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

author 1 author 2 ...

Last updated: Oct 09, 2019

Abstract

Deconvolution problems are ubiquitous and have a long history in many areas of science and engineering. In the context of genetic research with human cell lines, the ability to extract signals of individual genes from mixed gene-type samples is of paramount importance. However, the validation of existing deconvolution methods is often difficult because of the lack of datasets for testing, as well as the costs involved in the development of alternative approaches. To address the problem, we used an assay, called L1000, to generate a novel dataset of the response of 1000 genes to various perturbagens (shRNA and compounds) both in mixed samples and individually. We then run an open competition to explore different approaches using the dataset with individual measurements to validate the deconvolution methods. The competition produced several approaches to the problem based on a wide variety of techniques including machine-learning algorithms, such as Convolutional Neural Networks and Random Forests, and more traditional approaches, such as Gaussian mixtures and k-means. All top solutions achieved notable improvements over the benchmark.

Keywords: biology; open innoation competitions; crowdsourcing; deconvolution; gene expressions; cell lines.

Contents

1 Introduction

Over the last decade, technological advances in multiplexed, high-throughput experimental systems have made it possible to generate large perturbational datasets which in turn have helped accelerate therapeutic discovery. A large proportion of these technologies rely on using multiple analytes (or 'sensors') to measure biological signals. These assays are extremely powerful but in some applications are limited by the number of unique analytes they can detect.[1]

This limitation is more prominent for gene-expression profiling, given the high number of genes involved. Within this context, the Connectivity Map (CMap) has developed an assay, called L1000, that integrates traditional multianalyte detection methods and statistical analyses to measure simultaneously the gene expression for multiple genes using the same analyte type, thus multiplying the capacity of existing technologies [1].

A central component of this approach is the computational deconvolution of gene expression signals from multiple genes measured using the same analyte. Similar deconvolution problems are ubiquitous in biology and several solutions have been proposed in a variety of different contexts. Examples include algorithms to identify cell type–specific gene expression differences in complex tissues [2] or dynamic changes in cell populations [3], or ways to discover the target proteins of small molecules [4].

CMap's current deconvolution approach, called dpeak, is based on the k-means clustering algorithm, which is a flexible unsupervised learning procedure widely used in biology, as well as in many other fields. This approach automatically partitions a set of gene-expression measurements into k clusters by minimizing the within-cluster sum of squares. It then assigns the clusters to each gene within the sample (see Subramanian et al. [1] for the details). This approach works well in practice but has several limitations; it tends to split clusters incorrectly when the distributions are not reasonably well separated, it is sensitive to outliers, and it is computationally demanding (an efficient k-means algorithm, such as Lloyd's, has a running time that scales as $O(kn)$ where n is the number of observations and k the number of clusters).

Motivated by recent successes in the use of machine learning techniques in biology, we hypothesized better deconvolution approaches could likely be developed. To test this hypothesis, we generated a novel experimental dataset of the response of approximately 1,000 genes to 122 different perturbagens (shRNA and compounds) within 4 to 10 replicates that we obtained by using the L1000 platform. We varied the detection mode for acquiring the data between two genes per analyte type (DUO) and one gene per analyte type (UNI). We then used the data to run an open innovation competition (as described in Lakhani et al. [5]; Blasco et al. [6]) where we solicited different solutions to the deconvolution problem by offering cash incentives to competitors and by assessing the accuracy of different approaches using the UNI data as the ground truth.

[1] For example, Luminex's most advanced platform, called FLEXMAP 3D, can measure simultaneously a max of 500 bead types for an equal number of genes or proteins from a small sample.

The top approaches included very popular machine-learning methods, such as Random Forests and Convolutional Neural Networks (CNNs), as well as more traditional Expectation-Maximization (EM) approaches based on Gaussian mixtures. Overall, the results of the challenge enabled us to perform a cost-effective comparison of different methods under nearly identical conditions, which we report below.

2 Methods

We ran the competition on the Topcoder platform (Wipro, India) for a total of 21 days. A prize purse of $23,000 in cash was offered to competitors as incentive to be divided among the top 9 submissions ($8000, $6000, $4000, $2000, $1000, $800, $600, $400, $100).

To generate data for this contest, we profiled 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 any one. 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.

We randomly split the three pairs of data, as described above, into training, testing, and holdout subsets. The training data were available for all the contestants to develop and validate their solutions offline. The testing data were used to evaluate solutions during the contest and populate the live leaderboard. The holdout data were used to evaluate competitors' final submissions and to guard against over-fitting. Prizes were awarded based on performance on the holdout dataset. In each category, the UNI data served as the ground truth.

To evaluate the submissions, we developed a scoring function based on a combination of two different accuracy metrics; the Spearman rank correlation between the values and the ground truth and the area under the receiver operating curve (ROC) as a measure of aggregated classification performance in the prediction of extreme modulated genes. The scoring function also included the computational speed. (See S1 Appendix for the details.)

3 Results

The contest attracted 294 participants, who made 820 code submissions with an average of about 18 submissions per participant. The top finishers in the contest employed a variety of different analysis approaches, including descision tree regressors (DTR), Gaussian mixture models (GMM), convolutional neural networks (CNN), and customized versions of k-means, all with notably

improved performance relative to the benchmark (S3 Table provides descriptives of the approaches used by the top finishers).

The winning approach (`gardn999`) was a random forest algorithm that combined the predictions of 10 different trees. Each tree was trained using 60 different features of the data. These features included the actual distribution of the values for each bead type (binned into 50 different variables), as well as various plate- and experiment-level variables for which the data point belongs.

The second best approach (`Ardavel`) was a em xxx .

3.1 Accuracy and speed

We tested the accuracy and speed of the competitors' solutions on the holdout L1000 data obtained by applying the DUO detection method (two genes per bead color) and utilizing the data obtained by employing the UNI detection method (one gene per bead color) as the ground truth.

