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

Author 1 Author 2 ...

Last updated: Oct 08, 2019

Abstract

Report results fo open innovation competition aimed at solving a gene-related deconvolution problem.

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

Contents

Instructions for submission

TODOS:

- Check length requirements for Nat. Met.
- Intro [AB]
- Methdos TN [DONE]
- Clustering
- Add disaggregated data

> A Brief Communication is a more concise format used typically to report a significant improvement to a tried-and-tested method, its modification and adaptation to an important original application, or an important new tool or resource of broad interest for the scientific community. This format typically does not exceed 3 printed pages. Brief Communications begin with a brief unreferenced abstract (3 sentences, no more than 70 words), which will appear on Medline. The title is limited to 10 words (or 90 characters). The main text is typically 1,000-1,500 words, including the abstract and contains no headings with the exception of a single heading for Methods to point readers to the online Methods section providing all technical details necessary for the independent reproduction of the methodology. Brief Communications normally have no more than 2 display items, although this may be flexible at the discretion of the editor, provided the page limit is observed. As a guideline, Brief Communications allow up to 20 references, and article titles are omitted from the reference list.

Brief Communications include received/accepted dates. They may be accompanied by supplementary information. Brief Communications are peer reviewed.

1 Introduction

Over the last decade, the ability to analyze multiple analytes simultaneously from the same biological sample has contributed to creating large perturbational datasets that are essential to accelerate the discovery of novel therapeutics. However, a fundamental limitation of traditional multianalyte assays lies in the restricted extent and type of available analytes.[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 methods and statistical analysis to measure simultaneously gene expressions for multiple genes using the same analyte type, thus multiplying the capacity of existing technologies [1].

A central component of this approach is the deconvolution of signals of different gene types acquired from mixed-gene samples. 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 D-peak, 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 type within the mixed-type sample (see Subramanian et al. [1] for the details). This approach works well in practice but has several limitations as well; 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. 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 L1000 platform; we varied the detection mode for acquiring the data between two genes per analyte color (DUO) and one gene per analyte color (UNI); then we 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 by offering cash incentives to competitors and by assessing the accuracy of different approaches using the UNI data as the ground truth.

The top approaches, as they emerged from the challenge, included very popular machine-learning methods, such as Random Forests and Convolutional Neural Networks (CNNs), as well as more

[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.

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

Towards developing an open innovation competition, we first transformed the deconvolution problem into a supervised classification task. Using the L1000 assay, we profiled six 384-well plates, each containing different sets of compound and shRNA treatments (see Supporting Info, 8.1). The same samples were detected using two different methods. The first method, called UNI, associates each gene to one single bead color, which leads to higher costs, but does not require deconvolution. The second method, called DUO, meaures two different genes on the same bead color, which reduces costs but requires deconvolution to transform the composite signal into two separate gene-specific expression values. We then used the data obtained from the UNI method (one gene per bead color) as the "ground truth" for the competition; and the data obtained from the DUO method (two genes per bead color) as the input to predict the equivalent UNI data. The datasets are now publicly available [LINK to REPO].

We randomly split the generated data into training, testing, and holdout datasets of equal size (2 plates each). All the contestants had access to the training data to develop and their solutions offline. The testing data were used to evaluate solutions during the contest and populate the live leaderboard. Holdout data were used to evaluate competitors' final submissions and guard against over-fitting. Prizes were awarded based on performance on the holdout dataset.

To evaluate submissions, we developed a scoring function that combines standard measures of accuracy based on the agreement between the predicted values and the ground truth, as well as computational speed (see Supporting Info, 8.3).

The competition run on the platform Topcoder (Wipro, India) for 10 days. A total of $23,000 in cash prizes was offered to competitors as incentive to be divided among the top 9 submissions (prize split: $8000, $6000, $4000, $2000, $1000, $800, $600, $400, $100).

3 Results

3.1 Participation

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. Table 1 lists the top 9 finishers and the languages and algorithms each used.

Table 1: 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

3.2 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. 3*). 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.

3.2.1 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

3.2.2 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.

