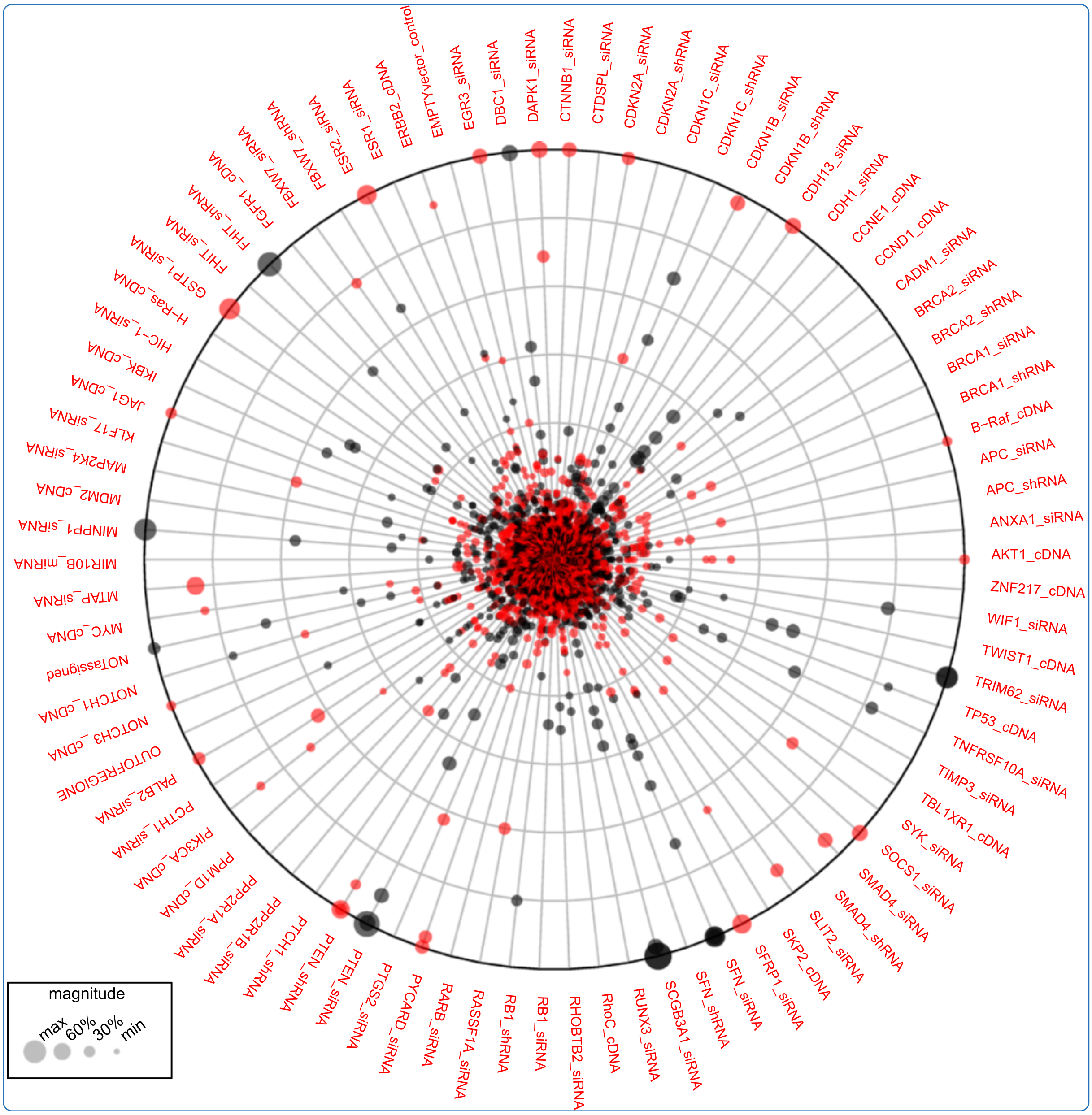
BMC
Bioinformatics



TOPS: a versatile software tool for statistical analysis and visualization of combinatorial gene-gene and gene-drug interaction screens

Muellner *et al.*

BioMed Central

Muellner *et al. BMC Bioinformatics* 2014, **15**:98
http://www.biomedcentral.com/1471-2105/15/98

BMC Bioinformatics

SOFTWARE Open Access

TOPS: a versatile software tool for statistical analysis and visualization of combinatorial gene-gene and gene-drug interaction screens

Markus K Muellner[1*†], Gerhard Duernberger[1,2†], Florian Ganglberger[1†], Claudia Kerzendorfer[1], Iris Z Uras[1], Andreas Schoenegger[1], Klaudia Bagienski[1], Jacques Colinge[1] and Sebastian MB Nijman[1*]

Abstract

Background: Measuring the impact of combinations of genetic or chemical perturbations on cellular fitness, sometimes referred to as synthetic lethal screening, is a powerful method for obtaining novel insights into gene function and drug action. Especially when performed at large scales, gene-gene or gene-drug interaction screens can reveal complex genetic interactions or drug mechanism of action or even identify novel therapeutics for the treatment of diseases.

The result of such large-scale screen results can be represented as a matrix with a numeric score indicating the cellular fitness (e.g. viability or doubling time) for each double perturbation. In a typical screen, the majority of combinations do not impact the cellular fitness. Thus, it is critical to first discern true "hits" from noise. Subsequent data exploration and visualization methods can assist to extract meaningful biological information from the data. However, despite the increasing interest in combination perturbation screens, no user friendly open-source program exists that combines statistical analysis, data exploration tools and visualization.

Results: We developed TOPS (Tool for Combination Perturbation Screen Analysis), a Java and R-based software tool with a simple graphical user interface that allows the user to import, analyze, filter and plot data from double perturbation screens as well as other compatible data. TOPS was designed in a modular fashion to allow the user to add alternative importers for data formats or custom analysis scripts not covered by the original release.

We demonstrate the utility of TOPS on two datasets derived from functional genetic screens using different methods. Dataset 1 is a gene-drug interaction screen and is based on Luminex xMAP technology. Dataset 2 is a gene-gene short hairpin (sh)RNAi screen exploring the interactions between deubiquitinating enzymes and a number of prominent oncogenes using massive parallel sequencing (MPS).

Conclusions: TOPS provides the benchtop scientist with a free toolset to analyze, filter and visualize data from functional genomic gene-gene and gene-drug interaction screens with a flexible interface to accommodate different technologies and analysis algorithms in addition to those already provided here. TOPS is freely available for academic and non-academic users and is released as open source.

Keywords: Double perturbation screens, Synthetic lethality, Functional genetics, Massive parallel sequencing, Luminex xMAP, Drug screens, Synergy, Epistasis

* Correspondence: mmuellner@cemm.oeaw.ac.at; snijman@cemm.oeaw.ac.at
†Equal contributors
[1]Research Center for Molecular Medicine of the Austrian Academy of Sciences (CeMM), Vienna, Austria
Full list of author information is available at the end of the article

Background

Genes operate in complex cellular networks, and elucidating this connectivity is critical for understanding normal physiology and disease. Functional genomic screens that combine two perturbations have been used with great success to uncover such genetic interactions, and also to reveal mechanisms of drug action [1,2]. This is generally done by perturbing a biological system, most often a cell, in a defined manner i.e. by overexpression or knockdown of a gene or by addition of a drug and subsequently administering a second set of perturbations and analyzing the system for non-additive readout (i.e. synergistic/antagonistic or synthetic lethality/synthetic rescue) [3].

These perturbation experiments have traditionally been performed in model organisms such as yeast where genetic manipulation is relatively easy compared to mammalian cells [2]. Now, with the advent of technologies including RNA interference (RNAi) and the recent emergence of nuclease-based genome engineering, human cells can also be used to perform functional genomic screens. Cellular fitness, often represented by cell number, division rate or death, is used as a proxy for many of these screens since it is a parameter that can be measured with relative ease. This proxy is also of particular relevance in case genetic interactions are sought as potential anti-cancer therapies. Here, the aim is to kill only cells carrying cancer mutations by targeting an interacting gene which is in synthetic lethal configuration to the mutation. This can be achieved by targeting this second gene product with a small molecule inhibitor or antibody. The non-cancer cells would remain unaffected from this therapy as inhibiting only one interaction partner will not lead to lethality. Indeed, many cancer mutations can be tested systematically against drugs to uncover selective resistance/sensitivity mechanisms, or against small interfering RNAs to find potential new drug targets [4-6].