Correlation. The Spearman rank correlations between gene-level expressions of the ground truth and those of each of the top nine algorithms and the benchmark were high overall ($\rho > 0.56$). For both shRNA and compound experiments, the empirical distribution of the solutions made by the competitors was right-shifted compared to the k-means benchmark, indicating more accurate predictions (*Fig. 2*, A and B); the median Spearman rank correlation of the competitors was significantly higher than the equivalent for the benchmark (*Fig. 2*, C and D), although the size of the observed improvements was modest (2-3 percentage points).

Next, we counted the number of genes with the highest Spearman rank correlation for each algorithm. Results (*Fig.* **??**) showed that the winning algorithm, based on a random forest, accounted for about one third (33%) of the genes; the second placed algorithm (based on EM) accounted for one fifth (20%); and the fourth placed algorithm (CNN) accounted for another approximately one fifth (20%). Thus, these three approaches achieved jointly the best performance for more than 70% of the genes in our samples.

Extreme modulations. We further tested the accuracy of the competitors' solutions on the detection of, so called, "extreme modulations." These are genes that are notably up- or down-regulated by perturbation, relative to the absence of perturbagens treatment and, hence, exhibit an exceedingly high (or low) differential expression (DE) values. We obtained the DE values by using a robust z-score procedure (as described by Subramanian et al. [1]) and evaluated the detection accuracy of each solution by computing the corresponding area under the curve (AUC); all the solutions achieved a good detection accuracy (AUC > 0.87). Again, the competitors' solutions achieved signicant improvements relative to the benchmark (*Fig.* **??**).

Clustering. 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 a two-dimensional projection of the UNI ground truth data and DUO-

derived benchmark and contenstant predictions for both $DECONV$ and DE data using t-distributed stochastic neighbor embedding (t-SNE, Maaten and Hinton [7]). We observe that in $DECONV$ data the samples primarily cluster by pertubagen type, with the exeption of the ground truth UNI data, which appears distinct from the deconvoluted samples (*Fig.* **??** A and B). Separating the samples by algorithm reveals commonalities in the predictions generated by similar algorithms (*Fig.* **??** 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. After the standard transformation to DE data we observe that the t-SNE projection is much more homogenous, indicating that perturbagen type and algorithm-specific affects have been greatly reduced (*Fig.* **??** D). This is reassuring, given that in production mode downstream analysis of this data will be based on DE. It also suggests that integrating multiple deconvolution algorithms into an ensemble method might be feasible.

Gene knockdowns. We further tested differences in the probability oof detection of the extreme modulations for knockdown genes (explain). For this comparison, we computed the metric also for the UNI data, and compared performance of... We observe a high performance in accuracy. xxx we computed the KD success frequency for each algorithm and the benchmark. This metric is the fraction of the 376 landmark-targeting shRNA experiments in the holdout data for which the predictions met these success criteria. We used the UNI ground-truth data to estimate the maximum achievable KD success frequency, which in this case was 0.8. We observe that all but 2 of the top 9 contestant algorithms achieve a higher KD success frequency than the benchmark solution (*Fig.* **??**). These results suggest that the algorithm improvements, as assessed by the accuracy metrics used in the contest, translate to improvements in biologically relevent metrics used in common applications of L1000 data.

Inter-replicate variance. Agreement between experimental replicates is crucial xxx [ref] "The inter-replicate variation of gene expression-quantities is of the utmost importance to biologists because lower variance means higher reproducibility." [refXXX]. Our data contain several replicates which enabled us to study improvements in inter-replicability variance. See figure *Fig. 4*.

Speed. Speed improvements over the benchmark were substantial (*Fig.* **??**). The benchmark took about 4 minutes per plates. In contrast, the fastest algorithm took as little as 4 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. Thus, at least within the context of the implemented solutions, we found a negligible trade-off between speed and accuracy.

3.1.1 Reduction variation across replicate samples

One of the issues with the benchmark k-means solution is that it does little to mitigate the discrepancy in prediction accuracy between the genes measued with high and low bead proportions.

Todos: - boxplots of replicate variance, stratified by algo and hi/lo bead - barplot of average variance per algo, showing that winning algo has lowest variance and discrepancy b/w hi/lo bead is minimized

3.1.2 Clustering by gene-Level performance

We observed that the global structure of DE data seemed independent of algorithm type in general. We additionally sought to understand whether there were differences between the algorithms at the individual gene level. To assess this, we identified the best performing algorithm (by correlation metric) for each of the 976 landmark genes. We observe that while the contest winner is the best performer for the majority of genes, over 70% of the genes achieve better correlation with a different algorithm, and all but 2 algorithms are the best performers for at least 5% of the genes. This suggest that there may be complementarity between the algorithms, which could potentially be leveraged by an ensemble approach.

3.2 Ensemble approaches

Figure 3. (A) Scatterplot runtime vs accuracy for ensamble (slides p. 163)

Speed vs accuarcy trade-off. Integration one or multiple methods?

3.3 Minors:

- signs of overfitting (compare traing vs testing)

4 Discussion

Motivated by recent successes in the use of machine learning techniques to do x, y and ,z, we hypothesized that better deconvolution approaches could be developed. To test this hypthesis, we first generated a novel experimental dataset with differential-expression measurements obtained by two different detection methods. The first that involves deconvoltion, which we used as the ground-truth, and another that measures 2 genes per analyte, which was used a training dataset. Then, we run an open innovation competition to enable a cost-effective xxx of the space of possible solutions using these experimental data. We report the results of the this challenge.

The top three approaches as they emerged from the challenge included different machine-learning methods such as Random Forests, ConvNet, and Gaussian Mixtures. We evaluted these methods on the holdout data. The developed methods achieved significant improvements in accuracy (correlation) of baseline gene expression values, as well as a better performacne in the detection of extremely modulated genes (e.g., xxxx). Compared to the benchmark, these developed methods were more consistent across replicates (a smaller inter-replicate variability), thus leading to more reliable predictions overall.

We evaluated the computational speed of all the approaches. Overall, we find small speed vs accuracy tradeoff. Parametric methods, such as Gaussian Mixture, achieved 60x speedup compared to the benchmark, without losing on the accuracy. Machine-learning methods, such Random Forest, achieved greater accuracy but were slower. Yet, the ratio of accuracy over speed improvement was relatively small.

