Supplementary Notes

A community effort to assess and improve drug sensitivity prediction algorithms

James C Costello[1,2,13,14], Laura M Heiser[3,14], Elisabeth Georgii[4,14], Mehmet Gönen[4], Michael P Menden[5], Nicholas J Wang[3], Mukesh Bansal[6], Muhammad Ammad-ud-din[4], Petteri Hintsanen[7], Suleiman A Khan[4], John-Patrick Mpindi[7], Olli Kallioniemi[7], Antti Honkela[8], Tero Aittokallio[7], Krister Wennerberg[7], NCI DREAM Community[9], James J Collins[1,2,10], Dan Gallahan[11], Dinah Singer[11], Julio Saez-Rodriguez[5], Samuel Kaski[4,8], Joe W Gray[3] & Gustavo Stolovitzky[12]

[1]Howard Hughes Medical Institute, Boston University, Boston, Massachusetts, USA. [2]Department of Biomedical Engineering, Boston University, Boston, Massachusetts, USA. [3]Department of Biomedical Engineering, Oregon Health and Science University, Portland, Oregon, USA. [4]Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, Espoo, Finland. [5]European Molecular Biology Laboratory-European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge, UK. [6]Department of Systems Biology, Center for Computational Biology and Bioinformatics, Columbia University, New York, New York, USA. [7]Institute for Molecular Medicine Finland FIMM, University of Helsinki, Helsinki, Finland. [8]Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland. [9]List of participants and affiliations appear at the end of the paper. [10]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, Massachusetts, USA. [11]National Cancer Institute, National Institutes of Health, Bethesda, Maryland, USA. [12]IBM T.J. Watson Research Center, IBM, Yorktown Heights, New York, USA. [13]Present address: Department of Pharmacology, University of Colorado Anschutz Medical Campus, Aurora, Colorado, USA. [14]These authors contributed equally to this work.

Correspondence should be addressed to S.K. (samuel.kaski@aalto.fi), J.W.G. (grayjo@ohsu.edu), or G.S. (gustavo@us.ibm.com).

Nature Biotechnology: doi:10.1038/nbt.2877

Table of Contents

Nature Biotechnology: doi:10.1038/nbt.2877

Supplementary Note 1: NCI-DREAM Drug Sensitivity Prediction Methods

Kernel method 1

Summary
Multiple views of the genomic datasets were generated, training and predictions were made using a kernelized regression method that combines multitask and multiview learning, and uses Bayesian inference to estimate model parameters.

Introduction
When there are multiple outputs or related tasks (here, predicting sensitivities of cell lines against a number of drugs), one can consider learning them simultaneously instead of treating them as independent problems, which is known as *multitask learning*. It is expected to perform better than learning independent models due to the possibility of capturing the correlation between different outputs or tasks during training. When multiple representations for data samples are available, instead of using a single representation, it is possible to learn a model of the underlying phenomena using all of the representations together, which is known as *multiview learning*. There are different multiview learning strategies, and *multiple kernel learning* is a principled way of combining multiview learning and kernel-based learning to introduce nonlinearity into the model[1]. Since the challenge problem included both multiple outputs and multiple input representations, we integrated multitask learning and multiview learning into a kernelized regression model. We formulate a novel probabilistic algorithm that uses a common similarity measure among the tasks (outputs) by sharing the weights over the kernels calculated on different representations.

Methods
Bayesian multitask multiple kernel learning: In order to obtain a Bayesian multitask multiple kernel learning algorithm, a fully conjugate probabilistic model is formulated and a deterministic variational approximation mechanism is used for inference. **Figure K1** illustrates the proposed probabilistic model with a graphical model. The main idea is to calculate intermediate outputs from each kernel using the same set of sample weight parameters and to combine these intermediate outputs using the kernel weights and the biases to estimate the target outputs[2].

Nature Biotechnology: doi:10.1038/nbt.2877

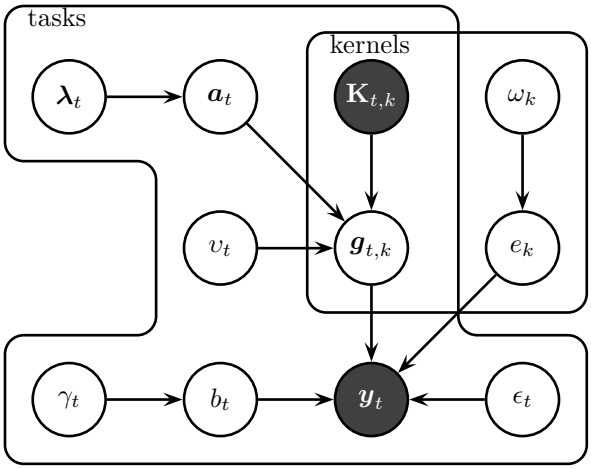


Figure K1 Graphical model of Bayesian multitask multiple kernel learning. The shaded nodes are observed variables while the other nodes are random variables.

The notation used is as follows: the subscripts t, i, and k index tasks, training samples, and kernels, respectively. The numbers of tasks, training samples for each task, and input kernels are denoted by T, N_t, and K, respectively. The $N_t \times N_t$ kernel matrices for each task-view pair are denoted by $K_{t,k}$. The $N_t \times 1$ vectors of weight parameters $a_{t,i}$ and their priors $\lambda_{t,i}$ are denoted by a_t and λ_t, respectively. The precision priors for intermediate outputs are denoted by υ_t. The $N_t \times K$ matrices of intermediate outputs are represented as G_t, where the columns of G_t are represented as $g_{t,k}$. The bias parameters and their priors are denoted by b_t and γ_t, respectively. The $K \times 1$ vectors of kernel weights e_k and their priors ω_k are denoted by e and ω, respectively. The precision priors for target outputs are denoted by ε_t. The $N_t \times 1$ vectors of target outputs are represented as y_t.

The distributional assumptions of the proposed model are defined as

$$\lambda_{t,i} \sim \mathcal{G}\left(\lambda_{t,i};\ \alpha_\lambda, \beta_\lambda\right) \quad \forall (t,i)$$
$$a_{t,i}|\lambda_{t,i} \sim \mathcal{N}\left(a_{t,i};\ 0, \lambda_{t,i}^{-1}\right) \quad \forall (t,i)$$
$$\upsilon_t \sim \mathcal{G}(\upsilon_t;\ \alpha_\upsilon, \beta_\upsilon) \quad \forall t$$
$$g_{t,k}|a_t, K_{t,k}, \upsilon_t \sim \mathcal{N}\left(g_{t,k};\ K_{t,k}a_t, \upsilon_t^{-1}I\right) \quad \forall (t,k)$$
$$\gamma_t \sim \mathcal{G}\left(\gamma_t;\ \alpha_\gamma, \beta_\gamma\right) \quad \forall t$$
$$b_t|\gamma_t \sim \mathcal{N}(b_t;\ 0, \gamma_t^{-1}) \quad \forall t$$
$$\omega_k \sim \mathcal{G}(\omega_k;\ \alpha_\omega, \beta_\omega) \quad \forall k$$
$$e_k|\omega_k \sim \mathcal{N}(e_k;\ 0, \omega_k^{-1}) \quad \forall k$$
$$\varepsilon_t \sim \mathcal{G}(\varepsilon_t;\ \alpha_\varepsilon, \beta_\varepsilon) \quad \forall t$$
$$y_t|b_t, e, G_t, \varepsilon_t \sim \mathcal{G}\left(y_t;\ \sum_{k=1}^{K} e_k g_{t,k} + b_t 1, \varepsilon_t^{-1} I\right) \quad \forall t$$

where $\mathcal{N}(\cdot;\ \mu, \Sigma)$ represents the normal distribution with mean vector μ and covariance matrix Σ, and $\mathcal{G}(\cdot;\ \alpha, \beta)$ denotes the gamma distribution with the

Nature Biotechnology: doi:10.1038/nbt.2877

shape parameter α and the scale parameter β.

Learning model parameters: Exact inference for the probabilistic model is intractable and using Gibbs sampling is computationally expensive. We instead formulate a deterministic variational approximation, which is more efficient in terms of computation time. The variational methods maximize a lower bound on the marginal likelihood using a factored approximation of the posteriors to find the joint parameter distributions[3]. We can write the factorable ensemble approximation of the required posterior by defining each factor in the ensemble just like its full conditional distribution. We can bound the marginal likelihood using Jensen's inequality and optimize this bound by maximizing with respect to each factor iteratively until convergence. For the proposed model, thanks to the full conjugacy, the resulting approximate posterior distribution of each factor follows the same distribution as the corresponding factor.

Formulating the challenge as a multitask multiple kernel learning problem: The problem of predicting cell line sensitivities against different drugs can be cast into multitask multiple kernel learning. In addition to the genomic measurement data, three types of knowledge-enhanced data views were computed from the measurement data: (i) gene set views aggregating measurements across functionally related genes as defined in C2 and CP collections from MSigDB[4], (ii) measurement combination views integrating expression with copy number variation or DNA methylation, using the PARADIGM tool[5] and gene-wise multiplication, and (iii) transformation of continuous data into present-absent calls. This resulted in 22 data views in total.

Each of the views were used to calculate a kernel between cell lines, using the Gaussian kernel for real-valued data and the Jaccard similarity coefficient for binary-valued data. Drugs are considered to be the tasks in multitask formulation. For this particular problem, the notation defined for tasks, kernels, and cell lines can be interpreted as follows:

- t: the index for drugs,
- k: the index for genomic views,
- i: the index for cell lines,
- T: the number of drugs,
- K: the number of genomic views,
- N: the number of cell lines in the training set.

After learning the shared kernel weights $\boldsymbol{e}$, the drug-specific cell line weights $\boldsymbol{a}_t$, and the drug-specific biases b_t, the sensitivity values for test cell lines can be calculated. The predicted sensitivity values are converted into rankings by sorting them.

Calculating similarities between cell lines: For real-valued genomic views, the training samples are first normalized to zero mean and unit standard deviation using z-normalization. Then, the similarity between cell lines is calculated using the Gaussian kernel, which is defined as

Nature Biotechnology: doi:10.1038/nbt.2877

$$k_{t,k}(x_{t,k,i}, x_{t,k,j}) = \exp\left(-\frac{||x_{t,k,i} - x_{t,k,j}||^2}{2\sigma_{t,k}^2}\right) \quad \forall(t,k,i,j)$$

where $\sigma_{t,k}^2$ is set to the dimensionality (*i.e.*, the number of features) of the corresponding genomic view.

For binary-valued genomic views, the similarity between cell lines is calculated on the original training samples using the Jaccard similarity coefficient, which is defined as

$$k_{t,k}(x_{t,k,i}, x_{t,k,j}) = \frac{x_{t,k,i}^T x_{t,k,j}}{x_{t,k,i}^T x_{t,k,i} + x_{t,k,j}^T x_{t,k,j} - x_{t,k,i}^T x_{t,k,j}} \quad \forall(t,k,i,j)$$

where it is guaranteed to take values between 0 and 1, similar to the Gaussian kernel.

Discussion

In post-analysis the nonlinearity due to the kernels turned out to have the largest impact on the model performance, followed by multitask learning and (weighted) multiple kernel learning. The learned combination of data views performed better than any individual data view. Gene expression was the most informative original data view, but further performance was gained by integrating it with gene set views of the same data. Prediction performance varied across drugs, but was above team average for drugs with wider dynamic range. Further improvements by designing views and selecting features are conceivable.

Kernel method 3

Summary

Separate normalizations were applied to each dataset, several SVM classifiers were independently trained (varying kernels and input data), and final predictions were made using a weighted average of all SVM outputs.

Introduction

The NCI-DREAM challenge is to train a robust classifier that predicts dose responses from the supplied genomic datasets. We have used support vector machines (SVM)[6] for this supervised regression problem, as SVMs are robust machine learning tools for learning supervised data and are able to capture linear, as well as various non-linear relationships with the use of different kernels. Publicly available libSVM library for MATLAB was used for our implementation.[7]

One challenge is that six genomic datasets are available, each of which could be processed in different ways, and further, different SVM kernels may be more or less appropriate for different datasets (see **Table K3**). For effective prediction, the optimal combination of these parameters needs to be discovered. To address this problem, we used an computationally expesive cross-validation (CV) approach that separately trains a classifier and predicts dose response for the test cell lines for all these combinations, evaluates the classifiers for each

Nature Biotechnology: doi:10.1038/nbt.2877

combination on the training dataset, and then integrates the predictions on the test cell lines based on the evaluations.

Methods

Setting model parameters: For each drug, several different models were evaluated based on dataset, data normalization, and SVM parameters (see **Table K3**).

Table K3: All combination setups of dataset, data normalization and SVM kernel parameter

Datasets	Data Normalization (L2 normalization of genes x cell lines matrix)	SVM kernel parameter
Gene expression	Raw	Linear x 3 (c = 1, 10, 100)
Methylation	Row (gene) normalization	Polynomial x 3 (c = 1, 10, 100)
RNAseq	Col (cell line) normalization	Radial x 3 (c = 1, 10, 100)
RPPA	Row-col normalization (normalized row first the column)	Sigmoid x 3 (c = 1, 10, 100)
Exome seq (Mismatch(alt) - Mismatch(ref) was used)	Col-row normalization (normalized column first the row)	

Learning parameters through cross-validation: For training purposes, the training drug response data was randomly divided into two sets: training and test. An SVM was trained on the training set using 20 iterations of bagging.[8] Each iteration of bagging used 80% of the training set, and the SVM model was used to predict drug response results on the held out test half. The median of the 20 values was used as the actual prediction. The full learning process was repeated 20 times. For each repeat, the SVM predictions on the test set were evaluated. For each drug, the number of times (out of 20 runs) the prediction had Pearson correlation coefficient > 0.5 and p-value < 0.05 was counted and used for weighting models to integrate predictions.

Integrating predictions from all models: SVM with 80% bagging (100 iterations) was run on the complete training data across different parameters outlined in **Table K3**. The predictions appeared to have a different distribution compared to the training data, so the predictions were linearly transformed such

Nature Biotechnology: doi:10.1038/nbt.2877

that the mean and the standard deviation of the dose response predictions for a compound was equal to that of the training set. The predictions were then combined using a weighted average over all SVM configurations. The counts (out of 20) for each model evaluated using CV were used as weights for this linear combination.

Dealing with trivial exceptions: Some drugs had flat GI50 distributions (Drugs: 5, 12, 13, 24, 26, 27). For these drugs, the median drug response was assigned to all test cell lines and ranks for these drugs were arbitrary.

Discussion

We used an approach that involves learning across a combination of several dataset/normalization/SVM parameter choices, and then integrating the predictions based on their CV performance. We found that the predictions from this approach were substantially better than random predictions. Furthermore, we found that this approach was more robust to over-fitting compared to the alternative approach of choosing the combination that shows the best performance or uniform averaging. Another advantage of this approach is that each SVM combination is independent, so the approach can be easily distributed across processors.

One difficulty in applying this approach (as well as many alternatives) is the relatively small number of training examples, which was exacerbated by cross validating with sub-sampled sets based on the drug response training data. Presumably, an SVM would perform significantly better when presented with larger numbers of training and test cell lines.

Kernel method 4

Summary

Bidirectional search was used to select features, training and prediction was done using a support vector machine (SVM; radial basis).

Introduction

Due to the large number of features in the NCI-DREAM datasets, it is computationally infeasible to enumerate all possible feature subsets from the different datasets to determine the optimal features for prediction. Therefore, a heuristic bidirectional search was used to find a solution close to the optima (e.g., subset of features)[9] that predicts the effect of the drug compounds on the untested breast cancer cell lines. Bidirectional search combines sequential forward and backward selections to find a locally optimal set of features. The bidirectional search was applied on the genomics profiling datasets (excluding exome seq) to create an ensemble approach in which the results were combined by averaging the ranks. The motivation for using an ensemble model is to facilitate the handling of the diverse sets of data,[10] where each exhibits different

Nature Biotechnology: doi:10.1038/nbt.2877

characteristics and properties. Support vector machine was used to assess the quality of the subsets of features.[7]

Methods

Feature Selection: Bidirectional search is a parallel implementation of sequential forward (performed from an empty set) and backward selection (performed from a full set). This feature selection method was applied on five different datasets (RNA seq, DNA methylation, RPPA, copy number variation, and **gene expression**). For the algorithm to converge to the same solution, features already selected by sequential forward selection were not removed by the sequential backward selection. Similarly, features removed by sequential backward selection were not selected by the sequential forward selection. Missing values in the dataset were removed prior to applying the bidirectional search.

Algorithm – Sequential Forward Selection (SFS)

1- Start with an empty set of features $F_{SFS} = \varnothing$
2- Select the best feature
   - $best_{feature} = arg \quad max \quad Evalfxn(F_{SFS_i} + best_{feature}) \mid best_{feature} \notin F_{SFS}$
   - $F_{SFS_{i+1}} = F_{SFS_i} + best_{feature}$

Sequential forward selection (SFS), a greedy search algorithm, initially starts with an empty set of features F_{SFS} and sequentially selects the best feature that minimizes the mean squared error, the difference between the predicted and true compound dose response.

Algorithm – Sequential Backward Selection (SBS)

1- Start with a full set of features $F_{SBS} = \{All\ Features\}$
2- Select the worst feature
   - $worst_{feature} = arg \quad max \quad Evalfxn(F_{SBS_i} - worst_{feature}) \mid worst_{feature} \in F_{SBS}$
   - $F_{SBS_{i+1}} = F_{SBS_i} - worst_{feature}$

Sequential backward selection (SBS), a greedy search algorithm, initially starts with a full set of features F_{SBS} and sequentially removes the worst feature that maximizes the mean squared error.

Algorithm – Bidirectional search

1- Apply sequential forward selection starting with $F_{SFS} = \varnothing$
2- Apply sequential backward selection starting with $F_{SBS} = \{All\ Features\}$
3- Select the best feature

$$best_{feature} = arg \quad max \quad Evalfxn(F_{SFS_i} + best_{feature}) \mid best_{feature} \notin F_{SFS} \;\&\; best_{feature} \in F_{SBS}$$

$$F_{SFS_{i+1}} = F_{SFS_i} + best_{feature}$$

4- Select the worst feature

Nature Biotechnology: doi:10.1038/nbt.2877

$$= arg \quad max \quad Evalfxn\left(F_{SBS_i} - worst_{feature}\right) \mid worst_{feature} \in F_{SBS} \,\&\, worst_{feature} \notin F_{SFS}$$

$$F_{SBS_{i+1}} = F_{SBS_i} - worst_{feature}$$

Bidirectional search applies sequential forward and backward selections until the subset of features converges.

Prediction and ranking: A support vector machine was used to predict the sensitivity of different breast cancer cell lines to the drug compounds. SVM was trained on 35 breast cancer cell lines (*i.e.*, training set) and applied to predict the sensitivity of the remaining 18 cell lines (*i.e.*, test set). Mean squared error was used to assess the quality of the features in the training set. Since the underlying structure of the data is nonlinear, SVM maps the data to a higher dimensional space through the kernel function and then applies linear regression in this mapped space to predict the sensitivity; the kernel used was radial basis function: $e^{-\gamma * |u-v|^2}$ where $\gamma = \frac{1}{\#\ of\ features}$.

The bidirectional search and SVM predictions provided five ranked lists that corresponded to the sensitivity of the breast cancer cell lines to each drug compound. The average rank of the different breast cancer cell lines across the five ranked lists were averaged and combined into a final ranked list. We sorted the 53 cell lines from the most sensitive to the least sensitive with respect to each individual drug.

Discussion

Bidirectional search is a good heuristic approach for obtaining solutions close to the optimal. Missing values presents a disadvantage to the method proposed and degrades its general performance. Therefore in future, strategies to address missing values could increase the general performance of the bidirectional search algorithm.

Nonlinear regression 1

Summary

Features were randomly selected to built an ensemble of un-pruned regression trees for each dataset, missing values were imputed, weights for the models were calculated, final predictions were made using a weighted sum of the individual models.

Introduction

The underlying methodology used is a Random Forest (RF) regression model.[11] Since RF generally provides good predictive performance for high number of variables, it is well suited for processing large-scale genomic data with more features than samples, as is the case with NCI-DREAM datasets. We also used a linear regression based approach to combine RF predictions from different

Nature Biotechnology: doi:10.1038/nbt.2877

datasets (e.g., gene expression, methylation, *etc*.). Missing data were imputed using an average of highest and lowest expressions for that feature.

Methods

Database combination: For each training database provided, we integrated them with the drug response training data to generate a merged database, which was used for training and prediction. The cell lines in the new database are categorized by three categories: (C1) cell lines that have both genomic characterization and drug sensitivity information, (R1) cell lines that have only genomic characterization, and (R2) cell lines that have only drug sensitivity information. The number of cell lines in category C1 is often less than 35 due to the presence of "NA" values in drug sensitivity data or missing genomic characterizations.

Feature selection: For each combined database, we use the C1 category to select features by random forests. First, we select n = 500 bootstrap samples from the original data. Then, we build an ensemble of un-pruned regression trees.[11, 12] Each of these trees is built on bootstrap samples. A random subset of all the features is used for splitting the tree nodes. For each node of the tree, we randomly select m features to base the decision at that node. The m was selected to be $\log_2 M+1$, where M is the number of input features.

Regression: After selecting the features, we apply random forest regression algorithm (a nonlinear multiple regression approach) to generate the prediction model. Random forests are built by growing trees depending on a random vector Θ, such that the tree predictor, $h(x,\Theta)$, takes on numerical values as opposed to class labels. $h(X)$ is the average tree response of k trees corresponding to a response variable Y. The mean-squared generalized error for any numerical predictor, $h(X) = E_{X,Y}(Y - h(X))^2$. The requirement for accurate regression forest is to lower correlation between residuals and low error trees.[12]

Prediction and ranking: Let f be the prediction model obtained by the above regression algorithm for each combined training database. To sufficiently utilize the information in different databases, we produce a weight-based integrated model for prediction. We use least square regression to estimate the weights for each f by minimizing $\sum_j (a_j - \sum_i W_i f_i)^2$, where a_j is the actual drug response and W_i is the corresponding weight of f_i. Additionally, for validation, we use leave-one-out cross-validation (CV) to estimate prediction errors of not only the weight-based integrated model but also the individual models that uses only its own training information. In other words, the N training genomic characterizations can generate 2^N-1 prediction errors and we select the model with the minimum prediction error as the final prediction model. Cell lines were ranked on their predicted GI50 values per drug.

Discussion

Nature Biotechnology: doi:10.1038/nbt.2877

Based on the leave-one-out CV, it appears that the gene expression and methylation datasets are more informative than the remaining datasets. The final model obtained for each drug usually ranged from 0 to 0.2, which represents a high accuracy prediction by RF. Since leave-one-out CV estimations for small samples can have a huge variance, for future work, we will test the robustness of the applied approach on other drug sensitivity datasets.

Nonlinear regression 2

Summary

Features were filtered based on their correlation to dose response, random forests were trained for each dataset, missing values were imputed, final rankings were based on a ensemble score from 4 individual dataset models.

Introduction

Our method represents a two-step modeling approach. First, each of the six classes of genomics data and the known drug response profiles were used to build dataset specific models for predicting cell line drug response. Second, the predicted ranks from each model were summarized in the form of rank-product to produce the final score. Based on our previous studies, the information provided from each class of genomics data are complementary and important. If we pool all genomic data to build a single model, datasets that contain fewer variables will be overshadowed by dataset that contain many variables, thus, our strategy is to build models separately, and then combine the predicted ranks. We used random forests to build models from each genomics dataset and used penalized regression to build a model from the known drug response profiling data.

Methods

In this model, we used all the provided genomics data, namely, gene expression, methylation, RNA seqs, RPPA, CNV, exome seq, and the drug response matrix. Additionally, we used an expanded set of drug response data provided in the supplementary file of Heiser, *et al*.[13]

Nature Biotechnology: doi:10.1038/nbt.2877

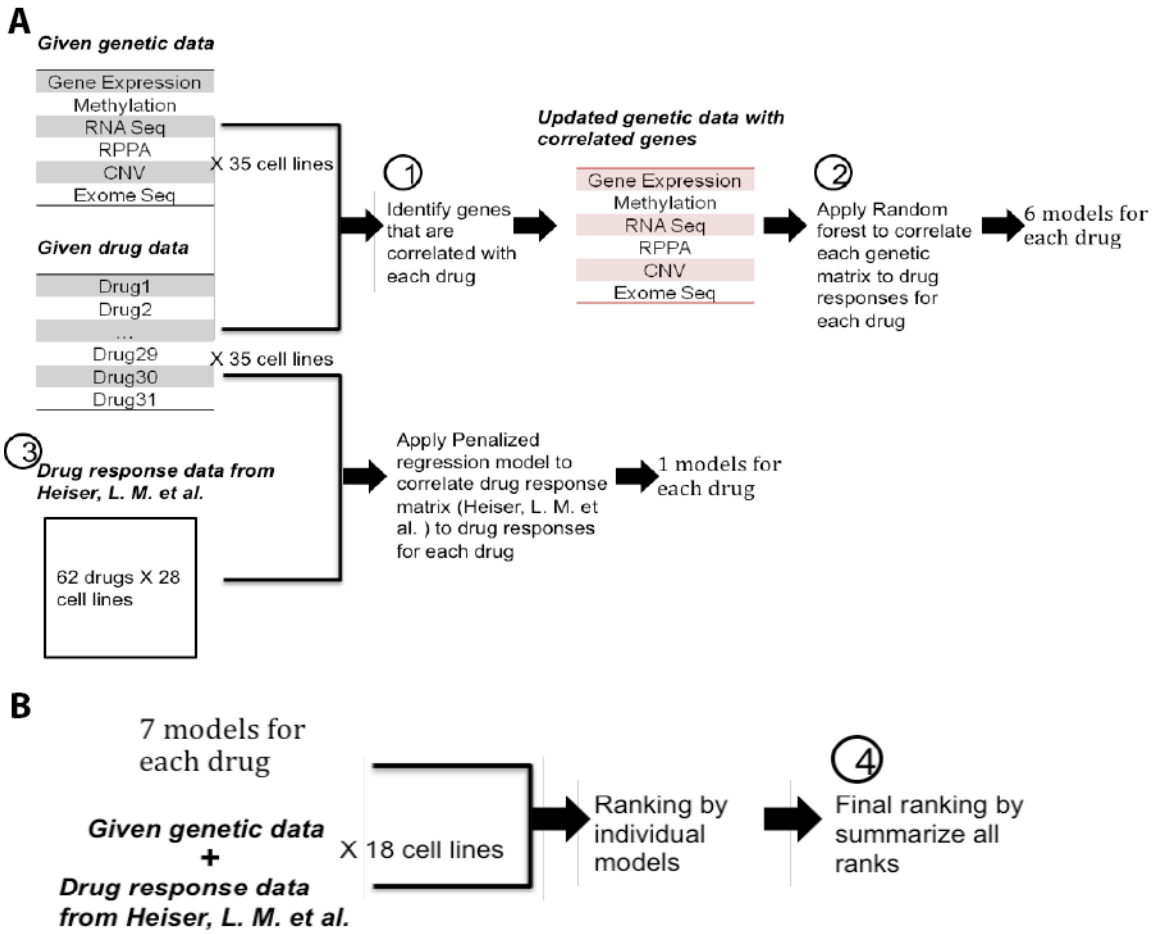


Figure N2. Schematic representation of the Nonlinear regression 2 method

The following steps describe our method, where the numbers listed below correspond to the numbers in **Figure N2**.

1. **Gene selection**: For each genomic dataset and drug pair, we calculated the correlation between each feature and the cell line drug responses and assigned *p*-values. The false discovery rate (FDR) was calculated from a beta-uniform mixture model.[14] All features that passed the FDR criteria (FDR<10%) were included in the model construction. If there are fewer than 100 features being identified at 10% FDR, we used the top 100 probes ranked according to *p*-values.

2. **Development of models based on genetic data**: For each genomic dataset and drug, a random forest regression model (randomForest R package, default parameters were used) was built to establish correlations between genomic data and drug responses. Cell lines with no drug data were deleted before modeling.

3. **Development of the models based on drug response profiles:** There is drug response data for 29 out of the 35 training set cell lines reported in

Nature Biotechnology: doi:10.1038/nbt.2877

Heiser, *et al*.[13] One cell line had too few measured GI50s, and was removed from the model construction. Similarly, drugs with few measurements on cell lines were also removed. The final drug response profiling matrix included 28 cell lines and 62 drugs. The *k*-nearest neighbors (KNN) approach was used to impute missing values in the profiling matrix, and a penalized regression model was built for each of the 62 drugs.

4. **Consensus ranking from individual predictions:** We used the product of the rank from each model to determine a composite score. The results from Exome Seq and Methylation data was not included in the composite score due to their poor performance in leave-one-out cross-validation in the training set. Finally, we rank cell lines based on the composite score.

Discussion

Based on our results from the leave-one-out cross-validation analysis in the training set, we found that the model developed from known drug response profiling data performs the best, while the model using RNAseq data showed the best performance among the 6 genomic models. Our result suggests that the information derived from the known drug response profiling data is important for accurately predicting cell lines' responses to new drugs. It could be due to the fact that all the genetic information provided was obtained at the baseline state (before treated with any compound). Only the model built from known drug responses provides direct information on cell lines' behavior under perturbation. Although the information is limited, as there are only data for 74 drugs from Heiser, *et al.*,[13] they are still very helpful.

Based on our results, we think it would be interesting to incorporate two types of information, and see whether they will be helpful to improve our current model. The first is to include pathway information in modeling. Knowledge from biological pathways can be useful for gene selection and gene clustering. The second is to utilize the provided cell lines that are resistant to compound treatments. Our current model only used this information during feature (gene) selection before any modeling. Another piece of information that cannot be applied in this challenge, but should be otherwise useful in real world screening, is the chemical features of the compounds. As similar compounds tend to trigger similar biological responses, a model built from a compound's chemical features will also provide predictive information on cell line responses.

Nonlinear regression 3

Summary

Features were filtered based on their correlation to dose response, random forests were trained for each dataset, missing values were imputed, final rankings were based on a composite score from 5 individual dataset models.

Methods

Nature Biotechnology: doi:10.1038/nbt.2877

This method is a modification of the approach presented in Nonlinear regression 2. In the Nonlinear regression 2 method, the final cell line ranking was based on the composite score from 4 genomics datasets, include gene expression, RNA seqs, RPPA, and CNV. The predictions made from this model are based on a final composite score including 5 genomics datasets, including gene expression, RNA seqs, RPPA, CNV, and methylation.

Nonlinear regression 4

Summary
Features were filtered based on their correlation to dose response, random forests were trained for each dataset, missing values were imputed, final rankings were based on a composite score from 5 individual dataset models.

Methods
This method is a modification of the approach presented in Nonlinear regression 2. In the Nonlinear regression 2 method, the final cell line ranking was based on the composite score from 4 genomics datasets, include gene expression, RNA seqs, RPPA, and CNV. The predictions made from this model are based on a final composite score including 5 genomics datasets, including gene expression, RNA seqs, RPPA, CNV, and exome seq.

