

Improving Deconvolution Methods in Biology through Open Innovation Competitions: an Application to the Connectivity Map

Andrea Blasco (Harvard) Ted Natoli (Broad) Rinat A. Sergeev (Harvard)

Steven Randazzo (Harvard) Jin H. Paik (Harvard) Max Macaluso (Broad)

Rajiv Narayan (Broad) Karim R. Lakhani (Harvard)

Aravind Subramanian (Broad)

Last updated: Dec 11, 2019

Abstract

The Connectivity Map (CMap) team has generated millions of perturbational gene expression profiles using an assay called L1000. L1000 uses a 500-plex cytometry platform to measure a reduced representation of the transcriptome comprising 1,000 genes. A key computational challenge is to deconvolute the 500 composite raw signal values for each profile into the 1,000 gene expression values. The current k-means approach is slow (~5 minutes per plate) and susceptible to errors. Here we describe new methods obtained via an open innovation challenge. The challenge used a novel dataset of L1000 profiles for 2,200 perturbation experiments for performance evaluation, and attracted 294 competitors from 20 countries. The top-nine performing methods ranged from machine learning (Convolutional Neural Networks and Random Forests) to more traditional solutions (Gaussian Mixtures and k-means). The winning method, a random forest regressor, achieved the highest accuracy with the lowest (inter-replicate) variation, detecting nearly a thousand more differentially-expressed genes than the k-means benchmark, while also improving precision.

Keywords: connectivity map; open innovation competitions; crowdsourcing; deconvolution; gene expressions; machine learning; transcriptome.

1 Introduction

Deconvolution problems are commonplace in many areas of science and engineering. In the context of biomedical research, a recurring issue is how to isolate specific gene expressions of distinct populations (cell types, tissues, and genes) from composite measures obtained by a single analyte or sensor. This problem often stems from the prohibitive cost of profiling each population separately [1, 2] and has important implications for the analysis of transcriptional data in mixed samples [3, 4, 5, 6], single-cell data [7], the study of cell dynamics [8], but it also appears in the analysis of imaging data [9].

Existing computational deconvolution approaches work well in many specific settings [10] but they might be suboptimal to the increasing availability of biological data. Machine learning techniques can potentially improve upon current methods. A typical advantage is the ability to capture complicated patterns that can be hard to model otherwise (especially in complex and massive datasets as those frequently used in biomedical research). However, introducing machine learning in the field presents several challenges; some of which are validation, adaptation to complex datasets, and identification of the best machine-learning approaches to specific problems.

Here we describe how we addressed these challenges in the context of the Connectivity Map (CMap). The CMap project is a massive catalog of human gene-expression signatures of genetic or pharmacologic perturbation with lots of applications in drug discovery and development [1]. The CMap’s team has generated millions of perturbation gene expression profiles using a new assay, called L1000. Based on a 500-plex fluorescence-based flow cytometry system, L1000 measures a reduced representation of transcriptome comprising 1000 genes that are highly representative of the state of human cells, thus achieving significant cost reductions compared to more traditional methods (RNA-sequencing). However, a key technical challenge is to deconvolute the 500 composite raw signal values for each profile into the 1000 gene expression values. The current approach used in the CMap pipeline is slow and can be susceptible to errors. Below we report improvements to the deconvolution algorithm (dpeak) obtained via an open innovation challenge.

For evaluation of new methods, we generated a novel experimental dataset with L1000 profiles for 122

different perturbagens (shRNA and compounds) and several replicates for a total of over 2,200 gene expression experiments. We varied the detection mode for acquiring L1000 data between the current procedure, called DUO, that obtains a raw composite measure of two genes per analyte color, and a more expensive method, called UNI, that measures one gene per analyte color. The deconvolution algorithm processes the DUO data and assigns the correct expression level to each of the two genes whose transcripts bind to beads of the same analyte color. This procedure is not needed with the UNI data. Hence, the UNI data provided ground truth that enabled us to evaluate empirically different deconvolution methods on the DUO data.

Leveraging this data set, we then explored different machine learning approaches through an open innovation competition [11, 12], or contest. We run the contest on Topcoder (Wipro, India), a popular crowdsourcing platform. The contest challenged participants to use the novel dataset to improve the deconvolution algorithm utilized by the L1000 platform. The contest drew about 300 competitors from 20 different countries and resulted in a diverse number of machine-learning approaches. The top approaches included machine-learning methods, such as Random Forests and Convolutional Neural Networks (CNNs), as well as more traditional Gaussian-mixture models and k-means. These approaches significantly performed better than the L1000 benchmark in various measures of accuracy and computational speed, and likely have application beyond gene expression.

2 Methods

The contest lasted 21 days. A prize purse of \$23,000 in cash was offered to competitors as an incentive to be divided among the top 9 submissions (prize split: 80, 60, 40, 20, 10, 8, 6, 4, and 1 hundred dollars). We provided competitors with a problem statement, access to training and testing data, and a well-defined scoring function. Figure 1 shows a schematic illustration of these three elements.

The problem statement described the L1000 procedure and the current solution. The key insight was that the distribution of any composite measure of two differentially expressed genes should have two peaks and the size of each peak should reflect the proportion of measurements per gene. L1000 takes full advantage of this fact by pairing genes optimally (trying to maximize the average difference in their expression levels) and by mixing genes in a 2:1 proportion to enable the assignment of the correct expression levels to each gene within each pair [see 1, for the details]. Then it uses a deconvolution algorithm, called dpeak, that detects the peaks of the distribution and assigns them to each gene. More specifically, the dpeak procedure was based on a k-means clustering algorithm [13]. This approach partitions the composite measurements for each profile into k clusters by minimizing the within-cluster sum of squares. It then assigns the largest cluster to the gene with higher bead proportion (and the smallest cluster to the gene with lower bead proportion).

The training and testing data consisted of six 384-well perturbagen plates, each containing mutually exclusive sets of compound and short-hairpin (shRNA) treatments (see S1 Table and S2 Table for a complete list of the perturbagens). Multiple treatment types were used to avoid potentially over-fitting to anyone. The compound and shRNA perturbagen plates were arbitrarily grouped into pairs, and to avoid any potential ‘information leakage’ each pair was profiled in a different cell line. The resulting lysates were amplified by Ligation Mediated Amplification (LMA, Subramanian et al. [1]). The amplicon was then split and detected in both UNI and DUO detection modes, resulting in three pairs of data generated under comparable circumstances. The training data was available for all the contestants to develop and validate their solutions offline. The

testing data was for submission evaluation during the contest and to populate a live leaderboard. The holdout data was for submission final evaluation evaluate, thus guarding against over-fitting. Prizes were awarded based on performance on the holdout dataset.

The scoring function combined measures of accuracy and computational speed (See S1 Appendix for more details). The accuracy metric was the product of two different metrics. The first was the average genewise Spearman’s rank correlation between the deconvolution gene expressions and the ground truth for each gene. The second was the Area Under the receiver operating curve (AUC) in the prediction of extreme modulated genes. The computational speed was measured as the average runtime in seconds per plate.