Motivated by the success of ensamble methods, we evaluated possible complementarity between the different approaches. We built an algorithm that combines the top methods selected by gene based on the results on the training data. On the holdout, we found that the ensamble further improved accuracy relative to xxxx.

4.1 Inspection of methods.

- gardn999. This was the winner, who holds a PhD In Physics from the University of Kansas. His solution was a Random Forest based on a total of 60 "features." These features included the actual values (50 variables?) and "various relationships of the averaged quantities in the Barcode, Experiment and Set for which the Pair belongs." The model was training using a total of 10 trees, to improve computational speed. Accuracy could be improved by increasing the number of trees.

- Ardavel used the EM algorithm for each pair and for a plate-wide distribution of expression and cluster size that is used for adjusted (especially of low-proportion genes); as well as further per-plate per-well adjustements. Find that it is better to do not assume a priory a probability of cluster size (i.e., 2:1 ratio), but it is better to estimate it from the data. Found many 3 peaked. Realized that plate-wide information was crucial to improve the solution, indicating where to swap the peaks and indicate false peaks.

- mkgenious used basic k-means tools tailered to the data to increase speed and accuracy. In other words, this was a fine-tuned implementation of the benchmark.

- Ramsez2 focused on the prediction of extreme modulations using as input 32bins histogram of the measurements for each pair. Neural Net predicts low and high values separately (2 subnetworks same architecture but trained separately). See figure. Used MSE loss and cross-entropy with Adam optmizer.

4.2 Future work.

- We have created a dataset of over 120 shRNA and compound experiments with measurements for about 1000 genes. This dataset constitutes a public resource to all the researchers in this area who are interested in testing their deconvoltuion appproaches.
- However, it remains to be seen performance on combining thre or more genes with single analytes. This is future work.
- Next, we will apply these resutls to over one million experiments and explore cost svings achieved by having a lower number of replicates

4.3 Older notes

Summary of the results presented in the methods section.

Discussion generality of the solutions

- Novel? Have any of these solutions previously been applied to deconvolution problems?
- Specific to this problem or general to others?

Discuss implications of these methods for CMap production

- Preliminary results on past data conversion
- Directions for pipeline integration and generation of future data
- Cost savings
- Implementation strategy and outcomes
- Increase in data processing throughput

5 Figures

5.1 Scoring accuracy

5.2 Accuracy vs. Speed

Spearman rank correlation

```
## reading data/UNI_DUO_gene_spearman_correlations_holdout_n20x976.gctx
```

```
## done
```

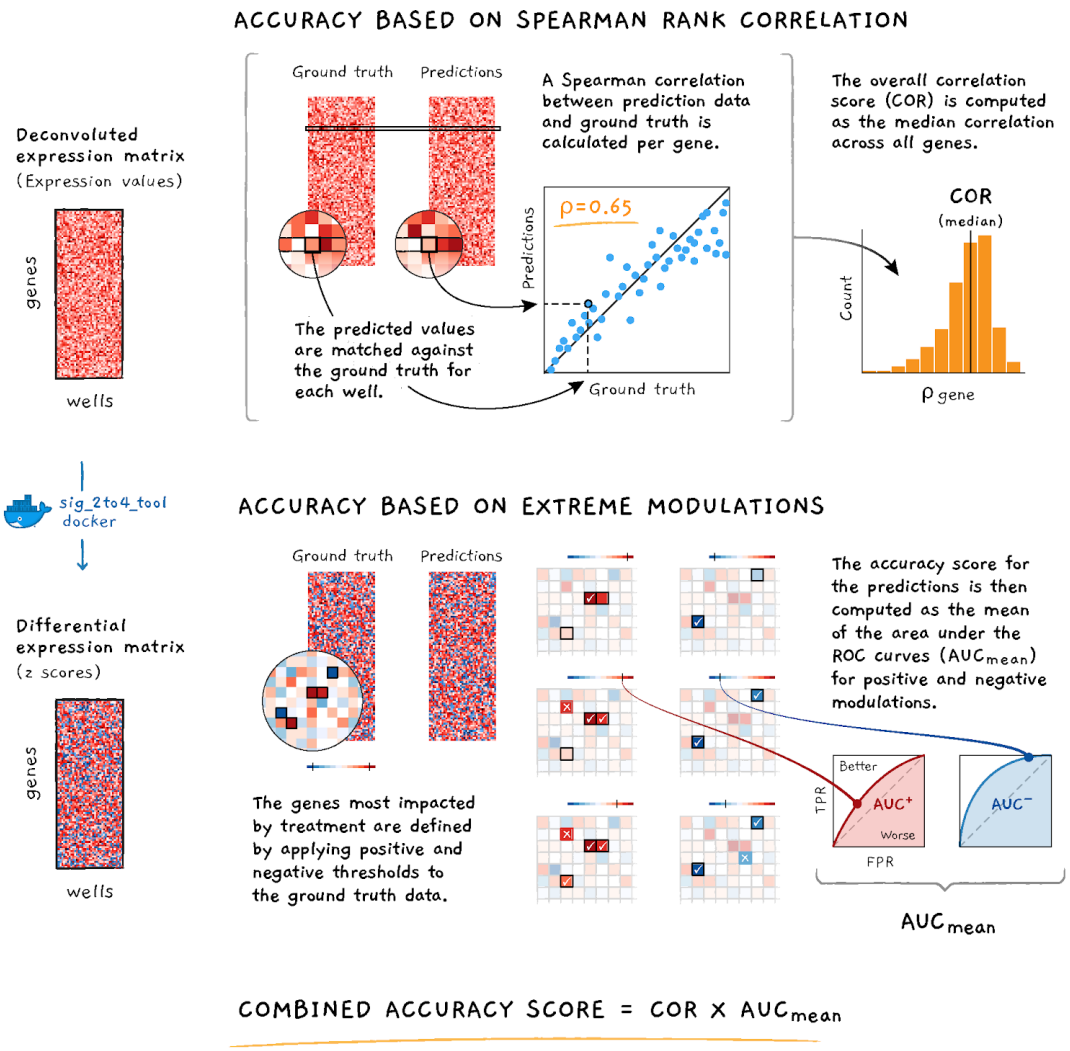


Figure 1: **Schematic illustrating accuracy components of scoring function**. The accuracy component is computed as the product of gene-wise Spearman correlations with ground truth and the area under the curve AUC of extreme modulations.