Nonlinear regression 5

Summary
Features were filtered based on their correlation to dose response, random forests were trained for each dataset, missing values were imputed, final rankings were based on a composite score from individual dataset models.

Methods
This method is a modification of the approach presented in Nonlinear regression 2. In the Nonlinear regression 2 method, the final cell line ranking was based on the composite score from 4 genomics datasets, include gene expression, RNA seqs, RPPA, and CNV. The predictions made from this model are based on a final composite score including all 6 genomics datasets, including gene expression, RNA seqs, RPPA, CNV, exome seq, and methylation.

Nonlinear regression 6

Summary
Gene features were selected using linear regression and maximal information coefficient, pathway information was also used to derive features, training and prediction was done using a random forest model

Introduction

Nature Biotechnology: doi:10.1038/nbt.2877

Our approach employs a basic machine-learning algorithm with features that show high correlation with drug sensitivity and features aggregated at the pathway level. Because no information was provided on the drugs used in the challenge, we focused on the similarities and differences among the cell lines to predict the drug sensitivity (GI50) of each cell line. Our fundamental assumption was that certain features such as expression level or mutation state of genes would directly affect drug sensitivity,[15, 16] while other features would affect drug sensitivity by changing the functional activity of a cell.[13, 17] Therefore, we selected and used two kinds of features—features highly correlated with drug sensitivity and pathway-level features. We used a random forest model based on selected highly correlated features and pathway-level features to predict the sensitivity of a cell line to each drug. We ranked each cell line according to the drug sensitivity values predicted by trained random forest models.

Methods

The overall machine learning process is illustrated in **Figure N6**. Our method can be divided into 3 parts: feature generation, model training, and prediction. Because the number of available features for each cell line is large relative to the number of cell lines, overfitting is a problem. We used feature filtering and feature elimination methods to address this problem. The random forest model was used for training prediction models and predicting drug sensitivity.

To filter the features of each cell line, we used linear regression and maximal information coefficient (MIC).[18] We used linear regression to select the features linearly correlated with the sensitivity for each drug (FDR < 0.05); however, with this approach, only linearly correlated features can be selected. To identify features that have nonlinear correlation to drug sensitivity, we applied the MIC^5 method on each dataset for each drug and selected features that had strong association with drug sensitivity (correlation coefficient > 0.8). After this feature-filtering step, features linearly or nonlinearly correlated with drug sensitivity for each drug were compiled for each cell line.

Nature Biotechnology: doi:10.1038/nbt.2877

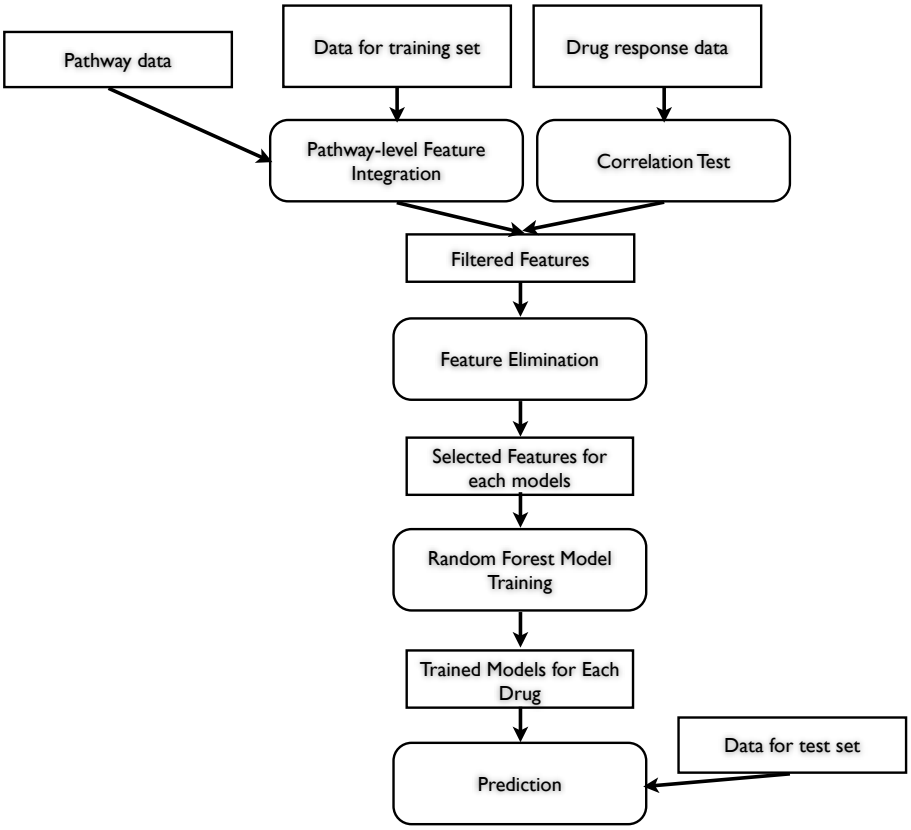


Figure N6. Schematic process diagram of the Nonlinear regression 6 method

Because we assume that the functional activity of a cell contributes to its drug sensitivity, we calculated the functional features using well-annotated pathway databases. Gene expression data and exome sequencing data were mapped to each pathway. We used the PathOlogist[17] tool to calculate each cell line's pathway-level activity and consistency using gene expression data. Pathway information from KEGG, NCI-PID, and Biocarta databases (total pathways: 621) were used as input to PathOlogist. Pathway level mutation status was also considered. The number of mutated genes in a pathway was calculated for each pathway using MsigDB: c2-cp pathway (total pathways: 880), and the numbers of mutated genes in each pathway were used as features along with the pathway activity and pathway consistency features calculated with PathOlogist.
We trained 1 to 3 models per drug owing to the incomplete data for each test cell line. We divided test cell lines into 3 groups as follows: cell lines with all data; cell lines with mutation, expression, copy number data; and cell lines with mutation data only. Random forest models were trained for each group of training cell lines if the number of available cell lines in each group was more than 13. We randomly sampled 3 cell lines from each group of cell lines to estimate the performance of each model. These sampled cell lines were excluded from the rest of the training procedures.

We performed the feature elimination procedure to reduce the total number of features for each model, which range from 100 to 5,000. We used the backwards

Nature Biotechnology: doi:10.1038/nbt.2877

feature selection method along with bootstrapping. The feature selection procedure was applied 25 times to all 3 models for each drug using the random forest learning algorithm.

We trained each model with selected features using a random forest model. We trained models with different settings for the number of variables randomly sampled as candidates at each split and selected the model within 1 standard error deviation to avoid overfitting. The performance of each model was estimated by calculating the average root means square error of predicted GI50 values of the 3 cell lines sampled before the training. For each cell line, we selected the model with the best performance for prediction among the models applicable to the cell line (according to the available data sets for the cell line). Final models for each drug were retrained using all available data sets and used for predicting the GI50 values of test cell lines for each drug. Furthermore, the final rank of each cell line for each drug was decided by the order of the GI50 value of each cell line.

Discussion
We used linearly or nonlinearly correlated features and composite features calculated using pathway databases. With these features, we trained random forest models using a feature elimination method. Because our method is based on a machine-learning algorithm, performance is dependent on the availability of data and drug characteristics.

Nonlinear regression 7

Summary
Random forests were constructed in a stacked approach, an ensemble of regression trees were constructed for all drug/dataset pairs, missing values were imputed, predictions were made for individual models and another random forest was used to combine the different predictions for the drugs to a final prediction.

Introduction
In bioinformatics, data obtained from microarrays, sequencing, images and other complex data types are often noisy, incomplete, high-dimensional and only slightly correlated with the biological process in question. For these reasons, there is an increasing trend to combine many different data sources in order to solve complex problems such as the inference of gene regulatory or protein-protein networks.[19] The NCI-DREAM drug sensitivity prediction challenge fits well in such a setting.

This challenge can be considered as a relational learning problem in which the interaction between two objects is to be predicted. In bioinformatics, machine learning techniques for approaching this kind of problem is well known, for example in chemogenomics.[20] This NCI-DREAM challenge setting deals with relating drug treatment to cell line response, with the GI50 concentraten as the

Nature Biotechnology: doi:10.1038/nbt.2877

relation to be learned. Typically, statistical models are based on a smoothness assumption for the different objects (*i.e.*, drugs and cell lines), namely, similar drugs will have a similar influence on the same cell line, and similar cell lines will react approximately the same to a given drug.

Methods based on a pairwise kernel representation incorporate features of both objects in an elegant way in order to build a predictive model.[21] One could use conditional ranking for this problem,[22] as for a given drug the challenge is to rank the cell lines according to their degree of drug inhibition. Unfortunately, since the identities of the different drugs were unknown, no feature representation could be constructed for these objects. The only available features were concerning the cell lines. As such, we have used a stacked approach to combine different sources of information. We take three layers in account: 1) Using features in one dataset to predict the GI50 for one drug, 2) combining the different features across datasets with a model to obtain a better estimate of the GI50 for each drug/cell line combination, and 3) predicting a final value for a particular drug using a model that leverages the estimates for all the different drugs as inputs.

In the second layer, the different features are combined, while the third layer attempts to process the dependencies between the drugs. Such a stacked approach requires building a considerable number of models, for which tuning can be computationally intensive. For these reasons, random forests, which are fast to train and do not require much tuning of the hyperparameters, were used. Furthermore, random forests are popular in bioinformatics for their ability to cope with high-dimensional feature vectors, an issue in this challenge. A final issue is that not all features are available for all of the cell lines. This results in missing values in the second layer. Although there are methods that can deal with missing values, we opted for using a matrix completion algorithm to infer the missing values as a preprocessing-step.

Methods

Let us denote a compound/drug as d_i. We were supplied with six genomics profiling datasets:

- f_1: DNA copy number variation
- f_2: Transcript expression values
- f_3: Whole-exome sequencing
- f_4: RNA sequencing data
- f_5: DNA methylation data
- f_6: RPPA protein quantification

For f_4 we have the $\log_2$ transformed estimates of gene-level expression (f_{4a}) and expression status values, indicating whether the genes were detected above background level (f_{4b}). Thus, we constructed a feature matrix with seven different datasets. Finally, let c_k denote the kth of the 53 cell lines.

Only the f_1 dataset contained some missing values. We imputed missing values using random forests through the MissForest package (with standard settings).[23]

Nature Biotechnology: doi:10.1038/nbt.2877

In the first step of the stacked model, a random forest was trained for each combination of drug and genomic dataset. Each of the 217 (7 datasets x 31 drugs) random forest models were trained using 15,000 trees. Every split of an individual tree was based on five randomly selected variables. This low number was needed to guarantee that each of the features was considered at least once, as some datasets had more than 50,000 features. The cell lines with known GI50 values for a particular drug were used to train the model. As such, predictions p_{ijk} were made for drug *i*, features *j,* and cell line *k*. These predictions where stored in a 1,643 x 7 matrix with each combination of a drug and cell line for rows and the features for columns. Since not all features were available for each cell line, this matrix contained missing values. These were inferred using the MissForest package with 5,000 trees.

The predictions were averaged in a supervised manner in the second layer. For each d_i, a random forest with 1,000 trees was used to predict the GI50 for a given cell line based on the seven previously obtained predictions. Again, the GI50 values that were given for cell lines and d_i served as a training set. These 31 models resulted in a complete response matrix for the 53 cell lines and 31 compounds.

In the third and final layer, information from the different drugs was combined. Similar to the previous layer, a random forest model was trained for each drug to predict the GI50 for a given cell line. This time the inputs for these models were the 31 GI50 values for the cell line obtained from the second layer. The random forest used standard settings for the number of trees and variables selected for regression of a dataset of this size. Finally, the predicted GI50 values were rank ordered for the final submission.

Discussion

We tackled the NCI-DREAM drug sensitivity prediction challenge using a stacked approach. The stacking allowed for exploiting the dependencies within different datasets without having to rely on overly complex techniques. By using a powerful matrix completion algorithm, we were not hindered by the partial nature of the data. Random forests are a popular method for these types of problems. One could probably do somewhat better by using techniques such as support vector machines with specialized kernels. In contrast to random forests though, these require intensive tuning and our method was fast and scalable.

Nonlinear regression 8

Summary

Features were ranked according to the absolute value of Spearman's correlation, the average rank of all cell lines was calculated according to the top features.

Introduction

Nature Biotechnology: doi:10.1038/nbt.2877

The challenge was to build a model capable of ranking the sensitivity of the remaining 18 cell lines to the 31 compounds (the *test set*): for each of the compounds, challenge participants were asked to predict the rank order of the 18 cell lines in the test set from the most sensitive to the least sensitive, in relation to the 35 cell lines in the training set. Considering the cell lines as cases to be ranked, the different genomic measurements can be treated as *features* of the cell lines. We focused our analysis on the features showing the highest absolute Spearman correlation with the response, across the 35 measured cell lines of the training set, to each of the 31 compounds.

Methods

The analysis pipeline we exploited for the current challenge is summarized in Figure N8; each step of the pipeline is presented below.

Data merging and filtering: For exome seq data, we merged the exome information at a gene level, by counting the number of mutations in each gene for each cell line. For DNA copy number variation data, we directly exploited the supplied, pre-processed dataset with gene-level changes in copy number.

We merged the different genomic datasets in one large matrix (Figure N8a), with the genomic features on the rows and the cell lines on the columns, both training and test sets. Since not all types of genomics profiling data were collected for every cell line**,** unmeasured cell lines in each dataset were labelled as *missing values*. Features across all datasets were filtered according to two criteria: 1) filter out features with 13 or more missing values (being 12 the maximum number of unmeasured cell lines across the different datasets), and 2) filter out features with less than 5 non-zero values across the cell lines (to retain the most informative signals and increase the robustness of the predictions).

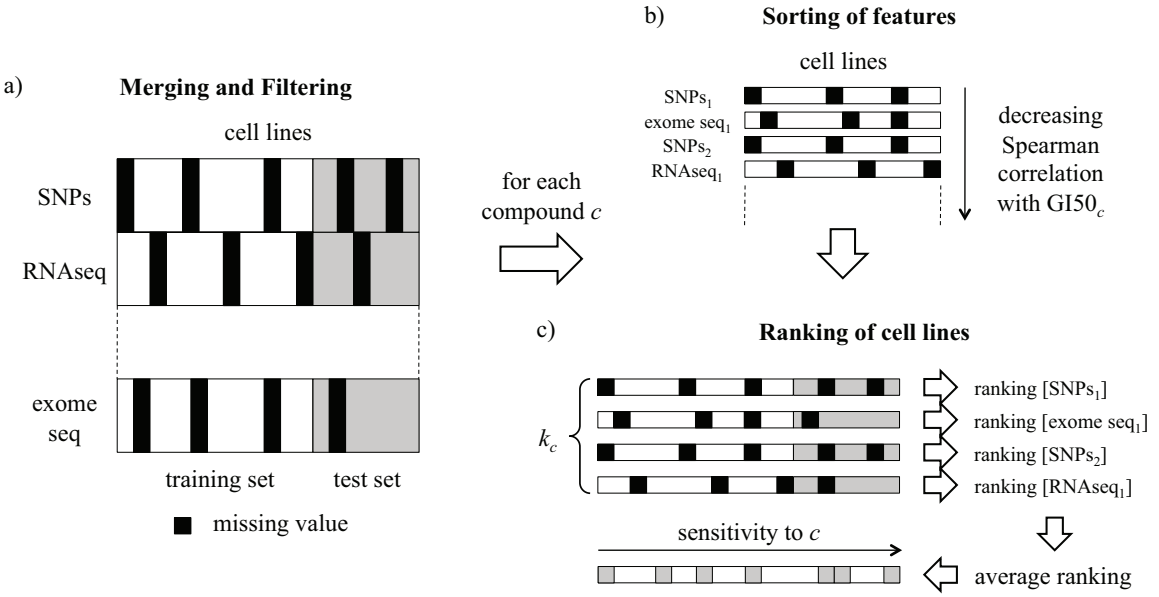


Figure N8. Schematic representation of the Nonlinear regression 8 method

Nature Biotechnology: doi:10.1038/nbt.2877

The next steps of the analysis were carried out separately for each compound *c*.

Sorting features: Considering only the cell lines in the training set (*i.e.,* with measured GI50 concentrations), we sorted all the features in decreasing order according to the absolute *Spearman correlation* with GI50 values (Figure N8b). Spearman correlation is based on rank and is not altered by shifting or scaling: no further normalization or transformation was thus needed.

Ranking cell lines: Cell lines from both the training and test set were then ranked according to the top k_c features, with the optimal number, k_c, for each compound *c* tuned as described below. Cell lines were separately ranked according to each of the k_c variables, reversing the ranks for features with negative correlation, then the ranks were averaged (Figure N8c): this way, we obtained a robust ordering of the 53 cell lines for each of the 31 compounds.[24] In case the relative order of the cell lines from the train set were different from the one measured through GI50 concentrations, we reordered those cell lines, keeping fixed the position of the cell lines from the test set.

Tuning of k_c using cross-validation: Since the optimal number of top correlated features, k_c, can vary across compounds, we designed a *cross-validation* strategy for tuning k_c and separately applied it to each compound. The tuning procedure considers only the 35 cell lines in the training set from which it samples 23 cell lines (~2/3) at random as an internal training set tr_i and leaves the remaining 12 as an internal test set ts_i. The sampling procedure is repeated 20 times, thus obtaining 20 pairs of internal training and test set (tr_i, ts_i) for each compound. Spearman correlation is then computed between each feature and the response of the cell lines *from the internal training set* tr_i; features are ranked in decreasing order of correlation. Starting from k_c = 1 and increasing it up to 100, the procedure then reorders the cell lines *from both* tr_i *and* ts_i according to their average rank across the top k_c correlated variables and records the Spearman correlation between the ordered cell lines *from the internal test set* ts_i and their experimentally determined GI50 values. Iterating the procedure 20 times, we obtain a distribution of the expected Spearman correlation of independent test sets. The optimal k_c for each compound is selected as the one maximizing the median correlation across the 20 internal test sets.

Discussion

We developed a procedure for learning how to rank multiple cell lines given several genomic profiling datasets and dose response measurements. Since the objective of the learning task was to rank cell lines, rather than to predict a numerical value for each cell line, our learning strategy was entirely based on relative rank; discriminative genomic features were selected based on their rank correlation with the drug response and cell lines were ordered by averaging their ranks according to the selected features.

Nature Biotechnology: doi:10.1038/nbt.2877

The optimal number of genomic variables to exploit for each compound was automatically identified with cross-validation. The number has been found to strongly depend on the specific compound, varying between 2 and 61. Not all types of data were selected by our procedure with the same frequency: RNA seq variables were selected for each of the 31 compounds, whereas proteins and copy number variations were never selected and SNPs were selected only on 2 cases. For the other datasets, some frequent patterns could be observed: gene expression and methylation variables were often selected together and as an alternative to exome seq data.

Nonlinear regression 10

Summary

Features were selected using a matrix approximation methods leveraging SVD, training and prediction were done using a regression tree models using gradient boosting.

Introduction

In measuring the comprehensive impact of a drug or small molecule on cells, numerous studies have identified that gene expression data can serve as an effective signature, with virtues of carrying sufficient variation and covering the whole genome. Recently, gene expression profiles measured from drug-treated cell lines have been successfully used to reposition established drugs through searching complement profiles between drug- and disease-generated gene expression data.[25, 26] Expression profiles have also been utilized directly to model drug sensitivity.[27] Based on these observations, we postulated that expression profiles could be useful in modeling drug response as well. In this analysis, we only used microarray profiles and RNA sequencing data as they cover all training and testing cell lines when combined together. We first performed feature selection by retaining the most important genes to make the dimensionality of feature space comparable to the size of training data, and then fit a nonlinear regression model by gradient boosting machine (GBM), which is found resistant to over-fitting in general and has shown promising applications in genome-wide association studies.[28] In terms of significance measurements of genes, we adapted the concept of normalized statistical leverage scores from a recent matrix approximation algorithm,[29] in which we keep features from original data matrix, rather than certain linear combinations of it as output from popular principal component analysis (PCA). The data generated by this method thus maintains the interpretability associated with each feature. This method has been used to pick up the most important genes and demonstrated excellent performance in a phenotype classification based on gene expression data.[29] In GBM, we explicitly specified interactions among features in each model, as the addition of feature-feature interactions could improve the accuracy of drug sensitivity prediction.[27]

Methods

Nature Biotechnology: doi:10.1038/nbt.2877

We built one regression model for each drug. To cover all test cell lines, we trained the models on mRNA variation data from 32 training cell lines and then made predictions on 14 test cell lines. For the remaining 4 test cell lines, specifically 21NT, 184A1, MX1, and 21MT1, we used to RNA sequencing data, from which 29 cell lines were included in the training. For each model, we form a data matrix *A* from expression profiles, where each row represents a cell line and each column represents a gene. Firstly we applied singular value decomposition (SVD) on *A*, $A = UDV^T$,where *U* and V^T are left and right matrix consisting of singular vectors, respectively, and *D* is a diagonal matrix. The normalized statistical leverage score S_i for gene *i* can be computed, $S_i = V_{i1}^2 + \ldots + V_{im}^2$, where V_{ij} is the element (i,j) in *V*, and *m* is the number of rows of *A*.[29] We retained the top *m* genes with the largest S_i to form a new data matrix *A'*, which contains many fewer columns than original one. Next, we fit a GBM regression against *A'* and drug response values by using the R package gbm. In the model parameter configuration, we specified interaction depth up to 3, which allows the model to include 3-way interactions among features. The underlying distribution was assumed to be Gaussian, and up to 3,000 trees was grown during the training. For a drug, the rank of each test cell line was determined by simply comparing its predicted GI50 value to other cell lines.

Discussion

From a machine learning perspective, one difficulty imposed by the data supplied in the NCI-DREAM challenge is that different classes of genomics data vary significantly in cell line coverage, and no single data source covers all training and test cell lines. In this analysis, to overcome the problem of missing data we chose to use gene expression profiles only, and this consideration is in part due to the fact that when combined together, microarray data and RNA sequencing data allowed us to train models using as many cell lines as possible. An evident pitfall of this kind of data selection, compared to a full use of all available sources, is that we may lose valuable information by not using data beyond gene expression. So it seems one possible improvement on the current approach could be achieved by integrating all sources together, either by imputing missing data or modifying present models to accommodate sources with partial coverage.

Nonlinear regression 11

Summary

Features were selected for individual cell lines by constructing random forests and pruning (recursive feature elimination), missing values were imputed, final predictions were made by training a random forest using features from all cell lines. In addition to cell line features, bioactivity spectra of the individual compounds were included as compound features.

Introduction

Coming from the field of preclinical drug discovery we immediately appreciated the problem as a bioactivity spectrum prediction. Previously we were successful

Nature Biotechnology: doi:10.1038/nbt.2877

in predicting these bioactivity spectra with regard to viral resistance. [30, 31] Our method (proteochemometrics, PCM) quantifies the similarity between targets (here cell lines) and drugs (here unknown drugs) using Random Forest with Recursive Feature Elimination (RFE). Subsequently PCM extrapolates from known activity values for combinations of the two in the training set to unknown activity values for combinations not present in the training set. Hence, our assumption was that the methods we previously used would work similar in the case of the NCI-DREAM challenge. However, the absence of a drug features (to provide a drug similarity metric) was the main problem to be tackled.

The main novelty for the application in the NCI-DREAM challenge was that we imputed drug features. Drug features were obtained from bioactivity spectra, the majority of the variation was contained in the drug similarity (on average a drug performed roughly similar on the cell lines, yet major differences between drugs existed). For each drug, the bioactivity values that were present were selected. The mean, skew, kurtosis, standard deviation, variance and percentile values were calculated. On these values multidimensional scaling was performed (raw, PCA, CMDS or distance matrices) resulting in the similarity being described in reduced dimensions. This produced an activity-based drug feature.

Methods
Calculations were performed in Pipeline Pilot 8.5.[32] For a schematic overview see **Figure N11**.

Feature Preprocessing: Features for the six genomic profiling datasets were preprocessed as follows:

- **RNA seq**: The values in the supplied file were used. When multiple measurements of a gene were present in a single cell line, the median of the measurements was used.
- **Gene expression**: The values in the supplied file were used. When multiple measurements of a gene were present in a single cell line, the median of the measurements was used.
- **RPPA**: Only the fully validated measurements from the RPPA file were used without further processing.
- **Methylation**: Methylation data was filtered to contain only measurements where Cct1 > 3 and CGct1 > 3. If multiple values were present, the median of the measurements was used.
- **Exome seq**: Only measurements with a confidence > 150 were used. For each measurement the following information was kept and used as unique identifier: Chromosome, Type, Summary, CancerGene and gene. Presence or absence of these identifiers was encoded for each cell line, when present, 1 was used, when absent, 0 was used. Finally, we calculated skew, kurtosis, variance and mean value of the bitstring (treating each measured mutation as a bit, sorted by chromosome, type and gene). Additionally, several more general parameters were calculated for each cell line using this feature set. These were: number of mutations,

Nature Biotechnology: doi:10.1038/nbt.2877

number of mutations in cancer genes, ratio between mutations in cancer genes and total mutations.

- **SNP**: The values in the supplied file were used; if duplicate measurements were present the median was used.
- **Cancerous or not flag**: A single flag was added indicating if the cell line was cancerous or not (based on literature research).

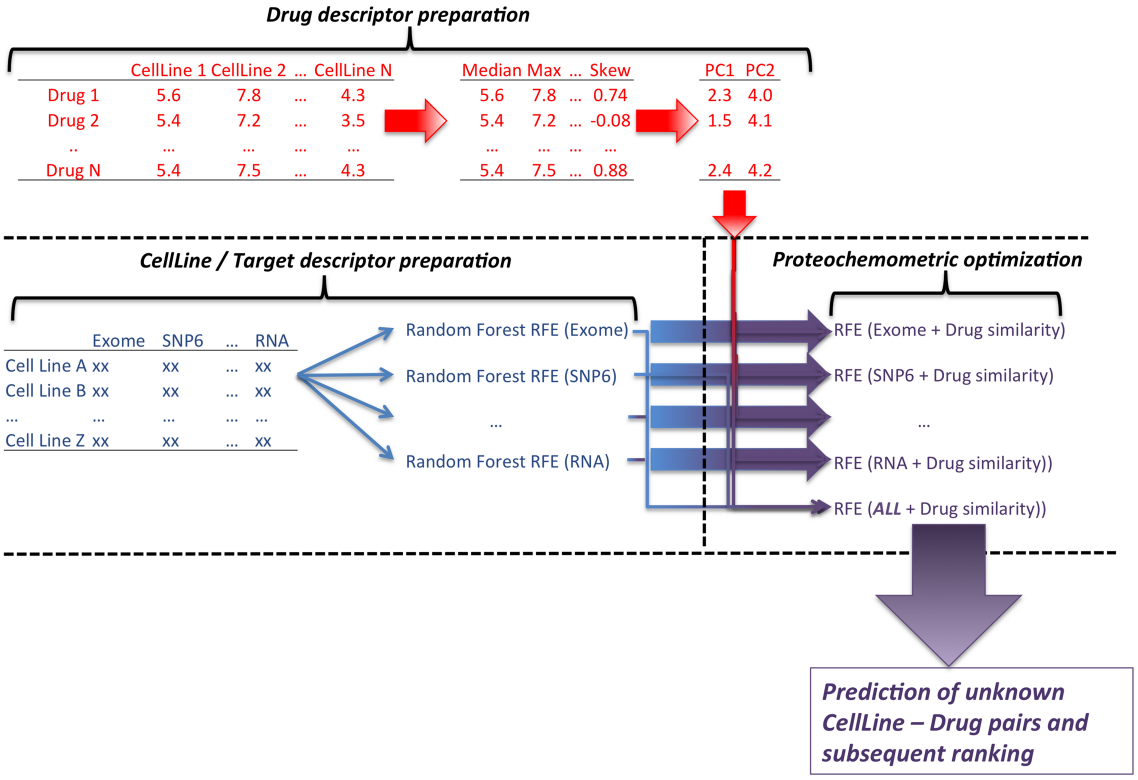


Figure N11: Schematic representation of the Nonlinear regression 11 method.

Feature selection using Recursive Feature Elimination (RFE): Relevant features were selected using Random Forest modeling and RFE. Features were kept that had a positive permutation importance using the permutation accuracy (for feature *n*, the average increase in error of the predictions when the values for features *n* were permutated). This procedure was run until the values stabilized. To ensure that the random forests converged, and to prevent a local minimum, the order of the data was randomized each iteration and the order of the features was randomized every 2 iterations. Within each loop, two external validation models were trained on different random subsets of the data using the same feature selection to get a realistic performance estimate using the subset. RFE was stopped when either of the following condition were met: 1) the difference between R_0^2 and $R_0^{2'}$ increased and the RMSE increased (indicating significantly reduced model quality), or 2) the number of iterations exceeded 10. The reduced feature set obtained from RFE was used as input for our PCM method.

Nature Biotechnology: doi:10.1038/nbt.2877

Imputating missing features: For all cell lines, the properties selected using the Random Forest RFE were imputed using the impute R package and *k*-nearest neighbors. Missing value imputation was done *after* RFE.

Training Random Forest models: Several (RF) models were trained, including:

1. Models on each individual dataset (6)
2. Models on combinations of two or three datasets (3)
3. Model trained on only cell lines for which each feature was available (1)
4. Model trained on all features (1, imputed by the RF algorithm)
5. Model trained on all features (1, imputed by 'Impute')

Model performance was estimated using a two-fold cross-validation. Given the 35 cell line drug response training set, a random 50% of cell lines were left out (fraction 2) and a model was trained on the remaining 50% (fraction 1). The cell lines were left out using stratified sampling. Performance measurements (R_0^2, $R_0^{2'}$, and RMSE) were calculated after grouping the predictions per drug. A second model was trained on fraction 2 and validated on fraction 1. The RMSE, R_0^2, and $R_0^{2'}$ of the 2 models were averaged to obtain a performance estimate.

The model trained on all data (model type 5) performed the best according to the CV evaluation (and would hence be recommended to predict the activity of all drugs per cell line). Still significant differences occurred when predicting activity of individual drugs using different feature sets. For the prediction of the unknown cell lines, all features were imputed for each cell line using the impute R package. Based on the predicted activity values and measured values the cell lines were ordered per drug and subsequently ranked (**Figure N11**).

Discussion

The most interesting observation from our approach was that the bioactivity spectra, which constitute an extremely simplified drug feature, were informative and allowed the training of predictive models. This is likely caused by the much larger inter-drug differences in GI50 than intra-drug differences. We observed our models to be highly predictive with regard to unknown combinations of drugs and cell lines; however, they were less capable of ranking the cell lines (which we did on the basis of the predicted GI50 values). This is likely caused by very small and zero differences between individual cell lines in the training data and the prediction error for the GI50 of 0.5 units. Yet, from a clinical perspective, the accurate prediction of drug activity would be more interesting than the ranking of cell lines.