A pairwise wise interaction screen yields a matrix of interaction (or fitness-) scores. For two non-overlapping sets of perturbations A and B the number of measurements and statistical tests T required to evaluate all unique pairwise combinations is given by:

$$T = |A| \times |B|$$

In practice, the number of data points typically generated is further increased to T x n by the requirement of n replicate measurements. Thus, a screen of, for example, 100 genes against 100 drugs measured in quadruplicate requires 40,000 measurements. A Dataset of this size is usually beyond the limits of what can be conveniently analyzed without the use of dedicated statistics software.

To generate these large datasets, multiplexing strategies greatly improve the throughput of the experiment, by combining genetic perturbations in a single well (multiplexing drug treatments is typically not feasible). A convenient multiplexing strategy is to introduce DNA barcodes into the genome of mutant cell lines such that each genetic perturbation corresponds to a single barcode sequence. In this scenario DNA barcodes act both as unique identifiers for each genetic perturbation and as a proxy for cell number/fitness. Barcodes can be encoded by short hairpin (sh)RNA sequences themselves or introduced into the genome as non-transcribed DNA sequences using lentiviral vectors [7,8]. Practically, a set of barcoded cell lines (A) can then be pooled together and divided into aliquots that are then screened against the second set (B) of perturbations.

In our laboratory we have generated multiplexed gene-gene and gene-drug interaction data using two distinct technologies [Figure 1]:

(1) Luminex xMAP – a hybridization based multiplexing technology that allows up to 500 measurements in a single sample. Here, DNA barcoded panels consisting of genetically modified cell lines (i.e. expressing an oncogene or knockdown of a tumor suppressor gene) and an unmodified control from the same genetic background are pooled in a single well of a multiwell plate (genetic perturbation, set A) and treated with either a drug or a second genetic perturbation (drug perturbation, set B). In this case only perturbation set A is multiplexed.

(2) Massive Parallel Sequencing (MPS). In this case, a panel of non-barcoded cell lines carrying perturbation set A from (1) are infected with a library of lentiviral shRNA vectors (genetic perturbation, set B). Here, the shRNA sequence that integrate into the genome serves both as perturbation and as DNA barcode. Further multiplexing can be achieved by indexing the second perturbation in a single sequencing run. For example, a MPS run yielding 10^8 aligned reads on average allows 100 cell lines to be screened against 100 drugs at a reasonable average sequencing depth of 10,000 reads per DNA barcode.

In both cases the abundance of DNA barcodes amplified from the genomic DNA is measured after a period of exponential cell growth and used as a score for cellular fitness. To deconvolute, analyze and visualize this type of data in a modular and intuitive manner we established TOPS.

Implementation

Programming Language, external software and system requirements

TOPS was developed as a graphical user interface for an analysis pipeline that we originally developed in R. In

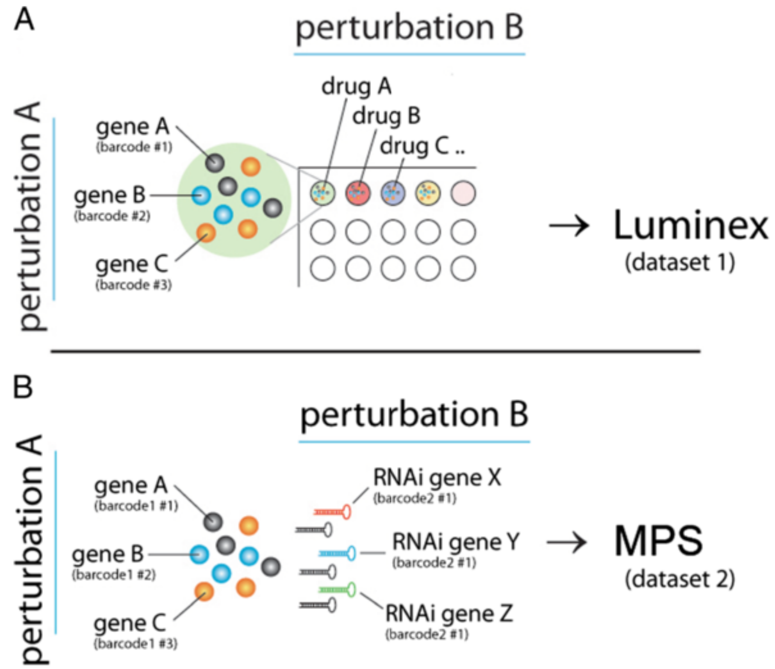


Figure 1 Overview of the two discussed multiplexing technologies used for gene-drug and gene-gene interaction studies. (A) Luminex multiplexing for a gene-drug interaction screen. The gene perturbation is multiplexed and encoded as a DNA barcode that can be hybridized to Luminex beads and quantified. Drugs are not multiplexed. **(B)** MPS double multiplexing screen for gene-gene interactions. The first (genetic) perturbation can be multiplexed as a Illumina index barcode, and the second perturbation is identified by sequencing the hairpin cassette, which acts as a second DNA barcode.

order to make these R scripts accessible to non-expert users, in particular scientists with little background in scripting languages, we took advantage of the R package rJava (http://www.rforge.net/rJava/) to run the scripts from a Java based interface. TOPS can therefore run its R scripts on all platforms for which rJava is available (currently MacOS, Linux and Windows).

The hardware requirements to run TOPS depend mostly on the volume of data to be analyzed and the algorithm to be used. We have developed and tested TOPS on two independent datasets derived from different experimental approaches, and on a standard office computer with a 2.4 GHz dual core processor and 2.5 GB of RAM. We note that for larger datasets the linear model (linearModel.R) algorithm provided with this release requires more memory than is usually available on an office computer. Therefore, we have provided an alternative that splits the analysis in smaller parts (linearModel_split.R) and later assembles the results. This modification necessitated an adjustment in the data normalization [Figure 2].

Output from the R interface, including and error messages, is stored in text files for troubleshooting. The overall progress of a session can be monitored and is logged separately.

User skills and installation

Installation of TOPS consists of unpacking the zipped file into a directory. For TOPS to run, Java, R and the package rJava are must also be installed. Upon first launch, definition of the paths in which R and rJava are located may be required. The most common locations for MacOS, Linux and Windows are checked automatically and the user will be prompted only if the libraries cannot be found. For convenience, alternative locations to be checked in this manner can be added in libPaths.txt in the TOPS directory.

TOPS uses a single semicolon separated values (CSV) data input file. This file has to be generated by the user and can for instance be exported from a Microsoft Office Excel spreadsheet. Alternatively the example datasets can be used for testing. To use TOPS no scripting skills are necessary.

Results and discussion

Example datasets provided

TOPS comes with two datasets to test the different procedures. Thus, users can explore the software without having to generate original data.

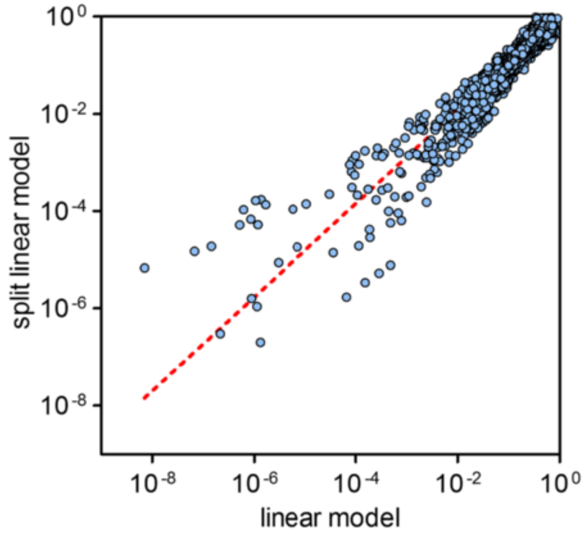


Figure 2 Correlation of p-values for linear and split linear model algorithms. Splitting the data set into smaller subsets and normalizing as well as analyzing each of them separately only introduces minor deviations in the resulting candidate hits (i.e. 39 of the top 50 hits in one method are among the respective top 50 of the other method).

Example dataset 1: a gene-drug interaction screen in breast cancer

To simulate oncogenic events in breast cancer we created a panel of 70 different isogenic cell lines ectopically expressing oncogenes or knocking down tumor suppressor genes commonly found in breast cancer. Findings based on this dataset have been published previously [7].

