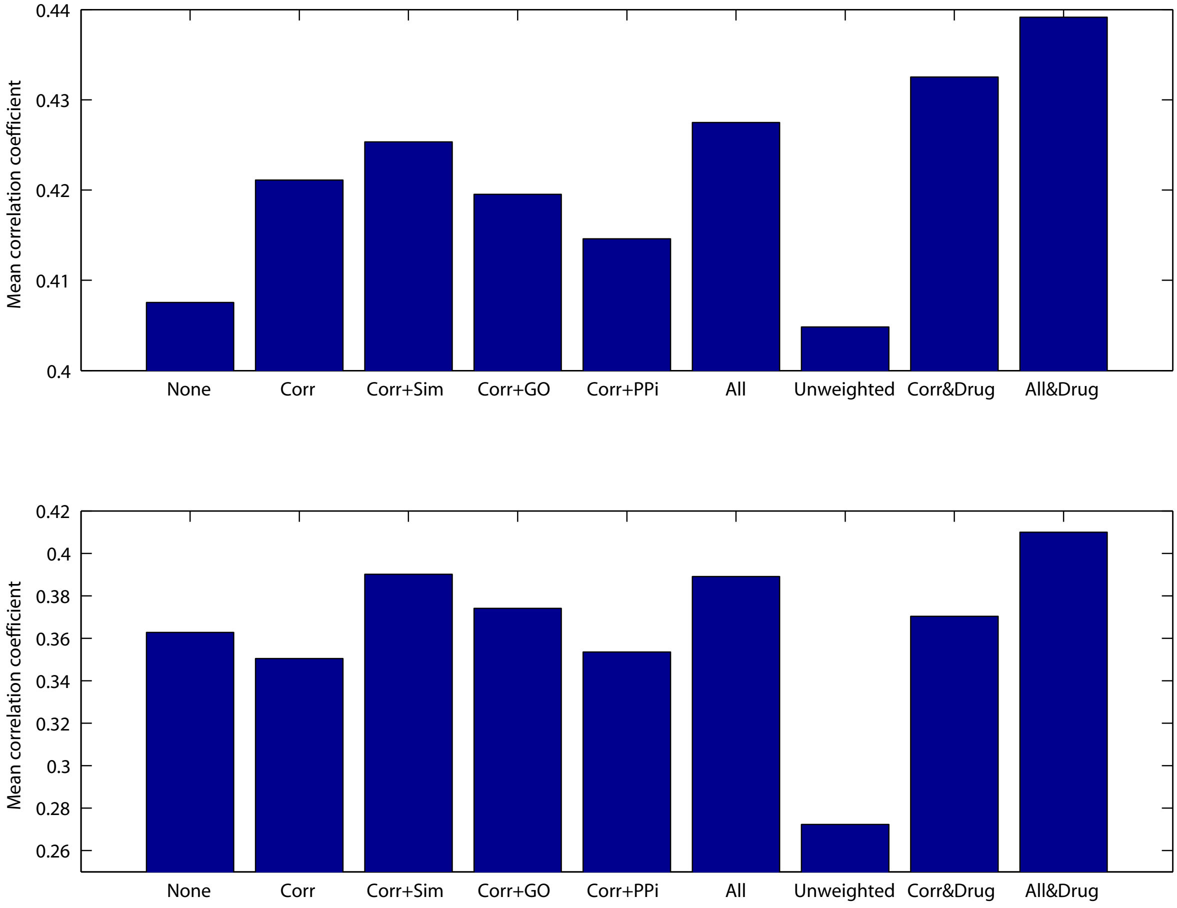


Figure 4

Improvement of prediction by noise reduction with various combinations of networks. The upper figure plots the results of PCR, the bottom of kNN. The mean correlation coefficients of prediction for 1,427 drugs before and after the off-subspace noise reduction are shown. Abbreviations are Corr: correlation coefficient for gene expressions; Sim: sequence similarity; GO: gene ontology; PPi: protein-protein interaction; All: Corr+Similarity+GO+PPi; Corr&Drug: input de-noising only via Corr and output de-noising; and All&Drug: input de-noising using All and output de-noising.