We also observed that restricted datasets (not using all data) were sometimes more predictive for individual drugs than was the large combined set. This observation shows that our method might be an interesting approach to extract predictive information from a reduced set of data. Future research applications of this work should include a dedicated drug feature (e.g., cheminformatics type), should select the best training set per drug, and should also include an extensive

Nature Biotechnology: doi:10.1038/nbt.2877

randomization validation, which we were unable to complete within the time frame.

Sparse linear regression 1

Summary

Features were simultaneously selected and a ranking model built for each drug by lasso regression

Introduction

A genomics-based approach to prediction of drug response was originally considered by Staunton, *et al*.[33] with some success. This approach accurately predicted the cell line chemosensitivity patterns of 88 drugs out of 232 considered. Unlike the current competition for which the objective is to rank the sensitivity of cell lines against each compound considered, this earlier study was concerned with the binary decision of whether a cell line was sensitive to a drug or not, irrespective of the relative order of sensitivity across all cell lines. In addition to the gene-expression profiles used by Staunton, *et al*.,[33] in accordance with more recent evidence presented,[13, 15, 16] the current competition also considers DNA mutation and methylation data, RNA and exome sequencing data, and RPPA protein quantification data.

As the challenge is to build a model capable of ranking the sensitivity of cell lines to 31 untested compounds, we followed an approach that can utilize raw GI50 concentration values for each drug without binarizing growth inhibition data. Additionally, the sample size was quite small and there were literally thousands of different features with which the model could be built, and which offer excessive flexibility during model learning that may significantly increase the risk of over-fitting the data. We do not believe that a model with a high *capacity,* such as models that use kernel machines or basis functions, would necessarily perform well in this task. Thus, we follow an "Occam's Razor" approach to model learning and adopt a strategy that uses a linear model, which is able to aggressively eliminate features during the learning process. The study by Bi, *et al*.[34] demonstrates that when a 1-norm loss function replaces the standard 2-norm loss function in support vector machines (SVMs), it can serve as a built-in feature-selection mechanism embedded into model learning. This type of model has been successfully applied in machine learning to a variety of problems. Our approach extends this idea to learning ranking functions from continuous-valued drug response data in an attempt to rank chemosensitivity of unknown cell lines relative to that of known cellular subtypes.

Methods

For each compound, we train a ranking function of the form $f(x) = w^T x$ where x is the feature vector characterizing each cell line. If a cell line characterized by x_1 is more sensitive to a given compound than a cell line characterized by x_2, then

Nature Biotechnology: doi:10.1038/nbt.2877

ideally $f(x_1)$ should be greater than $f(x_2)$. The coefficient vector w is optimized to satisfy as many such constraints imposed by the training data as possible via the following constrained optimization problem:

$$\begin{aligned} \min_{w \in R^d} \quad & |w| + C\sum_{i,j}\xi_{ij} \\ s.t. \quad & 0 \le \xi_{ij} \\ & w^T(x_i - x_j) \ge b - \xi_{ij} \\ & (i,j) \in \Omega \end{aligned} \tag{1}$$

where d is the size of the feature vector x, ξ_{ij} are the slack variables, b is some constant (we used b=1 in our experiments), $|w|$ indicates the 1-norm of the coefficient vector w, and C is the parameter that adjusts the trade-off between the two conflicting goals in the objective function: minimizing $|w|$ versus minimizing the total error committed by the ranking function. The constraint set Ω includes all pairs of cell lines for which the pair-wise distance between GI50 values was at least 0.1 greater for one cell line than for the other. The value 0.1 is arbitrarily chosen to indicate that the difference in GI50 values of two compounds may not be biologically significant if it is less than this value.

The learning goal is to minimize the sum of the ξ_{ij} variables while making sure w is as sparse as possible, *i.e*., most of its elements are zero. The 1-norm of the coefficient vector w plays a critical role in this optimization problem by generating sparse solutions for w. This makes our approach capable of jointly performing feature selection and learning model coefficients. We use the Matlab optimization toolbox to solve this constrained optimization problem. Since the training sample size is small, each run takes only between three to four minutes.

The penalty parameter of error, *C,* was optimized using a held-out approach. For this purpose we have sequestered about twenty-five percent of the eligible cell lines in the training set for validation. We used the remaining cell lines to solve the above problem for each compound. The ranking performance of the model is measured according to how well the cell lines in the hold-out set are ranked relative to cell lines in the training set. The ranking between a pair of cell lines x_1 and x_2 is considered correct if $f(x_1) > f(x_2)$ when the GI50 of x_1 is at least 0.1 greater than that of x_2 or $f(x_1) < f(x_2)$ when the GI50 of x_1 is at least 0.1 less than that of x_2. The ranking is considered incorrect if $f(x_1) > f(x_2)$ when the GI50 of x_1 is at least 0.1 less than that of x_2 or $f(x_1) < f(x_2)$ when the GI50 of x_1 is at least 0.1 greater than that of x_2. The ranking accuracy is computed by the ratio of the number of correct rankings divided by the total number of eligible pairs, *i.e*., the number of cell line pairs for which the GI50 for one compound is at least 0.1 greater or smaller than for the other compound.

Our intention was to start with the gene expression profiles and then explore the impact of additional information on ranking accuracy. However, owing to time

Nature Biotechnology: doi:10.1038/nbt.2877

limitations we could only test RNA sequence expression calls in combination with the gene expression profiles. In **Table S1**, we list the ranking accuracies within the hold-out set for a subset of the compounds using the combination of gene-expression profiles and RNA sequence expression calls. Although our final ranking functions use both gene expression profiles and RNA sequence expression calls, almost all features selected were from the gene expression batch, which suggests that RNA sequence expression calls were not particularly useful for ranking cell lines within the context of the proposed approach.

Table S1: Ranking accuracy for a subset of the compounds within the hold-out set using only gene-expression profiles or a combination of gene-expression profiles with RNA sequence expression calls.

Anonymized Drug IDs	Gene expression only		Combination of gene expression and RNA sequence calls	
	Accuracy (%)	# of features (selected from 18,631)	Accuracy (%)	# of features (selected from 55,585)
1	92.4	27	91.8	27
2	92.1	24	94.7	24
3	93.6	29	93.6	29
7	94.3	30	93.7	31
8	91.0	26	90.3	26
10	100.0	25	97.8	26
16	94.5	30	94.5	30
17	97.6	30	97.6	32
19	96.7	29	97.4	30
20	100.0	31	100.0	31

Discussion

For a majority of the drugs, the corresponding ranking functions achieved over 90% accuracy within the held-out set. For the remaining drugs, for which the performance was relatively poor (around 60-70%) the integration of additional features from DNA mutation and methylation data, exome sequencing, and protein quantification data may further improve the predictive accuracy of the ranking functions. However, the main challenge in this case would be combining different types of data into a single feature vector. Future research will explore recent advances in probabilistic topic models involving hierarchical latent Dirichlet allocation[35] and hierarchical Dirichlet processes,[36] and it will investigate their extension for the analysis of genomic data within the scope of chemosensitivity prediction. Each gene can be considered as a word in a vocabulary, and the expression level of the gene in a specific cell line can be treated as the frequency of a word in a document. In this case, gene expression profiles can be modeled by multinomial distributions, and a topic can be considered as a distribution over a subset of genes. The topics and the existing hierarchy across topics can be automatically discovered from a set of cell lines. Once topic proportions for each cell line are obtained, this information can be

Nature Biotechnology: doi:10.1038/nbt.2877

used to train ranking functions for each compound as described above. This offers a more compact representation of the data and can characterize cell lines at different abstraction levels that may effectively correlate with the actual pathways.

Sparse linear regression 2

Summary

Features were initially filtered based on linear regression to drug response, training and prediction was done using elastic nets.

Introduction

We have built predictive drug response models for untreated breast cancer cell lines using the given omics data, including microarray gene expression, RNA seq, exome seq, methylation, copy number variation (CNV), and reverse protein lysate array (RPPA) of cell lines with known drug responses. We used a linear regression method based on the elastic net algorithm[37] to select the significant and informative genomic features for ranking the drug responses of the 18 test breast cancer cell lines for 31 drugs.

We chose the elastic net algorithm because this method works well in cases where the number of features (*p*) far exceeds the samples size (*n*) and it has the ability to perform feature selection. Previous studies have demonstrated that this algorithm is a promising method to use to analyze the genomic features of untreated cancer cell lines and the associated drug response to develop a predictive model and apply it in the preclinical setting.[15, 16]

Methods

The genomics profiling data provided for both the training and test cell lines had missing values, which presented a challenge for our analysis strategy. For instance, MX1 has only RNA seq and exome seq data. Only 8 out of 18 test cell lines had complete data. It is possible to impute the missing values for the 10 test cell lines but we felt this would not be practical since more than half of the data in test set would be artificial and may add noise to the dataset. We therefore ignored the missing values and built the prediction models based on each dataset separately, then later combined the predicted results.

The first input we processed for the elastic net algorithm was an $n \times p$ matrix of genomic features (X) where n is the number of training cell lines and p is the number of features for each dataset (e.g., gene expression values from Affymetrix GeneChip Human Gene). The second input was the vector of drug responses (GI50, y) across the 35 training cell lines. We then applied these two input data (X and y) to the elastic net package in R to solve the following optimization problem:

$$\hat{\beta} = \arg\min_{\beta} \left\{ \left\| y - X\beta \right\|^2 + \lambda_2 \left\| \beta \right\|^2 + \lambda_1 \left\| \beta \right\|_1 \right\}$$

Nature Biotechnology: doi:10.1038/nbt.2877

Since there were more than 10,000 genes in the gene expression data, we did pre-filtering to select the significant genes that had *p*-values less than 0.05 based on the univariate regression against the drug responses (GI50) of training cell lines. After filtering, only the significant genes were kept and the size of the data was more manageable. We discretized the methylation data into 0 (unmethylated) and 1 (methylated) based on Illumina beta values with the cutoff beta=0.2. We also combined both RNA seq and exome seq calls into a new data matrix for representing the mutation status of cell lines. The missing values were ignored in our analysis. The compiled data matrices, having genomic features for each dataset, were then used for solving the optimization problem. In order to avoid over-fitting and get a robust model for predicting the drug response in the test cell lines, we did the cross-validation by splitting the training set into sub-training and sub-test datasets. This process was iterated until the model was optimized. In this way, we were able to pick the optimal steps for the regression.

The elastic net equation was solved for each dataset using the default parameters to generate a coefficient matrix *b*. The optimized model (the selected genomic features with associated non-zero *b* values) was used for predicting the drug responses (GI50) for each drug in the test cell lines. Finally, we analyzed the predicted drug responses from six different datasets for the test cell lines. We found usually one or two datasets give quite different values compared to the rest. In addition, we did not have further information about the quality of each dataset. Therefore, we took the median of the six drug response values as the final predicted drug response (GI50). The final ranking in 18 test cell lines and in all 53 cell lines were based on the order of these GI50 values.

Discussion

Our simple linear regression approach was among the top performers in this challenge. We felt that our way of handling missing values may be important. Although we used the elastic net algorithm in our study like others,[15, 16] our approach is different; in contrast to studies where all data are combined into a huge data matrix, we treated each dataset separately and assumed each would provide an equal level of information. Indeed, it looks like such an approach worked well. In the future we would like to investigate the mechanisms of drug action on molecularly distinct breast cancer subtypes and we believe it would be very helpful for finding new therapeutic targets and speed the development of new anticancer therapies for individual patients.

Sparse linear regression 3

Summary

Gene and pathway features were determined using a one dimension factor analysis, training and predictions were made with spike and slab multitask regression, drug dose response values were re-calcuated from raw growth curves

Nature Biotechnology: doi:10.1038/nbt.2877

Introduction

The search for genetic factors casual for a particular phenotype is plagued by small sample sizes and high dimensionality, and the NCI-DREAM drug sensitivity challenge is no exception. The 35 cell lines available in the training data is a tiny sample size when lined up against the tens of thousands of cell line characteristics available in the dataset. In addition, particular data modalities are typically not available across all cell lines. Our approach attempts to address these problems in three ways: biologically directed dimensionality reduction, multitask learning, and integrating multiple datasets.

Methods

An impressive array of data modalities is available for the NCI-DREAM cell lines, including copy number variation, methylation, gene expression microarrays, RNA seq, RPPA protein abundance measurements and oncogene mutations. However, this wealth of information makes it challenging to detect true casual signal since many features are expected to be correlated with drug sensitivity simply by chance. The high degree of missing data is also a challenge for standard prediction techniques. Dimensionality reduction can help alleviate this problem both by reducing the number of effective features, and providing a generative model that can be used to fill in missing measurement. Accordingly, we perform two stages of dimensionality reduction, both using biological knowledge. In the first stage, we use a one-dimensional factor analysis model for each gene, so that each dataset is considered an observed variable, explained by the "latent factor," which may be interpreted as an activation level of this gene. Note that this is distinct from simply averaging across the different modalities for each gene; for example, methylation might be negatively correlated with gene expression for a particular gene. Our model accounts for such cases. We use a probabilistic Bayesian approach that naturally copes with missing data. In the second stage, we use a hand-curated collection of 1,987 known pathways, collected from resources including GO, KEGG, and published GWAS hits, to construct a per pathway activation level (where the gene activation levels are explained by this second layer of pathway latent variables), again using a one-dimensional factor analysis.

The NCI-DREAM challenge differs from a standard regression task in that there are effectively 31 separate drug sensitivities that we wish to predict. While it is certainly valid to treat prediction for each drug independently, we can potentially view this as a "multitask" learning problem where we aim to share statistical power across the drugs. While there are many ways to achieve this goal, we choose to hypothesize a per-feature latent variable, β, representing the probability of a particular feature being useful. We use a "spike and slab" prior on our regression coefficients, where the probability of the coefficient being non-zero is β. Thus, if a feature appears to be predictive across multiple drugs this will be reinforced.

Our combined model for predicting drug sensitivities can thus be written:

Nature Biotechnology: doi:10.1038/nbt.2877

$$Y = XW + FB + \varepsilon,$$

where Y is the full matrix of drug sensitivities (k cell lines by d drugs). The matrix X is a ($k \times p$) matrix containing the learned latent pathway variables for p pathways in each cell line, with parameters W specifying the impact of each pathway on each drug. F and B represent latent factors capturing the residual similarity among drugs and cell lines, as described in the preceding paragraph. The pathway summary variables X are fixed ahead of time, and W, F, and B are learned jointly to maximize the likelihood of all observations in Y. These learned parameters are then also used to fill in the missing values of Y including the test set predictions for the NCI-DREAM challenge.

We did not take the GI50 values in the drug sensitivity matrix, Y, as supplied in the NCI-DREAM challenge. We re-analyzed the raw growth curve data by fitting a linear model to regress out the effect on log OD of background OD, zero concentration OD (OD0.1, *etc*.), and the matched plate OD measurements. We then calculate "active area": the area above the growth curve of log(OD) vs. log(concentration), with the top of the area defined by the average log OD at the lowest drug concentration used. This measure has the advantage of combining how small a concentration of drug is effective with the effect size. Unlike GI50, active area can differentiate between the sensitivity of two cell lines even if the growth of neither cell line is ever inhibited by 50% by the drug. In the end, our predictions were based on our own quantification rather than GI50.

Discussion

In summary, we approached the high dimensionality, significant missing data and small sample of the NCI-DREAM data using a combination of biologically meaningful dimensionality reduction and multitask learning. Our experiments suggest that using the pathway level summaries significantly improved predictive performance compared to using the gene level summaries or individual features. Our multitask approach and incorporation of the existing Heiser, *et al.* dataset appeared to give only modest improvement; we anticipate that more benefit could be gained if the identity of the drugs (and their inhibition targets) were available.

Sparse linear regression 4

Summary

Missing features were imputed, combinations of datasets were enumerated and used to train elastic net regression models, for each drug, final predictions were made using the best performing model.

Introduction

The development of new cancer drugs usually comprises multiple phases of clinical trials. Due to the expense and inefficiency of clinical trials, human cancer

Nature Biotechnology: doi:10.1038/nbt.2877

cell lines become mainstream resources for drug sensitivity analysis. In line with this, several studies have made progresses in identifying multiple potential genomic markers of drug sensitivity by systemically generating genomic profiles of cell lines and determining their response to candidate therapeutic compounds.[13, 15, 16] However, how to integrate multiple types of genomics data to maximize the predictive power is still an open problem. The data of NCI-DREAM drug sensitivity challenge has provided an unprecedented opportunity for researchers to develop powerful tools and assess the effect of each data type to drug response.

To achieve this goal, we adopted the elastic net regression framework to predict sensitivity of 18 cell lines in the test dataset. We note that the number of features is far greater than the number of cell lines in this task. The elastic net regression framework is particularly well suited for this kind of applications.[37, 38] It strikes a balance between obtaining a parsimonious model (through the L1 term) and retaining groups of correlated features (through the L2 term), such as genes co-expressed or co-localized within the same copy number amplification regions. Another major challenge in the current task is that not all types of genomic features are obtained for every cell line in both training and testing data. Thus we need to adaptively train an optimal model for each cell line in the test data.

Methods

In total, we used five of the genomics profiling datasets, including mutation, CNV, gene expression, methylation, and RPPA. For the drug response data, we used the *k*-nearest neighbor approach to impute missing GI50 values, where $k = 10$.[39]

In the training procedure, we enumerated all combinations of the five types of genomic data to train prediction models on cell lines. In total, there are $C_5^1 + C_5^2 + C_5^3 + C_5^4 + C_5^5 = 31$ combinations represented in 31 different feature matrices that can be used to train 31 different models. For each of the 31 drugs, the feature matrix and drug response vector are denoted as $X \in P^{N,p}$ and $y \in P^{N,1}$ respectively, where *N* is the number of cell lines, and *P* is the total number of genomic features. For each compound, we selected the best performing model (of the 31 total models) using cross-validation. The best performing model is then used for predicting the response of 18 cell lines in the testing data. In situations where some cell lines may not have the full set of five genomic datasets, we only adaptively selected the best performing model among those trained on the remaining data types (**Figure S4**).

Nature Biotechnology: doi:10.1038/nbt.2877

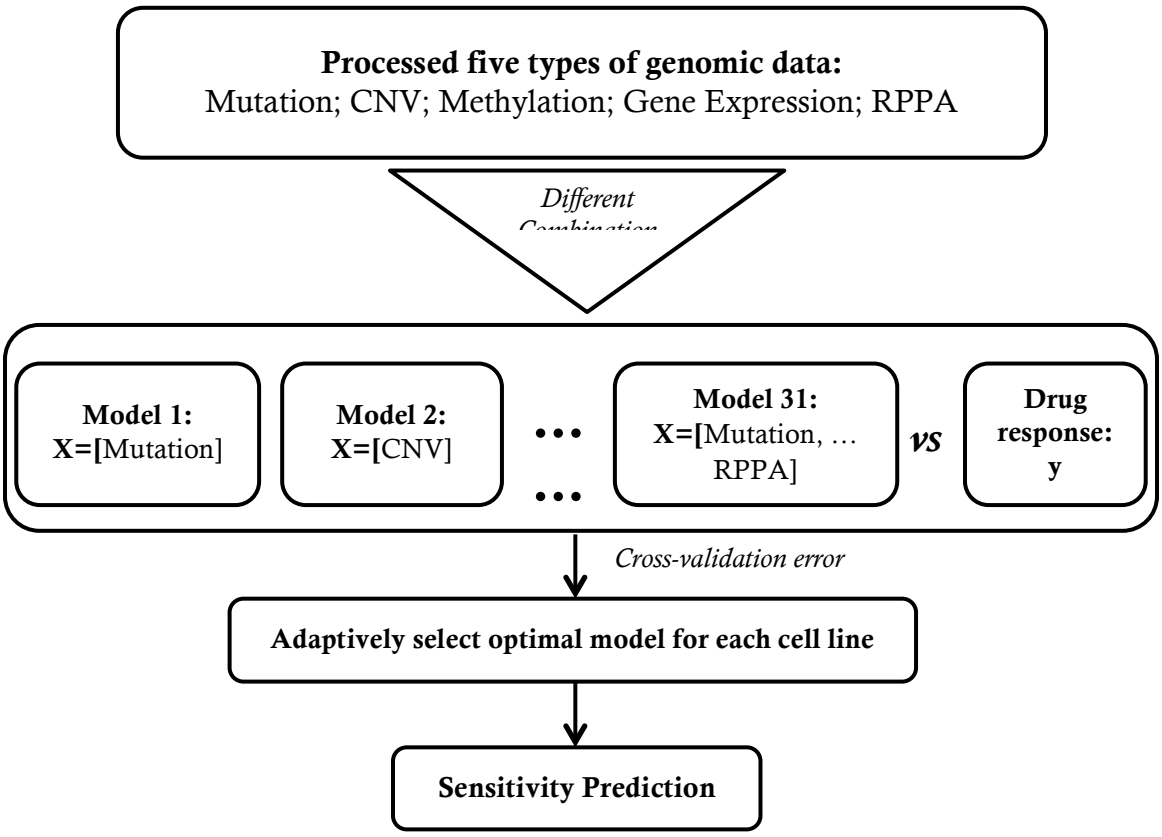


Figure S4. Illustration of the workflow for the Sparse linear regression 4 method.

Given the feature matrix X and response vector y, we employed the glmnet 1.8 software package to solve the following optimization problem:

$$\min_{(\beta_0,\beta)\in^{\circ}\ ^{p+1}} \left[\frac{1}{2N} \sum_{i=1}^{N} (y_i - \beta_0 - x_i^T \beta)^2 + \lambda P_\alpha(\beta) \right],$$

where

$$P_\alpha(\beta) = \sum_{j=1}^{p} \left[\frac{1}{2}(1-\alpha)\beta_j^2 + \alpha \mid \beta_j \mid \right].$$

In the elastic net model, α controls the relative strength of the L1 and L2 penalty terms, and λ controls the overall strength of the regularized regression penalty. The optimal setting for α and λ was chosen to minimize the root mean squared error (RMSE) using leave-one-out cross-validations for each (α,λ), with 20 different α uniformly sampled from [0.05, 1.0] and a default λ sequence calculated by the software package. After obtaining the best fit on the training dataset, we used the "predict.glmnet" function in the glmnet package to predict the response of the cell lines in the testing dataset.

Discussion

In this study, we adopted the elastic net regression framework to train the prediction models and adaptively select the most appropriate model based on

Nature Biotechnology: doi:10.1038/nbt.2877

existing data types of testing cell lines to predict their drug sensitivity. The results demonstrate that no genomic data type is consistently included in all the optimal models we selected. This partially implies the heterogeneity of the cancer cell lines and challenge of this prediction task. To capture this heterogeneity and improve the prediction ability, another key besides developing an efficient method is to expanding the cell-line panel.

Acknowledgement
This work was supported by the National Natural Science Foundation of China, No. 61379092.

Sparse linear regression 5

Summary
Gene and pathway features were determined using a one dimension factor analysis, training and predictions were made with spike and slab multitask regression drug dose response values were re-calcuated from raw growth curves, Heiser, *et al.* data[13] were used to train the model.

Methods
This method is a modification of the approach presented in Sparse linear regression 3. In the Sparse linear regression 3 method, the regression model was trained on the drug sensitivity data supplied in the NCI-DREAM challenge, though the drug sensitivity values were re-calculated based on the raw dose response curves. This implementation used additional outside information to expand the training set of drugs.

The experimental design of the challenge is the same as the original Heiser, *et al*. study[13] of which this challenge is an extension. While the drugs were not identified, the cell lines were. If we can find drugs in another dataset that have a similar sensitivity pattern across the common cell lines to those in the NCI-DREAM challenge, we should be able to say something about the expected sensitivity of the test cell lines. The original Heiser, *et al*. dataset included 12 of the test cell lines. We use matrix factorisation to transfer information between datasets: one way of viewing this method is that we embed each drug, and each cell line, in some low dimensional latent space. Nearby drugs show similar patterns of sensitivity, as do nearby cell lines. Using a probabilistic approach, we are able to cope with the fact that not all cell lines in the challenge were available in the Heiser, *et al.* dataset.

The combined model (as presented in Sparse linear regression 3) for predicting drug sensitivities can thus be written:

$$Y = XW + FB + \varepsilon,$$

Nature Biotechnology: doi:10.1038/nbt.2877

All matrices remain the same as described in Sparse linear regression 3, with the exception of *Y*. Here, *Y* is the matrix of drug sensitivities for *k* cell lines and *d* drugs that additionally includes drugs from the Heiser, *et al.* dataset for which NCI-DREAM cell lines were assayed.

Sparse linear regression 6

Summary

Features were removed with low dynamic range, missing feature values were imputed, training was done using lasso regression on individual datasets, final predictions were made using the weighted sum of regression models.

Introduction

The NCI-DREAM drug sensitivity challenge can be seen as a regression problem: we have *N* cell lines (*N* = 35 for the training set, *M* = 18 for the test set), each of which is described by a profile existing of *P -omics* features. For each compound *l* = 1,...,*L* and each cell line *i* = 1, . . . , *N* , we have one measurement $y_{i,l}$, corresponding to the GI50 concentration. These concentrations naturally define a rank, and thus the challenge to predict rank order can be seen as a regression problem. Here, we predict the results for each drug independently with a linear model. There are two main problems with this approach: first, not all variables are measured for all cell lines and all drugs and we have to define how to deal with these missing values, second, the number of features greatly outnumber the number of samples and therefore, we cannot use the classical regression. These problems are addressed in the proposed method.

Methods

Preprocessing: For all genomics profiling datasets, we first removed all features with a dynamic range of 0, *i.e.,* $\Delta(j) = \left[\max_{i\in\{1,\ldots,N\}} x_{i,j}, \max_{i\in\{1,\ldots,N\}} x_{i,j}\right] = 0$. In addition, we apply the following preprocessing scheme:

- **RNA seq**: We apply a filter, such that only above background expression values are kept.
- **RPPA**: We only keep fully validated data.
- **Methylation**: Values are filtered out as suggested, *i.e.,* rows are removed if *CpGs* < 3 or if *Cct*1 < 3.
- **Exome seq**: We summarized the given data as a binary data matrix that indicates whether a gene was mutated. We discard mutations if they were silent or in non-coding or intron regions and we apply filters on the confidence (> 120), the number of reads suggesting an alternative sequence (< 10) and the distance to the 3' end (> 0.1).
- After applying these filters, the values were normalized to have zero mean and a standard deviation of 1.

Nature Biotechnology: doi:10.1038/nbt.2877

Missing values: There are three types of missing values, each type requiring a different treatment. First, all datasets have not been acquired for all cell lines: the number of available datasets for a cell line varies between 1 and 6. For this reason, we cannot assemble every available feature into one matrix *X*. Instead we build one model for each dataset whose prediction results are then combined in the last step.

Second, there are missing GI50 values, *i.e.*, there are (drug, cell line) pairs for which no effect has been reported. As we work on drugs individually, we propose to remove the corresponding row (cell line) for the parameter estimation.

Third, there are missing values in the genomics datasets themselves. The missing values are distributed throughout the datasets, and we propose to impute missing values using the nearest neighbors, *i.e.* if a missing value occurs for a cell line, we take the value of the cell line which is closest in terms of profile features.

Estimation of the drug response: As mentioned above, we want to predict the GI50 values for each drug and each cell. In order to deal with missing data, we first predict the concentrations for each data source *s* separately and then combine the results in a second step. For simplicity, we only show the approach for one drug.

Let *x* be the feature vector from data source *s*. We can write the estimation of the GI50 concentration $\hat{y}_s$ as:

$$\hat{y}_s = f(x) = \beta_{s,0} + \sum_{j=1}^{P_s} \beta_{s,j} x_j = x^T \beta_s$$

where $x = (1, x_1, x_2, \dots, x_{P_s})^T$ is the feature vector and $\beta_s = (\beta_{s,0}, \beta_{s,1}, \dots, \beta_{s,P_s})^T$. As *N* >> *P*, we propose to use Lasso to determine the parameters of the model:

$$\hat{\beta}_s = \mathrm{argmin}[\|y_s - x^T \beta_s\|^2 + \lambda_s \sum_{j=1}^{P_s} |\beta_{s,j}|]$$

The parameter λ_s is obtained by leave-one-out cross-validation, and we write the minimal error (as obtained by leave-one-out) as $RSS_s^* = \left\|y_s - x^T \hat{\beta}_s\right\|^2$. From the minimal error, we can calculate the score $\alpha_s = 1 - \frac{RSS_s^*}{v}$, where $v = \sum_i (\bar{y}_s - y_{s,i})^2$. The score α_s therefore indicates how well the linear model fit the data compared to the intercept alone.

We know that the different data sources are unequally informative about the target variable, and we therefore weight the predictions coming from the different

Nature Biotechnology: doi:10.1038/nbt.2877

data sources with the fitting score: $\hat{y} = \sum_s \alpha_s y_s$. With $\hat{y}$ we can now establish the rank order of each cell line

Sparse linear regression 7

Summary
Statistically significant features were selected using Spearman correlation, training and prediction was done using an elastic net

Background
When predicting treatment response of untested compounds with new test cell lines, feature or gene selection is an important step for extracting the features that are associated with the treatment outcome. Until now, various models for predicting drug sensitivity were based on using all features or a few differentially expressed genes between the sensitive and resistant group.[15, 16] Methods based on using all features require much more computational time and may not be easily adapted to bigger datasets; whereas, methods based on differentially expressed genes suffer from the notion of setting cut-offs. In our approach, we extract a set of informative genes based on Spearman's rank correlation. Next, we apply the elastic net regression model to the training set of 35 breast cancer cell lines and finally we predict the response in the test set of 18 breast cancer cell lines.

Methods
Our methodology only uses gene expression data. A schematic representation of our algorithm is shown in in **Figure S7**.