For multiplexing purposes, each cell line was infected with a vector introducing a genetic barcode (xMAP tag) into the genomic DNA. The cell lines were then pooled, distributed among multi-well plates and treated with 87 individual drugs. After a period of exponential growth, genomic DNA was harvested and barcodes were quantified using Luminex xMAP beads. Raw bead data was converted to median measurements per interaction by calculating the truncated mean. The resulting data is in the format 'Cell line (name)', 'Drug (name)', 'Barcode score (numerical)'. The barcode score is a measure of the relative fitness as it is an estimate of the total number of cells for each isogenic cell line under the drug treatment condition.

Example dataset 2: a gene-gene interactions screen in breast cancer

We screened 27 isogenic cell lines expressing oncogenes against a library consisting of approximately 400 shRNA vectors covering 80 deubiquitinating enzymes (DUBs). DUBs represent an emerging class of cancer targets involved in many different cellular processes with therapeutic potential. After a period of exponential growth cells were harvested and genomic DNA was extracted in order to PCR-amplify the shRNA barcode cassettes. During the amplification process primers with 60 unique barcode sequences corresponding to the 27 cell lines in 4 replicates were used as described in [9]. The tagged PCR samples were then pooled, a sequencing library was generated and the samples were sequenced on two lanes (50 basepairs) of a Hiseq 2000 instrument (Illumina). The MPS raw data were converted to FASTQ format and split according to the indexing PCR primers with a custom Python script. Each sample's NGS reads were aligned to the shRNA sequences using shALIGN [10] allowing for zero, one, and two mismatches. The output CSV file is in the format: 'Cell line (name)', 'shRNA (name)', 'number of reads'. Number of reads corresponds to shRNA abundance and this is the readout for relative fitness.

Importing data

A typical input file consists of five columns with the following information:

> perturbation A; perturbation B; fitness score; replicate 1; replicate 2

In the provided example dataset 1, perturbation A corresponds to the gene (i.e. overexpression or knockdown) and perturbation B is a drug treatment. For dataset 2, perturbation B is an shRNA vector targeting a specific DUB and perturbation A is the gene. Fitness score represents the barcode abundance in these examples. Replicates 1 and 2 designate if measurements are replicates of each other. Replicate 1 identifies multiple perturbations towards the same outcome (i.e. multiple different hairpins for one gene, multiple vectors encoding the same cDNA) while replicate 2 identifies the replicate number for a given A-B combination.

TOPS has custom importers covering both types of files from the example datasets. These importers convert the data coming from Luminex or MPS to the input file format.

Entries with non-unique identifiers for perturbation A and perturbation B are automatically interpreted to be replicate measurements by the importers.

Pre-analysis filtering

We implemented a filtering method to allow the user to detect and remove bad replicates or samples before running the analysis as this can result in false positive hits. This filtering is based on the assumption that the frequency of gene-gene or gene-drug interactions is low among all possible combinations [11]. Therefore noisy perturbations can be identified by how well they correlate with others. For each perturbation a pairwise Spearman correlation is calculated against all other perturbations in the dataset. A poor correlation then indicates a technically noisy sample that the user may wish to exclude from further analysis. We perform this correlation for both A and B perturbations and correlation cutoffs can be chosen for each individually [Figure 3].

Analysis algorithms

TOPS is supplied with two analysis models that are well suited for interaction data of the type that is supplied in the example datasets. However, custom analysis methods can be conveniently plugged into TOPS. The first method is based on a multiple linear regression model and the second one is based on the Mann-Whitney U Test.

Statistical analysis: linear model

Linear models have been used previously to normalize [12,13] and analyze [14,15] high-throughput measurements. Accordingly, our analysis is composed of two consecutive linear models: the first one is used for data normalization; the second one identifies outliers that represent "hits" i.e. potentially biologically meaningful interactions that warrant further study.

Data normalization relies on the assumption that relative fitness values $intensity_i$ as determined by barcode quantification (via MPS or Luminex) are constituted of four components:

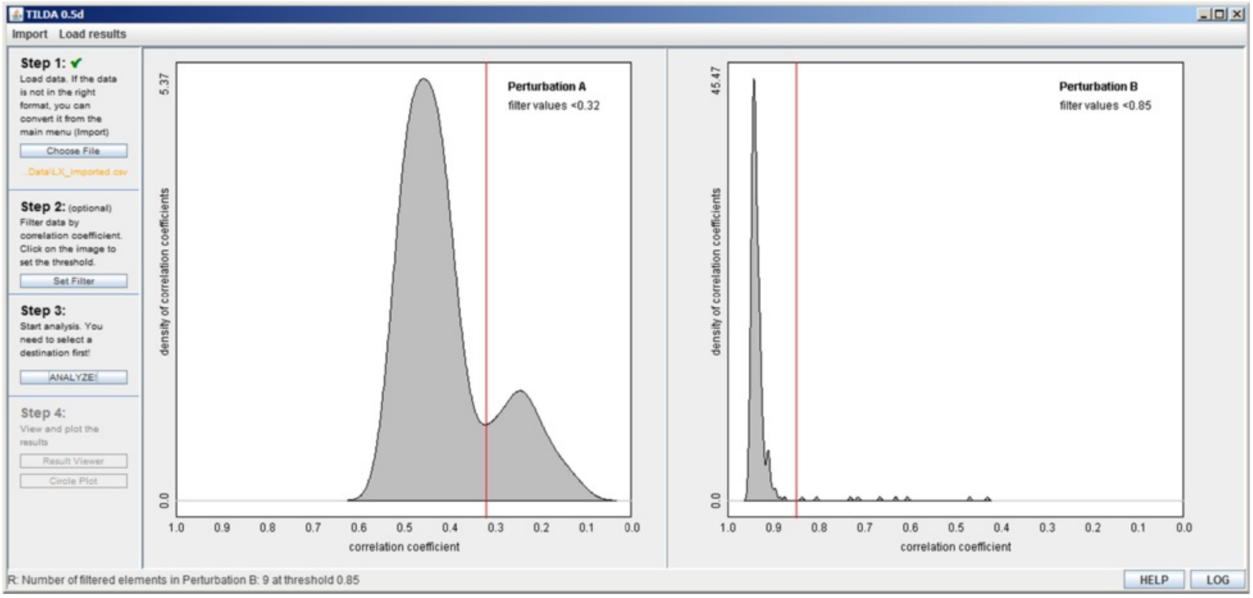
1) the systematic effect of perturbation A on fitness
2) the systematic effect of perturbation B on fitness
3) the signal that is due to the (synergistic) combination of perturbation A and B consistent across biological replicates, which is the signal of biological interest
4) noise caused by biological and technical variability between the replicates.

These four components can be modeled as:

$$intensity_i = \beta_A\, A + \beta_B\, B + s_i$$

where $intensity_i$ represents the measured fitness score, β_A, β_B are regression coefficients, A, B are categorical variables representing perturbations A and B respectively, and s_i is a remaining signal including both noise (biological and technical) and a potential synergistic contribution of A and B perturbations. The model is fit to the experimental data to estimate the first two components from the fitness score corresponding to the systematic, non-synergistic perturbations induced by A and B. The linear model was fit by robust regression using an M estimator (function rlm in R library "MASS") to avoid influence of outliers.

The second linear model further dissects the component s_i to identify potential synergy. Since we want to analyze the effect of each perturbation on the other one, s_i is dissected twice by fixing either the perturbation A or B first. Namely, fixing A first, we can reduce s_i to the signal observed for each successive value of A, which we



Step 1: ✓
Load data. If the data is not in the right format, you can convert it from the main menu (Import)

Step 2: (optional)
Filter data by correlation coefficient. Click on the image to set the threshold.

Step 3:
Start analysis. You need to select a destination first!

Step 4:
View and plot the results

R: Number of filtered elements in Perturbation B: 9 at threshold 0.85

Figure 3 Pre-analysis filtering for both sets of perturbations. Spearman correlations are plotted as a histogram for both sets of perturbations. Noisy perturbations can be removed before normalization by changing the cutoff slider according to the desired stringency.

denote by $s_{i|A}$. On this reduced data, the effect of perturbation B is estimated by the model:

$$\forall A : s_{i|A} = \beta_{B|A} B + \varepsilon$$

where A, B are categorical variables as above, $\beta_{B|A}$ the regression coefficient, and ε a term capturing the contribution of noise. The same procedure is applied when perturbation B is fixed first:

$$\forall B : s_{i|B} = \beta_{A|B} A + \varepsilon$$

The coefficients of these linear models ($\beta_{A|B}$ and $\beta_{B|A}$) capture the effect of a perturbation within the context of a specific other perturbation. The coefficients combine the signal of replicate experiments and estimate the magnitude of the biological effect thanks to the noise component ε.