3 Results

The contest attracted 294 participants who made 820 code submissions overall with a variety of different analysis methods (see Table 1 for an overview of the best methods). Here we report the top-four performing methods based on the score obtained on the holdout data.

The winning solution (by a competitor from the United States with a degree in Physics from the University of Kansas) was a random forest algorithm (combining 10 different trees). Each tree was trained using 60 different features of the data; 50 features were from the distribution of measurements within the gene pair, the remaining 10 features were measures of the association between the distribution of values within the pair and aggregate values at the plate and experiment level.

The second solution (by a competitor from Poland with a Master’s degree in Computer Science from The Lodz University of Technology) used the Expectation-Maximization (EM) method for an initial clustering of the observed values, assuming the distribution was a mixture of two log-normal distributions (Gaussian mixture model). It then fit a plate-wide distribution of the estimated expressions and cluster sizes for each gene, using the results to adjust the assignment of peaks to the low- and high-proportion genes.

The third solution (by a competitor from India with a bachelor’s degree in Computer Science) was a basic k-means algorithm with random initial points to avoid local minima along with other fine adjustments to minimize the impact of extreme outliers and improve speed.

The fourth solution (by a competitor from Ukraine with a bachelor’s degree in Computer Science from the Cherkasy National University) was a Convolutional Neural Network (CNN). The algorithm first filtered the data to remove extreme outlier measurements, transforming the filtered data into a 32-bin histogram for each pair of genes. The CNN consisted of two parts. The first part used the classic U-net architecture [14] (a contracting path to capture context and a symmetric expanding path) to provide adequate representation of the data. The output of this network was then used to label bins into one of the two genes for each pair, and to predict the exact value within the bin. This second step was achieved using two subnetworks with the same architecture but weights trained separately (using a mean squared error loss function).

3.1 Clustering by method and perturbagen type

Given the variety of methods represented amongst the prize-winning solutions, we sought to assess whether there were notable differences in the predictions generated. Using the holdout dataset, we generated the

contestants' deconvolution data (DECONV) and the corresponding differential expression (DE) values. We obtained the DE values by using the robust z-score procedure by Subramanian et al. [1]. We then compared the results using a two-dimensional t-SNE projection [15].

For the most part, the DECONV data clustered by pertubagen type, except for the ground truth UNI data, which appears distinct from the deconvolution samples (Figure 3, A and B). Separating the samples by method revealed commonalities in the predictions generated by similar approaches (Figure 3, C). For example, the decision tree regressor (DTR) algorithms have similar ‘footprints’ in the projection, as do the k-means and Gaussian mixture model (GMM) algorithms. This suggests that in general similar algorithms generate predictions with similar properties. However, after the transformation to DE data, the t-SNE projection appears more homogenous (Figure 3 D), indicating that pertubagen-type and algorithm-specific effects have greatly reduced. The reduced clustering at the DE level is reassuring given the downstream analysis is often focused on DE values.

3.2 Correlation accuracy

We tested the accuracy of the solutions by comparing the distribution of the genewise Spearman’s rank correlation (ρ) between the UNI data and the values obtained by the competitors through the deconvolution of DUO (holdout) data. We did so for the shRNA and compound experiments separately; and, within each set of experiments, we compared the correlation accuracy for genes in high and low bead proportion. In all these cases, we found that the winner’s distribution was shifted to the right of the benchmark’s, indicating significant improvements in accuracy. We also observed that, for each set of experiments, the greatest shift was for the distribution of genes in low bead proportions (compared to genes in high bead proportions). For the genes in low bead proportions, the winner’s average improvement was of 5 percentage points (with standard error of half a point) whereas the improvement was of 2 percentage points (with standard error of half a point) for the genes in high bead proportions.

The analysis of the top-nine performing algorithms showed similar results: the average Spearman’s rank correlation was generally greater for genes in high bead proportions (relative to low bead proportions) and for genes in shRNA experiments (relative to compounds); the benchmark performed consistently worse than all the others in the genes in low bead proportions, although it performed well in the high-proportion genes. Remarkably, the winning solution outperformed the other approaches in nearly all the comparisons.

Next, we ranked the top-nine algorithms by the average correlation metric for each of the 976 landmark genes and selected the best algorithm for each gene across all the experiments. The winning solution was selected as the best performer for about a third (30%) of the genes. The second-placed gaussian mixture model was selected as the best performer for about one fourth (20%). Thus, the top two submissions achieved the highest correlation in more than half of the genes in our sample. Even so, all but a few algorithms were the best performers for at least 5% of the genes, suggesting some complementarity between these algorithms.

3.3 Detection of extreme modulations

We further tested the accuracy of the competitors’ solutions on the detection of genes notably up- or down-regulated by perturbation, or *extreme modulations*. These genes exhibit exceedingly high (or low) DE values

(above a given threshold). Using the UNI DE values as the ground truth, we evaluated the detection accuracy of each solution by computing the corresponding area under the curve (AUC).

All the solutions achieved a good average detection accuracy ($AUC > .86$) with significant improvements over the benchmark in both the shRNA and compound experiments (??).

We further tested differences in accuracy for targeted gene knockdowns (KD). These are experiments involving an shRNA targeting one of the 976 landmark genes. We expected the targeted gene to exhibit a very low DE value (highly negative). For each such experiment, we define a successful KD as one in which the DE value is less than -2 and the gene-wise rank is less than 10, meaning that the targeted gene achieves one of its lowest DE values in the experiment where it was targeted. We computed the KD success frequency, or recall, for each competitor algorithm as well as the benchmark and UNI data. We used the *UNI* ground-truth data to estimate the maximum achievable recall, which in this case was 0.8. We observed that all (but 2) algorithms achieved a higher KD success frequency than the benchmark solution. These results suggest that the algorithm improvements translate to improvements in biologically relevant metrics used in common applications of L1000 data.

3.4 Reduced variation across replicate samples

Our data contain between 4 to 11 replicate samples for each perturbagen. Keeping low the inter-replicate variation is of crucial practical importance to biologists because it means higher reproducibility of results. With this goal in mind, the benchmark k-means solution is likely suboptimal because it does little to mitigate the discrepancy in variability between the genes measured with high and low bead proportions.

We compared the inter-replicate variance of the predictions made by each method for genes in low and high bead proportions. As expected, we found significant improvements in the inter-replicate variance of the competitors over the benchmark with the difference being more visible for genes in low bead proportions. Among the competitors, the winning method, which was the most accurate on average, was also the most efficient achieving the lowest inter-replicate variance.