3.3 Ensemble approaches

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

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

3.4 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

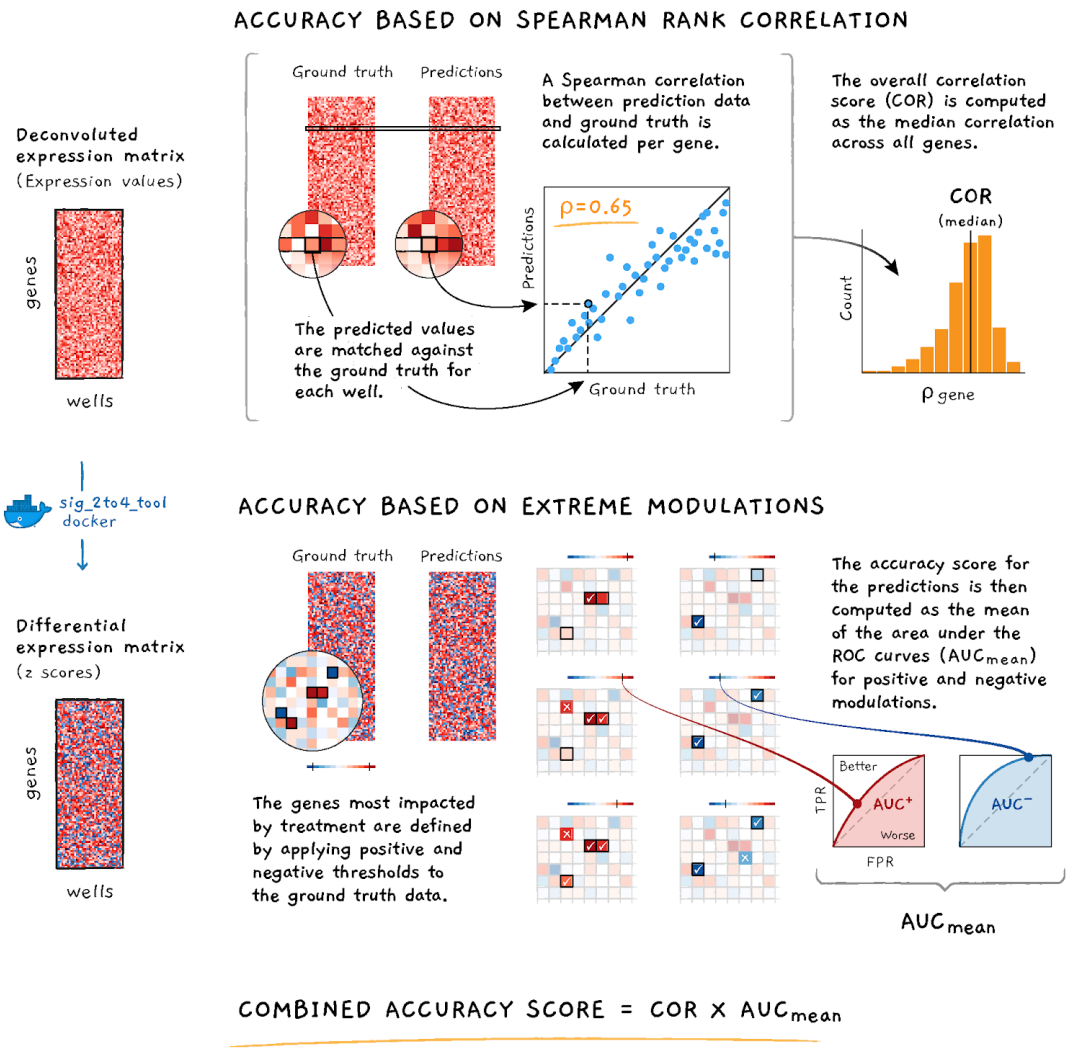


Figure 1: **Schematic illustrating accuracy components of scoring function**. The accuracy component is computed as the product of gene-wise Spearman correlations with ground truth and the area under the curve AUC of extreme modulations.