Nature Biotechnology: doi:10.1038/nbt.2877

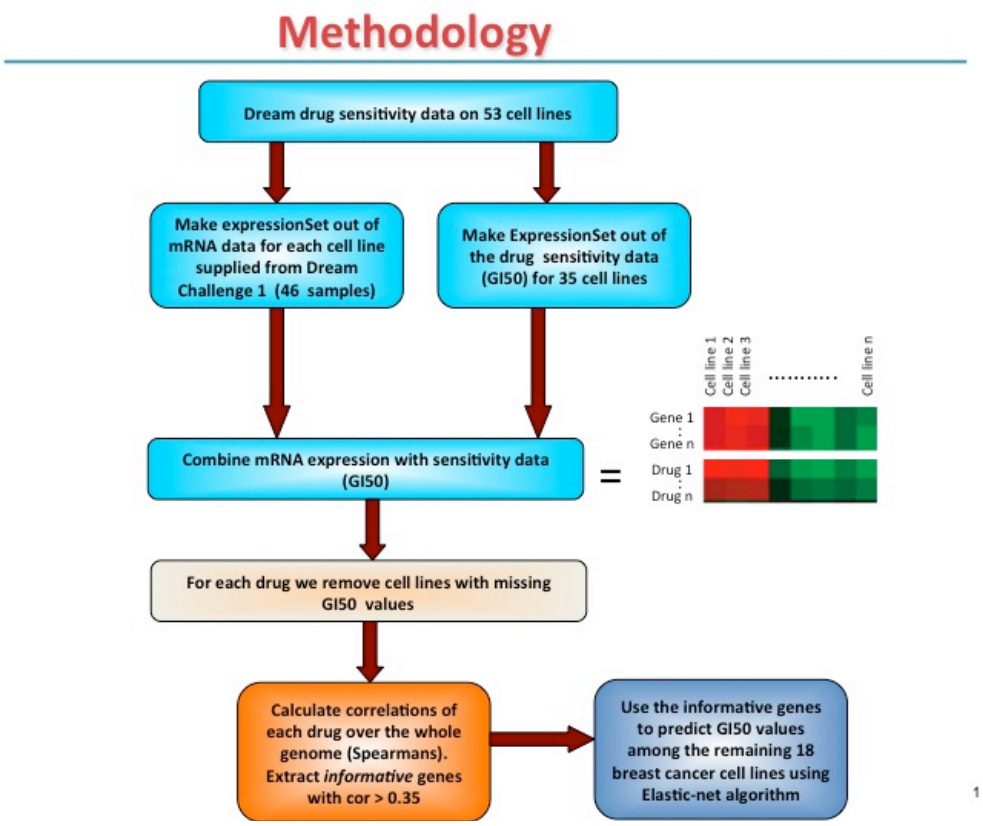


Figure S7. Schematic chart showing the work flow for the Sparse linear regression 7 method.

Elastic net[38] is a convex combination of the well-established methods of ridge[40] and lasso regressions.[41] This convex combination offers good predictive power and to be interpretable.[42] Elastic net is particularly useful when the number of features (*P*) is bigger than the number of observations (*N*). By contrast, lasso is not a very good predictor selection method in the $P > N$ case. Since the drug sensitivity data on the 31 drugs contains missing values for some drugs we made an implementation of both methods. Lasso performs effectively in a subset of the drugs with few informative genes, though it is greedy in the case of correlated predictors since it only picks the most correlated features. On the other hand, ridge regression keeps all predictors as non-zero with probability of one, and includes several highly correlated predictors.

The elastic net model parameters were chosen with cross-validation using the glmnet R package.[38] We also used the caret R Package[43] via resampling over a set of λ and α values. The values of α chosen vary between zero and one and it is assumed that an α close to one is “lasso-like”.

Discussion

In this work we restricted the number of informative genes used in the prediction to k=40. However, the computation can be parallelized by dividing the number of

Nature Biotechnology: doi:10.1038/nbt.2877

informative genes *k* into several groups. To further improve the elastic net model we should identify the accurate amount of informative genes needed for optimal prediction. To this end, it would be necessary to carry out statistical tests on the appropriate (*k*).

Sparse linear regression 8

Summary

Features were constructed by grouping genes according to GO terms, training and prediction was done using relaxed lasso regression

Introduction

Given the genomics profiling input data and the continuous output variable of drug response, the problem can be posed as a simple regression problem. However, the number of input dimensions greatly exceeds the number of examples (just the gene expression data has over twenty-thousand dimensions, while only roughly 30 training points are available per drug—the number is variable as many features are missing). Therefore, we focused on reducing the number of dimensions in order to apply simple regularized regression to the problem.

We worked only with the gene expression data (microarray and RNA seq). The fact that two different data sources were available for the same underlying phenomenon also led us to prefer this data. We grouped genes by their GO terms in order to obtain a smaller number of dimensions. After mapping expression values to GO terms, we used relaxed lasso.[44] Standard L_1 penalization achieves two goals: (1) sparsity in that many (even most) coefficients are exactly zero; and (2) regularization in that non-zero values are smaller (in absolute value) than in un-regularized regression. Due to the very large number of variables, a large penalization was needed to achieve sufficient sparsity, which leads to over-fitting. Relaxed lasso uses a strong penalization to choose coefficients and then a smaller penalty to find their final values.

Methods

We first combined the microarray and RNA seq data into a single measure per gene as follows:

1. Use the calls provided from the RNA seq data to determine which genes are changing. Only “active” genes were used.
2. Preprocess the RNA seq data with a log-transform, $r' = log(r + 1)$,where r' is the new value, followed by normalization to z-scores. Similarly, the microarray data was z-score transformed.
3. Combine the RNA seq and the microarray data into a single prediction by averaging the z-scored values.

Nature Biotechnology: doi:10.1038/nbt.2877

4. Any gene without both an RNA seq and microarray measurement was discarded.

We did not use this matrix directly. Instead we looked up GO terms for all genes (ignoring the Cellular Component vocabulary). All genes that map to the same GO term were combined by keeping only the largest value (in absolute terms), $maxabs(v_i) = argmax|v_i|$. For learning, we further processed the data using a threshold; we set the feature to 1 whenever the feature at that point is two standard deviations away from the mean (in either direction). We then select only features that are significantly correlated with at least twenty different drug outputs (*p*-value < 0.01, estimated by a permutation test). This allows for a modicum of information transfer between different drugs, which are otherwise treated separately; features that are informative about many drugs are less likely to be statistical artifacts than if this measurement was applied independently for each drug.

Finally, we used relaxed lasso for the optimization. A first lasso pass with $\lambda=2^{-12.12}$ was used for feature selection, and a second pass with $\lambda' = \lambda/10$ was used for the final learning. An initial attempt to use cross-validation to learn λ led to massive over-fitting, thus a value in the middle of the range was chosen.

On the output side, we normalize data by subtracting the per drug average from each entry, so that we regress on a centered value. Finally, coordinate descent was used for optimization, which ignores the regression error in the missing entries, i.e., we solve the following problem:

$$B^* = argmax_B \frac{1}{2n} \sum W_{ij} \left(Y_{ij} - (BX)_{ij}\right)^2 - \lambda |B|_1$$

Where W_{ij}represents the weight of example i, j. In our case, we set it to 1 if the example had data and to 0 if it is not (and we set Y_{ij} to an arbitrary value). Therefore, missing entries are ignored. For optimisation, we used coordinate descent as proposed by Friedman, ***et al***.[38] for this class of problems.

Discussion

The main driving force in the choices we made, was the pressure for feature selection and dimensionality reduction. For example, in the case of genes where one of the microarray or RNA seq measurements was missing, we decided to discard them instead of relying on lower-quality measurements.

One major problem with this approach is the need for setting parameters (the penalization factors λ). Cross-validation was a possible solution, but due to the small size of the training data, the variation between different folds was enormous and the final result was very unstable, a value for the regularizer was then hard-coded. A more robust solution would have been desirable. Similarly, other choices in the methodology (for example, the function to aggregate genes

Nature Biotechnology: doi:10.1038/nbt.2877

by GO term) were evaluated by cross-validation, but it would have been preferable to be able to rely on internal metrics.

Sparse linear regression 9

Summary
Gene and pathway features were determined using a one dimension factor analysis, training and predictions were made with spike and slab multitask regression, CCLE and NCI60 data were used to train the model, GI50 values were used.

Methods
This method is a modification of the approach presented in Sparse linear regression 3. In the Sparse linear regression 3 method, the regression model was trained on drug sensitivity values that were recalculated from the raw dose response curves. In this implementation of the method, the GI50 values supplied in the NCI-DREAM challenge were used as the drug sensitivity values.

Sparse linear regression 10

Summary
Features were selected using a regression with log penalty, which bridges the L0 and L1 penalty, missing values were imputed, penalized regression models were trained on indiviual datasets, final predictions were made using a weighted average

Introduction
The NCI-DREAM drug sensitivity prediction challenge is a typical high-dimensional problem where sample size *N* is much smaller than the number of features *P*. Recently, many penalization methods have been developed to address the challenging tasks of prediction and classification in such high dimensional settings.[45] It has been recognized that folded concave penalties such as SCAD[46] often deliver better performance than convex penalties such as the Lasso.[47] In addition, appropriate tuning parameter selection is crucial for the application of such penalization methods, and particularly, extended BIC has been developed to choose tuning parameters in high dimensional settings.[48] Guided by these recent methodology developments, we applied a penalized regression approach using a folded-concave penalty to select multiple genomic features that were associated with drug response. The tuning parameters of the penalty were selected by extended BIC.

Methods
Modification of existing penalization method: We employed the log penalty for our penalized regression, which is an example of the folded-concave penalties. Previous works has shown that the coordinate ascent algorithm for

Nature Biotechnology: doi:10.1038/nbt.2877

penalized Maximum Likelihood Estimation (pMLE) with the log penalty can be interpreted as iterative adaptive Lasso.[49] The numerical algorithm published before Sun, *et al*.[49] may encounter the problem of over-fitting when dimension is high and there are strong associations between covariates and response variable. We modified this algorithm by solving a least squares problem using a combination of coordinate descent and Local Linear Approximation (LLA).[50] Specifically, we updated the estimate of each regression coefficient sequentially (which is the coordinate decent part), and the solution of each coefficient is obtained after applying a local linear approximation.

Prediction model: We predicted drug sensitivities using 5 genomic datasets, including two DNA datasets (copy number variation and whole exome sequencing), two RNA datasets (gene expression and RNA sequencing) and one RPPA protein dataset (fully validated).

The missing values in the 35x31 drug response matrix were imputed by rank *k* SVD using the "Imputation" R package. The missing values were initialized with the corresponding column means then replaced by the values determined by the rank *k* SVD. This replacement was repeated until convergence. The optimal rank *k* was determined by cross-validation.

For each genomic dataset and each drug response, we selected features in two steps. First, we performed simple linear regression to select the top 1,000 features that were marginally correlated with drug sensitivities. Second, we further selected a subset of these 1,000 features by penalized regression using log penalty. The tuning parameters of the log penalty were selected by extended BIC. For the RPPA dataset, we used all 66 fully validated features for the penalized regression.

From previous steps, we had 5 predictions for each drug's sensitivities based on the five genomic data sets. We combined the 5 predictions using weighted average. The weights were selected either by PCA or by all possible weights of 0.5 or 1 (2^5-1 combinations). For each drug, we chose the weighting method that provided the most accurate estimation of ranks across all (training) cell lines; if two weights provided the same set of ranks, we chose the weight that provided the most accurate estimation of the drug's sensitivities across all (training) cell lines.

Discussion

In our penalized regression approach, we assumed an additive linear model, which is usually robust, but may be less sensitive when there are non-linear relationships between drug sensitivities and genomic features, or there are interactions among the genomic features; therefore, our method may be improved by model-free regression in the high-dimensional setting.[51]

Nature Biotechnology: doi:10.1038/nbt.2877

Sparse linear regression 12

Summary
Features were filtered on dataset specific criteria, missing values were set to random numbers, training and prediction was made using the interior point method for L1-regularization.

Introduction
We identified three major issues associated with the NCI-DREAM drug sensitivity challenge, which include:

1. The different molecular profile data from distinct measurement platforms were diverse, and it was difficult to integrate them.
2. There were a large number of genes/features within each of the above six datasets.
3. There were a large number of missing data for many cell lines that were randomly spread in all genomic and drug response datasets.

This motivated us to choose L1-regularized least squares regression for this challenge.

Methods
To overcome the above issue 1, statistical regression analysis was used and it combines all the genomic datasets irrespective of the molecular and platform differences and without preprocessing/normalizing individual datasets. Regarding issue 2, relevant genes were selected across samples from different datasets using various screening methodologies. Firstly, only those genes with standard deviation greater than 0.8 across samples in gene expression (both microarray and RNAseq) profiles were selected. Later, genomic identification of significant targets in cancer (GISTIC)[52] analysis was performed to select those genes with significant DNA copy number changes in 44 breast cancer cell lines. Finally, those genes that had correlation co-efficient greater than 0.75 across samples were selected from methylation dataset. The selected genes from the above datasets were combined and integrated with reverse phase protein array (RPPA) dataset for further analysis. We avoided exome seq data in this analysis.

Finally, a random value (-100) that was not present in any of the datasets was used in place for missing values, wherever applicable. In addition, this random value was chosen such that it did not compromise the drug prediction values to a greater level. This optimal approach dealt with the missing data. In cases where the GI50 values were missing in the training set of cell lines, those were considered as test cell lines.

The datasets were reduced as described above and combined into a single set “combined-dataset” (CDS, no preprocessing involved). Later, L1-regularized least squares regression was applied as described in Beroukhim, *et al*.[53] This form of training, in addition to providing “weights” for each feature in the CDS,

Nature Biotechnology: doi:10.1038/nbt.2877

also helped “sparsify” the parameters. The cost function reduced in this form of training is:

$$\min_x \|y - Ax\|^2 + \lambda \|x\|_1 ,$$

where *y* is the list of drug responses (GI50 values) provided for a particular drug in the training, *A* is the matrix of the CDS (where rows represent cell lines and columns represents all the genes/parameters), *x* is the final weights of each parameter to be determined, and *λ* is a penalizing coefficient. In other words, in addition to minimizing the distance between *y* and *Ax*, the algorithm also penalizes the 1-norm of *x*, thus, eliminating all but the most significant parameters needed to predict *y*. If the value of the coefficient *λ* is higher, a large number of *x* values (weights of genes that predict drug responses) will become insignificant, and *vice versa*. We used the implementation provided in Beroukhim, *et al.*,[53] where relative tolerance can be provided to solve the L1-regularized least squares problem within a given residual. During each iteration of the regularization process, the least squares problem was solved using the preconditioned conjugate gradient (PCG) method.

Discussion

Initially, we used 18 training cell lines and their data to identify molecular markers of drug response and later, we predicted drug responses for 35 test cell lines. Finally, we combined the drug responses from the 53 training and test cell lines and ranked them. There was a compromise on the prediction results due to the assignment of random number for missing data as discussed in issue 3. A better solution instead of assigning a random number for missing data in issue 3 discussed above could improve the results. In general, we observed gene expression profiles were better in predicting the drug responses compared to that of the other genomic datasets. This is probably true as gene and protein expression are final determinants of drug responses. Overall, in the post genomic era that generates high-throughput molecular and drug data, our algorithm performs drug response prediction analysis by integrating diverse data sources irrespective of different platforms being used.

Sparse linear regression 13

Summary Sentence: Features were selected using lasso regression in Gompertz growth model, and predictions were then made with the selected model.

Introduction

In post-genomics era, the widely used ‘omic’ technologies, including RNA-seq, genetics (SNPs), epigenetics (DNA methylation and histone modifications) and proteomics, are producing terabyte data related to human health. The multi-dimensional overwhelmed information requires being deciphered in order to identify associations between molecular subtypes, pathways, and drug response.

Nature Biotechnology: doi:10.1038/nbt.2877

We used Gompertz growth model to compress the dimensions generating genomic profiles of the 53 breast cancer cell lines, which will help us to determine their response to panels of candidate therapeutic compounds and make predictions from either of these data profiles in identifying single drug response in patients.

Methods

We first set up a statistical framework for inference and prediction. Let y_{ij} be the drug response in the ith cell line ($i = 1,2, \dots ,35$) for a given drug j. Here the drug response is defined as the coefficient b in the Gompertz growth model for each cell line such that $f(x) = ae^{-b^{-cx}}$, where x is the dose level of the drug. The coefficient b captures the speed of decline in growth, hence the sensitivity of a cell line to that drug. Linear regression models are fitted to model the drug response of each cell line to different genomic markers (e.g., gene expression, RNA sequence, methylation, and RPPA). Since there are far more genomic features than the number of cells, we regularize the linear regression models with a LASSO (Least Absolute Shrinkage and Selection Operator) penalty and solve it with the coordinate descent algorithm. Three-fold cross-validation is used to find the best tuning parameters for each of the regression models. The selected biomarkers are combined in the final model for building the relationship between drug response and genomic markers. This final model is robust as it eliminates the irrelevant biomarkers to the response and has good predictive performance. The drug responses for the 18 testing cell lines are predicted from the final model.

Discussion

For a specific type of genomic characterization provided in the drug sensitivity predction data, a statistical learning model is utilized to identify the genomics features most predictive of the dose response in the 35 training cell lines. We have limited the number of genomic features to about 10. We examined for 35 training breast cancer cell lines (1) the segmented genome copy number calls from the DNA copy number variation platform; (2) gene-level summaries from the transcript expression values platform; (3) mutation status from whole exome sequencing; (4) RNA sequencing data from whole transcriptome shotgun sequencing (RNA-seq); (5) DNA methylation data; (6) protein quantification data from Reverse phase protein array (RPPA); and (7) the drug response data. We found some interested results, for example the methylation level of ATP2A1, protein expression EIF4EBP1, SNP in C9f152 and TSPAN6 expression, they responded to all drugs, implicating they could be possibilities of biomarkers in predict response of other unknown drugs. We utilized the whole dose response data in addition to the GI50 concentration for the 31 anonymous compounds on 35 of the 53 cell lines.

Nature Biotechnology: doi:10.1038/nbt.2877

PLS or PC regression 1

Summary

Removed lowly expressed and/or low variance features, features were selected based on correlation to drug response, multiple partial least squares regression models were trained and consensus determined for final prediction

Introduction

Whole genome gene expression information is often selected as input data for building predictive models on drug response.[15, 16, 54-56] A recent FDA-led project [57] was conducted to evaluate methods using gene expression data to build models to predict clinical endpoints (MAQCII: MicroArray Quality Contol II). In the project, 36 independent teams analyzed six microarray data sets (including three human datasets on breast cancer, multiple myeloma, and neuroblastoma patient samples), to generate predictive models for classifying a sample with one of 13 endpoints. Using independent testing data, the study found that most teams' gene expression based predictive models perform very well on several endpoints, including estrogen receptor status and liver overall necrosis scores; on the other hand, all the teams' models poorly predicted overall survival in multiple myeloma [57].

We postulated that the poor prediction of overall survival in multiple myeoloma in the MAQC II study was due to that the pre-selection of an arbitrary 24 month cutoff for classifying patients [57]. Since both gene expression and overall survival data in the multiple myeloma case are continuous variables, one can also build a regression based model that may have greater ability to predict outcome of a continuous variable. In fact, the research group that generated the multiple myeloma dataset originally adopted a uni-variance Cox regression approach to analyze the data, and was able to identify a signature gene set as well as to observe a "high-risk" subgroup of ~14% patients [58]. This signature was later validated on several independent studies and on different regression-based approaches [59-62], highlighting the advantage of using a regression approach without predefining class memberships.

The original regression approach on the multiple myeloma study is not suitable for cell line panel situations, since the Cox regression was designed for handling survival data while drug response on cell line screens is characterized by GI50 values. Partial Least Squares Regression (PLSR) can be applied on different genomic profiling datasets, and it is well known to effectively handle high numbers of independent variables with minimal demands on sample size [63-65]. Therefore, we chose the PLSR approach as the basis for building our predictive modeling framework using the NCI-DREAM Challenge datasets. A specially designed splitting strategy was implemented in our PLSR framework to capture consensus features in the training dataset.

Nature Biotechnology: doi:10.1038/nbt.2877

Methods

Among the six genomic profiling datasets provided by the NCI-DREAM challenge, we focused on using gene expression data alone to build predictive models. The reason to choose a single type of data was mainly based on practical considerations – in real clinical trials, patient samples are hard to collect and very unlikely to be used to generate multiple types of profiling data. Choosing gene expression as the input data type for building a predictive model was because most of the publicly available profiling data was generated using gene expression platforms.

There are four cell lines in the test set that didn't have microarray gene expression data. To address this issue, the following evaluation was conducted: First, we downloaded Cancer Cell Line Encyclopedia (CCLE)[15] and compared microarray gene expression between NCI-DREAM and CCLE on the overlapping cell lines; second, we compared microarray and RNA seq data within the NCI-DREAM challenge on the same cell lines. We found that the correlation of microarray vs RNA seq data within the NCI-DREAM challenge was slightly better than the correlation of microarray data between NCI-DREAM and CCLE datasets. Therefore, we normalized the RNA seq data against the training set of microarray data for the cell lines with missing microarray data in the test set. This provided "educated guesses" for these cell lines and was used in our predictions.

There were nine out of the 31 drugs where the majority of the training cell lines have the same GI50 values, complicating predictive model construction. To address this limitation, we assigned the same GI50 values for all the test cell lines. In addition, several drugs' microarray based predictive models might not perform as well as others. In these cases, we also built RNA seq-based predictive models and compared the performance between microarray-based and RNA seq-based models, to identify the better performance for these drugs.

We developed a Partial Least Squares Regression (PLSR) [63-65] modeling framework that contains multiple steps on data preprocessing and normalization, data reduction, feature selection, a special splitting strategy to capture consistent features across the dataset, identification of independent models, determination of consensus gene weights, selection of a predictive model for each drug, and finally the predictions for the test set in the challenge.

The PLSR modeling framework

Data preprocess and normalization: In order to use RNA seq information for the four cell lines that lack microarray gene expression data, we first identified genes that overlap between microarray and RNA seq datasets, then performed a quantile based normalization on RNA seq data using microarray data as a reference. The merged data was used for subsequent model building and drug sensitivity predictions.

Nature Biotechnology: doi:10.1038/nbt.2877

Data reduction: Data reduction was done in two steps: First, we applied an intensity cutoff of 40% of the whole genome to remove genes that may not be present in the system. Secondly, we applied a variance cutoff of 0.3 to only keep genes whose intensities vary the most in the cancer cell line panel.

Feature selection: We performed feature selection by checking the correlation between each gene's expression profile and drug responses (GI50 values). Permutation was done on each gene by randomly assigning drug responses to the panel of cell lines. A raw *p*-value was calculated based on the permutation testing. Feature genes were selected using p-value < 0.05.

A specially designed cross-validation strategy: In model training/testing, a common cross-validation approach is to randomly divide the data into training and testing subsets, then evaluate model performance by checking the test set performance (e.g., mean or median of a performance distribution). Here, we developed a special approach to first do a "balanced split" that divides the data into training (70%) and test (30%) sets. The training set was further divided into sub-training (60% of training) and sub-test (40% of the training) sets by a "random split." Therefore, the whole training set is eventually divided into sub-training (42%), sub-test (28%) and test (30%) sets. We typically created 200,000 splits (models) per drug. In each split, we ran a 5-fold or 3-fold cross-validation (depending on how many cell lines have response data for an individual drug), to generate a PLSR model on the sub-training set. Then, we evaluated the model performance on this split for both sub-test and test sets. On each "balanced split", we evaluated performance using correlation and area under the curve (AUC), then selected top performing models.

Identifying top independent model*s:* Top models were selected from the following: 1) models should have top performance on the sub-test set for both AUC and correlation measures (correlation is weighted higher than AUC), 2) the test set should have much narrower performance distribution compared to sub-test set on both AUC and correlation measures, 3) top model performance in the test set should be better than, or at least similar to, the sub-test set, and 4) collectively, top models should have relatively high performance among all splits on the test set. After we identified top models using the above criteria, we checked the degree of overlap in the training sets of the top performing models. The rationale is that we aimed to identify high performing and independent models, which are expected to also capture the consistent relative importance of genes in the prediction.

Finding consensus genes weights and selecting a predictive model for each drug: After identifying top models from our cross-validation strategy above, we took the following steps to find a top predictive model for each drug: 1) removed top models that overlap with each other (sharing significantly common cell lines in training sets), 2) obtained consensus gene weights on the remaining top models using Singular Value Decomposition (SVD),[66] and 3) selected an

Nature Biotechnology: doi:10.1038/nbt.2877

individual model that had the highest similarity to the consensus model – this is our final model for an individual drug. We made predictions on the whole NCI-DREAM drug response dataset using the final model for each drug separately, then replaced training cell lines' response using the experimental data, and finally rank ordered the cell lines for each drug.

Discussion
When evaluating microarray vs RNA seq-based predictive models on the training dataset, we observed that microarray-based models tend to outperform the RNA seq models on the same drugs. This indicates that more work needs to be done on RNA seq data analysis to fully utilize its potential.

PLS or PC regression 2

Summary
Features were selected by using lasso regression and groups of genes predefined by core signaling pathways, predictions were made by linear regression of the reduced feature set to drugs response, predictor datasets were merged in advance of drug response prediction, and responses were predicted simultaneously sharing information among drugs.

Introduction
We entertained a number of models to rank cell lines by response to various drug treatments. Initially our effort was exploratory, in that we focused on preliminary processing of different genomic types and in finding features within these types that exhibited strong correlation with the response. The high covariate dimensionality, the low training set size, the low signal-to-noise ratio, and the extensive imbalance caused by missing data sources compelled us to reduce dimension and balance information sources as much as possible prior to constructing response predictions. The result of this effort was a covariate matrix (called *Z*flat) of dimension 53 lines by 22,227 genes holding line/gene summary scores of genomic variation. Importantly, this covariate matrix summarized all six genomic datasets in a maximally informative way, using whatever data happened to be measured on the given line and gene combination (the calculation was insulated from drug response data.) Subsequent prediction models used the genomic information in *Z*flat in a variety of ways. To make an informed judgment about which prediction model ought to be used for final rank prediction, we established a cross-validation (CV) system specifically designed to measure rank correlations on left-out samples. The prediction models that consistently showed strong test-sample performance were based on *pathway-index* calculations,[67] which first used lasso regression to select predictive genes within known pathways, then used these selections to create a response-dependent, pathway-specific covariate for each line. We reasoned that any dimension-reductions justifiable from good out-of-sample information would be beneficial, and so we focused attention on genomic information from a set of 15 core signaling pathways. Considering the limited response data per drug, we further reasoned

Nature Biotechnology: doi:10.1038/nbt.2877

that some benefits might be possible by combining information over drugs. We investigated methods that clustered drugs as well as methods that invoked shrinkage estimation via empirical Bayesian analysis. Ultimately a simple *regression-stacking* approach showed the best CV performance and was used to generate predictions.

Methods

Constructing covariate matrix *Z*flat: Preprocessing computations proceeded separately within each of the six genomic datasets. For instance, Affymetrix expression data were log-transformed; exome sequence data were reduced to binary gene-level indicators of some polymorphism. Data at each feature and within each genomic type were then standardized by removing feature-specific means across available data over all cell lines, and by dividing by the associated feature-specific standard deviations. Next, these `*z*-scores' were aligned into a large, 3-dimensional covariate array, *Z*, of dimensions 53 cell lines by 6 datasets by 22,227 genes. The alignment involved mapping feature ids to a common set of gene ids. We used *gene symbol* as coded in the R/Bioconductor library *org.Hs.eg.db* to enable this mapping. For protein data we parsed the feature names and matched where possible to the best-matching gene symbol. The final gene count of 22,227 represents those genes for which every cell line has at least some data for this gene (*i.e.*, in at least one of the six genomic datasets). Owing to idiosyncrasies of data generation and deficiencies of alignment, the large covariate array *Z* was still littered with missing data. However, at every cell line and every gene there were some data, and this enabled the final construction of the matrix, dimensioned 53 cell lines by 22,227 genes holding a summary gene/line variation score. Rather than simply average across *Z* to get *Z*flat, we processed the six genomic datasets using gene-specific principal components. We took the 53 by 6 matrix of *z*-scores at each gene and replaced this with a 53 by 1 vector holding the first principal component; the purpose was to find some linear combination of sources that retained maximal variation (over lines) at each gene. The construction thus allowed key variation at different genes to come from different sources.

Cross-validation system: Each prediction method was assessed within the 35 supplied training cell lines using a CV calculation tailored to the NCI-DREAM drug sensitivity challenge. Knowing that teams would be judged on rank correlation, and knowing the relative size of the supplied training set to the whole (35/53), we considered CV based on (test/training) splits of (10/25). We used (usually) 100 random test/training splits to assess a given prediction method. We trained the prediction method on the 25 training cell lines and predicted response on the 10 test cell lines. We then computed Spearman correlations between predicted test and actual test responses, separately for each drug. Using a smaller test size (e.g., leave-one-out), we reasoned that the performance quality measure (Spearman) would be inadequate. In addition to plotting all Spearman correlations from such a calculation, we computed a method score as a weighted average over drugs of average (over test/training splits) Spearman correlation, with weights equal to the observed response sample variances for the drugs,

Nature Biotechnology: doi:10.1038/nbt.2877

e.g., we always predicted drug 26 responses perfectly, but this high quality was eliminated from the method score since drug 26 was given zero weight. By using random leave-outs from the 35 training lines, we avoided a problem seen initially in the separate analysis of genomic datasets caused by lines with no data in that class of data.

Pathway index models: reducing dimension and combining across drugs: We found good predictions in methods that dramatically reduced dimensions using known pathway information in 15 core signaling pathways.[67] Aberrations in these signaling pathways are associated with cancer growth, and we reasoned that they might harbor key variation in the cell's response to drug. About 5% of genes are represented among these core pathways (1,093 of the 20,007). A response-guided covariate matrix (cell lines by pathways) was constructed by first applying lasso regression separately within pathways to identify genes that either (1) have a positive coefficient in the regression or (2) have a negative coefficient. (We used the R package *glmnet* for lasso regression.) Rather than use the coefficients (these can display inconsistency properties), we constructed a pathway index by taking the difference (for each line) between the average *Z*flat values at positive-coefficient genes minus the average *Z*flat values for negative-coefficient genes. The response-guided covariate matrix was then used to develop predictions; a key advantage is that we reduced from genome scale to 15 columns (pathways) and also we suppressed noise by eliminating data on genes that were not predictive (by lasso within pathway). Importantly, the response-guidance was done within training data, and thus was properly calibrated in the CV system.

To combine information among drugs, we stacked the matrix of response values (drugs by cell lines) into a long response vector. Similarly we stacked the drug-specific response-guided pathway index covariate matrices (cell lines by 15 pathways) into one big matrix ([lines x drugs] x 15). This large design matrix was further reduced by principal components analysis to a single first principal component, representing the linear combination of pathway-specific vectors that had maximal variation. We then used this single genomic predictor, in addition to an incidence matrix enabling drug-specific intercept terms, into a multiple regression against the stacked response vector. Predictions were generated from this fitted regression model.

Discussion

After playing with various prediction methods one line and one genomic type at a time, we sought a simple approach to integrate predictive information among lines and genomic sources. Additionally, we sought a CV system to compare the various prediction schemes under consideration. By an alignment of all the sources and a principal components (PC)-based combination, we produced a (gene/line) covariate with potential predictive power because of high variation over cell lines given by the first PC. Having reduced over sources we still had a huge number of genes and a seemingly low signal-to-noise ratio, so we

Nature Biotechnology: doi:10.1038/nbt.2877

reasoned that restriction to core signaling pathways might provide a useful structuring of prior knowledge. Our numerical experiments also showed that the pathway-index calculations had substantially better predictive performance than standard regression methods (e.g., lasso within drug or within data source). We did not extensively test approaches to combine across drugs, but the regression-stacking method had the potential to contain the relevant effects and it was much easier to deploy than model-based shrinkage methods.