3.5 Computational speed

The speed improvements over the benchmark were substantial. While the dpeak took about 4 minutes per plates, the fastest algorithm took as little as 5 seconds per plate (a 60x speedup compared to the benchmark) and the slowest was well below one minute per plate. We observed no particular trade-off between speed and accuracy. The fastest algorithm (**Ardavel**), that was based a gaussian mixture model, achieved a good level of accuracy as well, and ranked second overall. On the other hand, the algorithm with the best performance in terms of accuracy (**gardn999**), which was based on a decision tree regression, also achieved a decent speed performance compared to the benchmark.

3.5.1 Ensembles

Next, to assess the complementarity of the algorithms, we assessed the performance of ensembles composed of different number of models. For the present analysis, we focused on the subset of the data with shRNA

experiments (ignoring the data with compound experiments). For this subset, the competitors achieved a significantly lower correlation and AUC than the data with compound experiments. We used a basic approach to aggregate the predictions of each solution by using the median prediction. Then, we tested the performance using the Spearman correlation and the AUC metrics computed on the holdout dataset for that plate. Results (??) showed that the performance of the ensemble tends to increase with the number of models involved. However, the maximum performance in both metrics tends to plateau (or even decrease) after the ensemble reaches a size equal to 3 models.

3.6 Concluding remarks

In summary, we have created a novel dataset of over 120 shRNA and compound experiments with measurements for about 1,000 genes. This dataset constitutes a public resource to all the researchers in this area who are interested in testing their deconvolution approaches. Using an open innovation competition, we collected and evaluated multiple and diverse deconvolution methods.

The best approach was based on a random forest, which achieved (i) the highest “global” correlation between UNI and DUO data, (ii) the lowest inter-replicate variability and (iii), compared to the benchmark, was able to detect more than a thousand additional differentially-expressed genes, while improving the detection precision at the same time. This provides evidence of the tremendous potential of using machine-learning approaches for deconvolution methods in biology.

Next, we will apply these results to over one million experiments that constitute the Connectivity Map, and explore cost savings achieved by having a lower number of replicates. Another potential application is to leverage these algorithms to enable detection of three or more genes using the same analyte type.

4 Tables and Figures

Table 1

Top-nine performing solutions. This table lists the top-nine solutions and the languages and algorithms each used, as well as the average speedup per plate relative to the k-means benchmark.

rank	handle	language	method	category	speedup
1	gardn999	Java	random forest regressor	DTR	17x
2	Ardavel	C++	Gaussian mixture model	GMM	62x
3	mkagenius	C++	modified k-means	k-means	24x
4	Ramzes2	Python/C++	ConvNet	CNN	10x
5	vladaburian	Python/C++	Gaussian mixture model	GMM	7x
6	balajipro	Python/C++	modified k-means	k-means	21x
7	AliGebily	Python	boosted tree regressor	DTR	5x
8	LastEmperor	Python	modified k-means	k-means	7x
9	mvaudel	Java	other	other	55x

Figure 1

Schematic illustrating the computational problem, generated data, and scoring function. Panel (a) provides a schematic description of the L1000 DUO detection mode and the associated deconvolution problem to be addressed by the contestants. Panel (b) shows the data generated for the contest comprising 6 different sets of perturbational experiments. The experiments consisted of 3 plates of compound (CP) and shRNA treatments (KD) in 3 cell lines. Each plate was detected using DUO (2 genes per analyte) and UNI (one gene per analyte) with UNI serving as the ground truth. Contestants were given two plates of data for training their models offline. The second set of two plates was used during the contest for testing and to populate the live leaderboard, and the third set of two was used as holdout to determine the final contestant placements. Panel (c) illustrates the accuracy component of the scoring function that was used to evaluate the solutions submitted by the competitors. A solution's overall accuracy score was the product of the genewise Spearman's rank correlations with ground truth (DECONV data) and the area under the curve AUC of extreme modulations (DE data).

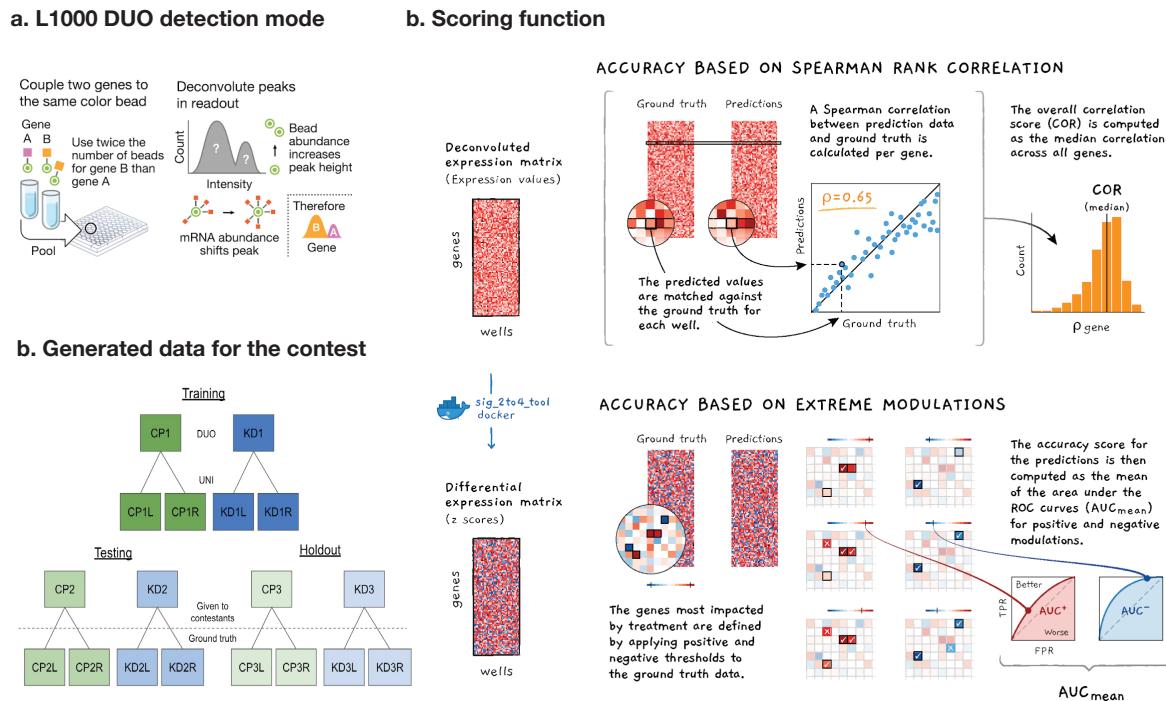
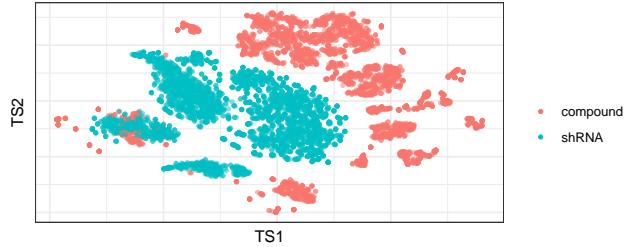


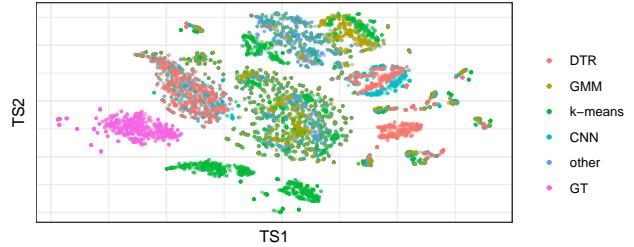
Figure 3

Clustering of solutions. Each point represents the two-dimensional projection of a sample generated by UNI ground truth (GT) or by applying a deconvolution algorithm to DUO data. t-SNE was run on the 2 plates of holdout data, one each containing compound and shRNA treatments. DECONV data colored by perturbagen type (A) and algorithm type (B). DECONV (C) and DE (D) data colored by algorithm type and stratified by each individual implementation.