The 'rlm' function of the R 'MASS' library is employed to compute a robust estimate of the coefficients of the linear model. This function also offers the possibility to calculate the statistical significance of each coefficient by performing a one-sided t-test ($\beta_{A|B} \neq 0$ or $\beta_{B|A} \neq 0$). For small sample sizes this test can suffer from false positives due to underestimating variance as described in Axelsson et al. [16].

As both linear models are applied after log transformation of the input data these additive coefficients reflect modeling a multiplicative effect of the fitness response of the pairwise fitness effects $\beta_A$and β_B [17,18].

In the case of the sLM split linear model aimed at reducing computation time and memory usage, the dataset is subdivided into groups of 50 perturbations (A and B), resulting in less data points available for the t-test, hence the reduced statistical power.

Statistical analysis: Mann-Whitney U-test

The second method we implemented to analyze fitness score data is based on the *Mann-Whitney U-test*, a more robust alternative to the *Student's t-test* that does not assume a normal distribution. For this analysis a distinct normalization procedure is employed.

First, numerical data was arranged in a matrix with rows and columns representing perturbations A and B, respectively. Next, we calculate a normalized value for each gene-gene or gene-drug "interaction" in the matrix as:

$$m_{ij} = \frac{A_{ij} B_{ij}}{median\left(A_j\right) median(B_i)}$$

Here, i and j represent row and column positions in the matrix, and A_j and B_i represent sets containing all values for row j and column i, respectively. Thus, each value in a row is divided by the corresponding row median, and each value in a column by the column median across the matrix. Each fitness score in the matrix is subsequently $\log_{10}$ transformed.

Next, the algorithm performs two-sided tests to calculate p-values. The model requires at least 3 replicates per interaction to perform this test. As with the linear method algorithm we calculate two p-values for each interaction, corresponding to perturbation set A or B.

Combining p-values

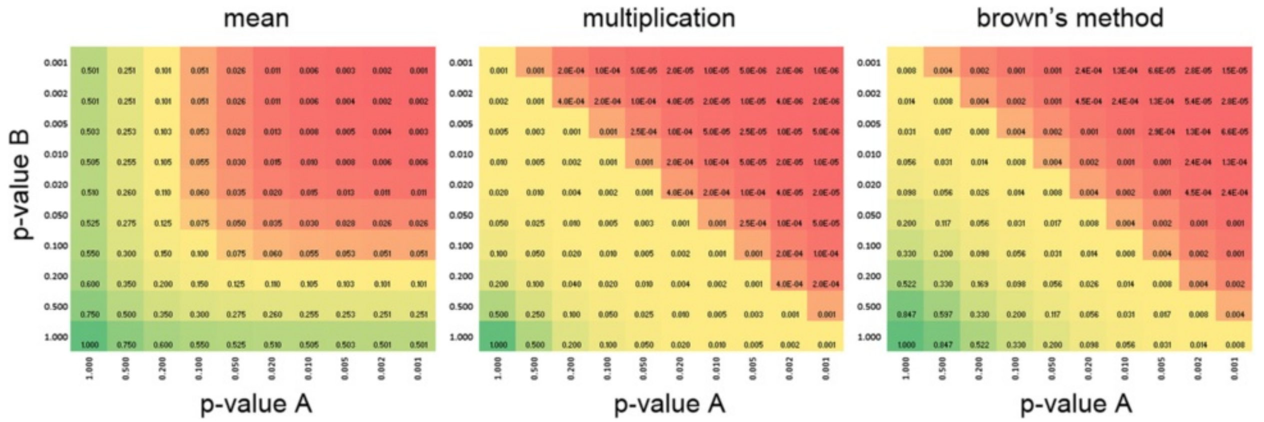
Both methods of analysis presented above calculate p-values in either direction of the interaction matrix. This means that either the distribution of A's or B's for a given A-B interaction matrix is used to derive the significance score. The distributions of A's and B's are likely to be different, depending on the assay technology used to generate the data. Therefore a pvalue derived from one distribution (pvalue A) is not necessarily equal to the pvalue derived from the second distribution (pvalue B) for the same A-B interaction. To obtain an overall score to rank hits, we tested three methods to combine p-values: (1) calculating the average of both values, (2) multiplication of both values with each other, and (3) Brown's Method [19], a modification of Fisher's method for combining dependent p-values [Figure 4]. Taking the average of two p-values strongly penalized fitness scores just below the significance level, potentially resulting in more false negatives while Brown's Method rewarded combinations where both p-values are close to significant. Multiplication of both p-values gave comparable but slightly more extreme results than Brown's. We therefore selected Brown's Method to calculate combined p-values as a conservative middle ground between the other two more extreme methods.

Importing analyzed data into the "results browser" and generating plots

Once data has been analyzed by either method, the results of the analysis are stored in a text file in the following format:

perturbation A; perturbation B; relative magnitude A; relative magnitude B; absolute magnitude A; absolute magnitude B; p-value A; p-value B; combined p-value by Brown's Method.

Where perturbation A and B are the identifiers for the tested perturbations, relative magnitude A and B gives an indication of the ratio of treatment versus the median of all treatments (control), absolute magnitude A and B are the raw signals from the assay and the p-values are calculated as discussed previously. Given this format, other datasets not generated with our analysis pipeline can also be imported, filtered and plotted using TOPS.



mean

p-value B	1.000	0.500	0.200	0.100	0.050	0.020	0.010	0.005	0.002	0.001
0.001	0.501	0.251	0.101	0.051	0.026	0.011	0.006	0.003	0.002	0.001
0.002	0.501	0.251	0.101	0.051	0.026	0.011	0.006	0.004	0.002	0.002
0.005	0.503	0.253	0.103	0.053	0.028	0.013	0.008	0.005	0.004	0.003
0.010	0.505	0.255	0.105	0.055	0.030	0.015	0.010	0.008	0.006	0.006
0.020	0.510	0.260	0.110	0.060	0.035	0.020	0.015	0.013	0.011	0.011
0.050	0.525	0.275	0.125	0.075	0.050	0.035	0.030	0.028	0.026	0.026
0.100	0.550	0.300	0.150	0.100	0.075	0.060	0.055	0.053	0.051	0.051
0.200	0.600	0.350	0.200	0.150	0.125	0.110	0.105	0.103	0.101	0.101
0.500	0.750	0.500	0.350	0.300	0.275	0.260	0.255	0.253	0.251	0.251
1.000	1.000	0.750	0.600	0.550	0.525	0.510	0.505	0.503	0.501	0.501

multiplication

p-value B	1.000	0.500	0.200	0.100	0.050	0.020	0.010	0.005	0.002	0.001
0.001	0.001	0.001	2.0E-04	1.0E-04	5.0E-05	2.0E-05	1.0E-05	5.0E-06	2.0E-06	1.0E-06
0.002	0.002	0.001	4.0E-04	2.0E-04	1.0E-04	4.0E-05	2.0E-05	1.0E-05	4.0E-06	2.0E-06
0.005	0.005	0.003	0.001	0.001	2.5E-04	1.0E-04	5.0E-05	2.5E-05	1.0E-05	5.0E-06
0.010	0.010	0.005	0.002	0.001	0.001	2.0E-04	1.0E-04	5.0E-05	2.0E-05	1.0E-05
0.020	0.020	0.010	0.004	0.002	0.001	4.0E-04	2.0E-04	1.0E-04	4.0E-05	2.0E-05
0.050	0.050	0.025	0.010	0.005	0.003	0.001	0.001	2.5E-04	1.0E-04	5.0E-05
0.100	0.100	0.050	0.020	0.010	0.005	0.002	0.001	0.001	2.0E-04	1.0E-04
0.200	0.200	0.100	0.040	0.020	0.010	0.004	0.002	0.001	4.0E-04	2.0E-04
0.500	0.500	0.250	0.100	0.050	0.025	0.010	0.005	0.003	0.001	0.001
1.000	1.000	0.500	0.200	0.100	0.050	0.020	0.010	0.005	0.002	0.001