5 Figures

5.1 Scoring accuracy

5.2 Accuracy vs. Speed

Median difference

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

```
## done
```

6 KD accuracy & recall

6.1 Inter-replicate variance

6.2 Variance by experiments

7 Runtime and speedups

7.1 Accuracy

7.2 Gene knockdowns

7.3 High / Low bead discrepancy

7.4 Clustering of solutions

7.5 Complementarity

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

```
## done
```

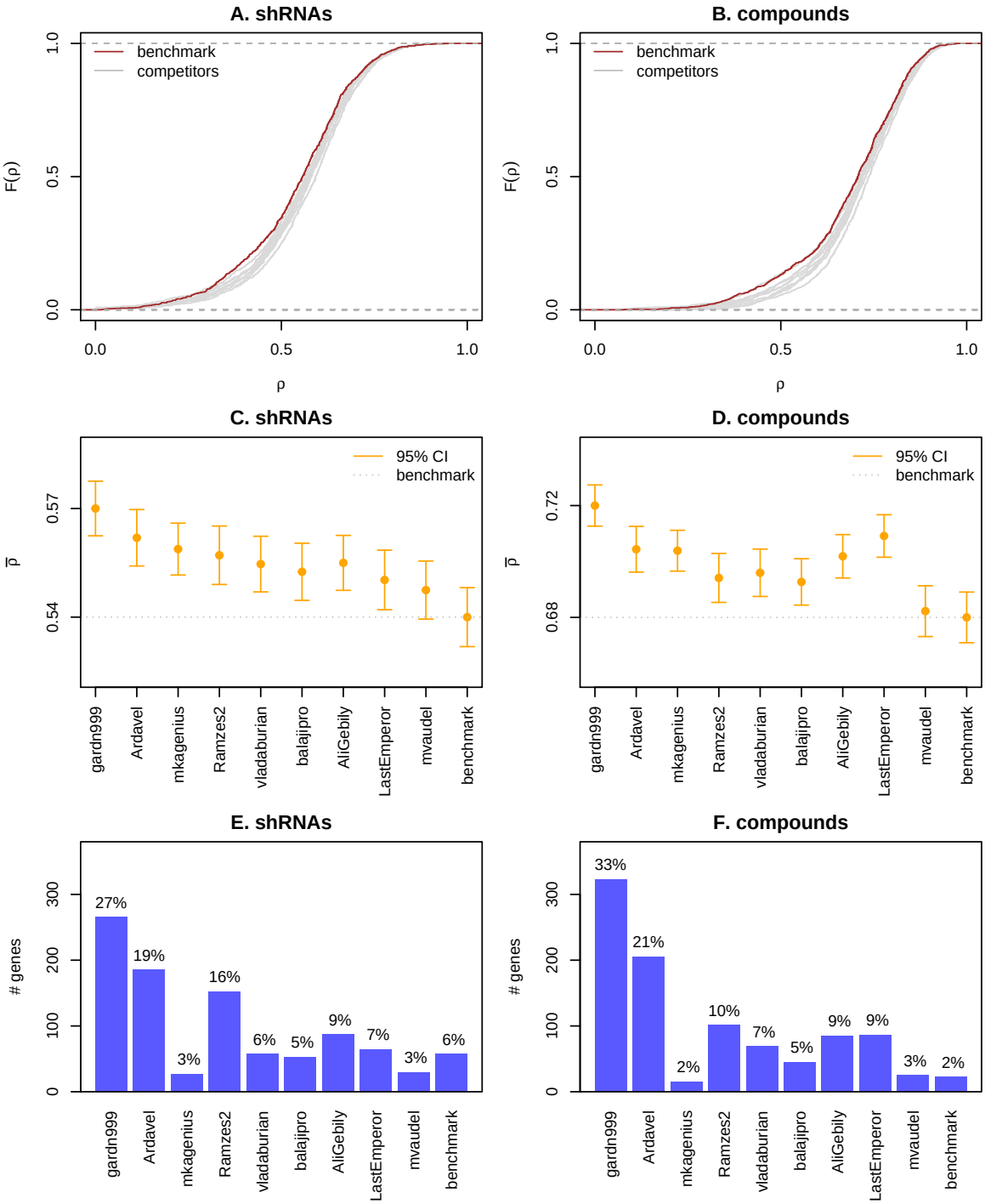


Figure 2: **Accuracy.** Top panels show empirical CDF of the distribution of the genewise spearman correlation (ρ) between the ground-truth gene-expressions (as detected by UNI) and predictions obtained by the competitors and the benchmark through the deconvolution of DUO data for the subset of shRNA (**A**) and compound experiments (**B**). The competitors (x-axis) are ordered by their final ranking in the contest. 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**). 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**).