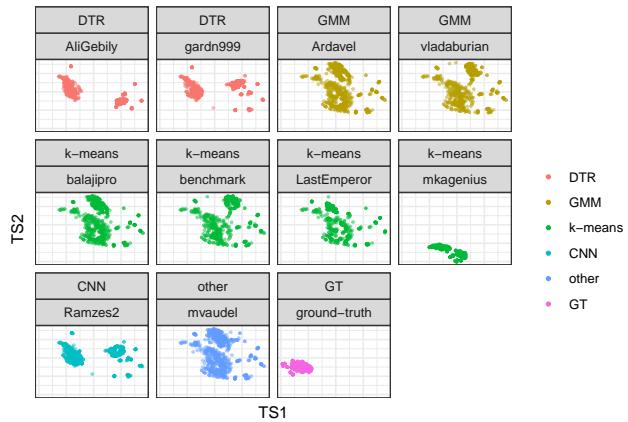
A DECONV colored by perturbagen type



B DECONV colored by algorithm type



C DECONV colored by algorithm type



D DE colored by algorithm type

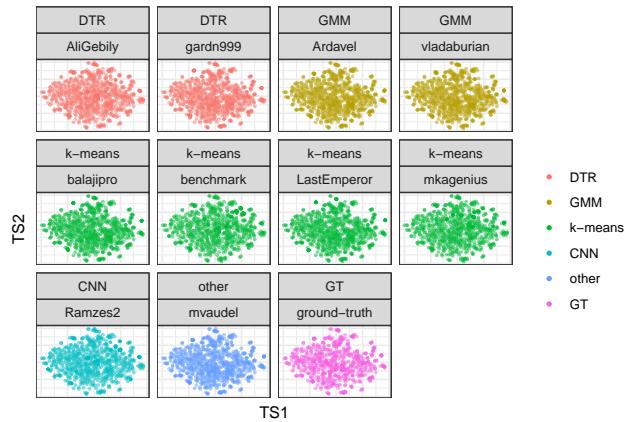
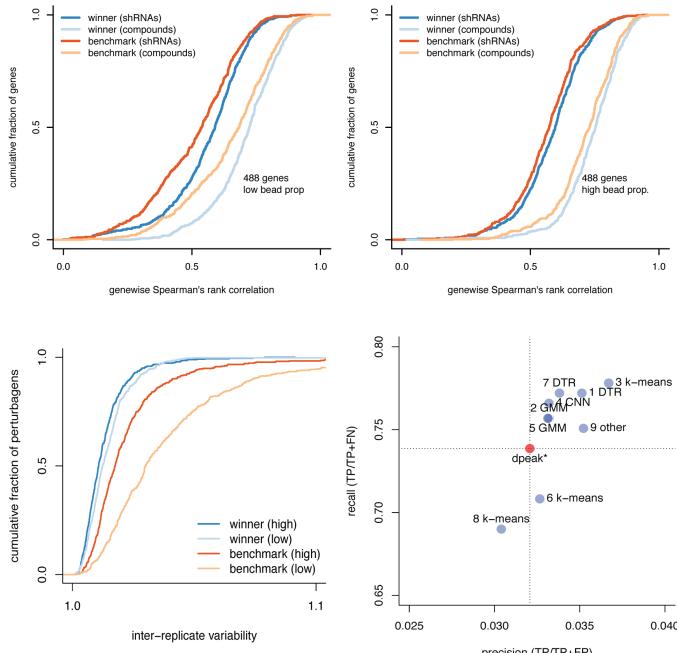


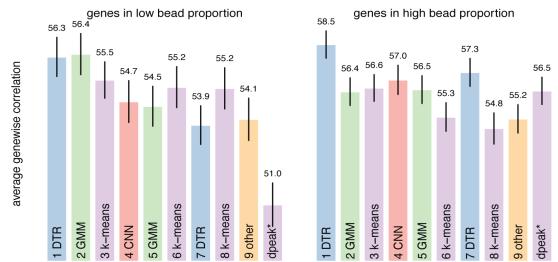
Figure 4

Improvements in accuracy XXX

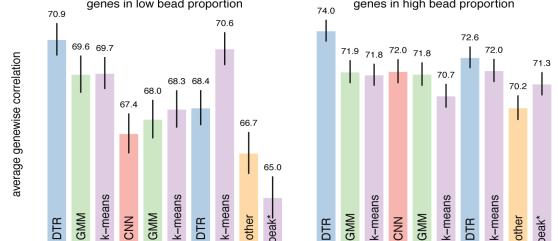
d. correlation accuracy (winner vs dpeak)



d. correlation accuracy (shRNAs)



d. correlation accuracy (compounds)



d. detection accuracy (extreme modulations)