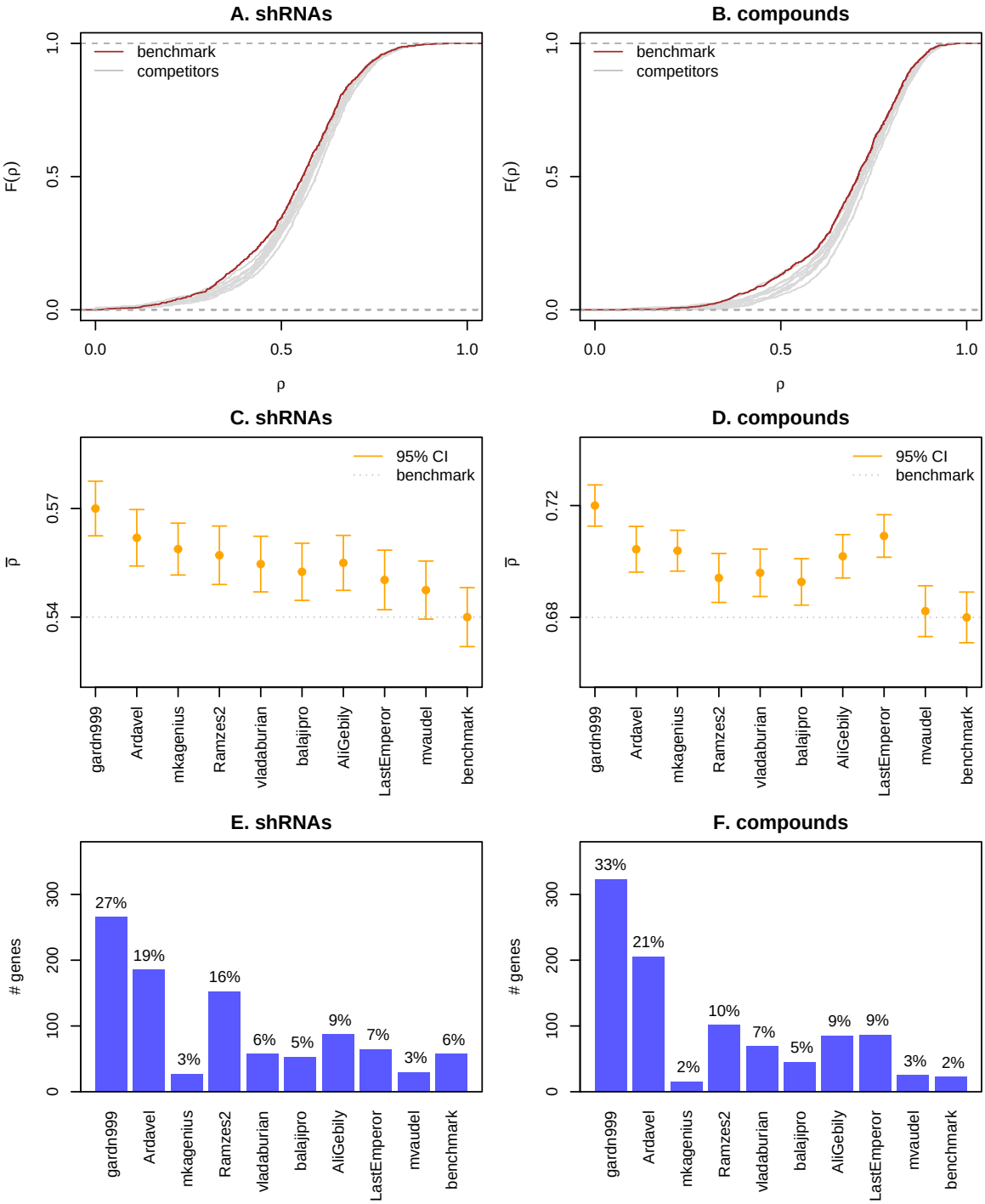


Figure 2: **Accuracy.** Top panels show empirical CDF of the distribution of the genewise spearman correlation (ρ) between the ground-truth gene-expressions (as detected by UNI) and predictions obtained by the competitors and the benchmark through the deconvolution of DUO data for the shRNA (**A**) and compound experiments (**B**). The competitors' CDFs are right-shifted compared to the benchmark, indicating more accurate predictions. Middle panels show the sample mean of the correlation coefficients ($\bar{\rho}$) with 95% confidence intervals for the shRNA (**C**) and compound experiments (**D**). The competitors (x-axis) are ordered by their final ranking in the contest. Compared to the benchmark, the mean correlation was significantly higher for nearly all top competitors. Bottom panels show the number of genes with the highest Spearman rank correlation for each algorithm. The combination of the top two ranked solutions (gardn999 and Ardavel) achieved the highest correlation in more than half of the genes in both experiments (**E** and **F**).