Table 1: The number of drugs given statistically-significant prediction of the responses. Here we define a drug that achieves the correlation coefficient more than 0.33 as a successfully predicted, which is derived from one sample t-test with the probability less than 0.01 examining the null hypothesis of "no correlation."

	None	Corr	Corr+Sim	Corr+GO	Corr+PPi	All	Unweighted	Corr&Drug	All&Drug
PCR	983	1,027	1,043	1,018	995	1,037	988	1,065	1,085
kNN	867	794	925	886	815	937	565	861	994

Table 2: Best hyper-parameters of our de-noising method

		Corr	Corr+Sim	Corr+GO	Corr+PPi	All	Unweighted	Corr&Drug	All&Drug
PCR	β_0	1	1	1	1	1	4	1	1
	β_1	0.05	0.05	0.05	0.05	0.05	0	0.05	0.05
kNN	β_0	1	1	1	1	1	1	1	1
	β_1	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05

Table 3: The top 10 drugs by PCR performance improvement. Each row contains: a Drug ID; the NSC number of the drug; #Cells: the number of available cells; R1, the correlation coefficient before noise reduction; R2, the correlation coefficient after noise reduction; p(R1), the p-value (by one sample t-test) for R1; and p(R2), the p-value for R2.

Top N	Drug ID	#Arrays	R1	R2	p(R1)	p(R2)	P(R1)/P(R2)
1	642049	49	0.42	0.71	2.71E-03	8.41E-09	3.22E+05
2	644945	59	0.51	0.69	3.85E-05	1.38E-09	2.79E+04
3	691277	57	0.42	0.65	1.18E-03	5.75E-08	2.05E+04
4	645803	60	0.53	0.68	1.39E-05	2.83E-09	4.91E+03
5	32946	60	0.37	0.59	3.51E-03	8.01E-07	4.38E+03
6	628672	59	0.59	0.71	9.04E-07	2.06E-10	4.39E+03
7	642710	60	0.57	0.7	1.55E-06	4.00E-10	3.87E+03
8	665128	57	0.58	0.69	2.34E-06	2.16E-09	1.08E+03
9	694265	60	0.49	0.63	6.35E-05	6.40E-08	9.92E+02
10	668334	60	0.62	0.72	9.44E-08	1.07E-10	8.79E+02

Table 4: Automatic selection of heterogeneous networks. Abbreviations are Corr: correlation coefficient for gene expressions; Sim: sequence similarity; GO: gene ontology; PPi: protein-protein interaction. Our de-noising method automatically selects networks. The value of t_k indicates whether or not a network is used for de-noising. If t_k is non-zero, we interpret that the *k*-th network is used.