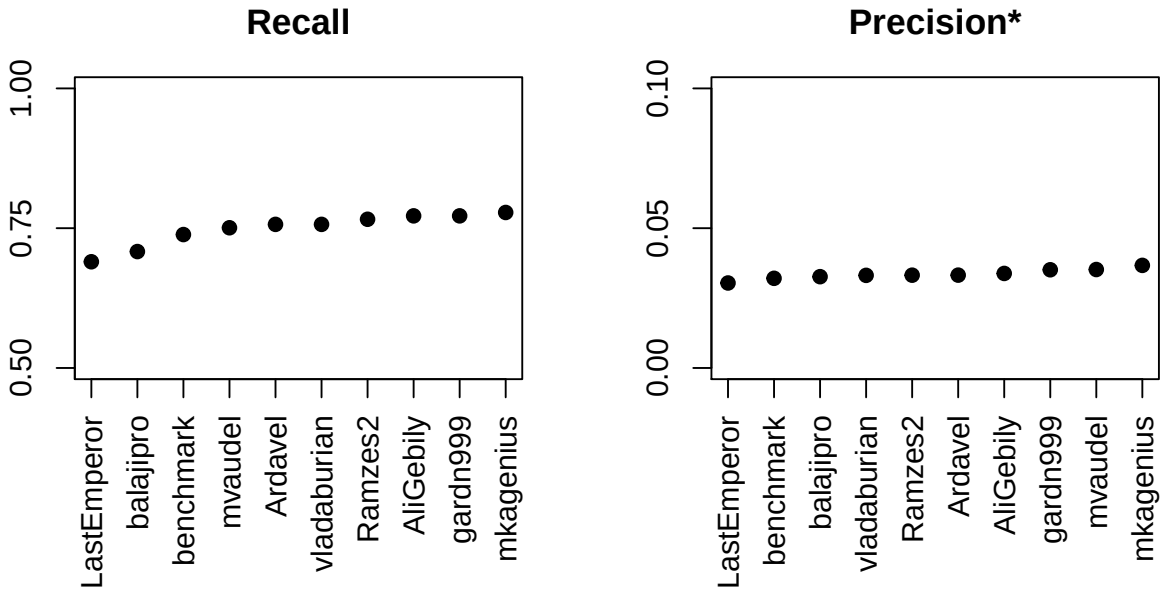


Figure 3: **Gene knockdown.** Precision and fraction of gene knockdown recalled by each algorithm. Precision is low because …

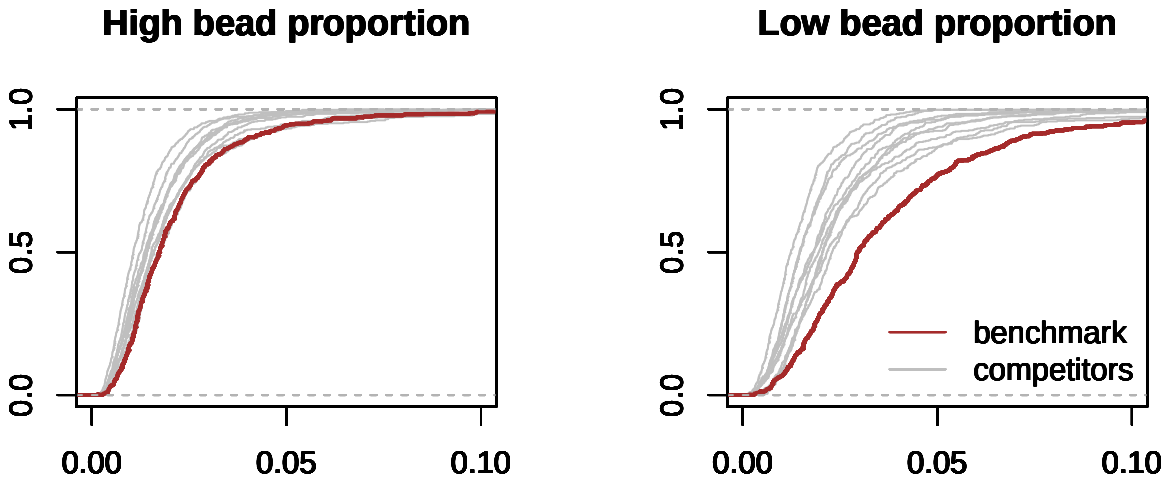


Figure 4: Inter-replicate variance.