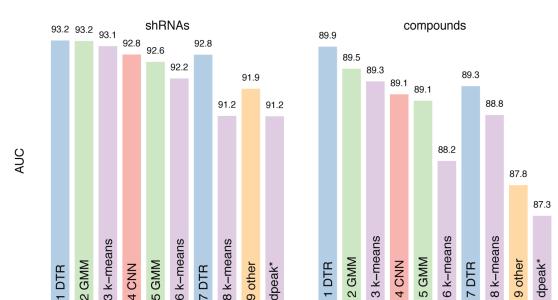
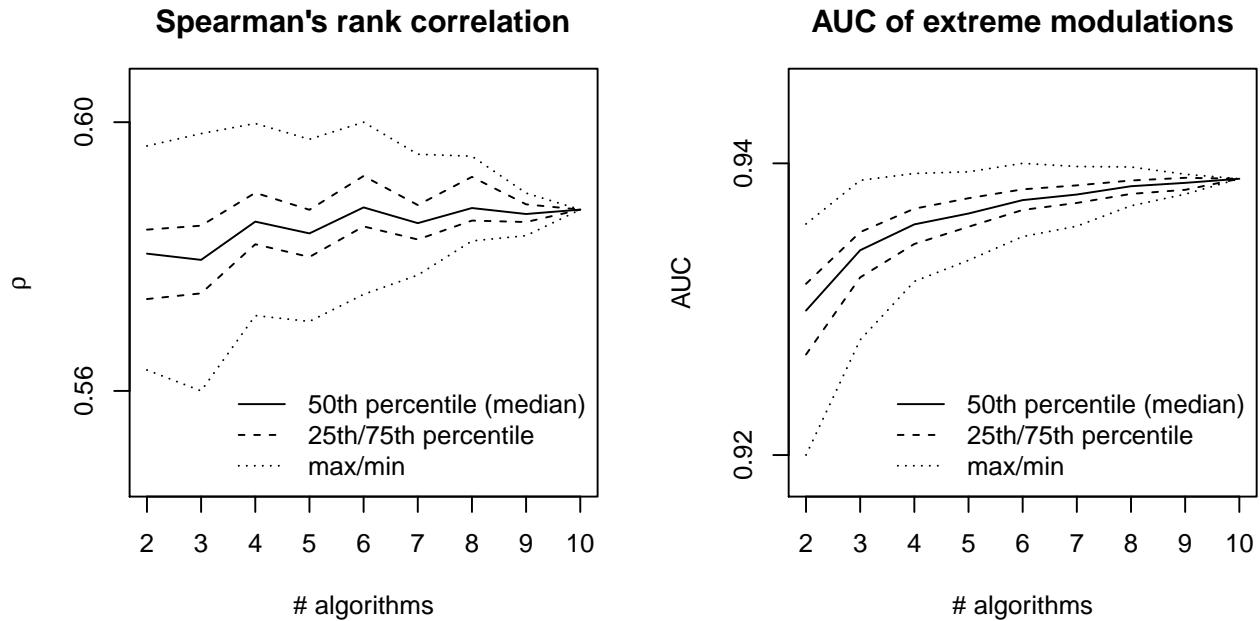


Figure 5

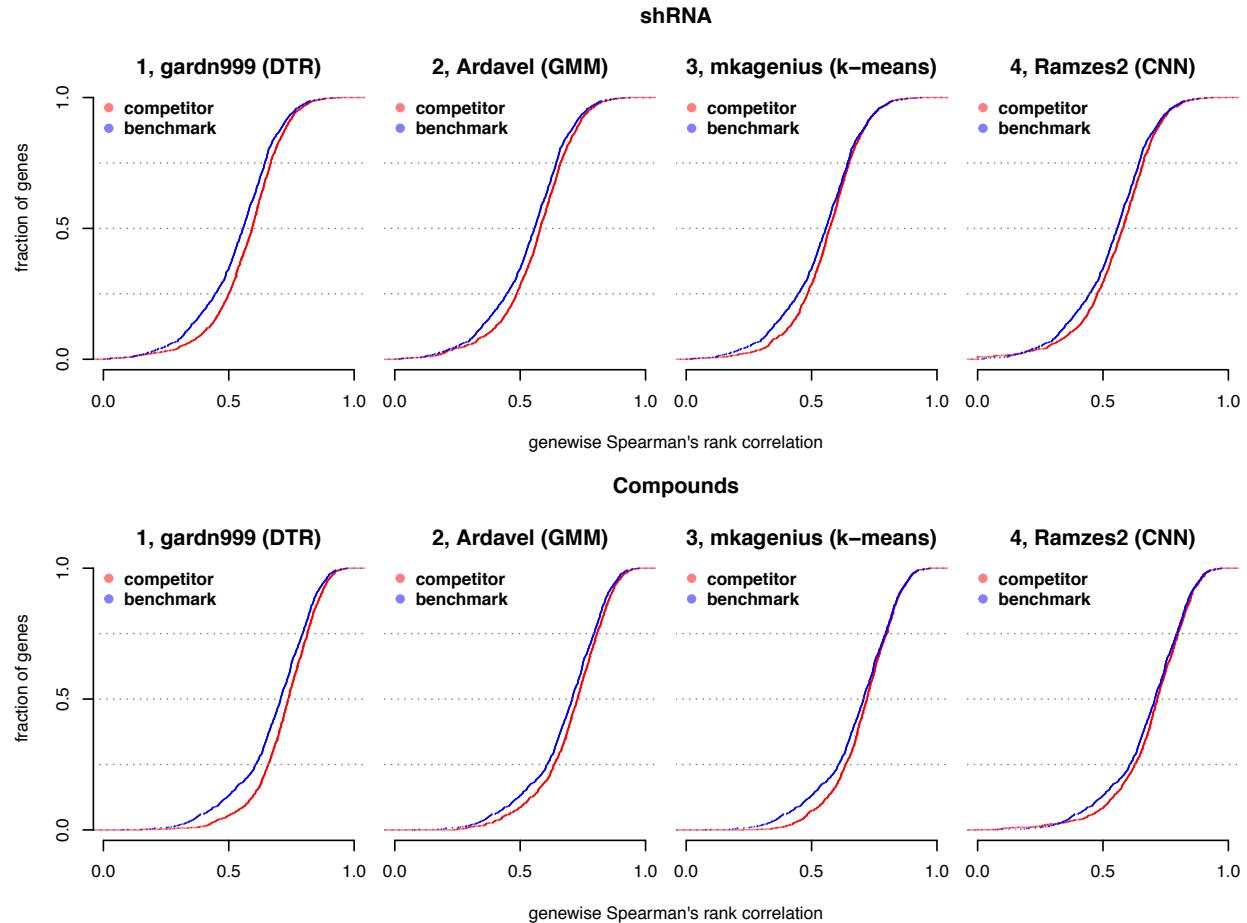
Performance of ensembles. This figure shows the performance in the **A** correlation metric and **B** the AUC metric of the ensemble based on the median prediction of all possible combinations of a given size of the top 10 algorithms (including the benchmark). The median performance of the ensemble tends to increase with its size. However, the maximum performance in both metrics tends to plateau (or even decrease) after the ensemble reaches a size equal to 3.