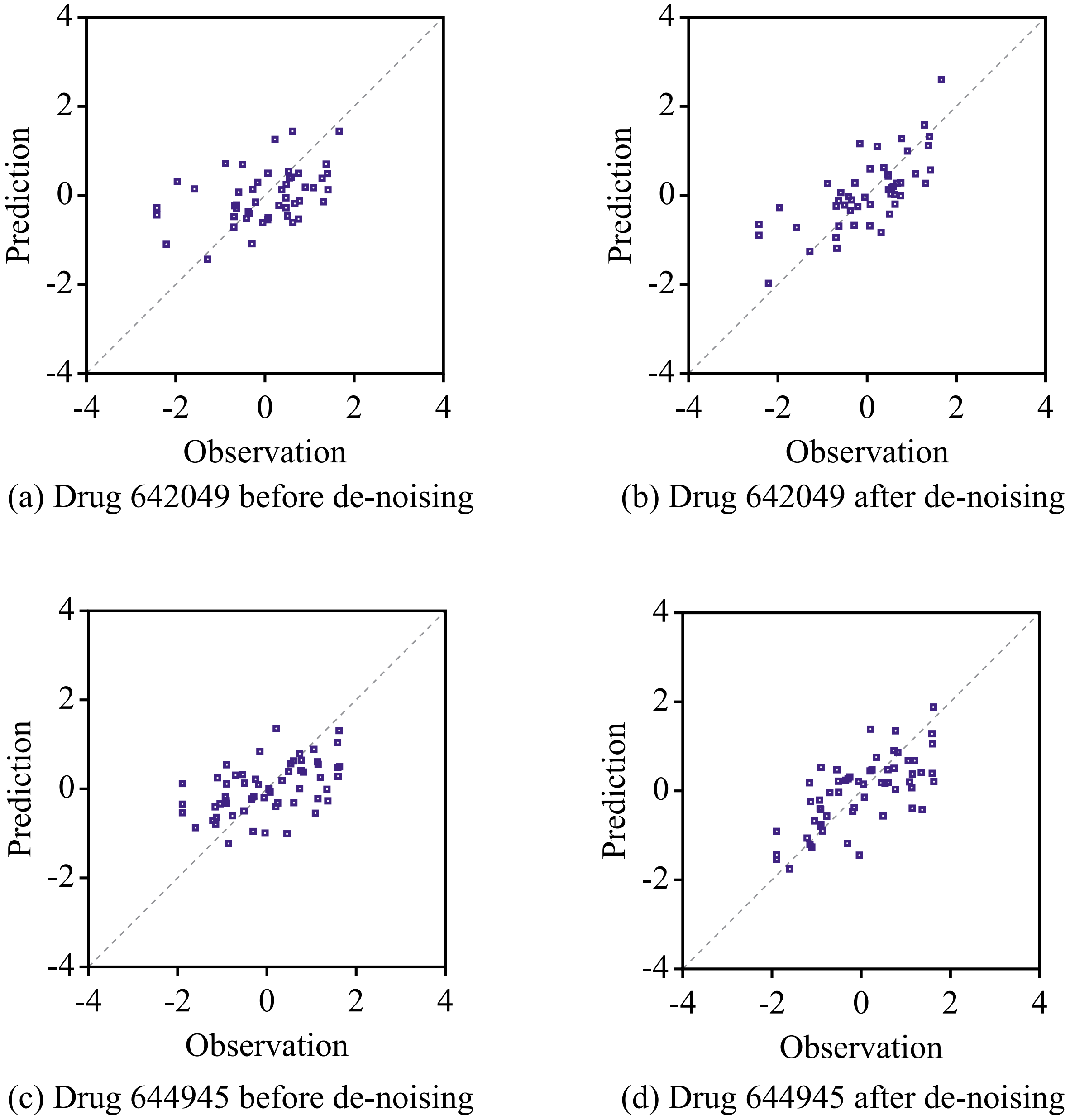
	Corr	Sim	GO	PPi
Used ($t \neq 0$)	2046	541	1256	75
Unused ($t = 0$)	0	9	115	64

names and their numerical data are shown on our supplemental web page.

Our de-noising method uses two hyper-parameters, β_0 and β_1. In the experiments reported here we determined their values by a grid search over 7 × 6 = 42 combinations and chose the one giving the best mean correlation coefficient for the 1,427 compounds. The best parameters are shown in Table 2. The most typical values are $\beta_0 = 1.0$ and $\beta_0 = 0.05$.

Importance of each network

In this study, we incorporated four types of heterogeneous data as networks. To analyze which net-work contributed to the noise reduction process, we checked the parameter t_k to count how often each network was used in the de-noising of the 2,046 expression profiles. If $t_k \neq 0$ at the optimal solution, we interpreted that the *k*-th network was used for de-noising. Our networks have *isolated nodes*, which do not have edges at all. The number of non-isolated nodes is as 2,046, 550, 1,371 and 196 for the net-



(a) Drug 642049 before de-noising

(b) Drug 642049 after de-noising

(c) Drug 644945 before de-noising

(d) Drug 644945 after de-noising

Figure 5
Compounds with improved regressions by PCR.

works 'Corr,' 'Sim,' 'GO' and 'PPi,' respectively. In deriving the importance of each network, we counted the number of drugs which use that network for de-noising.