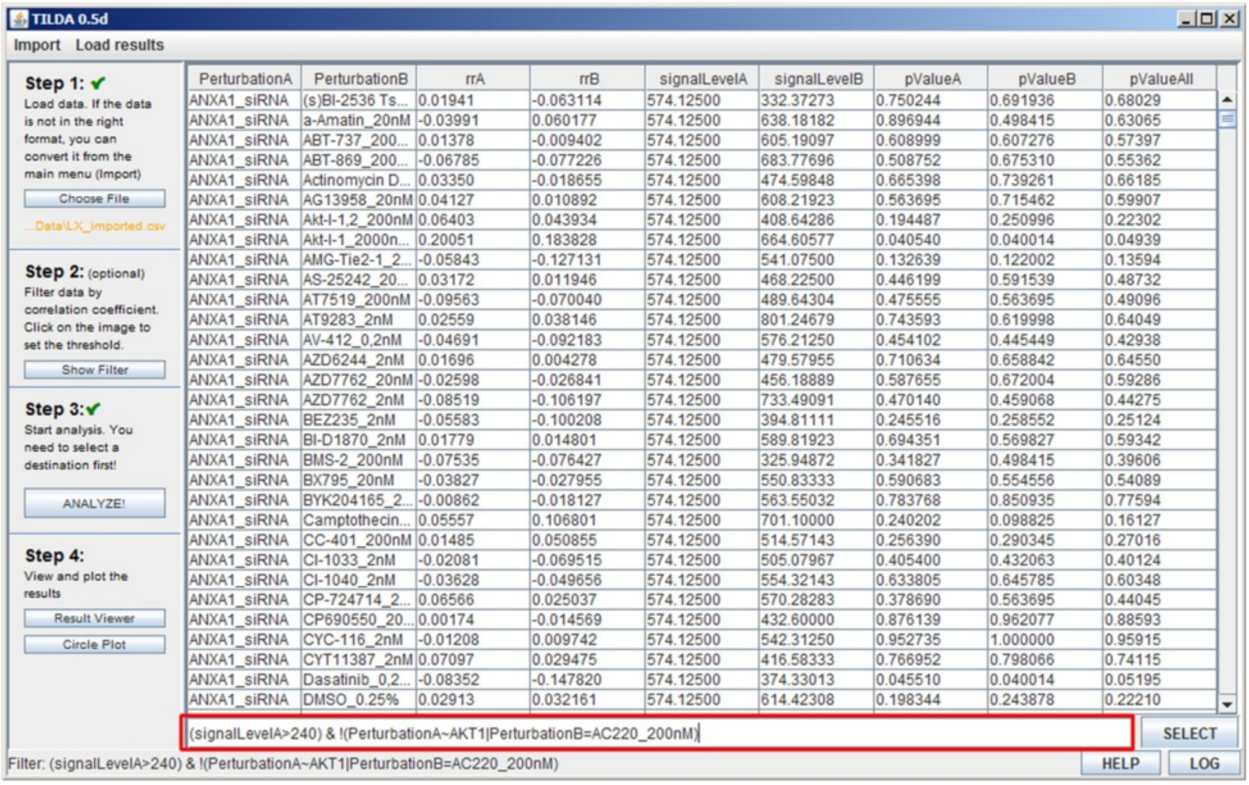
brown's method

p-value B	1.000	0.500	0.200	0.100	0.050	0.020	0.010	0.005	0.002	0.001
0.001	0.008	0.004	0.002	0.001	0.001	2.4E-04	1.3E-04	6.6E-05	2.8E-05	1.5E-05
0.002	0.014	0.008	0.004	0.002	0.001	4.5E-04	2.4E-04	1.3E-04	5.4E-05	2.8E-05
0.005	0.031	0.017	0.008	0.004	0.002	0.001	0.001	2.9E-04	1.3E-04	6.6E-05
0.010	0.056	0.031	0.014	0.008	0.004	0.002	0.001	0.001	2.4E-04	1.3E-04
0.020	0.098	0.056	0.026	0.014	0.008	0.004	0.002	0.001	4.5E-04	2.4E-04
0.050	0.200	0.117	0.056	0.031	0.017	0.008	0.004	0.002	0.001	0.001
0.100	0.330	0.200	0.098	0.056	0.031	0.014	0.008	0.004	0.002	0.001
0.200	0.522	0.330	0.169	0.098	0.056	0.026	0.014	0.008	0.004	0.002
0.500	0.847	0.597	0.330	0.200	0.117	0.056	0.031	0.017	0.008	0.004
1.000	1.000	0.847	0.522	0.330	0.200	0.098	0.056	0.031	0.014	0.008

Figure 4 Comparison of different methods for combining scores. Heatmap of the overall score out of the two individual p-values by either mean (left), multiplication (middle) or Brown's Method (right).

Post-analysis filtering

After the analysis is complete, the results are displayed in a table in the results browser [Figure 5] and can be sorted ascending or descending by clicking on the column headers. Here the user can sort and filter the obtained results before visualization by entering a string of filtering criteria. These criteria can be connected with the AND (&) or OR (|) operator and also allows for partial matches using the tilde (~) sign or exclusion using a logical NOT (!). Details on filtering strings can be found in TOPS's help files. The optimal filtering parameters vary with the technology used to obtain the data. In our experience, the mean Luminex signal (xMAP) and mean number of reads (MPS) are reasonable parameters to use for filtering out false positives.



Step 1: ✓
Load data. If the data is not in the right format, you can convert it from the main menu (Import)

...Data\LX_imported.csv

Step 2: (optional)
Filter data by correlation coefficient. Click on the image to set the threshold.

Step 3: ✓
Start analysis. You need to select a destination first!

Step 4:
View and plot the results

PerturbationA	PerturbationB	rrA	rrB	signalLevelA	signalLevelB	pValueA	pValueB	pValueAll
ANXA1_siRNA	(s)BI-2536 Ts...	0.01941	-0.063114	574.12500	332.37273	0.750244	0.691936	0.68029
ANXA1_siRNA	a-Amatin_20nM	-0.03991	0.060177	574.12500	638.18182	0.896944	0.498415	0.63065
ANXA1_siRNA	ABT-737_200...	0.01378	-0.009402	574.12500	605.19097	0.608999	0.607276	0.57397
ANXA1_siRNA	ABT-869_200...	-0.06785	-0.077226	574.12500	683.77696	0.508752	0.675310	0.55362
ANXA1_siRNA	Actinomycin D...	0.03350	-0.018655	574.12500	474.59848	0.665398	0.739261	0.66185
ANXA1_siRNA	AG13958_20nM	0.04127	0.010892	574.12500	608.21923	0.563695	0.715462	0.59907
ANXA1_siRNA	Akt-I-1,2_200nM	0.06403	0.043934	574.12500	408.64286	0.194487	0.250996	0.22302
ANXA1_siRNA	Akt-I-1_2000n...	0.20051	0.183828	574.12500	664.60577	0.040540	0.040014	0.04939
ANXA1_siRNA	AMG-Tie2-1_2...	-0.05843	-0.127131	574.12500	541.07500	0.132639	0.122002	0.13594
ANXA1_siRNA	AS-25242_20...	0.03172	0.011946	574.12500	468.22500	0.446199	0.591539	0.48732
ANXA1_siRNA	AT7519_200nM	-0.09563	-0.070040	574.12500	489.64304	0.475555	0.563695	0.49096
ANXA1_siRNA	AT9283_2nM	0.02559	0.038146	574.12500	801.24679	0.743593	0.619998	0.64049
ANXA1_siRNA	AV-412_0,2nM	-0.04691	-0.092183	574.12500	576.21250	0.454102	0.445449	0.42938
ANXA1_siRNA	AZD6244_2nM	0.01696	0.004278	574.12500	479.57955	0.710634	0.658842	0.64550
ANXA1_siRNA	AZD7762_20nM	-0.02598	-0.026841	574.12500	456.18889	0.587655	0.672004	0.59286
ANXA1_siRNA	AZD7762_2nM	-0.08519	-0.106197	574.12500	733.49091	0.470140	0.459068	0.44275
ANXA1_siRNA	BEZ235_2nM	-0.05583	-0.100208	574.12500	394.81111	0.245516	0.258552	0.25124
ANXA1_siRNA	BI-D1870_2nM	0.01779	0.014801	574.12500	589.81923	0.694351	0.569827	0.59342
ANXA1_siRNA	BMS-2_200nM	-0.07535	-0.076427	574.12500	325.94872	0.341827	0.498415	0.39606
ANXA1_siRNA	BX795_20nM	-0.03827	-0.027955	574.12500	550.83333	0.590683	0.554556	0.54089
ANXA1_siRNA	BYK204165_2...	-0.00862	-0.018127	574.12500	563.55032	0.783768	0.850935	0.77594
ANXA1_siRNA	Camptothecin...	0.05557	0.106801	574.12500	701.10000	0.240202	0.098825	0.16127
ANXA1_siRNA	CC-401_200nM	0.01485	0.050855	574.12500	514.57143	0.256390	0.290345	0.27016
ANXA1_siRNA	CI-1033_2nM	-0.02081	-0.069515	574.12500	505.07967	0.405400	0.432063	0.40124
ANXA1_siRNA	CI-1040_2nM	-0.03628	-0.049656	574.12500	554.32143	0.633805	0.645785	0.60348
ANXA1_siRNA	CP-724714_2...	0.06566	0.025037	574.12500	570.28283	0.378690	0.563695	0.44045
ANXA1_siRNA	CP690550_20...	0.00174	-0.014569	574.12500	432.60000	0.876139	0.962077	0.88593
ANXA1_siRNA	CYC-116_2nM	-0.01208	0.009742	574.12500	542.31250	0.952735	1.000000	0.95915
ANXA1_siRNA	CYT11387_2nM	0.07097	0.029475	574.12500	416.58333	0.766952	0.798066	0.74115
ANXA1_siRNA	Dasatinib_0,2...	-0.08352	-0.147820	574.12500	374.33013	0.045510	0.040014	0.05195
ANXA1_siRNA	DMSO_0.25%	0.02913	0.032161	574.12500	614.42308	0.198344	0.243878	0.22210