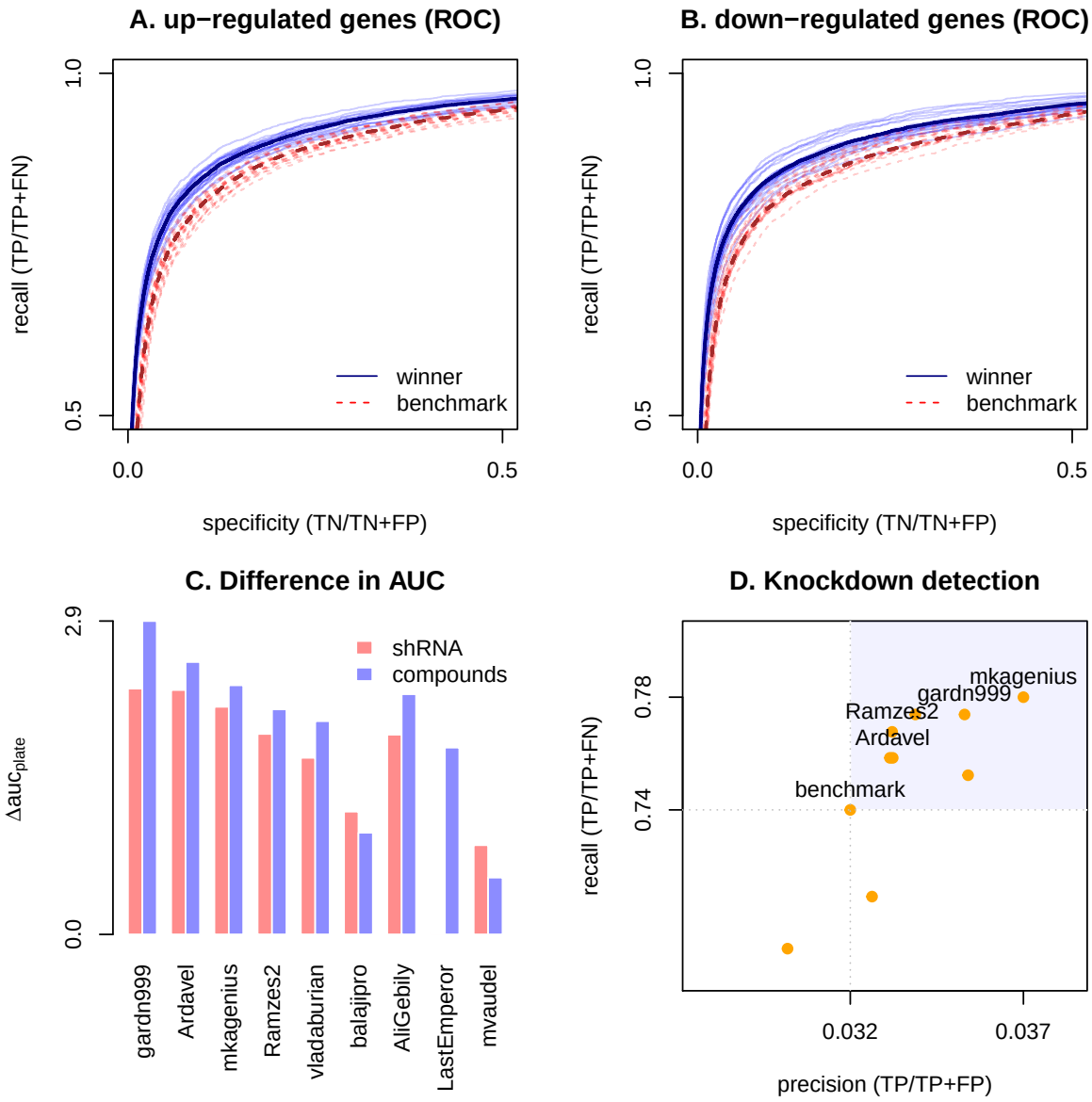


Figure 3: ROC, AUC and KD detection.

5.3 AUC plots

5.4 Inter-replicate variance

5.5 Variance by experiments

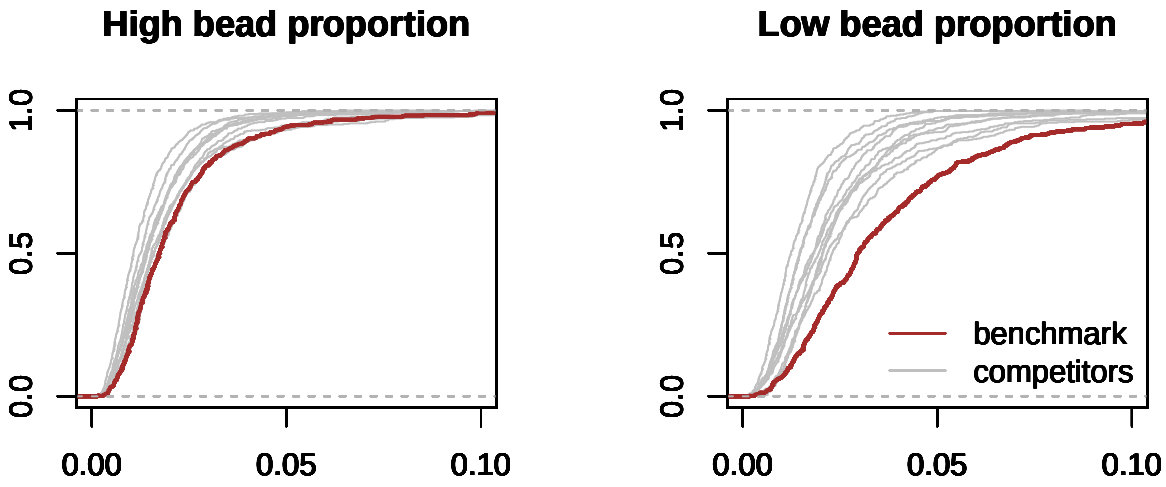


Figure 4: Inter-replicate variance.