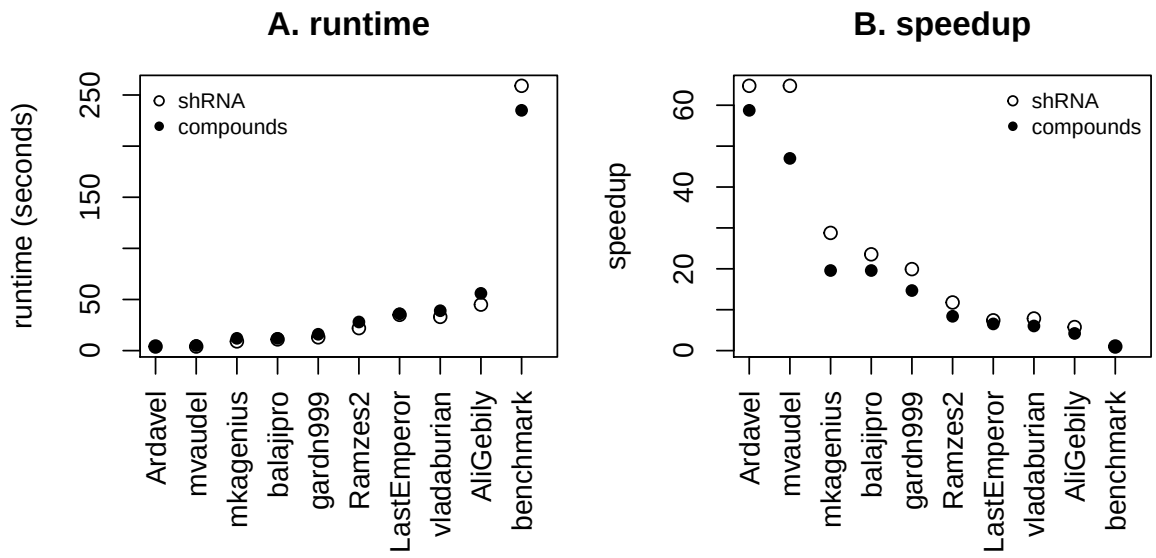


Figure 5: **Speed improvements.** Distribution of the per-plate runtime (in seconds) and speedups over the benchmark ($t_{\text{benchmark}}/t_{\text{competitor}}$) for each of the competitors' algorithms

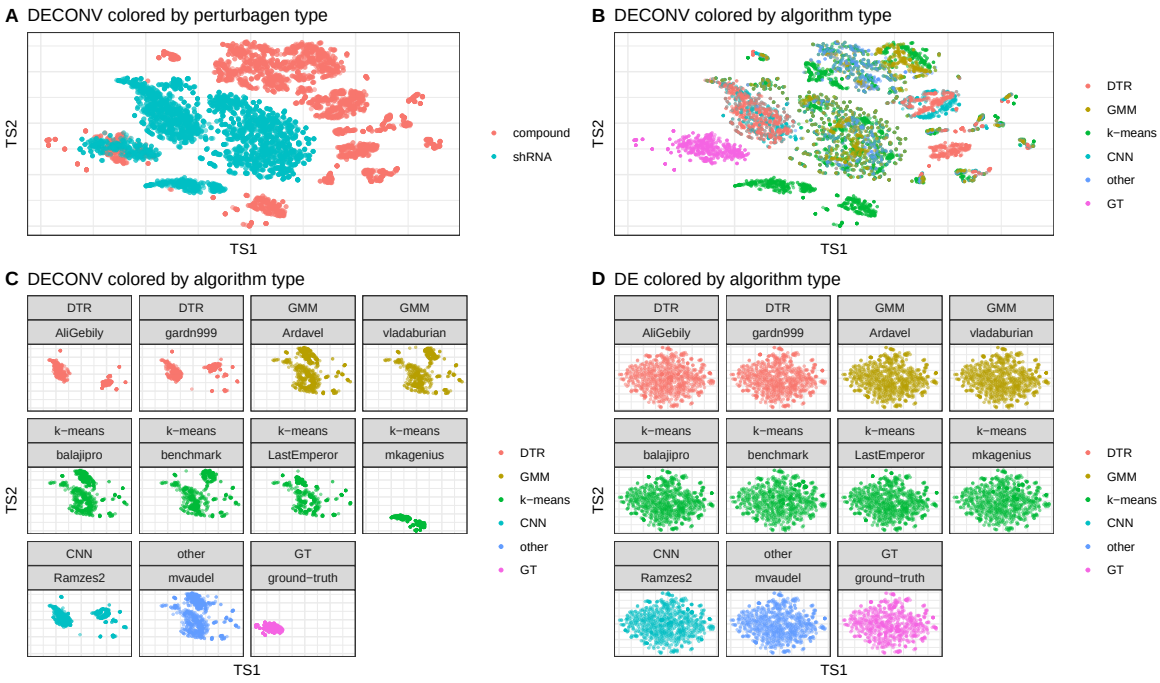
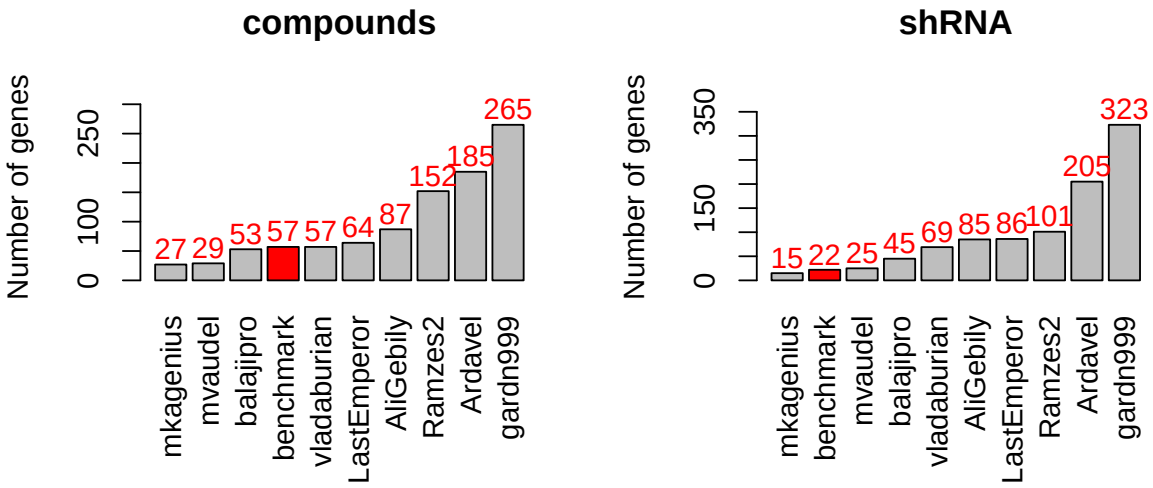
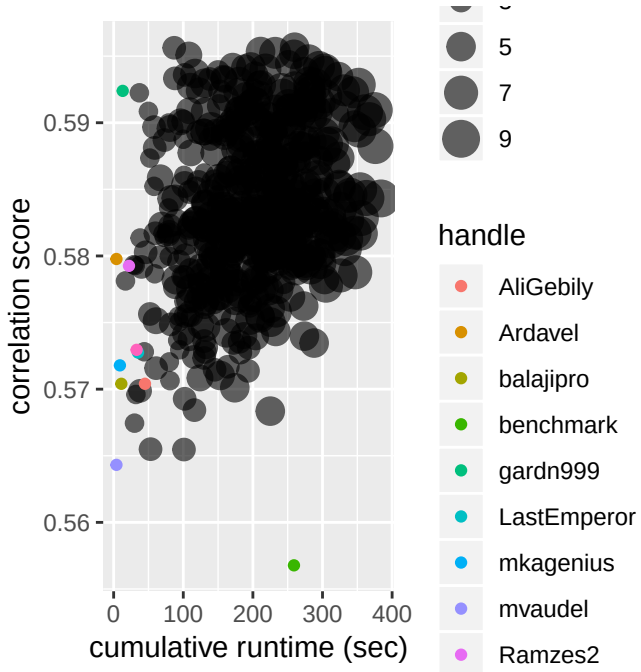