(signalLevelA>240) & !(PerturbationA~AKT1|PerturbationB=AC220_200nM)

Filter: (signalLevelA>240) & !(PerturbationA~AKT1|PerturbationB=AC220_200nM)

Figure 5 Screenshot of the results browser. Data from the analysis can be filtered by any of the parameters displayed in the header of the table. Filtering strings can be entered in the text field below the results table and data can be sorted by clicking on a column name.

Removing the bottom 5-10% of signals sorted by these parameters gave a satisfactory reduction in noise.

User modules

TOPS has been designed in a way that allows future users to easily write new importers in java to support other input data formats. We have also implemented the possibility to run user-written R-code for the data analysis which can then be called from within TOPS and fed into the data filtering and visualization [Figure 6]. The R scripts provided with this release can be used as templates to write custom modules as required.

Comparison of performance of the analysis models

In order to compare the performance of all three models (linear, split-linear, and Mann-Whitney U-test) we generated a randomized dataset with a distribution based on dataset 1. In this set we generated true positive hits with an average signal of 0.5, 0.7 (reduced fitness) 1.3 and 1.5 fold (increased fitness) compared to the average of true negatives.

To estimate false positive and false negatives, we used this dataset to determine receiver operating characteristics (ROC) for all three algorithms [Figure 7]. Overall, all algorithms show comparable characteristics with the U-Test performing slightly better at lower-fitness hits, and the linear models being more sensitive for increased fitness hits. In general, decreased fitness hits were more readily detected by our algorithms due to the non-normal distribution of the data. In terms of run-time performance the U-Test and Split Linear Model algorithm are practically identical while the Linear Model algorithm is slightly more computationally costly [Figure 8].

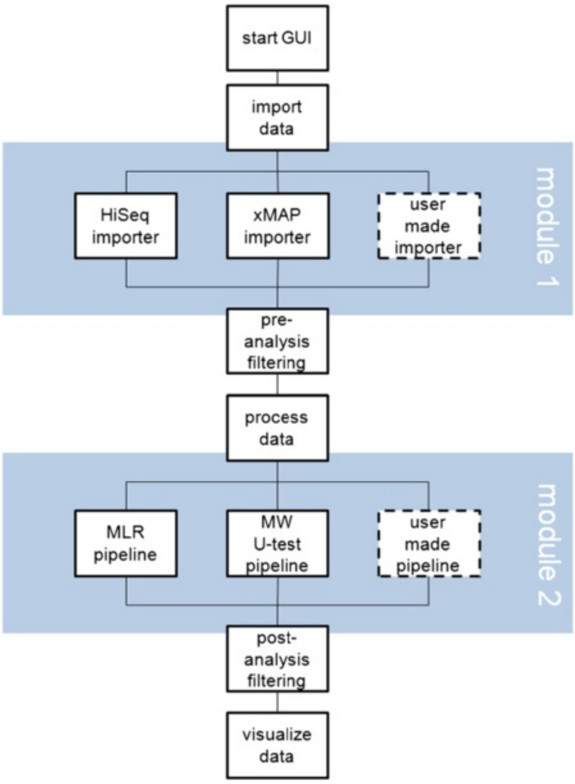


Figure 6 Basic layout of the analysis process in TOPS. TOPS has been written in order to be flexible with regard to user modules. User made importers can be written in Java and integrated with the GUI while custom analysis algorithms can be supplied as R scripts.

For dataset 1, interactions have also been validated experimentally. Comparing the three algorithms in terms of the verified true positives they can identify from a ranked list of the top 50 hits after filtering as described in [7] shows comparable performance [Table 1] and an acceptable true/false positive rate overall compared to other high-throughput screens [20].

In [Figure 9A-C] we show a scatter plot of P-values for perturbations A and B with the successful and failed validation indicated. We can observe an increasingly better spatial separation comparing the U-test with sLM and LM models, which could be for instance be captured by principal component analysis. However, since the localization of successful versus failed validation data points does not follow a simple pattern according to the original coordinates provided by the two P-values and is likely to be different for distinct datasets, this would necessitate to have available such a validation dataset to train a classifier performing better than simple P-value-based thresholds. As a generic tool TOPS does not offer this functionality but users might consider implementing it since we release the code of the system as open source. Combining the P-values for perturbations A and B does not have a negative effect on discriminative power but allows to provide a single interaction score [Figure 9D-F].

Data visualization

Data from gene-gene or gene-drug interaction screens can be difficult to inspect visually since the number of hits is comparatively small to the number of interactions. Traditional plotting methods from gene expression studies like heatmaps do not perform well in this setting. We therefore designed a circular bubble plot to display the data. An axis for every condition B is drawn on which the scores/p-values for the interactions with each condition A are plotted. The axes are arranged in a radial fashion such that the least significant hits are in the center of the plot and the most significant interaction scores are on the circumference of the circle. This allows the user to quickly spot outliers and problematic/noisy perturbations. In addition the magnitude of the effect can be encoded in the size of each dot, allowing the user to identify hits that are both significant and show a reasonable effect size. The direction of the interaction

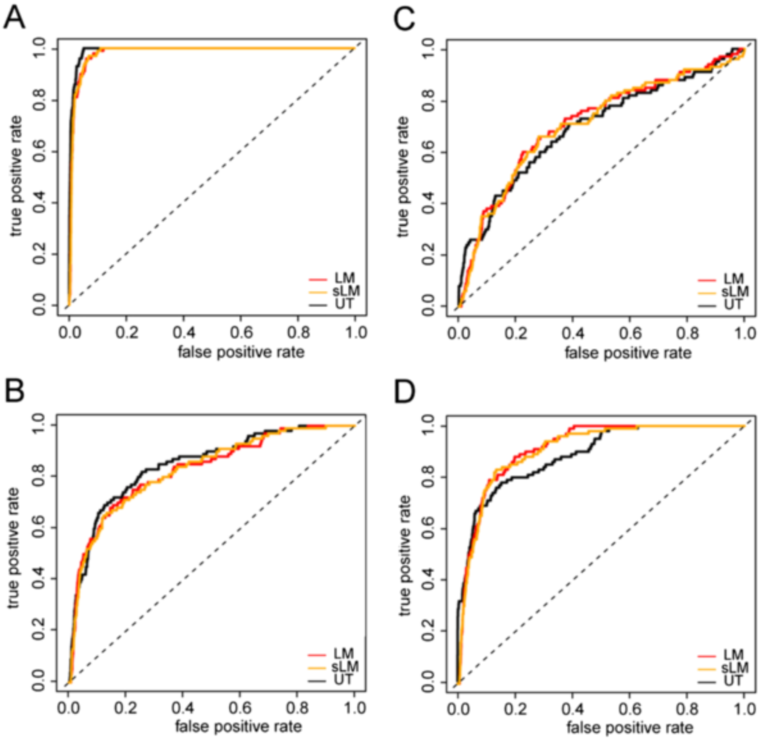


Figure 7 Receiver operating characteristics for the analysis algorithms. ROC curves are derived from a random 100x100 perturbations dataset in quadruplicate modeled after dataset 1 where 100 true positives were introduced and defined as either 0.5 **(A)**, 0.7 **(B)**, 1.3 **(C)** or 1.5 **(D)** times that of the true negatives.