Supporting information

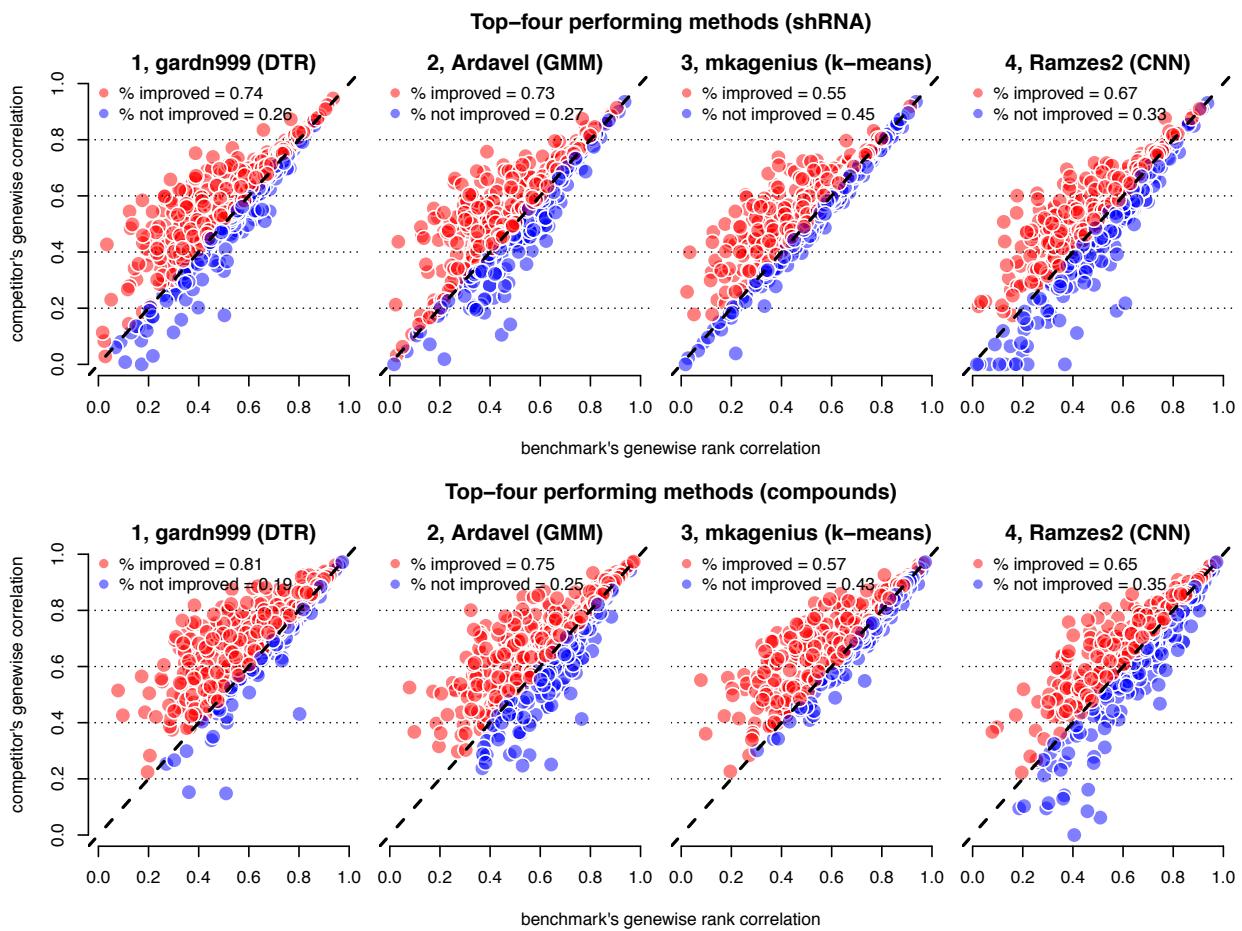
S1 Figure

eCDF of genewise Spearman's rank correlation. This figure shows the empirical Cumulative Distribution Function (eCDF) of the genewise Spearman's rank correlation for the top-four performing approaches and the k-means benchmark. The top panels present the eCDFs corresponding to the shRNAs, whereas the bottom panels present the eCDFs corresponding to the compounds.



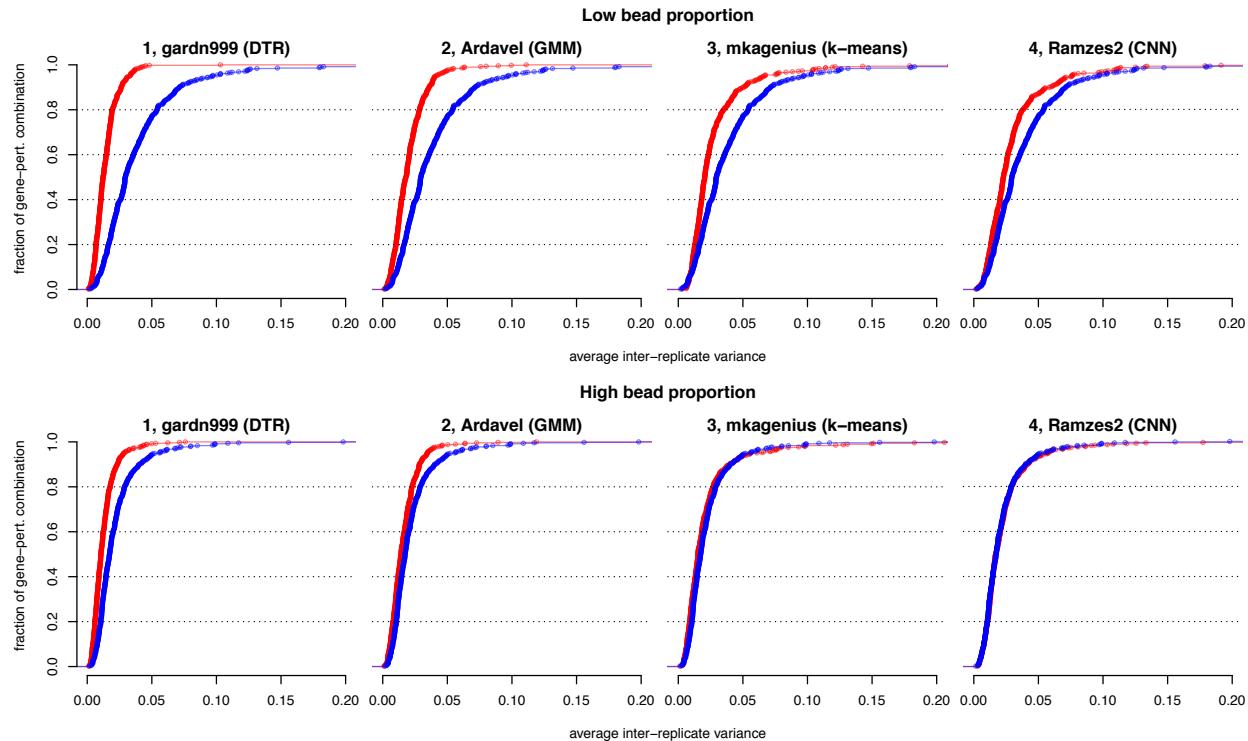
S2 Figure

Improvements in genewise Spearman's rank correlation. This figure shows the association between the genewise Spearman's rank correlation for the top-four performing approaches and the k-means benchmark; it also reports the fraction genes with a higher genewise Spearman's rank correlation than the benchmark. The top panels present the scatter plots corresponding to the shRNAs, whereas the bottom panels present the scatter plots corresponding to the compounds.