PLS or PC regression 3

Summary

Training and prediction was done using principal component regression for individual drugs.

Introduction

This is a textbook "leave some out"-style prediction task. Our strategy was to use as many data types as possible (in separate models) and to merge predictions from multiple data types and multiple models into an overall prediction. We chose principal component regression to handle the dimensionality mismatch between the input matrix (one column per gene) and the output vector (drug response across cell lines).

Methods

No data processing was performed other than organizing the various genomic data sources into tables with cell lines as rows and genes as columns. For each drug, we fitted the linear regression:

$$y_{ij} = X_j\, b_{ij} + e_{ij}$$

where y_{ij} is the response of drug i modeled by genomic data type j, X_j is the matrix of genomic data type j, b_{ij} are the coefficients to be fitted, and e_{ij} is the residual. To handle the dimensionality mismatch we used principal components regression up to dimension d of X_j. The parameter d was chosen by leave-one-out cross-validation (CV). Predictions were generated using the fitted coefficients b_{ij}. After fitting and predicting using each genomic dataset separately, we merged the set of responses for drug i into an overall prediction for drug i. The predicted responses were converted to ranks and merged into an overall prediction using the rank-mean to produce the submitted prediction for drug i.

Discussion

Using leave-one-out CV to determine parameter d in each regression led to gross over-fitting and little to no predictive accuracy on the blind test set. In retrospect, n-fold CV might have produced better generalization though we did not investigate possible improvements after the fact. Having extremely poor generalization despite encouraging accuracy using CV is a sobering reminder of the “self assessment” trap.[68] We were surprised to find how poorly this method performed.

Nature Biotechnology: doi:10.1038/nbt.2877

PLS or PC regression 4

Summary Sentence
Statistically significant features were selected using correlation, models were fit using principal componenet regression, final predictions were made using a weighted average of models.

Introduction
One of the most difficult challenges in clinical oncology is the selection of the most effective therapeutic agents for an individual cancer patient. The use of ineffective therapy in a certain proportion of patients confounds overall clinical trial interpretation. Ineffective therapies can also lead to diminished overall therapeutic outcomes in routine clinical practice while decreasing the quality of life in patients who do not benefit from therapy. In order to improve the selection of the best drugs for specific patients, we developed an *in vitro* cell line-based drug response prediction strategy, COXEN (CO-eXpression ExtrapolatioN)[69-73]. Recently, next generation sequencing techniques have been widely used to obtain accurate gene expression patterns in the whole genome. In this study, we introduce a strategy for predicting drug sensitivity of cancer cell lines based on gene expression data captured by either or both RNA seq and microarray profiling.

Methods
Quality control and gene annotation analysis: For microarray and RNA seq data, we first took a log-transformation to reduce their distributional skewness for our subsequent statistical analysis. Prior to the log-transformation, expression values of RNA seq originally truncated at zero were replaced by the minimum value among all non-zero expression values. Then, the distribution and correlations of gene expression across all cell lines were examined statistically and graphically to identify outlying cell lines or genes. In addition, we examined whether expression values of the same genes were consistent between RNA seq and microarray data; No significantly outlying cell lines were discovered but some genes were excluded from this analysis. We then matched gene annotations between RNA seq and Affymetrix microarray data using Hugo gene nomenclature definitions. For multiple matched genes between the two platforms, the pair with the highest expression correlation was selected. Compounds that showed no differential growth inhibition activities across most of training cell lines were excluded in our prediction analysis since statistical prediction models for such drugs could not be generated and/or were not meaningful.

Feature selection: We defined three different sets of genes: RNA seq-based biomarkers, microarray-based biomarkers, and concordant biomarkers (between the two). First, training cell lines with all required data-RNA seq, microarray, and drug response were split into three subsets for each drug. Two subsets with the same sizes were used for biomarker discovery and evaluation (*i.e.* training), and

Nature Biotechnology: doi:10.1038/nbt.2877

the last subset was held-out for cross-validation. For each RNA seq and microarray biomarker, we tested significance of correlation between gene expression and drug sensitivity in the first and second subsets sequentially in order to avoid a multiple comparisons pitfall in a large-screening analysis. Genes that were significantly correlated with drug activities in both subsets were ranked by their average correlation coefficients derived from the two subsets for our multivariate prediction modeling of drug sensitivity.

Modeling and evaluation: We adopted a multivariate dimension-reduction technique of principal component regression to avoid model over-fitting on each of RNA seq and microarray data. The two cell line subsets used in the gene selection were combined for prediction model training and evaluation. For each candidate biomarker set, the top 3 principal components were extracted from each RNA seq and microarray data, and used to build regression models of drug sensitivity against the principal components in a cross-validation manner. Multiple competing prediction models were built by adding candidate genes in the biomarker set sequentially. Each RNA seq and microarray data-specific prediction model was evaluated based on their prediction performance on the test set with rank correlation. We also examined the consistency of the prediction scores between the RNA seq and microarray data in the test set. Finally, we selected a final model that maximized the sum of the performance and consistency indices. The final prediction model for each drug was tested with the external training subset and applied to the test cell lines.

Statistical imputation of prediction score of cell lines with missing data: In the 18 test cell lines, there are three cell lines (16.6%) with missing RNA seq and four cell lines (22.2%) with missing microarray data. Since cross-platform prediction models cannot be made for these cell lines, we used a statistical imputation technique to impute one type of missing data, *i.e.* RANA-seq or microarray data. That is, for the cell lines with missing RNA seq data, a linear regression model was built from the training and test cell lines from microarray data. Then missing RNA seq data of the cell lines were imputed by predicted values of the microarray-based prediction scores. Likewise, we performed imputation for the cell lines with missing microarray data.

Imputation and Integration of RNA seq and microarray based predictions: Our final prediction for each test cell line was based on a weighted average between the RNA seq and microarray-based models as below. Let $r_{A,i}$ *and* $r_{S,i}$ denote prediction scores of microarray and RNA seq-based models of the *i*-th cell line, respectively, then weighted average score is defined as:

$$r_i = w_A r_{A,i} + (1 - w_A) r_{S,i}, \quad \text{where } 0 \le w_A \le 1$$

In particular, when a weight on microarray-based model, w_A, is 0 or 1, the final prediction depends only on the RNA seq-based model or the microarray-based model, respectively. For the weighted average, the optimal weight was obtained with a cross-validation analysis by gradually changing weights from 0 to 1 in the

Nature Biotechnology: doi:10.1038/nbt.2877

training cell lines with randomly generated 20% missing data. We then made our prospective prediction of the test cell lines by the weighted average for each drug and ranked them against the entire cell line panel accordingly.

Discussion

In the study, we proposed a strategy for predicting drug sensitivity with both RNA seq and microarray gene expression data. In particular, we used the weighted average prediction strategy to enhance prediction performance of single data type-based prediction. For cell lines with only one type of data, we used a statistical imputation method based on a linear regression. This imputation enabled us to generate cross-platform prediction scores for test cell lines with missing data. However, non-linear or non-parametric regression strategies may provide better imputation by capturing complex relationships between different molecular profiling techniques, which needs to be further investigated in a future study. Despite these challenges, we believe that a successful development of cross-platform prediction techniques will greatly improve prediction accuracy of drug activities, avoiding bias from one type of molecular profiling technique.

Ensemble/Model selection 1

Summary

Features were selected using correlation, dimensionality reduced using PCA, Lasso and Ridge method, several regression models were trained for individual drugs and the top cross-validated model was selected to make final predictions for each drug.

Introduction

The continuous nature of the response variable in the NCI-DREAM drug sensitivity challenge naturally falls in the regression formulation as a basic setup, where variable selection plays a critical role.[74] Although regression appears to be a natural setup, which aspect of the distribution to be regressed is a research topic by itself for the challenge data. In order to do so, a compromise was reached to keep the implementation computationally feasible without compromising quality significantly. In our earlier work, we have noted that the supervised principal component (SPCA[75]) method, combined with various types of regressions, enables efficient implementation of the variable search in a biological meaningful and statistically optimal manner. We adopted SPCA-based variable selection and population based data augmentation while we explore a wide range of regression models and input for this challenge.

Methods

All supplied genomic datasets were explored using basic data exploration techniques to study their inherent natures. In most cases the basic calibrations carried out on respective datasets appeared satisfactory. Further processing was carried out for the exome seq dataset.

Nature Biotechnology: doi:10.1038/nbt.2877

Feature Selection: For each drug and dataset, we used correlation between cell line level drug response and measurements on genomic features as a measure of association. From some datasets, all available data were used if the number of features were no too high; otherwise, features were initially truncated using the above correlation measure.

From this truncated list of features, we applied some of the well known model reduction methods like supervised principal component analysis (SPCA), L1-penalised regression (*i.e.*, Lasso), L2-penalised regression (*i.e.,* ridge regression) and (traditional) stepwise regression.

Predictive models: We have explored a range of regression models for mean and hazard for this problem, including multiple regression models, generalized linear models (GLM), Cox-proportional hazard models, *etc*. Different transformations of the basic data were also attempted. For GLM, different link functions for Gamma distribution were used. It is not known from the domain of this problem whether the predictor variable are effective on predicting mean (of the activity) or whether underlying hazard function can also be predicted by these regressors. Thus a Cox-proportional hazard model was used to elicit that aspect of the given data.

Three fold cross validation: To assess predictive ability of the proposed models, a cross validation (CV) technique was adopted.[76] A total of 35 cell lines were provided in the training set of drug responses; however, the actual training set sizes varied from drug to drug and also across data types. In each case, the cell lines were split into 3 disjoint sets of approximately equal sizes, with one set being designated the test set and the remaining cell lines in the training set.

Discussion
It appeared different drugs were best predicted by very different combinations of data types, amount of data and modeling choice. The overall results presented here are a picked from thousands of such models for each drug based on CV.

It was in general observed that the dissimilarity between the learning set and test set response had affected the prediction quality. However the predictors need to be checked before it can be ascertained firmly whether extrapolation has taken place or the underlying distribution has been violated in some of the test samples.

Ensemble/Model selection 3

Summary
Features were selected using Spearman's rank correlation, missing values were imputed, predictions were made using the best performing method (determined

Nature Biotechnology: doi:10.1038/nbt.2877

by cross validation on the training set) among an ensemble of methods (random forest, support vector machine and linear regression)

Introduction

Previous studies have indicated that the molecular mechanisms responding to different drug treatments are different. For example, mutations in cancer genes could be key biomarkers for targeted agents, while they are less informative for responses to cytotoxic chemotherapy[16]. Therefore, we addressed the drug sensitivity prediction challenge in a drug-centric manner: for each drug, we pre-selected the candidate genomic features according to their correlations with the drug response across cell lines. Given the fact that the effectiveness of machine-learning approaches varies in predicting drug sensitivity from genomic data,[15, 16] we applied three well-established machine learning methods: random forest, support vector machine (SVM) and linear regression to the training set and chose the one with the best performance based on cross-validation (CV) for the final prediction on the test set.

Methods

Figure E3 shows the overall scheme of our analysis, and the predictive models were built for each drug, respectively.

Feature Selection: For continuous features like DNA copy number variations (CNV), DNA methylation, gene expression, and RNA seq, we chose the top 100 features from each category based on their Spearman's rank correlation with the drug response across the training cell lines. If significant features (p-value < 0.05) were fewer than 100 for a category (e.g., RPPA), we only included significant features.

To select the informative features from the exome seq dataset, we chose two different strategies: (i) we counted the mutation number for each gene in each cell line and chose the top 100 genes ranked by the mutual information between their mutation number and drug response across the training cell lines; (ii) we used binary indicator (1 and 0) to represent the presence or absence of the mutation in the gene. We then chose the top 100 genes as ranked by the p-value from the t-test between the drug response of the "0" group and the "1" group across the training cell lines.

We then combined the top 100 informative features of each category into our final candidate feature list. The genomic data for all cell lines (including training and test sets) were formatted accordingly.

Nature Biotechnology: doi:10.1038/nbt.2877

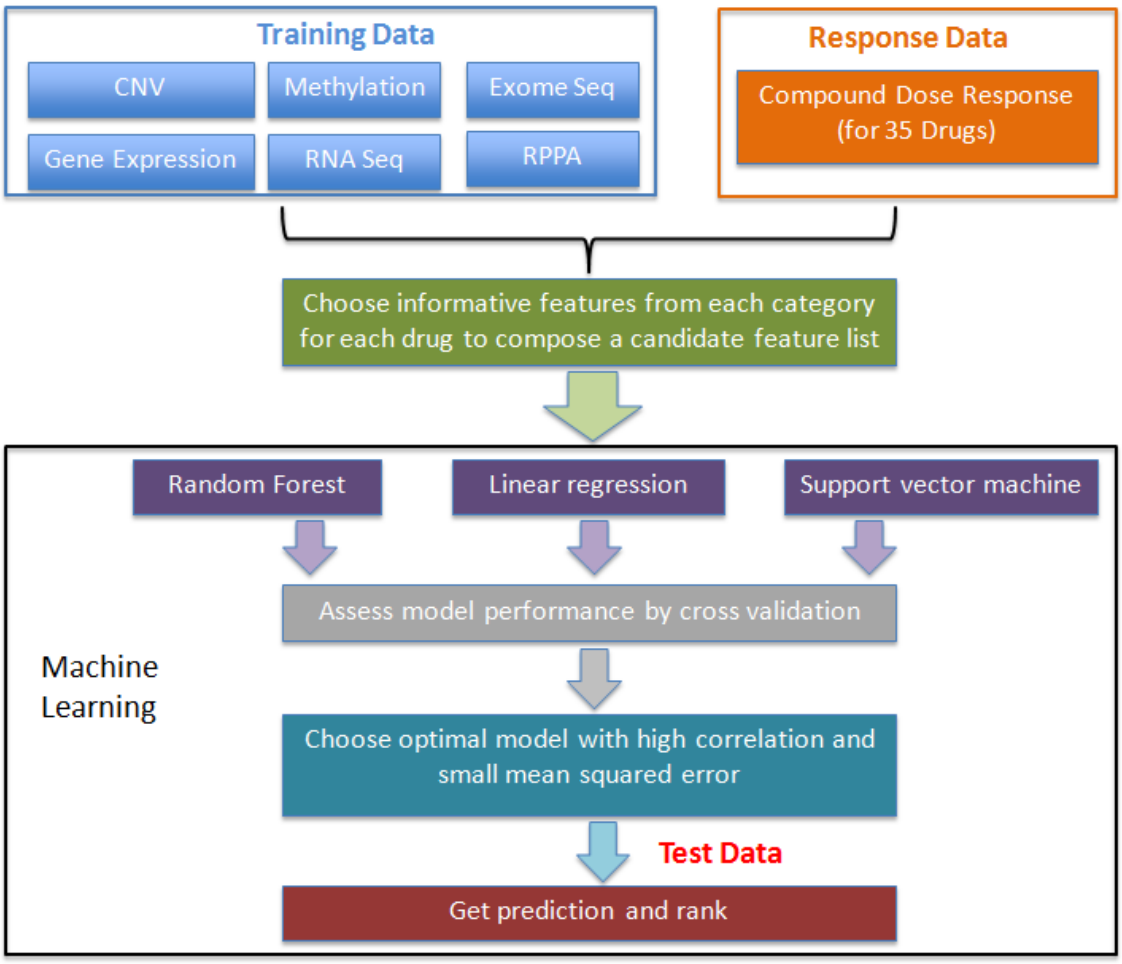


Figure E3. Schematic representation of the Ensemble/Model selection 3 method.

Nature Biotechnology: doi:10.1038/nbt.2877

Prediction Model: Before applying the machine learning methods, we first normalized the training and test data together and imputed missing values either directly or by principal component analysis (PCA) algorithm.

We applied three well-established machine learning algorithms (random forest, linear regression with PCA for feature selection, and SVM) to our training data to build the predictive models.

Based on leave-one-out CV, we assessed the model performance by calculating Spearman's rank correlation, as well as the mean square error (MSE) between the model prediction and the observed experimental data. The model with the highest correlation and the smallest MSE was applied to the test data for prediction, from which we ranked the sensitivity of the test cell lines to this specific drug.

Discussion

In this study, we used random forest, SVM and linear regression for drug sensitivity prediction using the top 100 genomic features correlated with drug response from diverse genomic profiling datasets. Among the genomic features across all datasets, gene expression from microarray provided the highest predictive power. However, when we restricted the analysis to the genes with both microarray and RPPA measurements, the latter usually provided better predictive power, suggesting that protein-level measurement is more informative about the drug sensitivity prediction when it is available. Noteworthy, the top 100 correlated genes from different platforms hardly overlapped, so they may complement each other and collectively contribute to the overall predictive power we observed. Among the three machine learning methods we used, SVM outperformed the other two in most of the cases according to our assessment method. In the future, we may (pre)select features in a more objective way, e.g., using false discovery rate (FDR) or the least absolute shrinkage and selection operator (LASSO), since the top 100 features might not be optimized.

Ensemble/Model selection 4

Summary

Gene and pathway features were compiled using outside data, an ensemble of prediction models were trained, final predictions were based on a rank-aggregation of combined prediction models.

Introduction

Our approach to solve this challenge was to use an ensemble method to aggregate the results of diverse prediction methods. We used methods that encompassed both *a priori* knowledge of cancer biology and modern predictive techniques. To incorporate *a priori* knowledge, we used Gene Set Enrichment Analysis to relate our feature sets to biologically relevant processes. As described in more detail below, we developed two new prediction methods:

Nature Biotechnology: doi:10.1038/nbt.2877

Difference Prediction and Cluster Similarity, both took advantage of the unique nature of the provided datasets.

Methods

Our models attempt to combine the results of multiple prediction methods in achieving an optimal prediction. We have implemented a set of feature selection methods available in the sklearn python package consisting of multiple types of *a priori* gene lists and computationally derived gene lists.[77,78] We applied these feature selection methods to all six genomics datasets provided in the challenge. We then used a collection of models to produce a final prediction by ordering the test cell lines according to drug sensitivity.

Feature Selection: Features consisted of the supplied measurements from the genomics datasets unless otherwise stated. Our computational feature selection methods fell into two main categories: correlation with GI50 values and pathway enrichment. For the correlation approach, we used step-wise regression[77,78] to determine the collection of *N* genes that have the lowest RMSE with the GI50 values. We collected gene lists for $N = 5, 10, 20$ for each drug and data type.

Pathway enrichment was done by grouping cell lines into 'susceptible' and 'resistant' classifications by fitting a two component Gaussian Mixture Model (GMM)[73] to the GI50 values for each drug. We then used RankProduct[79] to find significantly differentially expressed genes between the two groups. We used the DAVID web-tool[80] to determine gene-level annotations, where were grouped into 31 functional categories significantly enriched in the gene list. This produced two feature lists: a Boolean array of significant annotations and a list of the genes present in the significant annotations.

A priori gene lists were constructed using two methods. The first method was from our previous research with microarrays in deducing disease signatures for various subtypes of cancer and picking genes related to drug response.[81,82] The second method was to use computational approaches to cull lists from the Genetic Association of Disease database.[83] We used the ChEMBL database[84] to find the 20 most common genes targeted by chemotherapy drugs. We also identified the 20 most commonly mentioned gene names in the text of articles returned from the PubMed query "cancer drug targets".

Prediction: To process the nearly three thousand feature lists thus generated, we developed a pipeline, implemented in the IPython Notebook,[85] which fit the selected model, predicted the unknown data points and then used leave-one-out cross-validation (LOC) to predict the known data points. The sklearn python package provides many prediction methods, which we enhanced with two novel methods.

From the sklearn package we used the K-Nearest Neighbor Regression (KNR), Linear Least-Squares Regression (LR), Support Vector Regression (SVR), and

Nature Biotechnology: doi:10.1038/nbt.2877

Gradient Boosting Regression (GBR). We used the sklearn Grid Search technique to determine the optimal parameter sets for each of the feature lists using LOC. We used Kendall's Tau as the objective function.

Difference Prediction

This method we developed attempts to transform a regression problem into a binary prediction problem. The method consists of the following steps:

1. Calculate the difference between each pair of corresponding features. These differences become the new feature set.
2. From the GI50 values, determine the 'susceptible' and 'resistant' classification, which becomes the new response variable.
3. Train a simple classifier to predict the response variable based on the feature set from step 1.
4. Use the classifier to classify new cell lines into this context.
5. Determine the best rankings based on the response variables.
6. Extract GI50 value for these new cell lines.
7. This method has the advantage of reducing the regression problem into a binary classification problem at the same time that one increases the number of 'observations'. For this method we used either a logistic regression or a gradient-boosting classifier.

Cluster Similarity

This method uses a multi-step clustering approach to find a collection of features that preserve clusters created using GI50 values. The method is consists of the following steps:

1. Cluster cell lines by their GI50 values.
2. Find the features that preserve this clustering.
3. Use KNN regression with the features found in step 2.

Aggregation: The final predictions were based on combining the predictions of multiple prediction sets we generated. We assume that each of our predictions is correlated with the correct answer, yet independent of each other. Data aggregation attempts to find the results that are consistently near the top and adjusts the ranks accordingly. We excluded non-significant predictions (Kendall's Tau, $\tau > 1.0^{-5}$).

We implemented two aggregation methods: a linear-regression prediction and a weighted rank aggregation. For linear regression we use the following steps:

1. Use the predicted GI50 values from each method as a feature and the known GI50 as the response variable.
2. Train a linear regression model.
3. Predict unknown GI50 values.
4. Rank the predicted GI50 values.

The weighted rank aggregation method[86] has the following steps:

1. Convert all GI50 predictions to ranks.

Nature Biotechnology: doi:10.1038/nbt.2877

2. Use the LOC score as a 'weight' and calculate the weighted-mean of the ranks for each GI50 value.
3. Re-rank the weighted-means.

To evaluate these two methods we also use LOC. In practice the weighted-rank aggregate outperforms the linear regression method, $\tau = 1.0^{-15}$ vs. $\tau = 1.0^{-6}$. Our final predictions for submission used the weighed rank aggregate method.

Discussion

Our proposed method had a strong tendency to over-fit the training data. With a larger dataset of known GI50 values we would have possibly been able to avoid this pitfall. Incorporation of a Gaussian model to predict rankings could have potentially improved our final results.

Ensemble/Model selection 5

Summary

Features were selected using outside pathway and interaction data, missing values were imputed, individual drug predictions were made using the best model selected from an ensemble of methods

Background

Our model to rank order the drug response of 18 test breast cancer cell line in relation to 35 training breast cancer cell lines (*N*) was centered around two approaches. The first lies in variable selection. The genomic datasets provided would result in over 60,000 features (*P*). This results in a classic small *N*, large *P* problem in which a model's efficacy may be compromised as too many variables are left to explain a model,[87] which leads to over-fitting. In order to address this problem, we used a variable selection method that only selected features that had a high pathway-level impact on the cancer molecular interaction networks. Network or pathway-level impact has been used many times to prioritize genes for a variety of studies from detecting driver mutations[88] to identifying patient-specific pathways.[5] We also limited our approach to the gene expression dataset. We ranked all genes in the network using a method we recently developed called DawnRank (unpublished), selected the genetic variables that exhibited the highest network-level impact, and built our prediction model based on the 500 highest-ranking variables.

The model we used to rank breast cancer drug response was to select the best performing drug-specific classifier from multiple types of regression models. We decided to use this approach due to the fact that there is no "best" classifier for all types of data, and that the optimal classifier may change from drug to drug as they are independent samples. Using cross-validation, we identified the top performing model from SVM (Radial and Polynomial Kernel), RandomForest, Boosting, and Ridge Regression.[87] We opted for these particular classifiers due to their ability to handle large numbers of variables with only a relatively small number of observations.

Nature Biotechnology: doi:10.1038/nbt.2877

Methods

Feature Selection: We selected the features with the highest-ranking score using our DawnRank approach. We first started with a network from Ciriello, *et al*.[89], which represents an aggregated pathway drawn from large-scale curated databases such as KEGG[90], PID[91], and Reactome[92] as well as non-curated sources such as derived protein–protein interactions, gene co- expression, protein domain interaction, GO annotations, and text-mined protein interactions. For each drug, the NCI-DREAM gene expression profiles were used as input in our method; the top 500 genes were selected.

Imputing missing values: In addition to the variable and model selection, we also addressed other issues in data cleaning for our model regarding missing values. We observed that some of our models did not perform when using a small number of observations; therefore, we used a mean imputation to estimate missing values. Also, of the 53 cell lines, 3 did not have gene expression data. We used *K*-nearest neighbors impute these values. Regarding drug response values, we found that certain drugs were virtually identical across all cell lines, which caused models to fail. To correct for this, we added a small epsilon, a random fourth digit in which we generated from a uniform distribution (-0.0005, 0.0005) to provide a small amount of variance in those drug responses. We refrained from using drugs with many N/A values and little variation. All in all, 7 drugs (drugs 4, 6, 12, 13, 20, 26, and 27) were not used.

Model Selection: For each drug, we fitted five different potential models (SVM with a radial kernel, SVM with a polynomial kernel, Random Forest, Ridge Regression and Boosting) and used root mean square error (RMSE) loss to determine the top-performing model. Because the test response was hidden, we applied a 4-fold cross validation among the training dataset with each model for each of the 31 drugs. The model that exhibited the lowest RMSE for a given drug was selected to predict the test data for that drug. For the SVM models, we used both the radial and polynomial kernel in our analysis, and tuned our parameters for gamma from 2^{-7} to 2^{-2} and cost from 2^{-3} to 2^{-2}. Our random forest implementation was based on the normalized votes of 500 trees. Our ridge regression took on all lambda parameters from 1 to 1000 with increments of 1. The Generalized Boosted Regression models were the most restrictive in terms of our parameters even after imputing all missing values, and consequently, we relaxed the parameters to include 8 trees, a shrinkage of 0.1, and a bag-fraction of 0.9.

The most robust method for calculating drug sensitivity was Ridge Regression, the best-fit model for 18 of the 24 predicted drugs. Most of the remaining drugs were best fit using the SVM method with a polynomial kernel. SVM-Radial, Random Forest, and Boosting made only minimal impact on our model fitting. It is important to note that SVM radial and Boosting were selected as the optimal model for 3 of the 7 flagged datasets which may be indicative of their success in noisy data.

Nature Biotechnology: doi:10.1038/nbt.2877

Discussion
Our multi-model regression approach shows effectiveness for predictions drug response for a large number of drugs across many test cell lines. Nonetheless, the model can be improved upon as several drugs performed poorly on our model. Some suggestions for future work would be to address the model limitations by selecting other models and honing in on the variable selection. Additionally, an imputation for missing values could be an improvement as well. Overall, although our model in analyzing the NCI-DREAM dataset can be improved upon, the results using a statistical approach provide us with a crucial step to predict drug response reliably from genomic profiling datasets.

Other 1

Summary
Features were weighted based on Pearson's correlation to drug response, predictions were made using the correlation of the weighted features.

Introduction
This method could be considered somewhat distinct, in that it is rather simple and straightforward, while making no extraordinary effort to filter for the top predictive features. Instead, all features in each expression dataset examined were used, but each feature was weighted according to its correlation with sensitivity; features with no correlation would have essentially zero weight, but would not be explicitly excluded.

The following datasets were used in this model, gene expression, RNA seq, and RPPA. The decision was made to not use copy or mutation data, as these can be considered sparse and not conducive to examining global correlations.

Methods
For each expression dataset, features were log-transformed (if not already log-transformed) and centered across samples on the median.

The analysis described below was carried out for each of the three datasets individually. Results from the three datasets were then averaged in order to derive the final scores. For each dataset, all features profiled were used in the scoring (*i.e.*, there was no filtering or pre-selection of “best” correlates for the purposes of classifying).

With the given dataset and the known GI values, a matrix of correlations (by Pearson’s) was constructed (across the cell lines) between GI50 values and expression values. Within the matrix, each feature (e.g., gene or protein) had a correlation value for each of the 31 drug compounds; a strong positive correlation between feature and drug would suggest that the feature might be a marker of

Nature Biotechnology: doi:10.1038/nbt.2877

sensitivity (e.g. ERBB2/GRB7 for lapatinib GI50), and a strong negative correlation, a marker of resistance.

Using the above (gene X drug sensitivity) matrix (from which we get genomic profiles of "drug sensitivity"), the Pearson's correlation was computed between each drug sensitivity profile (derived from the training cell lines) and each genomic profile of the test cell lines. A high correlation between a drug sensitivity training profile and a test sample genomic profile would suggest that the test sample would be more sensitive to the drug (at least relative to the other cell lines).

Using the three datasets analyzed in the above manner (gene expression, RNA seq, and RPPA), the predicted sensitivity correlations for each drug X cell line were averaged across the three genomics datasets (the values being first z-normalized within each platform), in order to get a final predicted sensitivity correlation.

For the final results submission, the predicted sensitivity values for each drug were ranked across the cell lines. For the test samples, the ranking was assigned relative to the predicted values (across both training and test samples).

Discussion
For all its simplicity, the method ranked third overall. This could suggest that simpler approaches might be comparable in performance to more complicated approaches. Overall, the individual result sets using RPPA, gene array, and RNA seq datasets were largely correlated with each other, though averaging the three may have helped in reducing noise from outliers. Also, as the entire expression profile was used, the genomic information encoding drug sensitivity could involve hundreds if not thousands of genes; each profiling platform may provide information as well.

In the future, an examination of top correlated features for each drug, using a biological perspective, may be informative in terms of better understanding the biology of drug sensitivity in the cancer cells.

Other 2

Summary
Select gene features showing strong survival from the METABRIC dataset then hierachically cluster, build linear model to fit gene clusters to drug response, predict using regression model

Introduction
Cell lines are regularly used as models to understand tumor cells in human patients. Usually it is reasoned that if a compound is capable of reducing the viability of model cell lines, then the same compound should also have potential

Nature Biotechnology: doi:10.1038/nbt.2877

to increase patient survival. This motivates the question: Can this link also be reversed and utilized for the prediction of effects of combined compounds or for the effects of the same compound when applied to other cell lines? When exploring gene expression data of patients, one can usually identify many genes that are significantly correlated with patient survival, for example in a “higher expression is worse” pattern. In this example the assumption of the reversed reasoning would be: Any compound that is capable of decreasing the expression of these “bad genes” will also have potential to reduce viability of the tumor cell lines. In this report a prototype algorithm is presented that uses assumptions like these to transfer the information known about compounds for some cell lines to new cell lines and to make predictions about the compound effects on them.

Methods

Information on the patient side was learned from the METABRIC study[94]; more precisely, from the normalized $\log_2$(ratios), $M_{i,j}$, for all available genes I and for all available patients J in both the discovery and validation set (downloaded from European Bioinformatics Institute; accession: EGAS00000000083). Based on the follow-up information of disease-specific overall survival, genes were identified whose expression was significantly correlated with survival of breast cancer patients. These genes were partitioned into correlated genes $I^+ \subset I$ (showing a “higher is better” pattern) and anti-correlated genes $I^- \subset I$ (“lower is better”). Genes without significant correlation ($\alpha > 0.05$) were excluded.