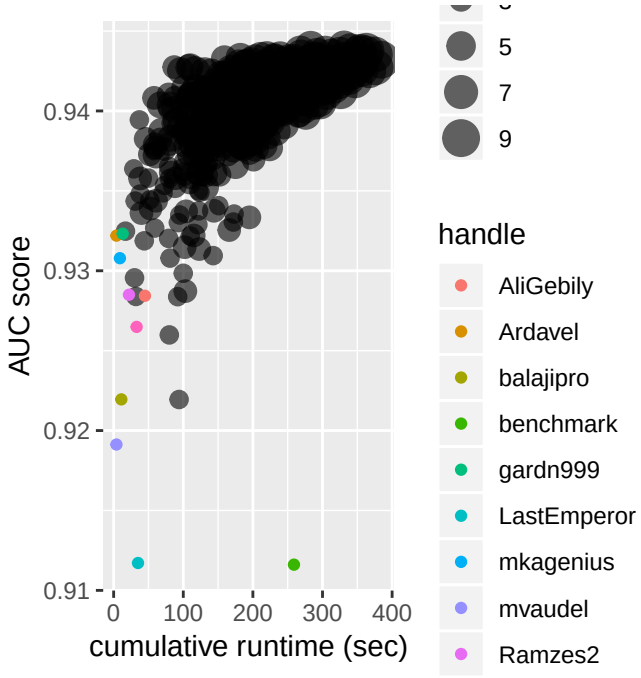
Figure 6: **t-SNE projection of deconvoluted data.** Each point represents the 2D projection of a sample generated by UNI ground truth (GT) or by applying a deconvolution algorithm to DUO data. t-SNE was run on the 2 plates of holdout data, one each containing compound and shRNA treatmens. DECONV data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