S3 Figure

Improvements in inter-replicate variability. This figure shows the empirical Cumulative Distribution Function (eCDF) of the genewise inter-replicate variance for the top-four performing approaches and the k-means benchmark. The top panels present the eCDFs corresponding to the genes in low bead proportion (recall genes are paired and mixed in 2:1 ratio within each pair), whereas the bottom panels present the eCDFs corresponding to the genes in high bead proportion.



S1 Table.

Compound perturbagens descriptives. This table shows compound perturbagen names (`pert_iname`), unique id (`pert_id`), time of treatment (`pert_itime`), dose (`pert_idose`), and number of replicates (`num_replicates`).

<code>pert_iname</code>	<code>pert_id</code>	<code>pert_itime</code>	<code>pert_idose</code>	<code>num_replicates</code>
abiraterone(cb-7598)	BRD-K50071428	24 h	10 um	11
acalabrutinib	BRD-K64034691	24 h	10 um	11
afatinib	BRD-K66175015	24 h	10 um	11
artesunate	BRD-K54634444	24 h	10 um	11
azithromycin	BRD-K74501079	24 h	10 um	11
betamethasone dipropionate (diprolene)	BRD-K58148589	24 h	10 um	11
CGS-21680	BRD-A81866333	24 h	10 um	11
chelidonine	BRD-K32828673	24 h	10 um	11
clobetasol	BRD-K84443303	24 h	10 um	11
digoxin	BRD-A91712064	24 h	10 um	11
disulfiram	BRD-K32744045	24 h	10 um	10
emetine hcl	BRD-A77414132	24 h	10 um	10
eplerenone	BRD-K19761926	24 h	10 um	11
epothilone-a	BRD-K71823332	24 h	10 um	9
flumetasone	BRD-K61496577	24 h	10 um	11
fluocinolone	BRD-K94353609	24 h	10 um	11
genipin	BRD-K28824103	24 h	10 um	11
hydrocortisone	BRD-K93568044	24 h	10 um	10
hyoscyamine	BRD-K40530731	24 h	10 um	11
indirubin	BRD-K17894950	24 h	10 um	10
L-745870	BRD-K05528470	24 h	10 um	10
nTZDpa	BRD-K54708045	24 h	10 um	11
oligomycin-a	BRD-A81541225	24 h	10 um	11
PRIMA1	BRD-K15318909	24 h	10 um	11
RITA	BRD-K00317371	24 h	10 um	11
spironolactone	BRD-K90027355	24 h	10 um	11
tanespimycin	BRD-K81473043	24 h	10 um	11
tretinoin	BRD-K71879491	24 h	10 um	10
UB-165	BRD-A14574269	24 h	10 um	11
ursolic-acid	BRD-K68185022	24 h	10 um	11

pert_iname	pert_id	pert_itime	pert_idose	num_replicates
WAY-161503	BRD-A62021152	24 h	10 um	11
ZM-39923	BRD-K40624912	24 h	10 um	11

S2 Table.

Non-compound perturbagens descriptives. This table shows non-compound perturbagen names (`pert_iname`), unique id (`pert_id`), and number of replicates (`num_replicates`).

<code>pert_iname</code>	<code>pert_id</code>	<code>num_replicates</code>
ABCB6	TRCN0000060320	4
ADI1	TRCN0000052275	4
ALDOA	TRCN0000052504	4
ANXA7	TRCN0000056304	4
ARHGAP1	TRCN0000307776	4
ASAHI	TRCN0000029402	4
ATMIN	TRCN0000141397	4
ATP2C1	TRCN0000043279	4
B3GNT1	TRCN0000035909	4
BAX	TRCN0000033471	4
BIRC5	TRCN0000073718	4
BLCAP	TRCN0000161355	4
BLVRA	TRCN0000046391	4
BNIP3L	TRCN0000007847	4
CALU	TRCN0000053792	4
CCDC85B	TRCN0000242754	4
CCND1	TRCN0000040038	4
CD97	TRCN0000008234	4
CHMP4A	TRCN0000150154	4
CNOT4	TRCN0000015216	4
DDR1	TRCN0000000618	4
DDX10	TRCN0000218747	4
DECR1	TRCN0000046516	4
DNM1L	TRCN0000001097	3
ECH1	TRCN0000052455	4
EIF4EBP1	TRCN0000040206	4
EMPTY_VECTOR	TRCN0000208001	15
ETFB	TRCN0000064432	4
FDFT1	TRCN0000036327	4
GALE	TRCN0000049461	4
GFP	TRCN0000072181	16

pert_iname	pert_id	num_replicates
GRN	TRCN0000115978	4
GTPBP8	TRCN0000343727	4
HDGFRP3	TRCN0000107348	4
HIST1H2BK	TRCN0000106710	4
IKBKAP	TRCN0000037871	4
INPP4B	TRCN0000230838	4
INSIG1	TRCN0000134159	4
ITFG1	TRCN0000343702	3
JMJD6	TRCN0000063340	4
LBR	TRCN0000060460	4
LGMN	TRCN0000029255	4
LPGAT1	TRCN0000116066	4
LSM6	TRCN0000074719	4
MAPKAPK2	TRCN0000002285	4
MAPKAPK3	TRCN0000006154	4
MAPKAPK5	TRCN0000000684	4
MIF	TRCN0000056818	4
MRPL12	TRCN0000072655	4
NT5DC2	TRCN0000350758	4
NUP88	TRCN0000145079	4
PARP2	TRCN0000007933	4
PLCB3	TRCN0000000431	4
POLE2	TRCN0000233181	4
PPIE	TRCN0000049371	4
PRKAG2	TRCN0000003146	4
PSMB10	TRCN0000010833	4
PTPN6	TRCN0000011052	4
RAB11FIP2	TRCN0000322640	4
RALB	TRCN0000072956	4
RHEB	TRCN0000010425	3
RNF167	TRCN0000004100	4
RPN1	TRCN0000072588	4
SLC25A4	TRCN0000044967	4
SNX11	TRCN0000127684	4

pert_iname	pert_id	num_replicates
STK25	TRCN0000006270	4
STUB1	TRCN0000007525	4
STXBP1	TRCN0000147480	4
SYPL1	TRCN0000059926	4
TATDN2	TRCN0000049828	4
TM9SF3	TRCN0000059371	4
TMEM110	TRCN0000127960	4
TMEM50A	TRCN0000129223	4
trcn0000014632	TRCN0000014632	4
trcn0000040123	TRCN0000040123	4
trcn0000220641	TRCN0000220641	4
trcn0000221408	TRCN0000221408	4
trcn0000221644	TRCN0000221644	4
TSKU	TRCN0000005222	4
UGDH	TRCN0000028108	4
USP14	TRCN0000007428	4
USP6NL	TRCN0000253832	4
VAT1	TRCN0000038193	4
VDAC1	TRCN0000029126	4
WIPF2	TRCN0000029833	4
YME1L1	TRCN0000073864	4
ZW10	TRCN0000155335	4

S1 Appendix

Scoring function. This appendix describes the scoring function used in the contest to evaluate the performance of the competitors' submissions.