6 Runtime and speedups

6.1 Accuracy

6.2 Clustering of solutions

6.3 Complementarity

```
## reading data/UNI_DUO_gene_spearman_correlations_holdout_n20x976.gctx
## done
```

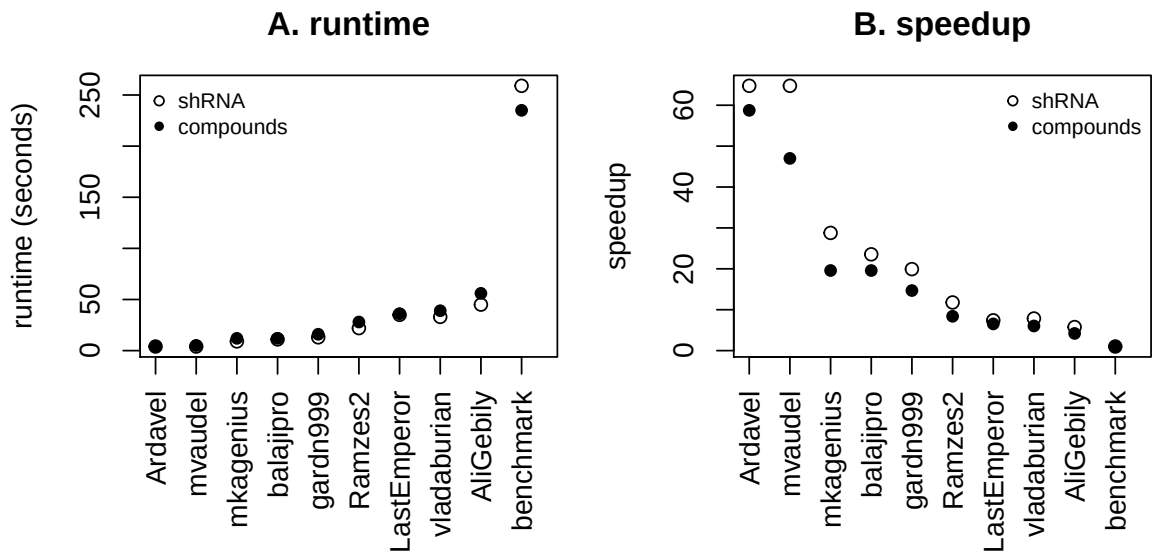


Figure 5: **Speed improvements.** Distribution of the per-plate runtime (in seconds) and speedups over the benchmark ($t_{\text{benchmark}}/t_{\text{competitor}}$) for each of the competitors' algorithms

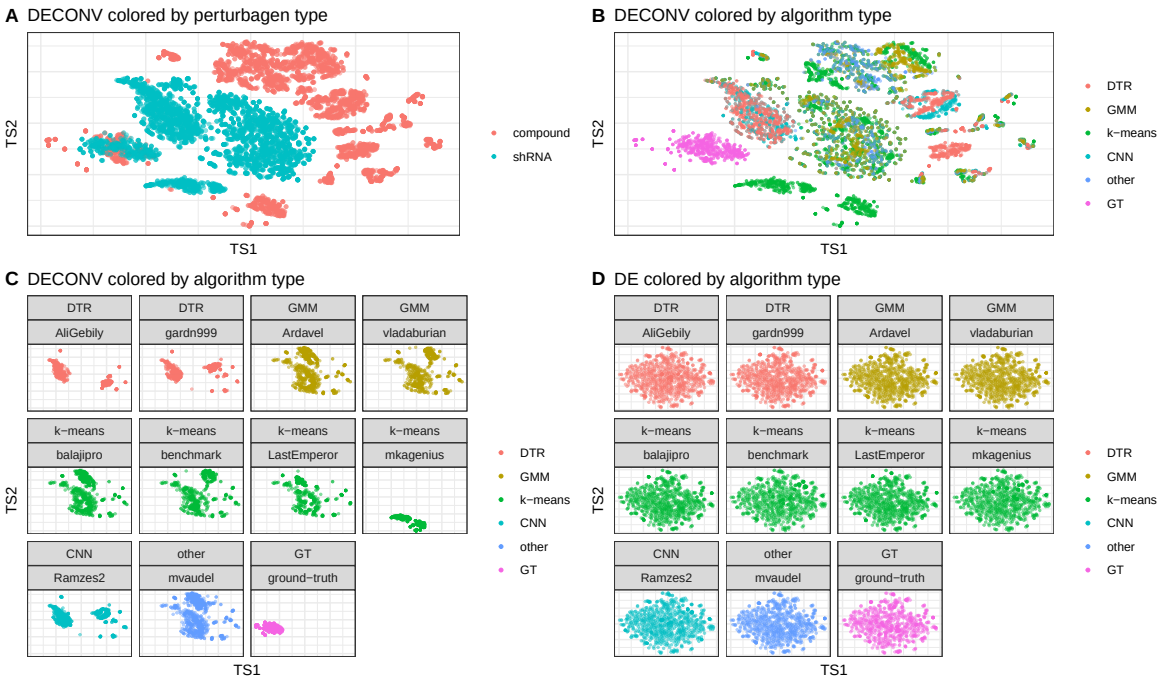
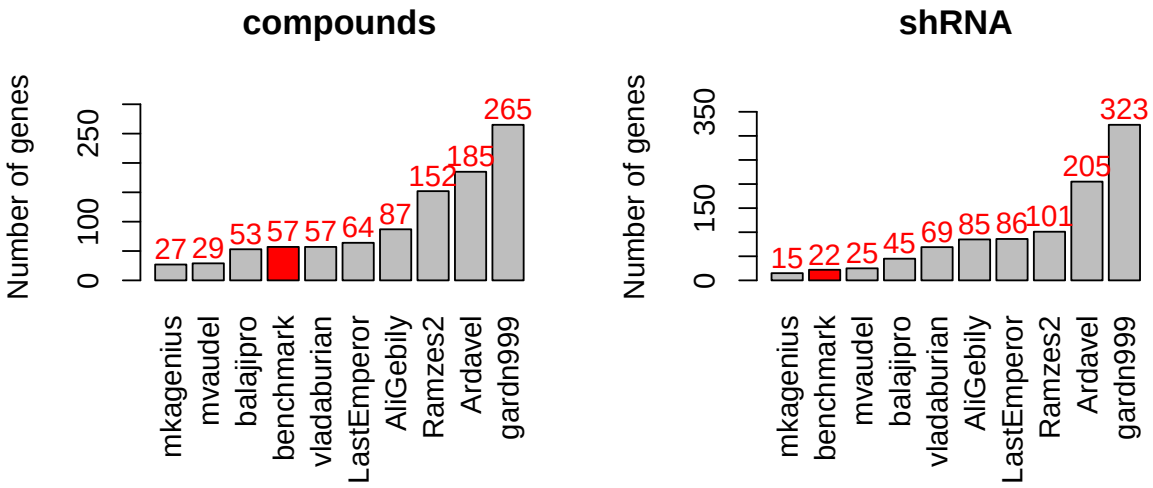
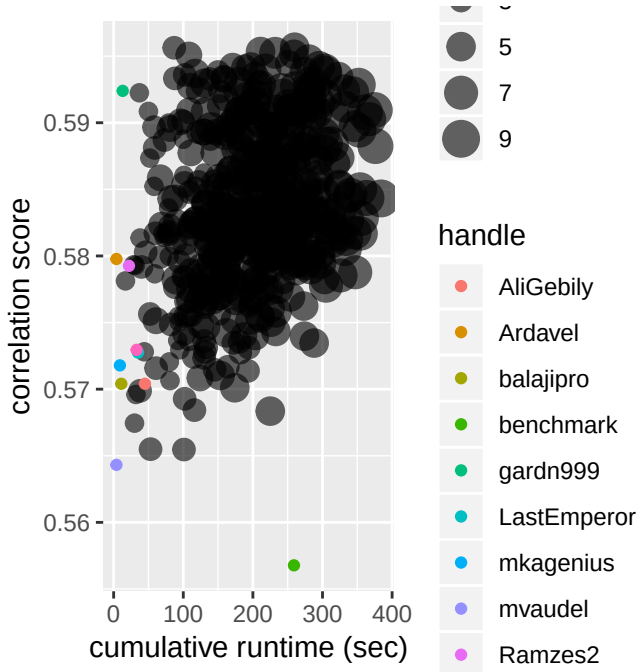