As information based on a single gene is rather uncertain, hierarchical clustering of the gene expression profiles was used, separately for both identified gene partitions and based on the correlation metric with Ward linkage. Several signatures of co-regulated genes $G_k^+ \subset I^+$ and $G_k^- \subset I^-$ were defined manually by visual inspection and selecting clusters in the resulting hierarchical cluster dendrograms. Only well-separated clusters were selected for further analysis. For each identified gene signature, a survival analysis for the average signature expression was conducted and survival slopes $s_k^\pm$ were calculated for every signature, defined as the expected linear increase (or decrease) in survival per difference in the signature’s average gene expression. This encodes the information about how much a higher expression of a particular set of co-regulated genes G_k^+ is better (or worse for G_k^-) for patient survival.

For drug sensitivity prediction, two sets of information are required: By how much does a higher average expression of the same gene signatures $G_k^\pm$ influence the decrease in viability of a cell line following the administration of a specific compound? Ideally, one would have measurement data before and after treatment for many cell lines and compounds to estimate the compound effects on gene expression and viability robustly. In this challenge normalized gene expression profile (GEP) data, $\widetilde{M}_{i,l}$, for genes I and cell lines L were available before compound administration, but not after compound administration. Therefore instead of trying to find the viability slope over the average signature expression separately for every single cell line, this was done for all cell lines

Nature Biotechnology: doi:10.1038/nbt.2877

simultaneously as follows:

Let $\overline{\widetilde{M}(G_k^{\pm}, l)}$ be the average expression of signature $G_k^{\pm}$ for cell line l and let $c_{\mathrm{GI50},l,n}$ be the concentration of compound n necessary to inhibit proliferation of cell line l by 50% (after 72h). In order to calculate the GI50 concentration of a compound n, predictors had to be defined based on the average gene expression of every signature $G_k^{\pm}$: Linear models and least squares fits were used on the points $\left(\overline{\widetilde{M}(G_k^{\pm}, l)}, c_{\mathrm{GI50},l,n}\right)_l$ for all available cell lines and resulted in linear functions $c_{k,n}^{\pm}$. The combination of the predictions for all signatures $G_k^{\pm}$ was defined as $c_n := \sum_k w_{k,n}^{+} c_{k,n}^{+} + \sum_k w_{k,n}^{-} c_{k,n}^{-}$, where the weights were normalized to 1 for each sum and were defined to be proportional to the influence $s_k^{\pm}$ of cluster $G_k^{\pm}$ on the patient survival and to the goodness of fit for $c_{k,n}^{\pm}$ (measured via correlation of the underlying data points). Now predictions of the expected GI50 concentration for a compound n administered to a new cell line $\tilde{l} \notin L$ were possible based on its average expressions for the gene clusters:

$$c_n(\tilde{l}) = \sum_k w_{k,n}^{+} c_{k,n}^{+} \left(\overline{\widetilde{M}(G_k^{+}, \tilde{l})}\right) + \sum_k w_{k,n}^{-} c_{k,n}^{-} \left(\overline{\widetilde{M}(G_k^{+}, \tilde{l})}\right)$$

Finally, the cell lines were ranked for each compound from the most sensitive (having the lowest predicted GI50 concentration) to the least sensitive.

Discussion

By developing a method to address the NCI-DREAM challenge, it was possible to test the conjecture that it is possible to quantitatively infer the decrease in cell line viability from the effects of the same compound on many other cell lines via gene expression patterns that are associated with patient survival.

Although more data was provided, only the gene expression and GI50 concentration data were used in order to specifically test this conjecture. (Note that the cell lines 184A1, 21MT1, 21NT, HCC1569, MX1, SUM229PE, T47DKBLUC did not have valid GEP data and thus could not be ranked.) Despite this restricted information the ranking predictions were significant (*p*=9e-4). However, a similar model used for the sister sub-challenge about DLBCL in order to predict effects of pairs of compounds on a single cell line and this model did not produce significant predictions. Taking both results together, it is probable that we can learn something new about the tested conjecture:

Gene clusters $G_k^{\pm}$ that are relevant for patient survival can indeed be used to transfer the information about compound effects to a new cell line, but only if these gene clusters were stratified by weights based on the consistency of the compound effects on many cell lines of the disease. If their weights were only determined by experiments with different compounds, but the identical cell line, then predictions of the effects of combinations of compounds on the same cell line was not possible. Maybe this can be explained as follows: For a single cell line, several of the $G_k^{\pm}$ signatures that were identified based on patient survival are not applicable, since the cell line is only representative for a specific subtype

Nature Biotechnology: doi:10.1038/nbt.2877

of the disease. This is especially true for diseases like DLBCL that are known to be genetically heterogeneous. Additionally, a single compound only affects the expression of a specific subset of genes and the overlap with the $G_k^{\pm}$ might be small for most of the tested compounds. Taking together, there might simply not be enough information to robustly define the weights for the $G_k^{\pm}$ in the "single cell line, multiple compounds" scenario.

Clearly, from an analysis point of view it would be ideal if one had GEP measurements before and after compound administration like in the DLBCL sub-challenge, but for many cell lines of the disease like in this breast cancer sub-challenge. Then it would be possible to combine both weighting schemes which should result in a self-stabilizing effect and might also allow predicting the effects of pairs of compounds based on patient survival (not just for a single cell line but for all used in the training phase plus new ones).

Other 3

Summary
Missing features were imputed, signatures were extracted for each dataset, predictions were made using 1-nearest-neighbor to training cell lines via Pearson's correlation between signatures for each data type, final predictions are the weighted sum of the individual datasets

Introduction
A cell response to an external stimulus such as a small molecule or drug is mediated via a cascade of interacting proteins and expressed genes. The temporal state of these genes (e.g., their expression values or methylation state) and genomically encoded information (e.g., single nucleotide variation) affect a cell's response to a drug. We thus suggest a plausible assumption that similar states of the drug-induced genes across different cell lines would result in similar phenotypic drug response. While knowledge about the complete set of genes participating in each drug response remains incomplete, we may approximate this set by looking for genes whose similar state across cell lines corresponds to similar drug response of those cells.

Building on these assumptions and leveraging the plethora of genomic measurements for each cell line supplied in the NCI-DREAM drug sensitivity challenge, we exploited nearest neighbor similarities between cell lines to infer the drug response on unknown cell lines. Thus, the final scheme obtained a unique signature for each dataset comprised of genes whose similar state across cell lines matched similar GI50 values. The main novelty in this work is the integration of the different genomic datasets into a unified prediction scheme.

Methods
We used the following six data types: (i) gene expression; (ii) RNA seq; (iii) Reverse protein lysate array (RPPA); (iv) methylation; (v) gene level copy

Nature Biotechnology: doi:10.1038/nbt.2877

number variation; and (vi) exome seq. For the first five datasets, we used the raw measurements, but for the exome seq dataset, we assigned each gene a Boolean value denoting whether it had mutations or not in the corresponding cell line.

We began by computing the pairwise cell line similarity between the 35 cell lines in the training set using Euclidean distance over the GI50 drug response measurements. Missing GI50 values were imputed using *k*-nearest neighbors. We validated that pairs of cell lines with the most similar GI50 measures across all drugs were typically closest also on a single drug level.

Performing our algorithm per dataset (*i.e*. gene expression, methylation, CNV, *etc*.), we selected a signature set of genes by selecting the top 5% genes whose induced cell line pairwise had minimal pairwise similarities according to the GI50 drug response (with the exception of the exome seq, where all genes were used due to sparseness of the data). Specifically, we (i) converted the values of each cell line across the genes to *z*-scores, (ii) computed the minimal absolute difference in the expression, methylation or copy number values between each pair of cell lines on each gene individually, and (iii) assigned each gene a score reflecting the total difference in drug response GI50 values for the closest pairs of cell lines according to that gene.

Next, based on the selected gene signatures for each datset, we computed Pearson's correlation between test (unknown drug response) and train (known drug response) cell lines to determine the closest test set cell line to each training set cell line. Each of the 18 test cell lines were assigned the GI50 values of the closest cell line appearing in the training set.

In order to combine the different datasets, we tested their performance in inferring the correct GI50 values by cross-validation, where we randomly split the 35 training cell lines into training and test sets. The accuracy of each dataset relative to a random selection of the closest pairs was assessed in a minimal square error scheme and this accuracy was further used for weighting the results obtained from each dataset. The final values for each test cell line were calculated as the weighted sum of the GI50 scores obtained from individual data types that contained measurements for that cell line. Our methodology was implemented in Matlab.

Discussion

Analyzing the weights assigned for individual datasets, we observed that gene expression provided the best accuracy, while gene copy number variation provided the worst. We believe that additional data like enrichment of selected signatures in pathways could enhance the performance. Furthermore, knowledge of the anonymized drugs tested in this challenge could help by including their targets or possible gene expression response signatures from outside, available datasets (e.g., the Connectivity Map[95]).

Nature Biotechnology: doi:10.1038/nbt.2877

Other 4

Summary
Features were selected using dataset specific criteria, missing values were imputed, predictions were made using KNN.

Introduction
Integrating diverse sources of data for the prediction of drug response of cell lines is a difficult problem. In this challenge, the aim was to integrate various genomic datasets, including copy number variation, expression profiles, methylation, RNA seq, RPPA and Exome sequencing, from 53 cell lines to predict their response to a variety of drugs. Given that most of the datasets had many missing values, a key challenge was to impute the data for feature extraction and prediction. Our main idea was to organize genomic features and drug sensitivities in a matrix and use the matrix completion algorithm to predict the missing values in the matrix. The hope is that similar genomic features would be associated with drug sensitivities. In this work, we use the K-nearest neighbor (KNN) method to impute the missing values across all datasets. Finally we use these features to predict drug sensitivities by KNN.

Methods
Feature Extraction: Since the datasets have high dimensions, we first extracted features by performing dimensionality reduction.

1. For each of the gene expression, methylation and RNAseq datasets, we selected ~1500 of the most variable genes according to their expression levels and then performed a fuzzy k-means clustering algorithm and obtained three clusters on all available samples. The clustering results were highly concordant with the subtypes of breast cancer. Here we got 3*3=9 features for each sample.
2. We performed a similar clustering on the RPPA dataset with all observed protein abundances.
3. For the CNV dataset, we selected ~1500 of the most variable genes and calculated a genomic instability index over these genes for each cell line. The genomic instability index is calculated as the percentage of highly unstable genes (absolute values greater than 1).
4. We calculated a mutation index as the number of SNPs from the Exome sequence dataset. We also calculated a cancer-related mutation index as the number of cancer-related SNPs.

Feature Imputation: We created a feature matrix (53 cell lines by 15 features) to organize the features calculated in the previous steps for all cell lines. Many features are missing in this matrix. To predict the genomic features, we then ran a KNN algorithm with k=5 to complete the feature matrix.

Prediction: Given the imputed features, there are many possible ways to predict drug sensitivity. By stacking the drug sensitivity matrix (53 cell lines by 31 drugs,

Nature Biotechnology: doi:10.1038/nbt.2877

missing values) with the imputed feature matrix, we used KNN with *k*=5 to predict the missing values in the drug sensitivity matrix.

Discussion
After we finished the imputation, we found that the sensitivity of each drug is highly correlated with specific genomic features or cancer subtypes. Thus, our future work will be to select the important features for each drug and build drug-specific predictors with linear regression. Furthermore, we could use other available information, such as the structure and chemical similarity of the drugs, to improve prediction.

Other 5

Summary
Features were filtered using dataset specific criteria, an ensemble of Cox regression models were constructed using random sampling from top performing features, final predictions is the average of all models

Introduction
Drug sensitivity in the NCI-DREAM challenge is measured by GI50, which is derived from a nonlinear curve that describes the relationship between drug concentration and survival of cells tested. The GI50 measure is probably inherently nonlinear, with respect to genomic and proteomic features in the data. In the first sub-challenge of NCI-DREAM, participants are requests to predict the order of drug sensitivity, rather than specific GI50 values. Regression methods that aim to predict specific values are likely over-fit the training data. Methods that explicitly model the ordering may work better. Therefore, we chose to use Cox regression in survival analysis, because GI50 is conceptually related to survival, and Cox regression is more suitable for prediction of rank order.

Methods
The method we used contains several steps and is described as follows:
Feature filtering and selection: Exclude features that appear to not contain predictive power.

- **RNA seq**: RNA seq calls were used to identify genes that are never expressed in the training samples. RNA seq data was used to identify genes whose RPKM exceeds 10 in less than 10 samples. The RNA seq data for those genes were excluded. The RPKM data for the remaining genes were then transformed to log-scale, and normalized to 0-mean-1-var for each gene.
- **Methylation**: Methylation features were excluded if they always indicated unmethylated (<0.4) or always indicated methylated (>0.5). Beta values were then normalized to 0-mean-1-var for each methylation feature.
- **Copy number**: For copy number data at the gene level, we excluded genes whose copy numbers have many NaN entries (>10). Features for copy numbers at cbs segments level were all excluded. Gene-level copy number data were then normalized to 0-mean-1-var.

Nature Biotechnology: doi:10.1038/nbt.2877

- **Mutation:** Mutation data were collapsed to gene level, counting how many mutations existed in each gene for each cell line.
- **Others:** No features in other platforms were excluded at this stage.

Next, we identified predictive features within the genomics datasets. For each drug, we performed univariate Cox regression using its GI50 and each of the remaining features from the filtering step, *p*-values were assigned to each feature, and features were rank ordered. Next, we computed the mutual information between GI50 and each feature, then rank ordered all features according to the mutual information values. We selected the top 100 overlapping features between the two lists.

Build an ensemble of predictors: From the top 100 overlapping features, randomly select a subset, and use multivariate Cox regression to build prediction models. The models are used to predict the "risk" of each cell line to respond to the drug. The order of the risks should be consistent to the order of GI50 values.

The reason for using multiple predictors is because of the missing data. If we have a Cox regression model that uses many features and some of the features are NaN's for one testing cell line, the Cox model will not produce a risk value that is comparable to the risks for other testing cell lines. For all possible arrangement of NaNs, we need at least one regression model.

Final rank order: Each model in the above ensemble provides a partial ranking order of a subset of the cell lines (due to NaN entries). We summarize those into a square matrix, where the (*i,j*) element is the proportion of models that indicate $risk_i > risk_j$ minus the proportion of models that indicate $risk_i < risk_j$. Ideally, if all models in the ensemble perfectly agree with each other, this square matrix should be composed of ones and negative ones. In practice, elements of this square matrix are between -1 and 1. If we reshuffle the rows and columns to the correct order, we should see that the upper right triangle contains mostly positive values, and the lower left triangle contains mostly negative values. Therefore, we solve an optimization problem: maximizing (the sum of upper right triangle minus the sum of lower left triangle) by reshuffling the order of the cell lines. The ordering from this optimization problem is the final ordering we report.

Note: the GI50 values for some of the training samples can also be NaN. Instead of placing the GI50-NaN training samples at the end of the final list, we placed them with respect to the training samples with known GI50 values. In other words, in addition to ordering the testing samples, we also order the training samples with unknown GI50.

Discussion

The novelty of our approach is to view drug response data as survival time without censoring, so that we can apply survival analysis to predict "risk" of drug

Nature Biotechnology: doi:10.1038/nbt.2877

sensitivity. From our analysis, gene expression (microarray and RNAseq), copy number and methylation are the most informative platforms.

Other 6

Summary
Features were selected using the concordance index, predictions were made using an integrated voting strategy based on each feature's ability to predict the order of pairs of cell lines.

Introduction
Given a set of six genomics profiling datasets, our task was to learn the patterns or rules from 35 training cell lines to predict the drug response for 18 test cell lines. More formally, we would like to learn a ranking function f from a suitable function class F, such that $f(x_i) > f(x_j)$ implies that the drug response of cell line i is larger than the sample j.

Methods
We developed a two-step procedure to rank cell lines according to their predicted drug response. This method is a rank correlation based data integration method and the procedure is described as follows.

Feature Selection: We note that the genomics datasets are heterogeneous and in high dimension; therefore, we propose a simple and efficient method to choose predictive features for drug response. If a genomic feature is predictive to drug response, it should correlate well with the drug response profile across all cell lines. To simplify the problem, we focused on the rank of genomic feature and drug responses instead of their measured values. Secondly, we considered the pairwise rank of cell lines by genomic feature and drug response. We then defined a concordance index to determine if a genomic feature is predictive of drug response. Suppose there are N cell lines, thus $N(N-1)/2$ pairwise relationships. For each pair of cell lines (A, B), we first compared the drug response GI50 values for A and B to derive an order. Next, we compared the values of a genomic feature for A and B to derive a second order. If the two orders were concordant, the comparison was assigned the value 1, otherwise zero. The sum of all possible pairs is denoted as S. The concordance index is defined as: $CI=S/(N(N-1)/2)$. Features were selected according to a genomic dataset specific threshold and each predictive feature was weighted by the concordant index.

Voting methodology: We adopt a simple majority voting and integrative strategy as shown in **Figure O6**. For each feature, its concordance with drug response allows to predict a ranking between two cell lines. A set of rankings is compiled for all predictive features within a genomics dataset. The ranking procedure was done for all pairwise cell line comparisons. The final step is to assembly the overall ranking of cell lines from the pairwise cell line rankings. We design a

Nature Biotechnology: doi:10.1038/nbt.2877

majority voting algorithm to achieve this object. To break the possible ties during ranking, we adopt some heuristic algorithms.

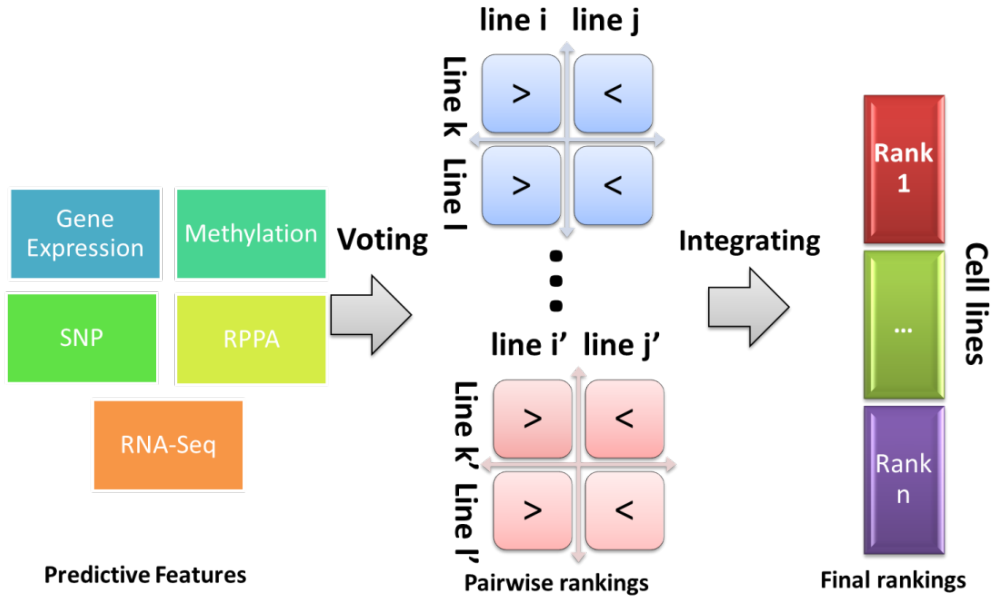


Figure O6. Schematic representation of the Other 6 method.

Discussion

We developed a rank correlation based data integration method to predict the sensitivity of cancer cell lines to drugs. We considered the rank of genomic features, which was more reliable than their measured values. Both voting and ranking approaches are simple and can be easily interpreted, thus increasing the biological interpretability of features underlying final predictions.

Supplementary Note 2: Supplemental Scoring Analysis

The NCI-DREAM challenge was to rank order cell lines from most to least sensitive for each of the 28 tested drugs. In a post-challenge analysis, we explored the possibility of scoring teams based on their ability to predict the classes of sensitive and resistant cell lines. In developing this new scoring scheme, we first clustered the cell lines associated with each drug into 3 classes: sensitive, resistant, and intermediate. The intermediate class captures cell lines that do not reliably cluster into the sensitive and resistant classes, but fall in between. The results of this mapping can be found for each team in **Supplemental Fig. 6**. Next, we mapped the sensitive, resistant, and intermediate labels onto a team's predictions and scored them based on balanced accuracy. Finally, we compared this measure to the wpc-index described in the main text. The measures were highly correlated, $\rho = 0.78$, and the results comparing all 44 teams can be found in **Supplementary Table 4**.

Nature Biotechnology: doi:10.1038/nbt.2877

We used the following approach to identify sensitive and resistant sub-populations of cell lines for each compound tested. For each vector of $-\log_{10}(GI_{50})$ values, we used Partitioning Around Medoids (PAM) clustering to identify three groups, which we interpret to be sensitive, intermediate, and resistant cell lines. In the case of compounds where cell lines did not reach GI50, we ensured that these clusters were not artificially influenced by the maximum concentration tested. PAM was implemented in R with the cluster package (version 1.14.2). Results of the sensitive, intermediate, and resistant classification for all cell lines and drugs can be found in **Supplementary Table 10.**

For scoring, we first mapped the sensitive and resistant calls to a team's ranked list of predictions. Next, we counted true positive (TP) and true negatives (TN), where sensitive cell lines were considered the positives and resistant cell lines were considered negatives. Within the test dataset, there are 105 cases of cell lines being sensitive to the 28 drugs (Total Positives) and 137 cases of cell lines being resistant to the 28 drugs (Total Negatives). For each drug, d, we have P_d positives defined by the gold standard. For each team, we counted the number of true positives TP_d that were in the top P_d positions of the team's submitted ranked list for d. Next, the overall number of TPs was computed as: TP = $\sum_{d=1}^{n} \mathrm{TP}_d$, where $n = 28$ drugs. The same approach was used to calculate the number of TNs. The balanced accuracy is then the average between the TP rate and the TN rate: balanced accuracy = $(\frac{1}{2}\left(\frac{\mathrm{TP}}{\text{Total Positives}} + \frac{\mathrm{TN}}{\text{Total Negatives}}\right))$.

Supplementary Note 3: Weighted probabilistic c-index (*wpc*-index)

For a given drug, d, the *c*-index between a predicted ranked list of cell lines (n =18 for the set of test cell lines), $R_d = \{r_1, r_2, \ldots, r_n\}$, where r_i is the rank order of cell line i, and the gold standard list of dose response values for the same n cell lines, $G_d = \{g_1, g_2, \ldots, g_n\}$, where g_i is the mean across replicate measurements of $-\log_{10}(GI_{50})$ values for cell line i. This is a non-standard formulation of the c-index; for example, in this implementation, r_i = 1, r_j = 2, g_i = 5, and g_j =4.5, represents a concordant comparison. The *c*-index is calculated as:

c-index $= c(G_d, R_d) = \frac{2}{n(n-1)} \sum_{i<j} h\left(g_i, g_j, r_i, r_j\right)$,

where

$$h\left(g_i, g_j, r_i, r_j\right) = \begin{cases} 1, & \text{if } \left(g_i > g_j \,\&\, r_i < r_j\right) \vee \left(g_i < g_j \,\&\, r_i > r_j\right) \\ 0.5, & \text{if } \left(g_i = g_j\right) \\ 0, & \text{if } \left(g_i > g_j \,\&\, r_i > r_j\right) \vee \left(g_i < g_j \,\&\, r_i < r_j\right) \end{cases}$$

Nature Biotechnology: doi:10.1038/nbt.2877

The *c*-index does not account for variance within the gold standard dataset; therefore, we modified the *c*-index to account for this variance. The probabilistic *c*-index (*pc*-index) is calculated as:

pc-index= $pc(G_d, R_d, s_d^2) = \frac{2}{n(n-1)} \sum_{i<j} hp\left(g_i, g_j, r_i, r_j, \sqrt{s_d^2}\right)$,

where

$$hp\left(g_i, g_j, r_i, r_j, s_d\right) = \begin{cases} \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{g_i - g_j}{2s_d}\right)\right), & \text{if } (r_i < r_j) \\ 0.5, & \text{if } (r_i = r_j) \\ \frac{1}{2}\left(1 + \mathrm{erf}\left(\frac{g_j - g_i}{2s_d}\right)\right), & \text{if } (r_i > r_j) \end{cases}$$

and

$$\mathrm{erf}(a) = \frac{2}{\sqrt{\pi}} \int_0^a e^{-t^2}\, \mathrm{d}t.$$

In this formulation of the *pc*-index, if the predicted rank of two cell lines, $\langle r_i, r_j \rangle$, is concordant with the gold standard, $\langle g_i, g_j \rangle$ (in our case $\left(g_i > g_j \,\&\, r_i < r_j\right) \vee \left(g_i < g_j \,\&\, r_i > r_j\right)$, the function $hp(\ldots)$ returns a value in the range (0.5,1] based on the error function. Keep in mind that R_d represents rank order and G_d are -$\log_{10}(GI_{50})$ values. We assume the variance measured in dose response follows a Gaussian distribution. Conversely, if a predicted rank of two cell lines, $\langle r_i, r_j \rangle$, is disconcordant with the gold standard, $\langle g_i, g_j \rangle$ (in our case $\left(g_i > g_j \,\&\, r_i > r_j\right) \vee \left(g_i < g_j \,\&\, r_i < r_j\right)$), the function $hp(\ldots)$ returns a value in the range [0,.5).

For a given team, the *pc*-index was calculated separately for each drug, d. The final team score was calculated as the weighted average of the *pc*-index scores across the evaluated drugs (*wpc*-index).

wpc-index= $S = \frac{\sum_d w_d \cdot pc_d}{\sum_d w_d}$

Each drug, d, has a different measured variance of dose responses, s_d^2, across the n cell lines, and a different number of missing values, thus our ability to calculate a reliable *pc*-index varied between drugs. To account for this, weights for each drug, w_d, were calculated. For each drug d a random ranking of n items was made, R_d^r, and the *pc*-index was calculated, $pc(G_d, R_d^r, s_d^2)$. This procedure was repeated 10,000 times to create an empirical null distribution which followed a Gaussian distribution with median and standard deviation (μ_d, σ_d). The gold standard dataset was converted to a ranked list of cell lines, R_d^*, based on the mean across replicate measurements of the $-\log_{10}(GI_{50})$ values of those cell lines, and the *pc*-index was calculated, $pc_d = pc(G_d, R_d^*, s_d^2)$. The drug weight, w_d, was calculated as the z-score of the gold standard ranking, R_d^*, compared to the null distribution (μ_d, σ_d), $w_d = \frac{pc_d - \mu_d}{\sigma_d}$. The maximum of the wpc-index will not be 1

Nature Biotechnology: doi:10.1038/nbt.2877

due to experimental variation in the dose response measurements. In order to make some results interpretable in the range [0,1] we also calculated a scaled version of the wpc-index which maps the wpc values to the range [0,1], using the transformation

scaled *wpc*-index = $\frac{wpc_i - wpc_{max}}{wpc_{max} - wpc_{min}}$

where the *wpc*-index score for team i is scaled to the max *wpc*-index wpc_{max}, which is computed using the gold standard ranking for each drug, and the min *wpc*-index, which is computed using the inverse of the gold standard ranking for each drug.

Supplementary Note 4: NCI-DREAM Challenge Criteria

Drug inclusion/exclusion criteria

The NCI-DREAM drug response datasets were selected based on data availability and novelty. Regarding data availability, on average, drugs were tested on 80% of the 53 cell lines. Regarding novelty, to provide an unbiased assessment of team predictions, we required that a drug's response data be unpublished, not distributed throughout the community of participants, and not available from other sources (e.g., the CCLE). Most of the drugs included in this data set are experimental compounds that have not been tested clinically in breast cancer, and therefore have the potential to serve as novel therapeutics.

The NCI-DREAM data as presented to participants contained 31 drugs; however, 3 of these drugs had completely flat profiles (i.e., the GI_{50} for all cell lines were the same) and were thus unscorable. Since the wpc-index uses a weighting scheme to summarize team performance over all 31 drugs, the weights of these drugs were 0, thus having no influence on a team's score. The intention of keeping these drugs was to explore team predictions and determine if any insights could be gained on the response of these 3 drugs; however, we were not able to glean any additional information from team predictions. Therefore, to avoid confusion, we have excluded these 3 drugs from all analyses presented in the manuscript. The original challenge data can be found at: http://www.the-dream-project.org/challenges/nci-dream-drug-sensitivity-prediction-challenge

NCI-DREAM community participation

All participating teams were contacted directly with the criteria necessary to be a member of the NCI-DREAM community and listed as contributors on this manuscript. To be listed as a community member, teams were required to submit a detailed write-up of their submitted methodology, which the DREAM organizers reviewed, edited, and compiled. This set of method write-ups comprises the supplementary methods. Teams were given the option to opt out

Nature Biotechnology: doi:10.1038/nbt.2877

of being an NCI-DREAM community member, and as such, these individuals were not included in the authors list and their full method descriptions were not included in these supplementary methods. In an effort to report the most comprehensive analysis possible, we have included the full set of teams for all analyses included in the main text. In total, there were 44 teams that submitted predictions with short descriptions as listed in **Table 1**. Of these teams, 38 provided method write-ups and comprise the NCI-DREAM community.

References

1. Gönen, M. & Alpaydin, E. Multiple Kernel Learning Algorithms. *J Mach Learn Res* **12**, 2211-2268 (2011).
2. Gönen, M. in International Converence on Machine Learning (ICML 2012) (Edinburgh, Scotland, UK; 2012).
3. Beal, M.J. in The Gatsby Computational Neuroscience Unit (Univversity College London, London, UK; 2003).
4. Liberzon, A. et al. Molecular signatures database (MSigDB) 3.0. *Bioinformatics* **27**, 1739-1740 (2011).
5. Vaske, C.J. et al. Inference of patient-specific pathway activities from multi-dimensional cancer genomics data using PARADIGM. *Bioinformatics* **26**, i237-245 (2010).
6. Cortes, C. & Vapnik, V. Support-vector machines. *Mach Learn* **20**, 273-297 (1995).
7. Chang, C.C. & Lin, C.J. LIBSVM: a library for support vector machines. *ACM Trans on Intell Sys Technol* **2**, 1-27 (2011).
8. Breiman, L. Bagging predictors. *Mach Learn* **24**, 123-140 (1996).
9. Saeys, Y., Inza, I. & Larranaga, P. A review of feature selection techniques in bioinformatics. *Bioinformatics* **23**, 2507-2517 (2007).
10. Hijazi, H., Wu, M., Nath, A. & Chan, C. Ensemble classification of cancer types and biomarker identification. *Drug Dev Res* **73**, 414-419 (2012).
11. Breiman, L. Random Forests. *Mach Learn* **45**, 5-32 (2001).
12. Ding, Z. in Computer Science, Vol. Masters (West Virginia University, 2008).
13. Heiser, L.M. et al. Subtype and pathway specific responses to anticancer compounds in breast cancer. *Proc Natl Acad Sci USA* **109**, 2724-2729 (2012).
14. Pounds, S. & Morris, S.W. Estimating the occurrence of false positives and false negatives in microarray studies by approximating and partitioning the empirical distribution of p-values. *Bioinformatics* **19**, 1236-1242 (2003).
15. Barretina, J. et al. The Cancer Cell Line Encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature* **483**, 603-607 (2012).
16. Garnett, M.J. et al. Systematic identification of genomic markers of drug sensitivity in cancer cells. *Nature* **483**, 570-575 (2012).