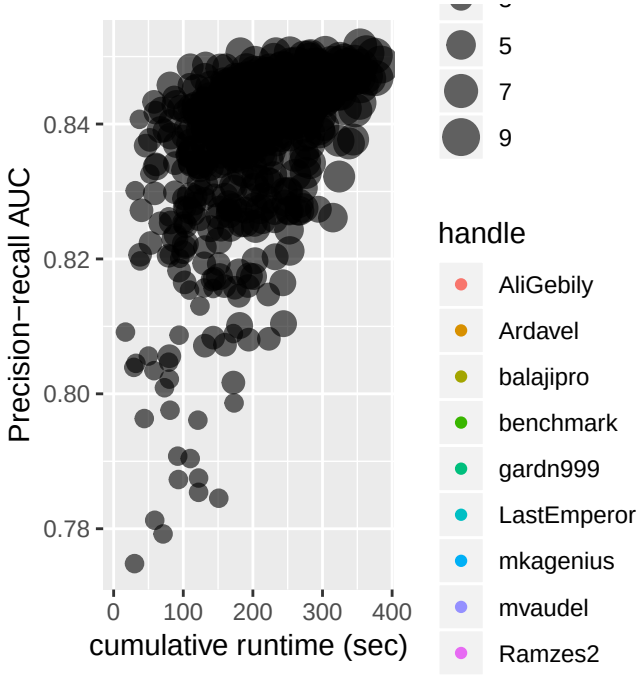
7.6 Ensemble

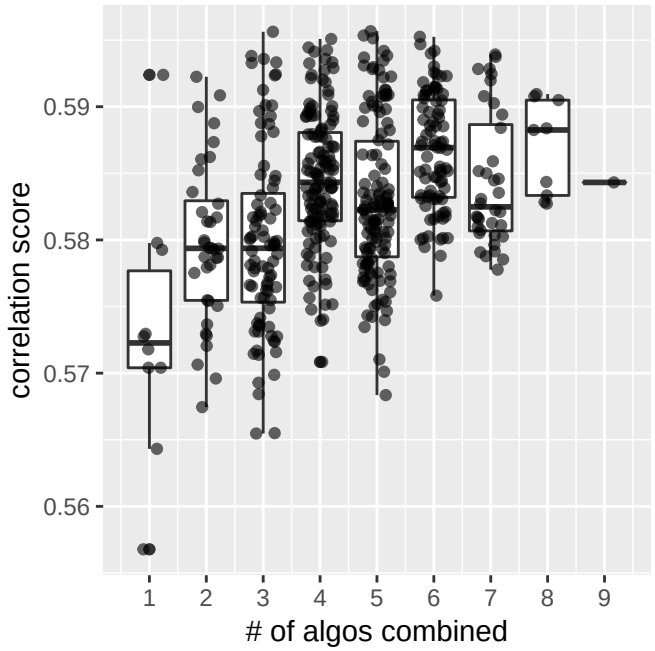


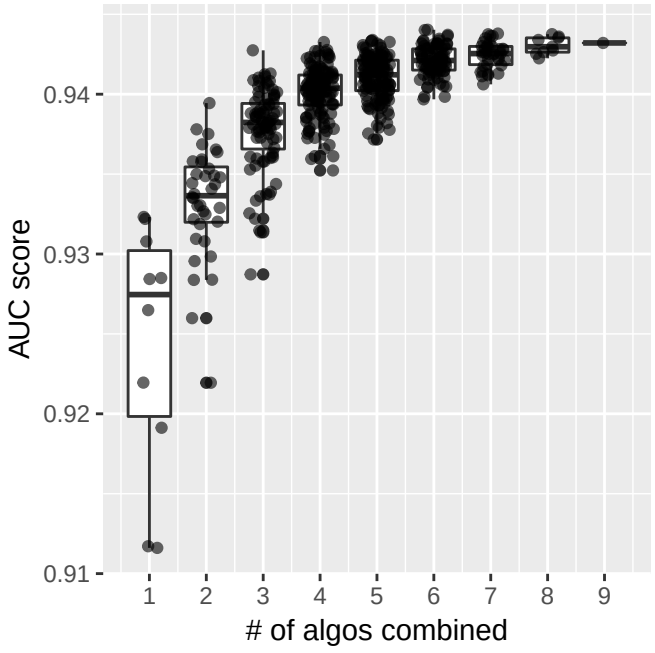



```
## Warning: Removed 10 rows containing missing values (geom_point).
```