Table 4 shows the importance of the data sources. We found that expression correlation contributed most frequently to the noise reduction of the entire 2,046 genes, which is not surprising because this network is by far the densest among the four, and because it is the unique data which are not only the input of prediction, but also used for de-noising themselves. Sequence similarity also seems very important; contributing to noise reduction in

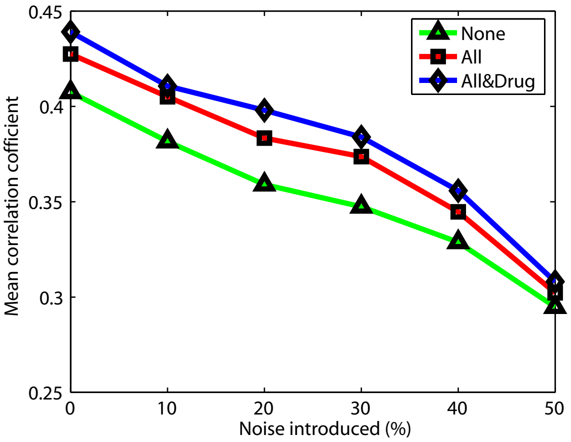


Figure 6
Effect of noise reduction for various levels of introduced noise.

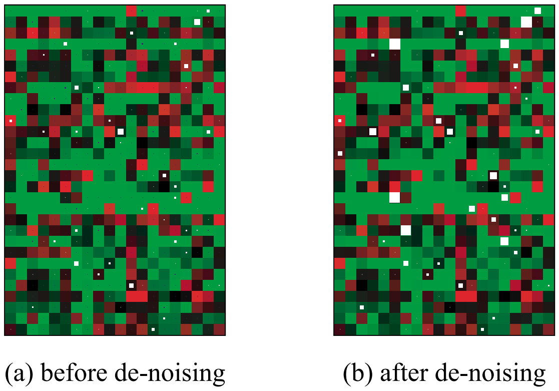


Figure 7
Examples of noise reductions on microarray data. The matrices show our microarray data for selected genes. The size of the white boxes indicate the magnitude of the added noise. In (a), 10% of the expression values are contaminated. The subplot (b) shows the result after de-noising using all of the networks.

98.4%(541/550) of the genes. The figure for the GO data was 91.6%(1256/1371). In contrast, only protein-protein interaction data contributed in 54.0% at the genes. We think this is mainly due to the known problem of protein-protein interaction data coming from unreliable methods such as yeast two-hybrid system. In addition this data source gave only all-or-none gene relations (1-0 type weighting scheme), which might conceivably lead to more a binary type of incorporation of data.

Robustness against perturbed array data

To test our de-noising method in more noisy situations, we added white noise to the expression profiles. The noise is derived from a normal distribution with twice the standard deviation as the one calculated from all the data. The fraction of noise-contaminated genes was changed from 10% to 50%. PCR is used for this simulation. The degradation of the prediction accuracy by noise is shown in Figure 6. The positive effect of noise reduction is clear for all noise levels but drops off significantly when the noise level reaches 50%. Indeed, applying noise reduction to both expression and drug response ('All&Drug') gives a higher correlation coefficient (0.411) at the 10% noise level than the simple PCR ('None') without noise contamination (0.408). Finally, we show a visual example of de-noising in Figure 7.

Discussion and conclusion

The prediction of drug response data is critical for the field of cancer therapeutics, which demands improved diagnostics for determining the appropriate choice and dosage of anti-cancer drugs. Combining gene relations from various biological resources to adjust values of gene expressions or drug response data is a new approach in this field. This approach requires effective methods, such as the one presented here, for utilizing heterogeneous data.

This algorithm is invariant if the network weights are multiplied by a constant, as shown in Eq. 1. However, the change of the ratio among weights may have an influence on the de-noising performance. Although various weighting schemes could be considered, we did not systematically investigate that issue in this work. However we did confirm that off-subspace noise reduction with the continuous weights defined above for sequence similarity, expression correlation, and GO heterogeneous data sources was more effective than using a 0–1 weighting scheme based on some threshold. The current weighting scheme might not yet be optimal and tuning would yield improvement to some extent.

We extended the off-subspace noise reduction method of Tsuda and Rätsch [1] and applied it to the noise reduction of gene expression data in the context of real-value prediction to drug response data. Our results show the method to be robust to noisy data and more effective than the traditional principal component regression, improving the prediction of 868 out of 1,427 drugs. We expect it will prove generally useful for correcting the values of noisy microarray data.