Nature Biotechnology: doi:10.1038/nbt.2877

17. Greenblum, S.I., Efroni, S., Schaefer, C.F. & Buetow, K.H. The PathOlogist: an automated tool for pathway-centric analysis. *BMC Bioinformatics* **12**, 133 (2011).
18. Reshef, D.N. et al. Detecting novel associations in large data sets. *Science* **334**, 1518-1524 (2011).
19. Marbach, D. et al. Wisdom of crowds for robust gene network inference. *Nature Methods* **9**, 796-804 (2012).
20. Jacob, L. & Vert, J.P. Protein-ligand interaction prediction: an improved chemogenomics approach. *Bioinformatics* **24**, 2149-2156 (2008).
21. Waegeman, W. et al. A kernel-based framework for learning graded relations from data. *IEEE Trans Fuzzy Sys* **99**, 1 (20012).
22. Pahikkala, T., Airola, A., Stock, M., De Baets, B. & Waegeman, W. Efficient regularized least-squares algorithms for conditional ranking on relational data. *arXiv.org:1209.4825* (2013).
23. Stekhoven, D.J. & Buhlmann, P. MissForest--non-parametric missing value imputation for mixed-type data. *Bioinformatics* **28**, 112-118 (2012).
24. Di Camillo, B. et al. Effect of size and heterogeneity of samples on biomarker discovery: synthetic and real data assessment. *PloS One* **7**, e32200 (2012).
25. Shigemizu, D. et al. Using functional signatures to identify repositioned drugs for breast, myelogenous leukemia and prostate cancer. *PLoS Comput Biol* **8**, e1002347 (2012).
26. Sirota, M. et al. Discovery and preclinical validation of drug indications using compendia of public gene expression data. *Sci Transl Med* **3**, 96ra77 (2011).
27. Torkamani, A. & Schork, N.J. Background gene expression networks significantly enhance drug response prediction by transcriptional profiling. *Pharmacogenomics J* **12**, 446-452 (2012).
28. Walters, R., Laurin, C. & Lubke, G.H. An integrated approach to reduce the impact of minor allele frequency and linkage disequilibrium on variable importance measures for genome-wide data. *Bioinformatics* **28**, 2615-2623 (2012).
29. Mahoney, M.W. & Drineas, P. CUR matrix decompositions for improved data analysis. *Proc Natl Acad Sci USA* **106**, 697-702 (2009).
30. van Westen, G.J.P., Wegner, J.K., Ijzerman, A.P., Van Vlijmen, H.W.T. & Bender, A. Proteochemometric Modeling as a Tool for Designing Selective Compounds and Extrapolating to Novel Targets. *Med Chem Commun* **2**, 16-30 (2011).
31. Van Westen, G.J.P. et al. Significantly Improved HIV Inhibitor Efficacy Prediction Employing Proteochemometric Models Generated From Antivirogram Data. *PLoS Comput Biol*, In Press (2013).
32. Accelrys Software Inc, Edn. 8.5 (Scitegic).
33. Staunton, J.E. et al. Chemosensitivity prediction by transcriptional profiling. *Proc Natl Acad Sci USA* **98**, 10787-10792 (2001).

Nature Biotechnology: doi:10.1038/nbt.2877

34. Bi, J., Bennett, K., Embrechts, M., Breneman, C. & Song, M. Dimensionality reduction via sparse support vector machines. *J Mach Learn Res* **3**, 1229-1243 (2003).
35. Blei, D.M., Griffiths, T.L. & Jordan, M.I. The nested Chinese restaurant process and Bayesian nonparametric inference of topic hierarchies. *J ACM* **57**, 7 (2010).
36. Teh, T.W., Jordan, M.I., Beal, M.J. & Blei, D.M. Hierachical Dirichlet processes. *J Amer Stat Assoc* **101** (2006).
37. Zou, H. & Hastie, T. Regularization and variable selection via Elastic Net. *J R Stat Soc B* **67**, 301-320 (2005).
38. Friedman, J., Hastie, T. & Tibshirani, R. Regularization Paths for Generalized Linear Models via Coordinate Descent. *J Stat Software* **33**, 1-22 (2010).
39. Acuna, E. & Rodriguez, C. The treatment of missing values and its effect on classifier accuracy. *Classification, Clustering, and Data Mining Applications*, 639-647 (2004).
40. Draper, N.R. & Smith, H. Applied Regression Analysis. (John Wiley & Sons, Inc., New York; 1998).
41. Tibshirani, R. The lasso method for variable selection in the Cox model. *Statistics in medicine* **16**, 385-395 (1997).
42. Zou, H. & Zhang, H.H. On the Adaptive Elastic-Net with a Diverging Number of Parameters. *Annals of Statistics* **37**, 1733-1751 (2009).
43. Kuhn, M. Building predictive models in R using the caret package. *J Stat Software* **28**, 1-26 (2008).
44. Meinshausen, N. Relaxed Lasso. *Computational Statistics & Data Analysis* **52**, 374-393 (2007).
45. Fan, J. & Lv, J. A selective overview of variable selection in high dimensional feature space. *Stat Sin* **20**, 101-148 (2010).
46. Fan, J. & Li, R. Variable selection via nonconcave penalized likelihood and its oracle properties. *J Amer Stat Assoc* **96**, 1348-1360 (2001).
47. Tibshirani, R. Regression shrinkage and selection via the lasso. *J R Statist Soc B* **58**, 267-288 (1996).
48. Chen, J. & Chen, Z. Extended Bayesian information criteria for model selection with large model spaces. *Biometrica* **95**, 759-771 (2008).
49. Sun, W., Ibrahim, J.G. & Zou, F. Genomewide multiple-loci mapping in experimental crosses by iterative adaptive penalized regression. *Genetics* **185**, 349-359 (2010).
50. Zou, H. & Li, R. One-step Sparse Estimates in Nonconcave Penalized Likelihood Models. *Annals of Statistics* **36**, 1509-1533 (2008).
51. Sun, W. & Li, L. Multiple loci mapping via model-free variable selection. *Biometrics* **68**, 12-22 (2012).
52. Kim, S.-J., Koh, K., Lustig, M., Boyd, S. & BGorinevsky, D. An interior-point method for large-scale L1-regularized least squares. *IEEE J Selected Topics in Signal Processing* **1**, 606-617 (2007).

Nature Biotechnology: doi:10.1038/nbt.2877

53. Beroukhim, R. et al. Assessing the significance of chromosomal aberrations in cancer: methodology and application to glioma. *Proc Natl Acad Sci USA* **104**, 20007-20012 (2007).
54. Liu, J.C. et al. Seventeen-gene signature from enriched Her2/Neu mammary tumor-initiating cells predicts clinical outcome for human HER2+:ERalpha- breast cancer. *Proc Natl Acad Sci USA* **109**, 5832-5837 (2012).
55. Navab, R. et al. Prognostic gene-expression signature of carcinoma-associated fibroblasts in non-small cell lung cancer. *Proc Natl Acad Sci USA* **108**, 7160-7165 (2011).
56. Shi, L. et al. The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nature Biotechnol* **28**, 827-838 (2010).
57. Shi, L. et al. The MicroArray Quality Control (MAQC)-II study of common practices for the development and validation of microarray-based predictive models. *Nat Biotechnol* **28**, 827-838 (2010).
58. Zhan, F. et al. The molecular classification of multiple myeloma. *Blood* **108**, 2020-2028 (2006).
59. Shaughnessy, J.D., Jr. et al. A validated gene expression model of high-risk multiple myeloma is defined by deregulated expression of genes mapping to chromosome 1. *Blood* **109**, 2276-2284 (2007).
60. Zhan, F., Barlogie, B., Mulligan, G., Shaughnessy, J.D., Jr. & Bryant, B. High-risk myeloma: a gene expression based risk-stratification model for newly diagnosed multiple myeloma treated with high-dose therapy is predictive of outcome in relapsed disease treated with single-agent bortezomib or high-dose dexamethasone. *Blood* **111**, 968-969 (2008).
61. Decaux, O. et al. Prediction of survival in multiple myeloma based on gene expression profiles reveals cell cycle and chromosomal instability signatures in high-risk patients and hyperdiploid signatures in low-risk patients: a study of the Intergroupe Francophone du Myelome. *J Clin Oncol* **26**, 4798-4805 (2008).
62. Mulligan, G. et al. Gene expression profiling and correlation with outcome in clinical trials of the proteasome inhibitor bortezomib. *Blood* **109**, 3177-3188 (2007).
63. Wold, S., Sjostrom, M. & Eriksson, L. PLS-regression: a basic tool of chemometrics. *Chemometrics and Intelligent Laboratory Systems* **58**, 109-130 (2001).
64. Janes, K.A. et al. The response of human epithelial cells to TNF involves an inducible autocrine cascade. *Cell* **124**, 1225-1239 (2006).
65. Lindgren, F., Geladi, P. & Wold, S. The kernel algorithm for PLS. *J Chemometrics* **7**, 45-59 (1993).
66. Demmel, J. & Hahan, W. Accurate singular values of bidiagonal matrices. Society for Industrial and Applied Mathematics. *J Sci Statist Comput* **11**, 873-912 (1990).

Nature Biotechnology: doi:10.1038/nbt.2877

67. Eng, K.H., Wang, S., Bradley, W.H., Rader, J.S. & Kendziorski, C. Pathway index models for construction of patient-specific risk profiles. *Stat Med* **32**, 1524-1535 (2013).
68. Norel, R., Rice, J.J. & Stolovitzky, G. The self-assessment trap: can we all be better than average? *Mol Sys Biol* **7**, 537 (2011).
69. Smith, S.C., Baras, A.S., Lee, J.K. & Theodorescu D. The COXEN principle: translating signatures of in vitro chemosensitivity into tools for clinical outcome prediction and drug discovery in cancer. *Cancer Res* **70**, 1753-1758 (2010).
70. Williams, P.D. et al. Concordant gene expression signatures predict clinical outcomes of cancer patients undergoing systemic therapy. *Cancer Res* **69**, 8302-8309 (2009).
71. Williams, P.D., Lee, J.K. & Theodorescu, D. Genomancy: predicting tumour response to cancer therapy based on the oracle of genetics. *Curr Oncol* **16**, 56-58 (2009).
72. Lee, J.K. et al. Prospective comparison of clinical and genomic multivariate predictors of response to neoadjuvant chemotherapy in breast cancer. *Clin Cancer Res* **16**, 711-718 (2009).
73. Lee, J.K. et al. A strategy for predicting the chemosensitivity of human cancers and its application to drug discovery. *Proc Natl Acad Sci USA* **104**, 13086-13091 (2007).
74. Bhattacharjee, M. & Sillanpaa, M.J. A bayesian mixed regression based prediction of quantitative traits from molecular marker and gene expression data. *PloS One* **6**, e26959 (2011).
75. Bair, E., Hastie, T., Paul, D. & Tibshirani, R. Prediction by supervised principal components. *J Am Stat Assoc* **101**, 119-137 (2006).
76. Stone, M. Cross-validatory choice and assessment of statistical predictions. *J Am Stat Assoc* **36** (1974).
77. Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. Gene selection for cancer classification using support vector machines. *Mach Learn* **46**, 389-422 (2002).
78. Pedregosa, F. et al. Scikit-learn: Machine learning in Python. *J Mach Learn Res* **12**, 2825-2830 (2011).
79. Hong, F. et al. RankProd: a bioconductor package for detecting differentially expressed genes in meta-analysis. *Bioinformatics* **22**, 2825-2827 (2006).
80. Dennis, G., Jr. et al. DAVID: Database for Annotation, Visualization, and Integrated Discovery. *Genome Biol* **4**, P3 (2003).
81. Dawany, N.B., Dampier, W.N. & Tozeren, A. Large-scale integration of microarray data reveals genes and pathways common to multiple cancer types. *Intl J Cancer* **128**, 2881-2891 (2011).
82. Gormley, M., Dampier, W., Ertel, A., Karacali, B. & Tozeren, A. Prediction potential of candidate biomarker sets identified and validated on gene expression data from multiple datasets. *BMC Bioinformatics* **8**, 415 (2007).

Nature Biotechnology: doi:10.1038/nbt.2877

83. Zhang, Y. et al. Systematic analysis, comparison, and integration of disease based human genetic association data and mouse genetic phenotypic information. *BMC Med Genomics* **3**, 1 (2010).
84. Gaulton, A. et al. ChEMBL: a large-scale bioactivity database for drug discovery. *Nucleic Acids Res* **40**, D1100-1107 (2012).
85. Perez, M.T. & Sommaruga, R. Interactive effects of solar radiation and dissolved organic matter on bacterial activity and community structure. *Environmental Microbiology* **9**, 2200-2210 (2007).
86. Fagin, R., Kumar, R. & Sivakumar, D. Efficient similarity search and classification via rank aggregation. *ACM*, 301-312 (2003).
87. Hastie, T., Tibshirani, R. & Friedman, J.H. The elements of statistical learning: data mining, inference, and prediction. (New York: Springer-Verlag, 2001).
88. Bashashati, A. et al. DriverNet: uncovering the impact of somatic driver mutations on transcriptional networks in cancer. *Genome Biol* **13**, R124 (2012).
89. Ciriello, G., Cerami, E., Sander, C. & Schultz, N. Mutual exclusivity analysis identifies oncogenic network modules. *Genome Res* **22**, 398-406 (2012).
90. Kanehisa, M., Goto, S., Sato, Y., Furumichi, M. & Tanabe, M. KEGG for integration and interpretation of large-scale molecular data sets. *Nucleic acids Res* **40**, D109-114 (2012).
91. Schaefer, C.F. et al. PID: the Pathway Interaction Database. *Nucleic acids Res* **37**, D674-679 (2009).
93. Croft, D. et al. Reactome: a database of reactions, pathways and biological processes. *Nucleic Acids Res* **39**, D691-697 (2011).
94. Curtis, C. et al. The genomic and transcriptomic architecture of 2,000 breast tumours reveals novel subgroups. *Nature* **486**, 346-352 (2012).
95. Lamb, J. et al. The Connectivity Map: using gene-expression signatures to connect small molecules, genes, and disease. *Science* **313**, 1929-1935 (2006).

Nature Biotechnology: doi:10.1038/nbt.2877

Supplementary Figures

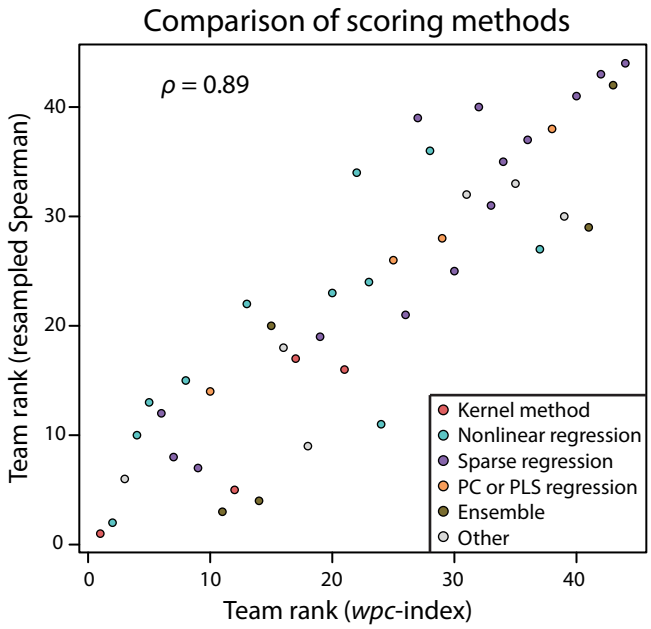
A community effort to assess and improve drug sensitivity prediction algorithms

James C Costello[1,2,13,14], Laura M Heiser[3,14], Elisabeth Georgii[4,14], Mehmet Gönen[4], Michael P Menden[5], Nicholas J Wang[3], Mukesh Bansal[6], Muhammad Ammad-ud-din[4], Petteri Hintsanen[7], Suleiman A Khan[4], John-Patrick Mpindi[7], Olli Kallioniemi[7], Antti Honkela[8], Tero Aittokallio[7], Krister Wennerberg[7], NCI DREAM Community[9], James J Collins[1,2,10], Dan Gallahan[11], Dinah Singer[11], Julio Saez-Rodriguez[5], Samuel Kaski[4,8], Joe W Gray[3] & Gustavo Stolovitzky[12]

[1]Howard Hughes Medical Institute, Boston University, Boston, Massachusetts, USA. [2]Department of Biomedical Engineering, Boston University, Boston, Massachusetts, USA. [3]Department of Biomedical Engineering, Oregon Health and Science University, Portland, Oregon, USA. [4]Helsinki Institute for Information Technology HIIT, Department of Information and Computer Science, Aalto University, Espoo, Finland. [5]European Molecular Biology Laboratory-European Bioinformatics Institute, Wellcome Trust Genome Campus, Cambridge, UK. [6]Department of Systems Biology, Center for Computational Biology and Bioinformatics, Columbia University, New York, New York, USA. [7]Institute for Molecular Medicine Finland FIMM, University of Helsinki, Helsinki, Finland. [8]Helsinki Institute for Information Technology HIIT, Department of Computer Science, University of Helsinki, Helsinki, Finland. [9]List of participants and affiliations appear at the end of the paper. [10]Wyss Institute for Biologically Inspired Engineering, Harvard University, Boston, Massachusetts, USA. [11]National Cancer Institute, National Institutes of Health, Bethesda, Maryland, USA. [12]IBM T.J. Watson Research Center, IBM, Yorktown Heights, New York, USA. [13]Present address: Department of Pharmacology, University of Colorado Anschutz Medical Campus, Aurora, Colorado, USA. [14]These authors contributed equally to this work.

Correspondence should be addressed to S.K. (samuel.kaski@aalto.fi), J.W.G. (grayjo@ohsu.edu), or G.S. (gustavo@us.ibm.com).

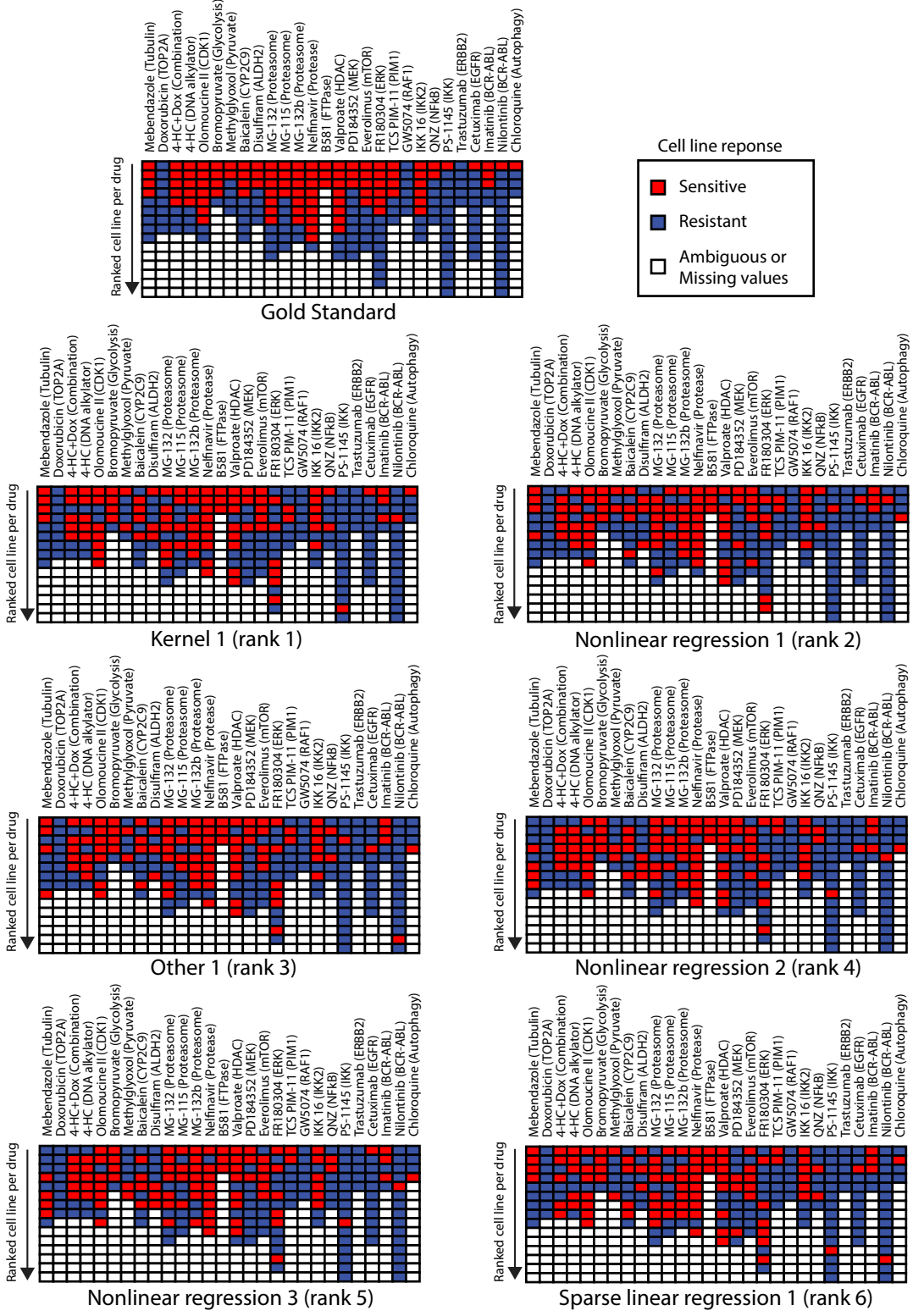
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 1. Comparison of independent team scoring methods

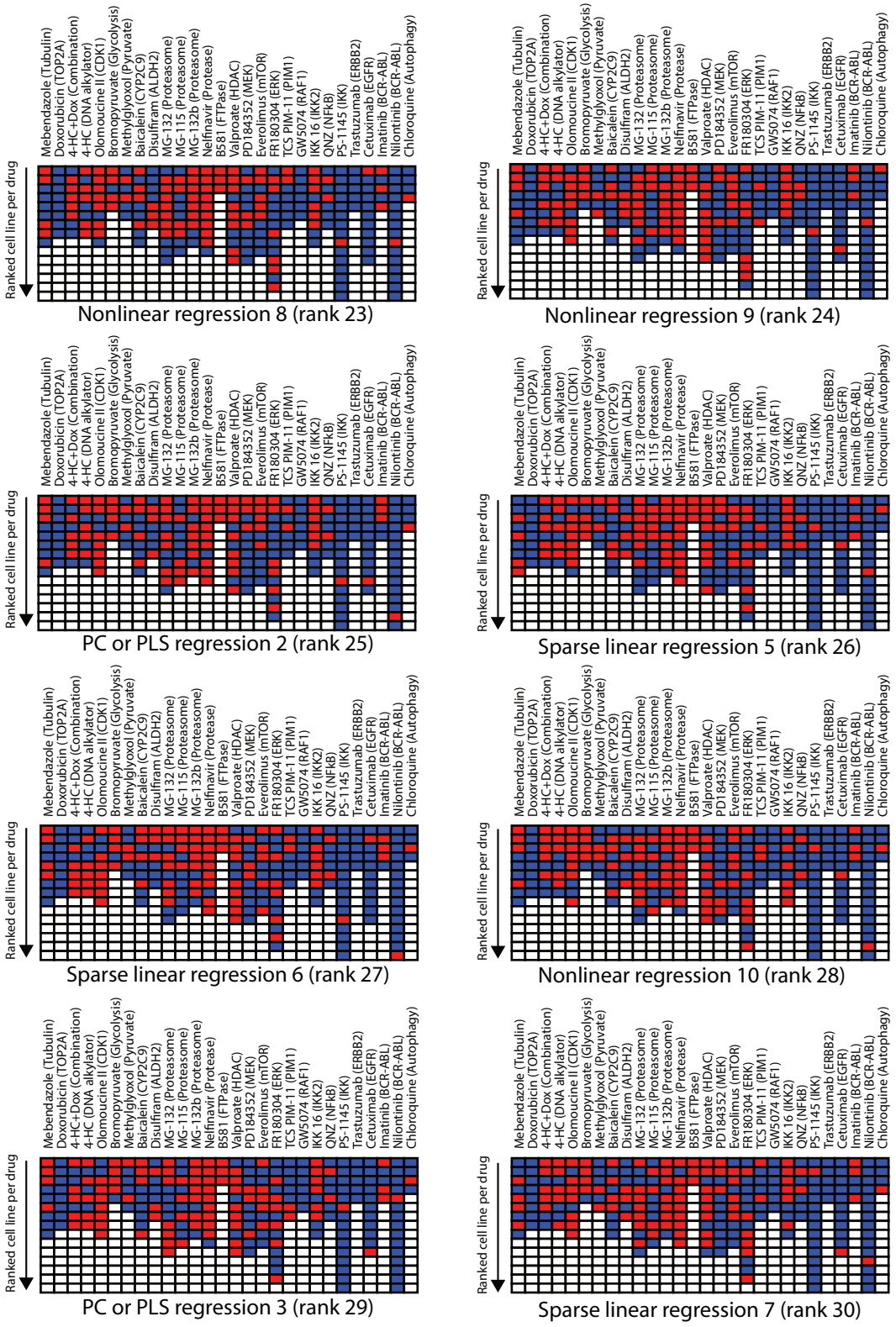
Two scoring methods, the weighted, probabilistic concordance index (*wpc*-index) and the resampled Spearman correlation, were calculated for each of the 44 submissions. Both scoring methods showed highly correlated results, with the top 2 teams being ranked first and second by both approaches. Scoring methods are described in the Online Methods.

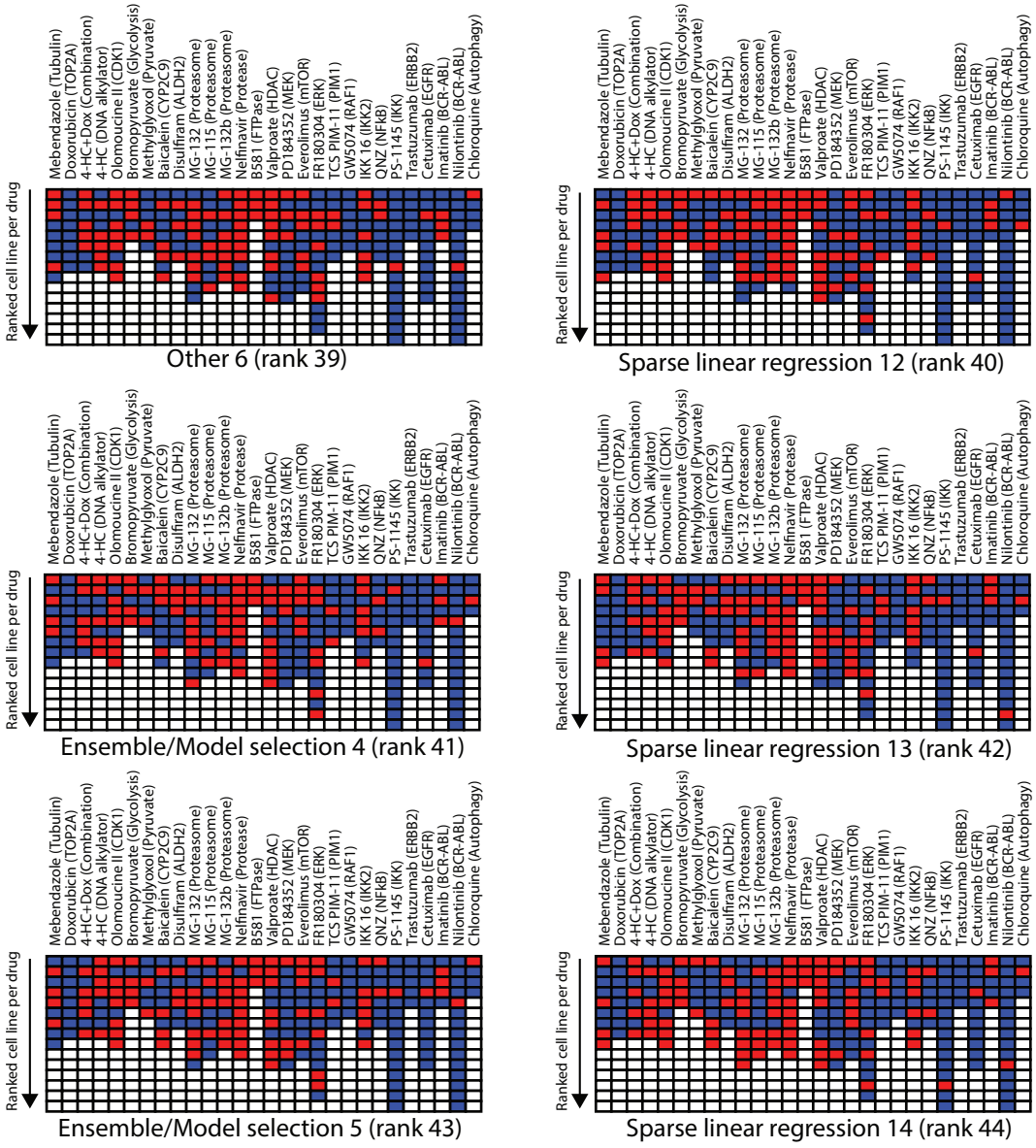
Nature Biotechnology: doi:10.1038/nbt.2877



Nature Biotechnology: doi:10.1038/nbt.2877

Nature Biotechnology: doi:10.1038/nbt.2877

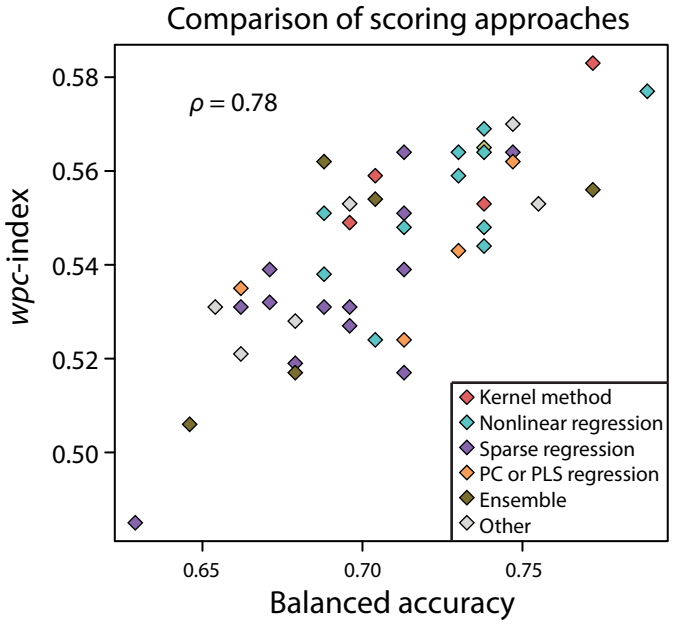




Supplementary Figure 2. Team-by-team predictions for sensitive and resistant cell lines

For each drug, cell lines were clustered into 3 classes: sensitive, ambiguous, and resistant. The ambiguous class captures cell lines that did not reliably cluster in the sensitive or resistant classes. For each team and each drug, cell lines were ordered according to the predicted rank. Cell lines were then color coded according to the sensitive (red), ambiguous (white), or resistant (blue) classes. Details of the cell line clustering can be found in the Supplemental Methods.