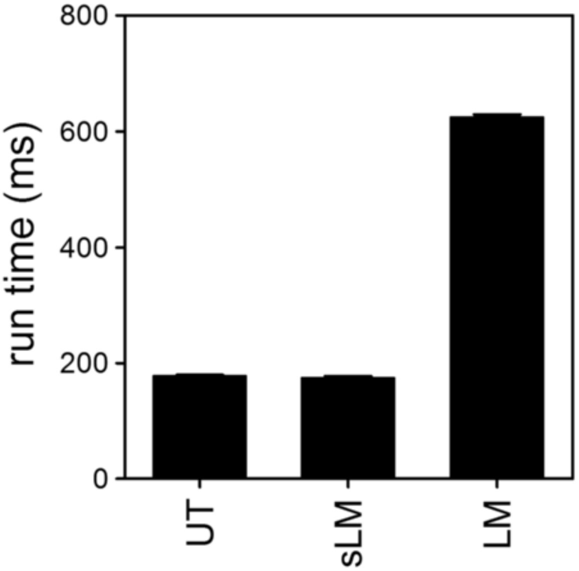


Figure 8 Runtime performance of the three algorithms. The U-Test (UT), split linear model (sLM) or linear model (LM) algorithms were tested for runtime performance analyzing a set of 200×200 interactions.

(increased or decreased fitness as compared to control) is encoded by different colors [Figure 10]. The identifiers of one set of perturbations is plotted at the circumference of the circleplot. The second perturbation can be identified by hovering the mouse over a datapoint in the plot. TOPS' plots can be exported as bitmaps in PNG format or as vector graphics as a PDF file.

Table 1 Experimentally validated true and false positive hits from dataset 1

Model (pvalue)	True positives	False positives	Unverified
LM (pvA)	8	2	40
LM (pvB)	1	1	48
LM (pvAll)	6	2	42
sLM (pvA)	8	2	40
sLM (pvB)	2	0	48
sLM (pvAll)	8	1	41
UT (pvA)	7	2	41
UT (pvB)	4	2	44
UT (pvAll)	6	2	42

The top 50 interactions from a ranked hit list are categorized by experimentally validated as true or false positive. Hits that were not experimentally tested are indicated as "unverified". The true and false positive rates for the linear model (LM), split linear model (sLM) or U-Test (UT) algorithms for either p-value A (pvA), p-value B (pvB) or the combined p-value (pvAll) are shown.

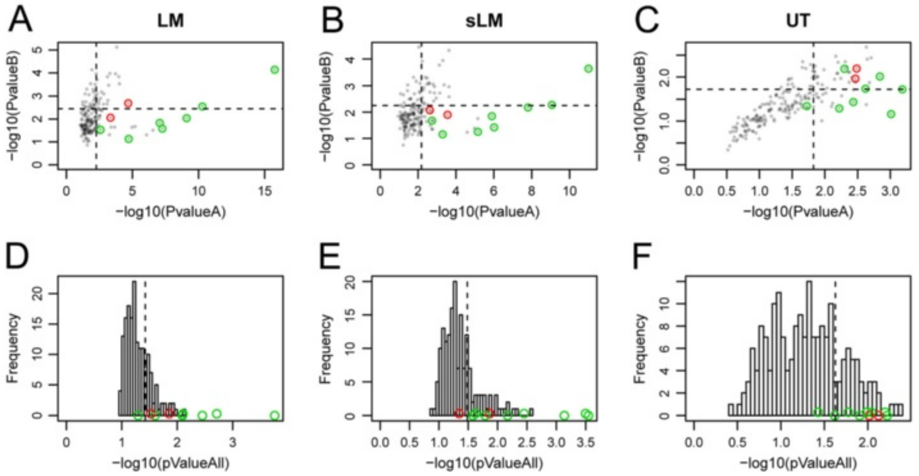


Figure 9 Separation of successfully versus failed validations of selected hits. The cut-off for the top 50 hits are indicated by dashed lines. **(A-C)** Position of validations in the P-value **A** and **B** space (log-scale). **(D-F)** Distribution of all the combined P-values (**A** and **B** combined according to Brown's method) with positions of validated hits. Experimentally validated true positives are indicated in green, false positive hits in red.

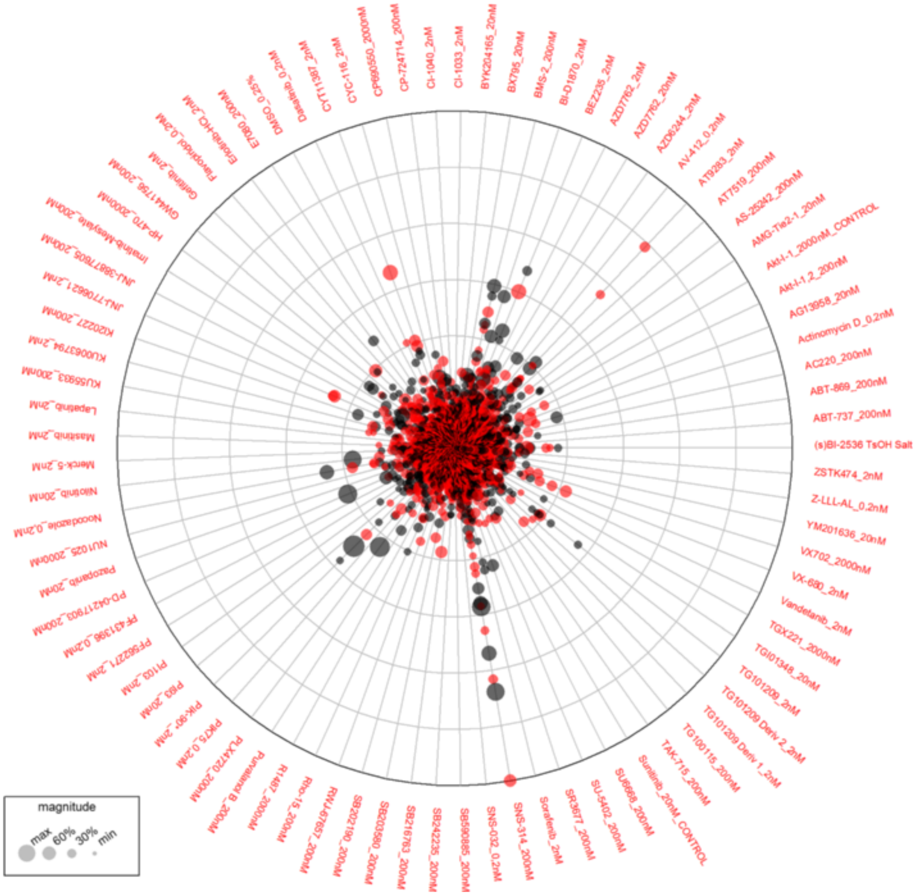


Figure 10 TOPS's graphical output. Results of the analysis are displayed on a circular bubble plot with the color encoding the direction (reduced fitness, red or increased fitness, black) of the interaction, distance from the center indicates score/pvalue and bubble size corresponds to the magnitude of the measured effect. Labels at the circumference indicate the perturbation identifier, in this example drugs.

Conclusions

We present a user-friendly software package to analyze and visualize interaction data from functional genomic dual perturbation screens. Although multiple tools for analyzing cell viability screens are available [16,21-24] these have their limitations by either being based on commercial software (mostly MATLAB) or requiring command line skills. TOPS incorporates statistical models designed for the analysis of pairwise interactions of larger gene/drug sets. Furthermore it is fully based on free software and provides a graphical user interface. The software is easily accessible and offers a powerful analysis tool for the benchtop scientist while being expandable enough to be attractive to users who would like to run their own analysis methods. Importantly, not only Luminex xMAP and Sequencing data can be analyzed with the presented methods but in principle any data from other technologies can be imported as long as the data can be reduced to a "perturbation A", "perturbation B", "score" format and true positives are relatively rare in comparison to true negatives which is a necessity due to the nature of normalization of the in-built analysis. We have included two analysis pipelines based on different methods to demonstrate the versatility of TOPS. We have also included two importers for Luminex xMAP data and for preprocessed screening data that makes these two technologies particularly easy to use with the software.

Availability and requirements

Project name: TOPS.
Project home page: https://sourceforge.net/p/topscemm/wiki/Home/.
Operating system(s): Win32/OSX/Linux.
Programming language: Java and R.
Other requirements: Java 6 or newer. R 2.14 or newer.
License: Creative Commons Attribution ShareAlike License V3.0.
Any restrictions to use by non-academics: Only those imposed by the license.

Competing interests

The authors declare that they have no competing interests.