Authors' contributions

TK designed the core of the method and carried out the computational experiments. YM and KM conceived of the necessity of the study and contributed the understanding of the problem involved in the cell study using microarrays. PBH and KT rewrote much of the manuscript, restructuring the exposition and refining the wording. WF prepared the biological data, carried out the statistical analysis, coordinated the project, and wrote the first draft of the manuscript. All authors helped to draft the manuscript and approved the final manuscript.

Acknowledgements

The authors deeply thank to two anonymous reviewers for careful reading and plenty of helpful suggestions.

References

1. Tsuda K, Rätsch G: **Image Reconstruction by Linear Programming.** In *Advances in Neural Information Processing Systems 16* Edited by: *Thrun S, Saul L, Schölkopf B. Cambridge, MA: MIT Press*; 2004.
2. Furey TS, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D: **Free full text support vector machine classification and validation of cancer tissue samples using microarray expression data.** *Bioinformatics* 2000, **16(10):**906-914.
3. Staunton JE, Slonim DK, Coller HA, Tamayo P, Angelo MJ, Park J, Scherf U, Lee JK, Reinhold WO, Weinstein JN, Mesirov JP, Lander ES, Golub TR: **Chemosensitivity prediction by transcriptional profiling.** *Proc Natl Acad Sci USA* 2001, **98(19):**10787-10792.
4. Kaneta Y, Kagami Y, Katagiri T, Tsunoda T, Jinnai I, Taguchi H, Hirai H, Ohnishi K, Ueda T, Emi N, Tomida A, Tsuruo T, Nakamura Y, Ohno R: **Prediction of sensitivity to STI571 among chronic myeloid leukemia patients by genome-wide cDNA microarray analysis.** *Jpn J Cancer Res* 2002, **93(8):**849-856.
5. Okutsu J, Tsunoda T, Kaneta Y, Katagiri T, Kitahara O, Zembutsu H, Yanagawa R, Miyawaki S, Kuriyama K, Kubota N, Kimura Y, Kubo K, Yagasaki F, Higa T, Taguchi H, Tobita T, Akiyama H, Takeshita A, Wang YH, Motoji T, Ohno R, Nakamura Y: **Prediction of chemosensitivity for patients with acute myeloid leukemia, according to expression levels of 28 genes selected by genome-wide complementary DNA microarray analysis.** *Mol Cancer Ther* 2002, **1(12):**1035-1042.
6. Mariadason JM, Arango D, Shi Q, Wilson AJ, Corner GA, Nicholas C, Aranes MJ, Lesser M, Schwartz EL, Augenlicht LH: **Gene expression profiling-based prediction of response of colon carcinoma cells to 5-fluorouracil and camptothecin.** *Cancer Res* 2003, **63(24):**8791-8812.
7. Gruvberger-Saal SK, Eden P, Ringner M, Baldetorp B, Chebil G, Borg A, Ferno M, Peterson C, Meltzer PS: **Predicting continuous values of prognostic markers in breast cancer from microarray gene expression profiles.** *Mol Cancer Ther* 2004, **3(2):**161-168.
8. Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, Altman RB: **Missing value estimation methods for DNA microarrays.** *Bioinformatics* 2001, **17(6):**520-525.
9. Bo TH, Dysvik B, Jonassen I: **LSimpute: accurate estimation of missing values in microarray data with least squares methods.** *Nucleic Acids Res* 2004, **32(3):**e34.
10. Tipping M, Bishop C: **Mixtures of Probabilistic Principal Component Analyzers.** *Neural Computation* 1999, **11(2):**443-482.
11. Alberts B, Johnson A, Lewis J, Raff M, Roberts K, Walter P: *Molecular Biology of the Cell. Microarrays Monitor the Expression of Thousands of Genes at Once, III. Methods/8. Manipulating Proteins, DNA, and RNA/Studying Gene Expression and Function* 4th edition. *New York: Garland Publishing 2002 chap* .
12. Guttman L: **Some necessary conditions for common-factor analysis.** *Psychometrika* 1954, **19:**149-161.
13. Tibshirani R: **Regression selection and shrinkage via the lasso.** *Journal of the Royal Statistical Society Series B* 1996, **58:**267-288.