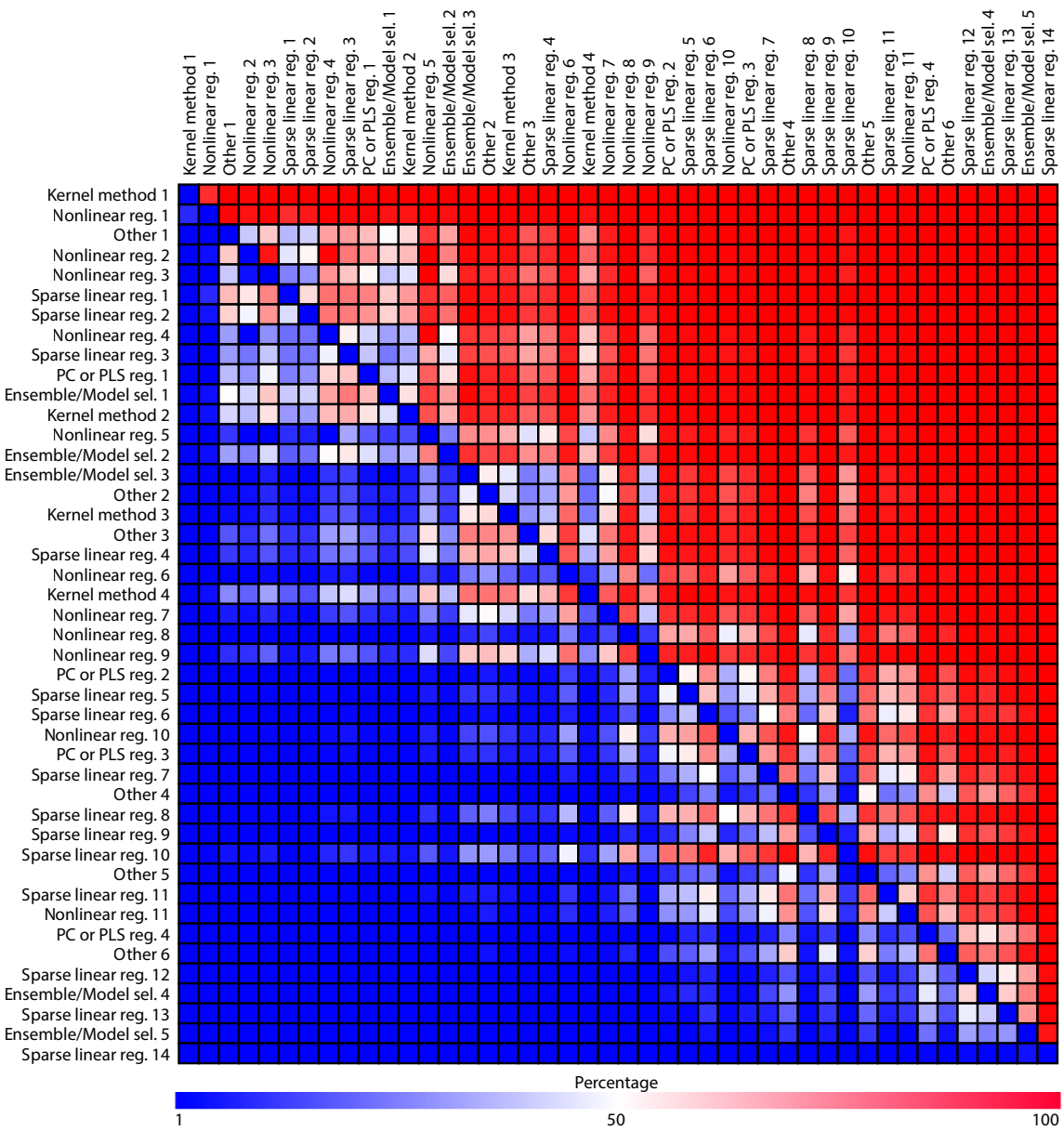
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 3. Comparison of the wpc-index and balanced accuracy scoring strategies

Two scoring schemes were compared, namely the weighted, probabilistic concordance index (*wpc*-index) and the balanced accuracy. The *wpc*-index scores a team based on the predicted rank order compared to the gold standard rank order. For the balanced accuracy, cell line responses in the gold standard were clustered and scored on two categories, sensitive and resistant. All 44 teams are presented and the color-coding corresponds to **Table 1**. Details of the *wpc*-index can be found in the Online Methods and balanced accuracy can be found in the Supplemental Methods.

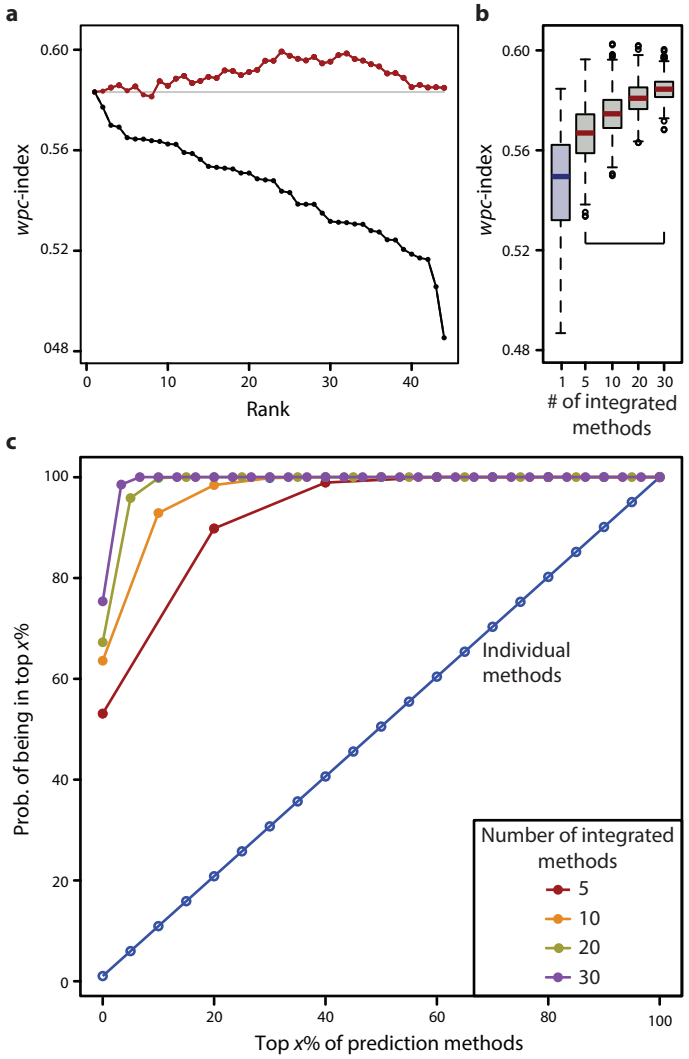
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 4. Team rank comparisons over resampled dose response

Team names correspond to those listed in **Table 1**. Teams are ordered according to their wpc-index scores that are also listed in **Table 1**. The gold standard set of test cell lines was subsampled to test the robustness of team ranks. 10% of the gold standard dataset was randomly masked, then teams were rescored and ranked according to the wpc-index. A total of 10,000 iterations were run and the colors reflect the percentage of times team *i* (row) outranked team *j* (column).

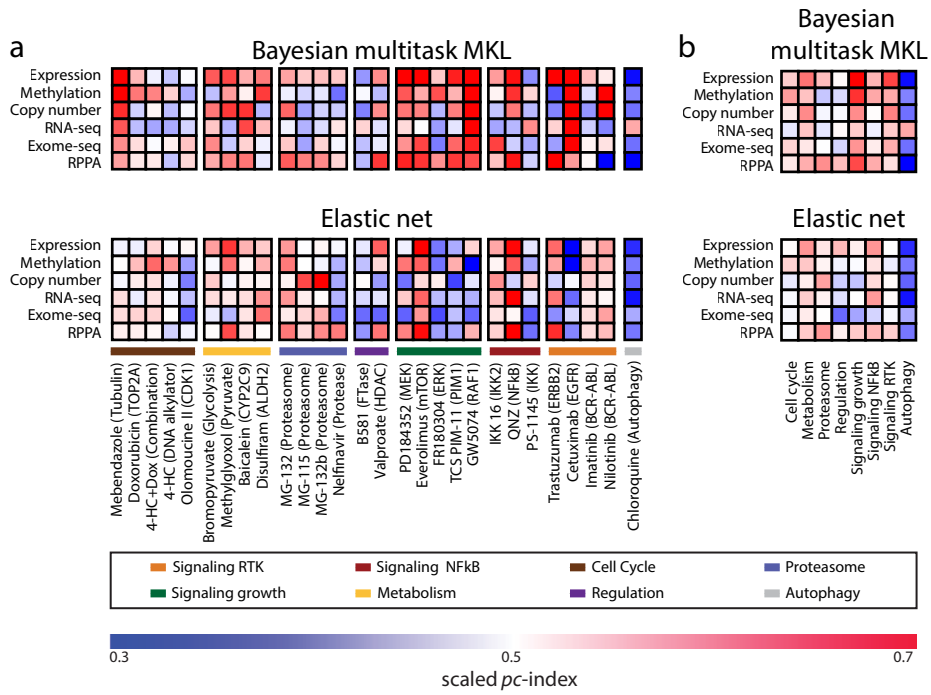
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 5. Wisdom of the crowd analysis: Integrated team submissions provide robust predictions

(**a**) Teams were ranked according to their weighted, probabilistic concordance index (*wpc*-index; black line). Integrated predictions were calculated by taking the average rank prediction of groups of predictions. The performance of the top two teams integrated is shown as the second ordered red point, the top three teams integrated as the third ordered red point, *etc*. (**b**) Random groups of $n = 5$, 10, 20, and 30 teams were integrated and scored. The boxplots represent the distribution of 1,000 random groupings. (**c**) As in (**b**), random team groupings were scored, and then the integrated group prediction was compared to the performance of the constituent members of the same group. Over 1,000 iterations, the points represent the probability of the integrated prediction being ranked first (left most point), second (next point), *etc.*

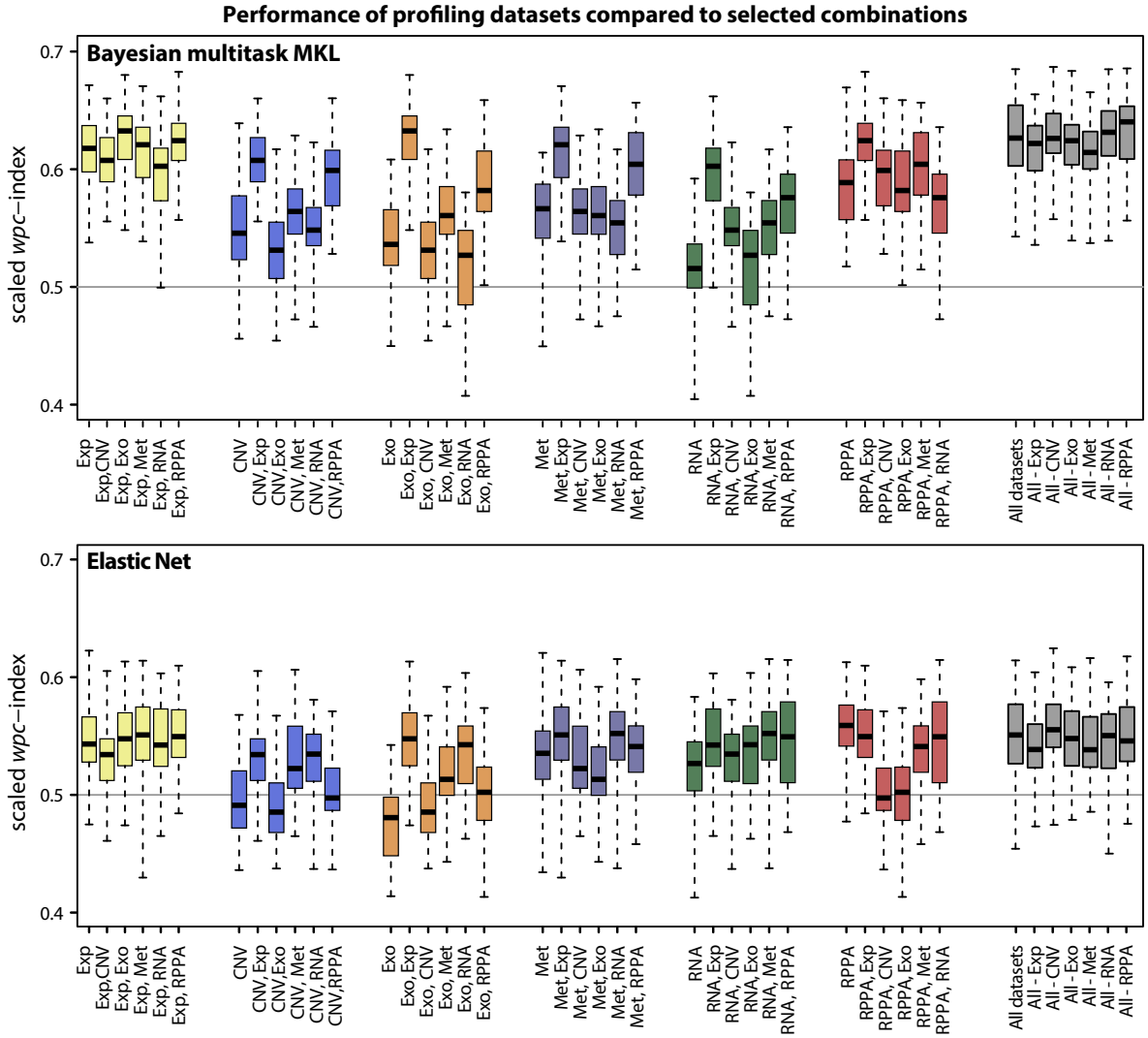
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 6. Variation in predictive performance across datasets and drugs

Two methods, Bayesian multitask MKL (Kernel 1) and an elastic net, were trained and predictions made using the 6 individual profiling datasets (rows). Each cell is the average scaled pc-index over 50 independent trials (color-coding in the bottom). (**a**) Performance for each of the 28 drugs (columns; grouped according to drug classes). (**b**) The performance of a drug class was calculated as the average performance of all compounds that constitute the drug class itself.

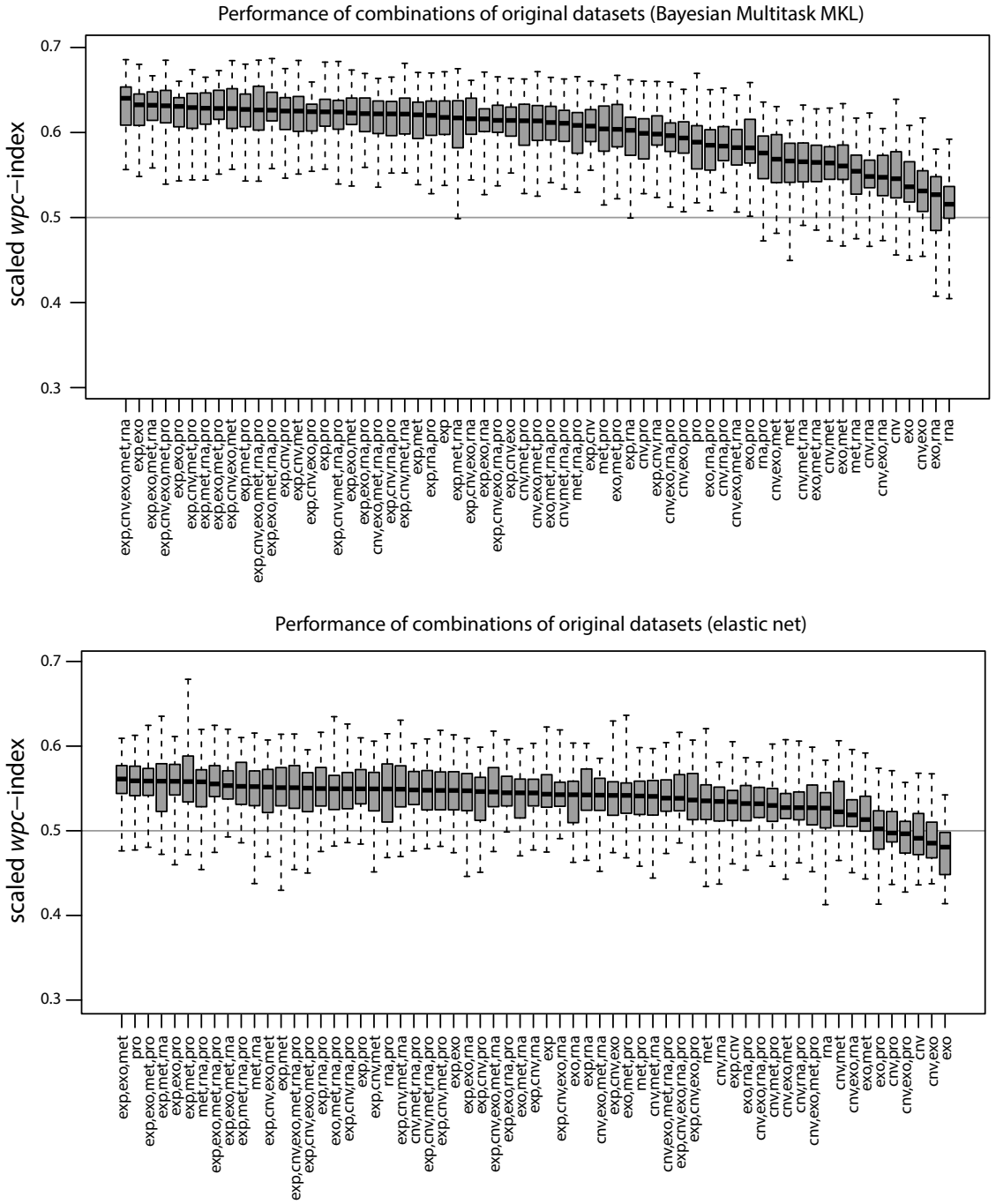
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 7. Exploration of data complementarity and redundancy

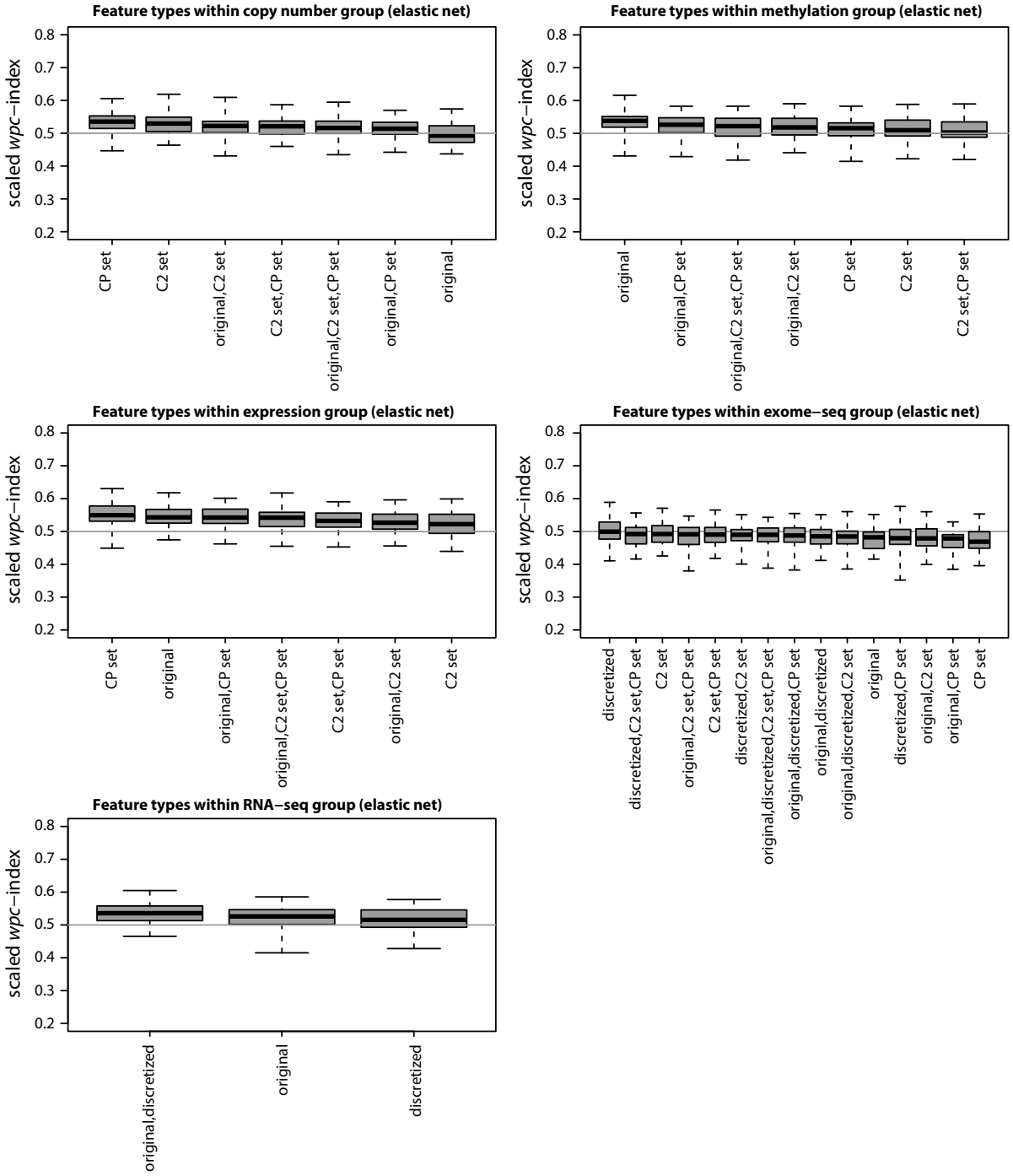
For two methods, Bayesian multitask MKL (Kernel 1) and an elastic net, two types of comparisons were made. First, and displayed as the colored box plots, both models were trained and predictions were made using individual profiling datasets (Exp = gene expression, CNV = Copy Number Variation, Exo = exome sequencing, Met = methylation, RNA = RNA sequencing, and RPPA = Reverse Phase Protein Array) and pairwise combinations of profiling datasets, revealing added value. Predictions were scored over 50 independent, random splits of the data and the distribution of performances are plotted. Second, and displayed as the grey box plots, the models were trained and predictions were made by leaving one of the 6 profiling datasets out in turn, indicating the dataset's nonredundant value when compared to the performance using all 6 datasets.

Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 8. Identification of informative dataset combinations
Predictive performance of all combinations of profiling datasets across 50 independent, random data splits was investigated for two methods, Bayesian multitask MKL and elastic net, extending the dataset comparison in **Fig. 4.** The combinations are ordered according to their median performance (exp = gene expression, cnv = Copy Number Variation, exo = exome sequencing, met = methylation, rna = RNA sequencing, and pro = Reverse Phase Protein Array).

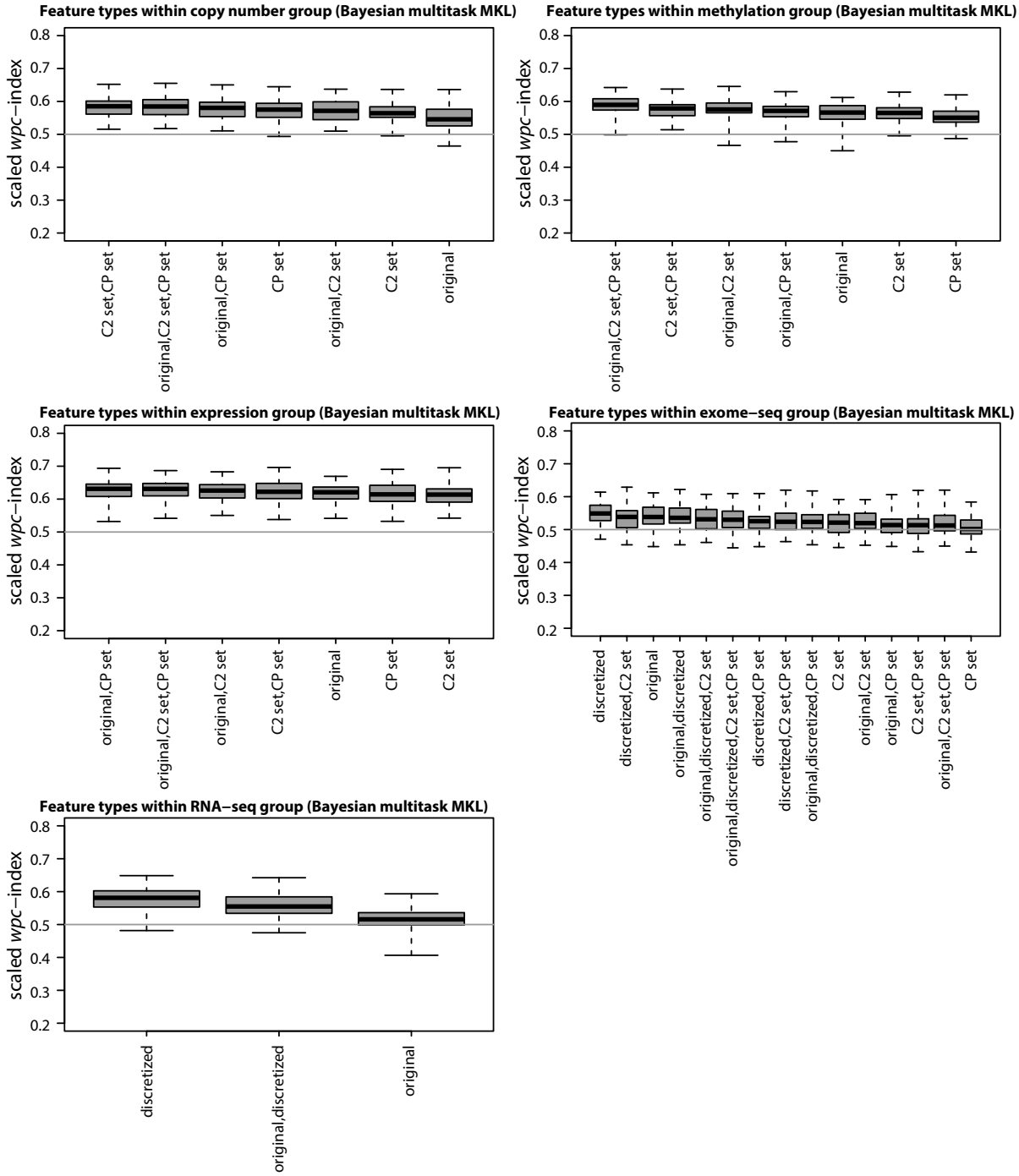
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 9. Identification of the most informative data view(s) within a profiling dataset for an elastic net

An elastic net was trained and predictions were made using various data views and combinations of data views within each of the 6 profiling datasets. By comparing each of these box plots to the original data view, the gain or reduction in performance contributed by different data views (or combinations of views) can be seen. The data views and combinations are ordered according to their median performance across 50 random data splits.

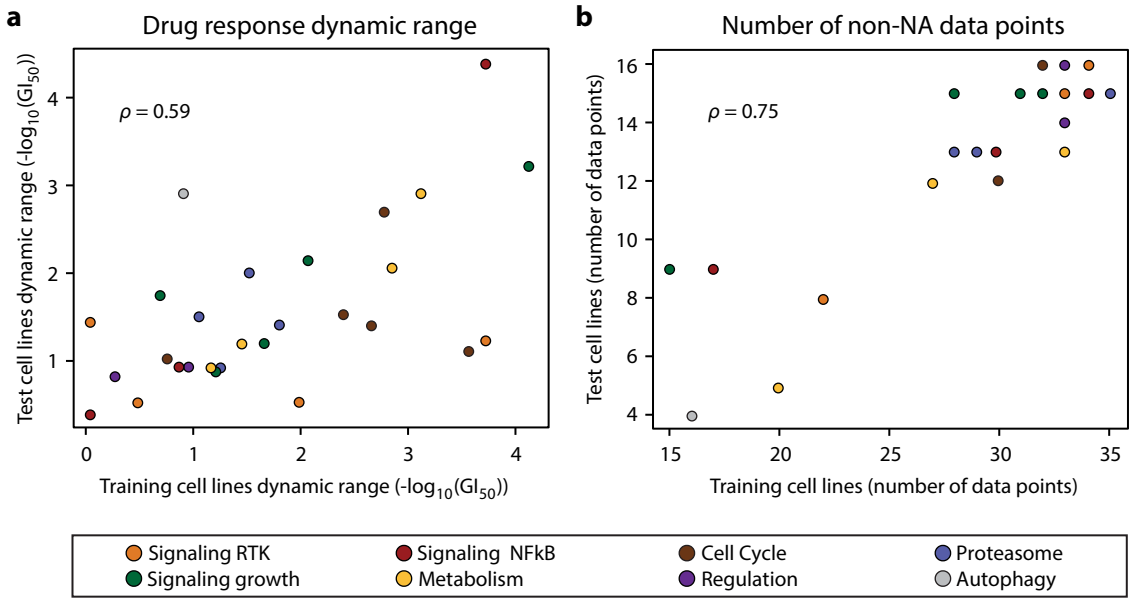
Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 10. Identification of the most informative data view(s) within a profiling dataset for Bayesian multitask MKL

Bayesian multitask MKL (Kernel 1) was trained and predictions were made using various data views and combinations of data views within each of the 6 profiling datasets. By comparing each of these box plots to the original data view, the gain or reduction in performance contributed by different data views (or combinations of views) can be seen. The data views and combinations are ordered according to their median performance across 50 random data splits.

Nature Biotechnology: doi:10.1038/nbt.2877



Supplementary Figure 11. Comparison of the training and test cell line features

(**a**) The dynamic range (minimum to maximum $-\log_{10}(GI_{50})$) and (**b**) the number of missing values in the training cell lines were compared to the test cell lines. In both instances, there is a highly statistically significant relationship, which shows the test data are representative of the training data.

Nature Biotechnology: doi:10.1038/nbt.2877