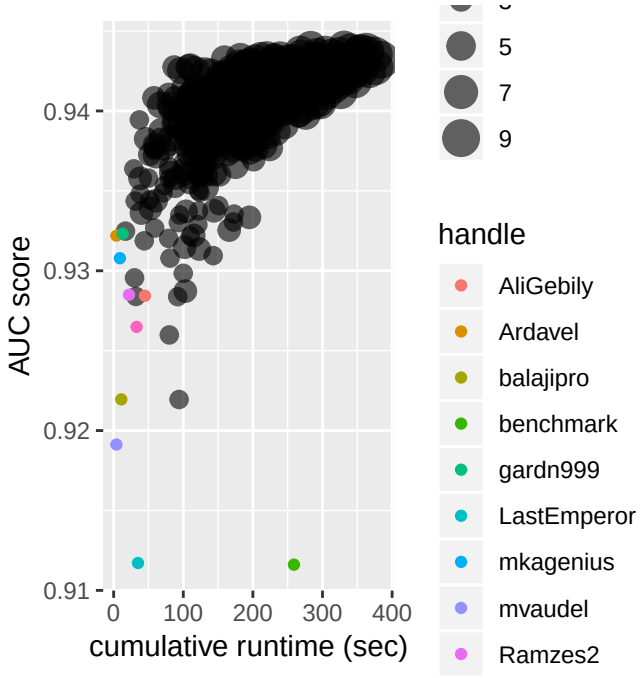
Figure 6: **t-SNE projection of deconvoluted data.** Each point represents the 2D projection of a sample generated by UNI ground truth (GT) or by applying a deconvolution algorithm to DUO data. t-SNE was run on the 2 plates of holdout data, one each containing compound and shRNA treatmens. DECONV data 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.



TODO: barplot of # of genes for which each algo. gives best prediction - possible group by plate/low vs. high