Submissions were scored based on a scoring function that combines measures of accuracy and computational speed. Accuracy measures were obtained by comparing the contestant's predictions, which were derived from *DUO* data, to the equivalent *UNI* ground truth data generated from the same samples.

The scoring function combines two measures of accuracy: correlation and AUC, which are applied to deconvoluted (*DECONV*) data and one to differential expression (*DE*) data, respectively.

DE is derived from *DECONV* by applying a series of transformations (parametric scaling, quantile normalization, and robust z-scoring) that are described in detail in Subramanian et al. [1]. The motivation for scoring *DE* data in addition to *DECONV* is because it is at this level where the most biologically interesting gene expression changes are observed. Of particular interest is obtaining significant improvement in the detection of, so called, “extreme modulations.” These are genes that notably up- or down-regulated by perturbation and hence exhibit an exceedingly high (or low) *DE* values relative to a fixed threshold.

The first accuracy component is based on the Spearman rank correlation between the predicted *DECONV* data and the corresponding *UNI* ground truth data.

For a given dataset p , let $M_{\text{DUO},p}$ and $M_{\text{UNI},p}$ denote the matrices of the estimated gene intensities for $G = 976$ genes (rows) and $S = 384$ experiments (columns) under DUO and UNI detection. Compute the Spearman rank correlation matrix, ρ , between the rows of these matrices and take the median of the diagonal elements of the resulting matrix (i.e., the values corresponding to the matched experiments between UNI and DUO) to compute the median correlation per dataset,

$$\text{COR}_p = \text{median}(\text{diag}(\rho(M_{\text{DUO},p}, M_{\text{UNI},p}))).$$

The second component of the scoring function is based on the Area Under the receiver operating characteristic Curve (AUC) that uses the competitor's DE values at various thresholds to predict the UNI's DE values being higher than 2 (“high”) or lower than -2 (“low”).

For a given dataset p , let $\text{AUC}_{p,c}$ denote the corresponding area under the curve where $c = \{\text{high}, \text{low}\}$ (either higher than 2 or less than -2); then, compute the arithmetic mean of the area under the curve per class to obtain the corresponding score per dataset:

$$\text{AUC}_p = (\text{AUC}_{p,\text{high}} + \text{AUC}_{p,\text{low}})/2.$$

These accuracy components were integrated into a single aggregate scores:

$$\text{SCORE} = \text{SCORE}_{\text{max}} \cdot (\max(\text{COR}_p, 0))2 \cdot \text{AUC}_p \cdot \exp(-T_{\text{solution}}/(3 \cdot T_{\text{benchmark}})),$$

where T_{solution} is the run time in seconds for deconvoluting the data in each plate, and $T_{\text{benchmark}}$ is the deconvolution time required by the benchmark D-Peak implementation.

S2 Appendix

L1000 Experimental Scheme The L1000 assay uses Luminex bead-based fluorescent scanners to detect gene expression changes resulting from treating cultured human cells with chemical or genetic perturbations [Subramanian 2017]. Experiments are performed in 384-well plate format, where each well contains an independent sample. The Luminex scanner is able to distinguish between 500 different bead types, or colors, which CMap uses to measure the expression levels of 978 landmark genes using two detection approaches.

In the first detection mode, called *UNI*, each of the 978 landmark genes is measured individually on one of the 500 Luminex bead colors. In order to capture all 978 genes, two detection plates are used, each measuring 489 landmarks. The two detection plates' worth of data are then computationally combined to reconstruct the full 978-gene expression profile for each sample.

By contrast, in the *DUO* detection scheme two genes are measured using the same bead color. Each bead color produces an intensity histogram which characterizes the expression of the two distinct genes. In the ideal case, each histogram consists of two peaks each corresponding to a single gene. The genes are mixed in 2:1 ratio, thus the areas under the peaks have 2:1 ratio (see Figure 1), which enables the association of each peak with the specific gene. The practical advantage of the *DUO* detection mode is that it uses half of the laboratory reagents as *UNI* mode, and hence *DUO* is and has been the main detection mode used by CMap. After *DUO* detection, the expression values of the two genes are computationally extracted in a process called ‘peak deconvolution,’ described in the next section.

References

- [1] Aravind Subramanian et al. “A next generation connectivity map: L1000 platform and the first 1,000,000 profiles”. In: *Cell* 171.6 (2017), pp. 1437–1452.
- [2] Brian Cleary et al. “Efficient generation of transcriptomic profiles by random composite measurements”. In: *Cell* 171.6 (2017), pp. 1424–1436.
- [3] Shai S Shen-Orr et al. “Cell type-specific gene expression differences in complex tissues”. In: *Nature methods* 7.4 (2010), p. 287.
- [4] Yi Zhong and Zhandong Liu. “Gene expression deconvolution in linear space”. In: *Nature methods* 9.1 (2012), p. 8.
- [5] Aaron M Newman et al. “Robust enumeration of cell subsets from tissue expression profiles”. In: *Nature methods* 12.5 (2015), p. 453.
- [6] Konstantin Zaitsev et al. “Complete deconvolution of cellular mixtures based on linearity of transcriptional signatures”. In: *Nature communications* 10.1 (2019), p. 2209.
- [7] Yue Deng et al. “Scalable analysis of cell-type composition from single-cell transcriptomics using deep recurrent learning”. In: *Nature methods* 16.4 (2019), p. 311.
- [8] Peng Lu, Aleksey Nakorchevskiy, and Edward M Marcotte. “Expression deconvolution: a reinterpretation of DNA microarray data reveals dynamic changes in cell populations”. In: *Proceedings of the National Academy of Sciences* 100.18 (2003), pp. 10370–10375.
- [9] Stephan Preibisch et al. “Efficient Bayesian-based multiview deconvolution”. In: *Nature methods* 11.6 (2014), p. 645.

- [10] Shai S Shen-Orr and Renaud Gaujoux. “Computational deconvolution: extracting cell type-specific information from heterogeneous samples”. In: *Current opinion in immunology* 25.5 (2013), pp. 571–578.
- [11] Karim R Lakhani et al. “Prize-based contests can provide solutions to computational biology problems”. In: *Nature biotechnology* 31.2 (2013), p. 108.
- [12] Andrea Blasco et al. “Advancing computational biology and bioinformatics research through open innovation competitions”. In: *PLOS ONE* 14.9 (Sept. 2019), pp. 1–17. doi: 10.1371/journal.pone.0222165. URL: <https://doi.org/10.1371/journal.pone.0222165>.
- [13] Stuart Lloyd. “Least squares quantization in PCM”. In: *IEEE transactions on information theory* 28.2 (1982), pp. 129–137.
- [14] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention*. Springer. 2015, pp. 234–241.
- [15] Laurens van der Maaten and Geoffrey Hinton. “Visualizing data using t-SNE”. In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.