Authors' contributions

MKM and FG designed and FG wrote the code for TOPS and the help files. CK and MKM created the datasets. IU, MKM and CK performed wetlab validation experiments. GD and JC designed and GD wrote the Linear Model based algorithm. FG and MKM designed and wrote the U-test based algorithm. KB and AS helped with processing the sequencing data. GD produced Figure 8, MKM produced the remaining figures and together with SN wrote the manuscript with the help and contributions from the other authors. SN, MKM and JC co-supervised the project. All authors read and approved the final manuscript.

Acknowledgements

We would like to thank Luminex Corp. for support with the Luminex assay, Helen Pickersgill for help in the preparation of this manuscript, and colleagues from CeMM who acted as beta-testers for the software.

Author details

[1]Research Center for Molecular Medicine of the Austrian Academy of Sciences (CeMM), Vienna, Austria. [2]Current address: The Research Institute of Molecular Pathology (IMP), Vienna, Austria.

Received: 17 May 2013 Accepted: 28 March 2014
Published: 8 April 2014

References

1. Hillenmeyer ME, Ericson E, Davis RW, Nislow C, Koller D, Giaever G: **Systematic analysis of genome-wide fitness data in yeast reveals novel gene function and drug action.** *Genome Biol* 2010, **11**(3):R30.
2. Hillenmeyer ME, Fung E, Wildenhain J, Pierce SE, Hoon S, Lee W, Proctor M, St Onge RP, Tyers M, Koller D, Altman RB, Davis RW, Nislow C, Giaever G: **The chemical genomic portrait of yeast: uncovering a phenotype for all genes.** *Science* 2008, **320**(5874):362–365.
3. Nijman SM: **Synthetic lethality: general principles, utility and detection using genetic screens in human cells.** *FEBS Lett* 2010, **585**(1):1–6.
4. Kessler JD, Kahle KT, Sun T, Meerbrey KL, Schlabach MR, Schmitt EM, Skinner SO, Xu Q, Li MZ, Hartman ZC, Rao M, Yu P, Dominguez-Vidana R, Liang AC, Solimini NL, Bernardi RJ, Yu B, Hsu T, Golding I, Luo J, Osborne CK, Creighton CJ, Hilsenbeck SG, Schiff R, Shaw CA, Elledge SJ, Westbrook TF: **A SUMOylation-dependent transcriptional subprogram is required for Myc-driven tumorigenesis.** *Science* 2011, **335**(6066):348–353.
5. Scholl C, Fröhling S, Dunn IF, Schinzel AC, Barbie DA, Kim SY, Silver SJ, Tamayo P, Wadlow RC, Ramaswamy S, Döhner K, Bullinger L, Sandy P, Boehm JS, Root DE, Jacks T, Hahn WC, Gilliland DG: **Synthetic lethal interaction between oncogenic KRAS dependency and STK33 suppression in human cancer cells.** *Cell* 2009, **137**(5):821–834.
6. Barbie DA, Tamayo P, Boehm JS, Kim SY, Moody SE, Dunn IF, Schinzel AC, Sandy P, Meylan E, Scholl C, Fröhling S, Chan EM, Sos ML, Michel K, Mermel C, Silver SJ, Weir BA, Reiling JH, Sheng Q, Gupta PB, Wadlow RC, Le H, Hoersch S, Wittner BS, Ramaswamy S, Livingston DM, Sabatini DM, Meyerson M, Thomas RK, Lander ES, *et al*: **Systematic RNA interference reveals that oncogenic KRAS-driven cancers require TBK1.** *Nature* 2009, **462**(7269):108–112.
7. Muellner MK, Uras IZ, Gapp BV, Kerzendorfer C, Smida M, Lechtermann H, Craig-Mueller N, Colinge J, Duernberger G, Nijman SM: **A chemical-genetic screen reveals a mechanism of resistance to PI3K inhibitors in cancer.** *Nat Chem Biol* 2011, **7**(11):787–793.
8. Brummelkamp TR, Bernards R: **New tools for functional mammalian cancer genetics.** *Nat Rev Cancer* 2003, **3**(10):781–789.
9. Marcotte R, Brown KR, Suarez F, Sayad A, Karamboulas K, Krzyzanowski PM, Sircoulomb F, Medrano M, Fedyshyn Y, Koh JL, van Dyk D, Fedyshyn B, Luhova M, Brito GC, Vizeacoumar FJ, Vizeacoumar FS, Datti A, Kasimer D, Buzina A, Mero P, Misquitta C, Normand J, Haider M, Ketela T, Wrana JL, Rottapel R, Neel BG, Moffat J: **Essential gene profiles in breast, pancreatic, and ovarian cancer cells.** *Cancer Discov* 2012, **2**(2):172–189.
10. Sims D, Mendes-Pereira AM, Frankum J, Burgess D, Cerone MA, Lombardelli C, Mitsopoulos C, Hakas J, Murugaesu N, Isacke CM, Fenwick K, Assiotis I, Kozarewa I, Zvelebil M, Ashworth A, Lord CJ: **High-throughput RNA interference screening using pooled shRNA libraries and next generation sequencing.** *Genome Biol* 2011, **12**(10):R104.
11. Tischler J, Lehner B, Fraser AG: **Evolutionary plasticity of genetic interaction networks.** *Nat Genet* 2008, **40**(4):390–391.
12. Fang Y, Brass A, Hoyle DC, Hayes A, Bashein A, Oliver SG, Waddington D, Rattray M: **A model-based analysis of microarray experimental error and normalisation.** *Nucleic Acids Res* 2003, **31**(16):e96.
13. Schadt EE, Li C, Ellis B, Wong WH: **Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data.** *J Cell Biochem Suppl* 2001, **Suppl 37:**120–125.
14. Huber P: *Robust Statistics.* Hoboken, NJ, USA: Wiley-Interscience; 2003.
15. Marazzi A: *Algorithms, Routines, and S-Functions for Robust Statistics.* London, UK: Taylor & Francis; 1993.
16. Axelsson E, Sandmann T, Horn T, Boutros M, Huber W, Fischer B: **Extracting quantitative genetic interaction phenotypes from matrix combinatorial RNAi.** *BMC Bioinforma* 2011, **12:**342.
17. Mani R, St Onge RP, Hartman JL, Giaever G, Roth FP: **Defining genetic interaction.** *Proc Natl Acad Sci U S A* 2008, **105**(9):3461–3466.

18. Laufer C, Fischer B, Billmann M, Huber W, Boutros M: **Mapping genetic interactions in human cancer cells with RNAi and multiparametric phenotyping.** *Nat Methods* 2013, **10**(5):427–431.
19. Brown M: **A method for combining non-independent, one-sided tests of significance.** *Biometrics* 1975, **31**(4):987–992.
20. Ilouga PE, Hesterkamp T: **On the prediction of statistical parameters in high-throughput screening using resampling techniques.** *J Biomol Screen* 2012, **17**(6):705–712.
21. Boutros M, Bras LP, Huber W: **Analysis of cell-based RNAi screens.** *Genome Biol* 2006, **7**(7):R66.
22. Luo B, Cheung HW, Subramanian A, Sharifnia T, Okamoto M, Yang X, Hinkle G, Boehm JS, Beroukhim R, Weir BA, Mermel C, Barbie DA, Awad T, Zhou X, Nguyen T, Piqani B, Li C, Golub TR, Meyerson M, Hacohen N, Hahn WC, Lander ES, Sabatini DM, Root DE: **Highly parallel identification of essential genes in cancer cells.** *Proc Natl Acad Sci U S A* 2008, **105**(51):20380–20385.
23. Collins SR, Schuldiner M, Krogan NJ, Weissman JS: **A strategy for extracting and analyzing large-scale quantitative epistatic interaction data.** *Genome Biol* 2006, **7**(7):R63.
24. Baryshnikova A, Costanzo M, Kim Y, Ding H, Koh J, Toufighi K, Youn JY, Ou J, San Luis BJ, Bandyopadhyay S, Hibbs M, Hess D, Gingras AC, Bader GD, Troyanskaya OG, Brown GW, Andrews B, Boone C, Myers CL: **Quantitative analysis of fitness and genetic interactions in yeast on a genome scale.** *Nat Methods* 2010, **7**(12):1017–1024.

doi:10.1186/1471-2105-15-98
Cite this article as: Muellner *et al.*: TOPS: a versatile software tool for statistical analysis and visualization of combinatorial gene-gene and gene-drug interaction screens. *BMC Bioinformatics* 2014 **15**:98.