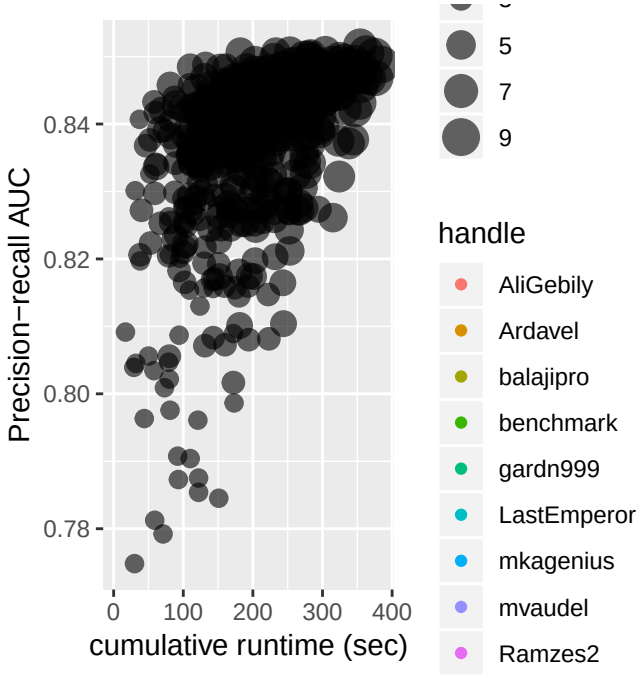
6.4 Ensemble

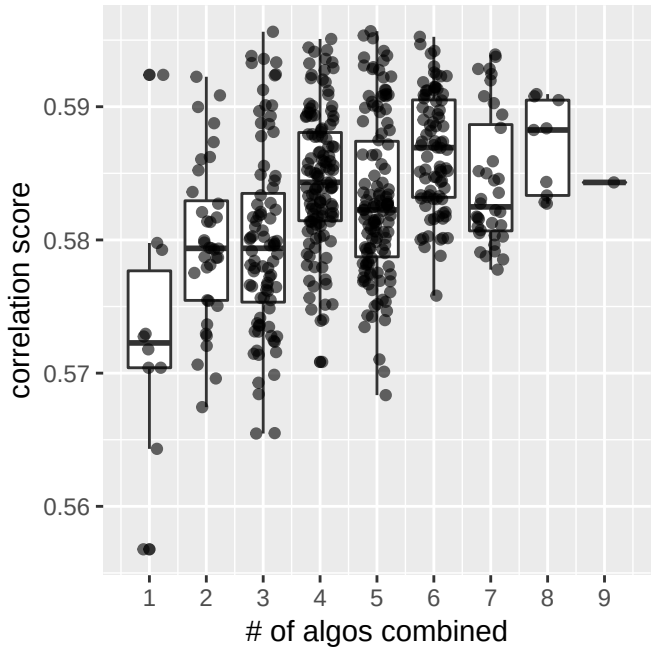


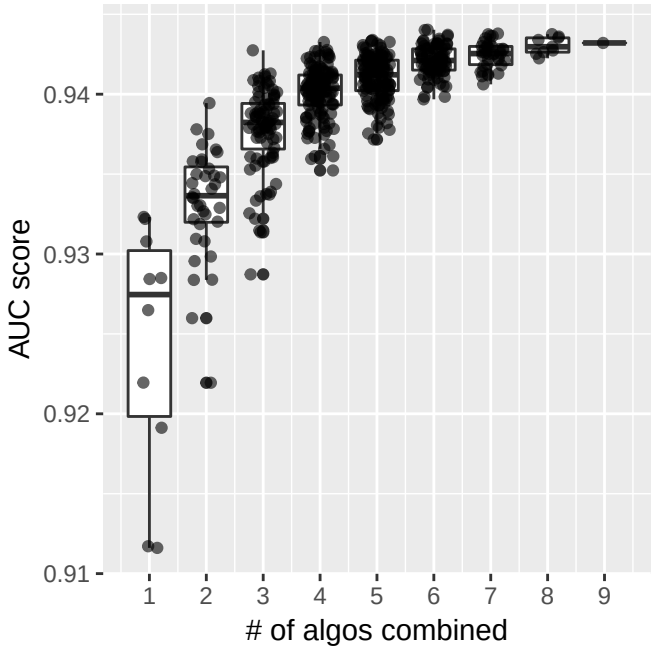



```
## Warning: Removed 10 rows containing missing values (geom_point).
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```

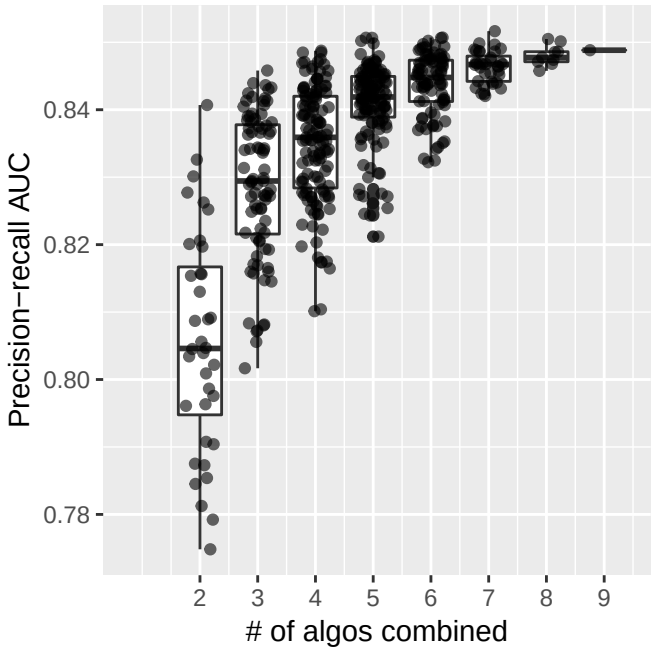







```
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
```

```
## Warning: Removed 10 rows containing missing values (geom_point).
```


Precision−recall AUC
0.84
0.82
0.80
0.78
2
3
4
5
6
7
8
9
# of algos combined

Supporting information

S1 Table.

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

pert_iname	pert_id	pert_itime	pert_idose	num_replicates
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

pert_iname	pert_id	pert_itime	pert_idose	num_replicates
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
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`).

pert_iname	pert_id	num_replicates
ABCB6	TRCN0000060320	4
ADI1	TRCN0000052275	4
ALDOA	TRCN0000052504	4
ANXA7	TRCN0000056304	4
ARHGAP1	TRCN0000307776	4
ASAH1	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

pert_iname	pert_id	num_replicates
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
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

pert_iname	pert_id	num_replicates
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
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

pert_iname	pert_id	num_replicates
WIPF2	TRCN0000029833	4
YME1L1	TRCN0000073864	4
ZW10	TRCN0000155335	4

S3 Table

Top finishers. This table lists the top 9 finishers and the languages and algorithms each used.

Table 3: Summary of contestant solutions

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

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 bilogically 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 pertubation 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,

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

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:

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

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.

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.

.0.1 Benchmark k-means solution

CMap's current solution to this problem is based on a k-means clustering algorithm called $dpeak$ that works as follows:

For each measurement, the dpeak partitions the list of realizations into $k >= 2$ distinct clusters and identifies two of the clusters whose ratio of membership is as close as possible to 2:1. The algorithm then takes the median intensity of each of the two clusters, assigning these values to the appropriate gene (i.e., matching clusters with more observations to the gene mixed in higher proportion).

After deconvoluting each sample on a plate, dpeak then uses the plate-wide distributions to perform adjustments on a per-well basis, correcting peaks that may have been misassigned (see Appendix).

Known problems with the current approach are that k-means is generally a biased and inconsistent estimator of the peaks of a bimodal distribution [ref]. It also sometimes fails to detect peaks with few observations or it incorrectly identifies these peaks as extraneous and disregards them. Another limitation is that it is computationally expensive (the current Matlab implementation takes about 30 minutes on a 12-core server to process one set of 384 experiments).

Additionally, because half the landmark genes are measured using two-fold less bead than the other half, these low-proportion genes are subject to increased variability, which the benchmark k-means solution does not mitigate. Hence, there is an unequal noise distribution in the benchmark's deconvoluted expression profiles.

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] Shai S Shen-Orr et al. "Cell type–specific gene expression differences in complex tissues". In: *Nature methods* 7.4 (2010), p. 287.

[3] 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.

[4] Georg C Terstappen et al. "Target deconvolution strategies in drug discovery". In: *Nature Reviews Drug Discovery* 6.11 (2007), p. 891.

[5] Karim R Lakhani et al. "Prize-based contests can provide solutions to computational biology problems". In: *Nature biotechnology* 31.2 (2013), p. 108.

[6] Andrea Blasco et al. "Advancing Computational Biology and Bioinformatics Research Through Open Innovation Competitions". In: *bioRxiv* (2019), p. 565481.

[7] Laurens van der Maaten and Geoffrey Hinton. "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov (2008), pp. 2579–2605.