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

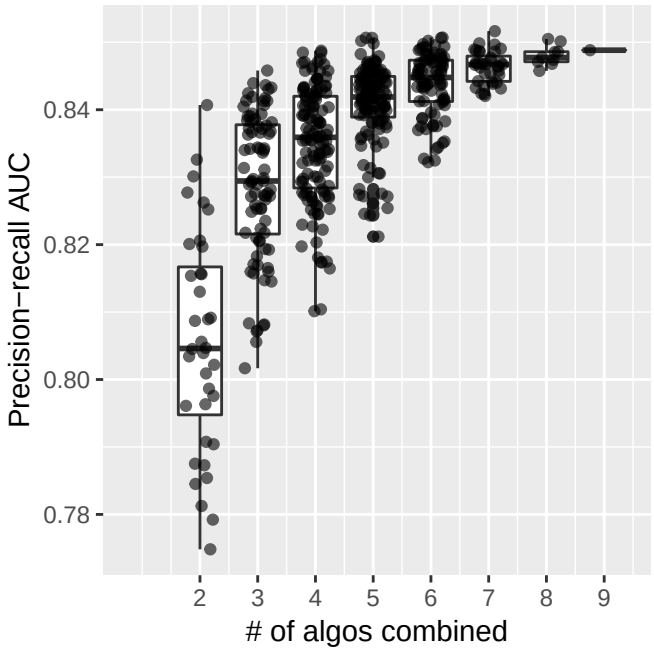







```
## Warning: Removed 10 rows containing non-finite values (stat_boxplot).
```

```
## 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

8 Supporting information

TODO: update equations to use Latex

8.1 Data generation for contest

To generate data for this contest, we profiled six 384-well perturbagen plates, each containing mutually exclusive sets of compound and shRNA treatments. 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 2017). The amplicon was then split and detected in both UNI and DUO detection modes. The three pairs of data were arbitrarily assigned to training, testing, and holdout categories, where in each case the UNI data served as the ground truth. Training data were made available for all the contestants to develop and validate their solutions offline. The testing data were used to evaluate solutions during the contestn and populate the live leaderboard. Holdout data were used evaluate competitors' final submissions and guard against over-fitting. Prizes were awareded based on performance on the holdout dataset.

Table 2: Data generated

Category	Type
training	Compounds
testing	Compounds
holdout	Compounds
training	shRNA
testing	shRNA
holdout	shRNA

8.2 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.

8.2.1 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.

8.3 Scoring function

Contest submissions were scored based on accuracy and speed.

8.3.1 Accuracy

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 (See figure XX).

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. 2017[ref].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.

8.3.1.1 Accuracy based on Spearman correlation

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 MDUO, p and MUNI, 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 between the rows of MDUO, p and the rows of MUNI, p; 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:

CORp = median(diag(spearman(MDUO, p, MUNI, p))

8.3.1.2 Accuracy based on AUC of extreme modulations

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 bead type a in a given dataset p, let AUCp, c denote the corresponding area under the curve where c = { high | low }, where high means UNIDE, p >= 2, and low means UNIDE, p <= -2; then, compute the arithmetic mean of the area under the curve per class to obtain the corresponding score per dataset:

$$\mathrm{AUC}_p = (\mathrm{AUC}_p, hight + \mathrm{AUC}_p, low)/2.$$

8.3.2 Speed

For a given dataset p, the speed component of the score is computed as the run time in seconds for deconvoluting the data in each plate.

8.3.3 Aggregegated score

The accuracy and speed components were integrated into a single aggregate scores as follow:

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

where $T_{\text{benchmark}}$ is the deconvolution time required by the reference D-Peak implementation.

8.3.4 Accuracy based on knockdown predictions

In addition to the Spearman correlation and extreme modulation AUC metrics used in the contest, we were also able to asses each algorithms ability to correctly predict the successful knockdown (KD) of landmark genes. The shRNA experiments used to generate the contest data were constructed such that each shRNA specifically targeted one of the landmark genes. Hence, the expectation in each of those experiments is that the targeted landmark gene should exhibit a greatly reduced expression level, which should manifest in a very low z-score in DE data. Additionally, we expect that the targeted landmark gene should be most dramatically down-regulated in the samples in which it was directly targeted. To assess this, we compute a gene-wise rank by DE z-score across all samples in a given plate. Criteria for indicating a successful KD are $zs <= -2 ANDrank <= -10$.

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.