TIMMA-R : an R package for predicting synergistic multi-targeted drug combinations in cancer cell lines or patient-derived samples

He, Liye

2015-06-01

User instructions and tutorial of the *timma* package

Liye He, Krister Wennerberg, Tero Aittokallio and Jing Tang
Institute for Molecular Medicine Finland (FIMM)
University of Helsinki

January 9, 2015

Contents

1 Summary

Utilizing drug-target interaction and single-drug sensitivity data, we have developed a logic-based approach called TIMMA (Target Inhibition interaction using Minimization and Maximization Averaging) for predicting the effects of targeted-drug combinations for a given cancer cell line or patient-derived sample. The TIMMA approach starts by identifying an optimal set of targets which are predictive of the single-drug sensitivities. This target set is then considered as essential in driving the cancer survival. By combining the targets of each individual drug, a drug combination is modelled as its corresponding binding profile on the selected target set. The cellular response of such a binding profile can be estimated by a model-based averaging on the sensitivities of drugs with similar binding profiles. The similarity between two drugs is defined according to the set relationship between the binding profiles, rather than using a conventional distance-based metrics such as Spearman correlation or Tanimoto distance. By comparing the predicted sensitivities for drug combinations and the observed sensitivities of the single-drug components, the TIMMA approach further calculates the scores for quantifying the synergistic effect, which can be experimentally tested using drug combination screens. It also provides a network view of the target inhibition interactions, formulating a data-driven hypothesis on the most promising multi-targeted therapies to effectively treat the cancer cells of

a particular type. TIMMA can be an ideal computational tool for prioritizing, explaining and discovering novel multi-targeted therapies in personalized cancer treatment (Tang *et al.*, 2013).

2 Quick Start

timma is an R package that implements the TIMMA model prediction and its supporting functions. In this section, you will learn how to get a quick start of using the *timma* package to make a first prediction.

1. Install the *timma* package from CRAN.

   ```
   > install.packages(timma)
   ```

2. Load the *timma* package in the R environment.

   ```
   > library(timma)
   ```

3. Prepare the input data. Two types of data are needed. One is the drug-target interaction data, usually obtained from drug-target binding assays. The current TIMMA algorithm works with categorical drug-target interactions, meaning that the raw data from quantitative binding activity assays (such as Ki, Kd and IC_{50} readouts) may need to be preprocessed by binarization or multi-class classification. The other input data is a drug sensitivity vector for a given cancer cell line or a patient-derived sample. The drug sensitivities can be continuous, as they are usually measured as cell viability IC_{50} or the area under the dose-response curve. The drug sensitivity data should be scaled into $[0, 1]$, where a higher value indicates a stronger effect. You can load an example data from Tyner *et al.* (2013) by typing:

   ```
   > data(tyner_interaction_binary)
   > # load a binary drug-target interaction data
   > data(tyner_interaction_multiclass)
   > # load a multi-class drug-target interaction data
   > data(tyner_sensitivity)
   > # load a drug sensitivity data
   ```

4. Run the main *timma* function with the given drug-target interaction data and a selected drug sensitivity vector. The other input parameters are optional (See more details about the full list of parameters in section 3.2). An example command is shown below:

   ```
   > timma(x = tyner_interaction_binary, y = tyner_sensitivity[,1],
    sp = 1, max_k = 8, verbosity = T)
   > # x: the drug-target interaction data
   > # y: the drug sensitivity data for the first sample
   > # sp: the starting point for the search algorithm to
    navigate the target set space. By default it starts with
    the first target in the drug-target interaction data.
   > # max_k: the maximal number of targets to be included in
   the selected target set. The search algorithm will stop
   once the max_k has been achieved.
   > # verbosity: a parameter specifying whether the progress is displayed or not.
   ```

5. Three components are implemented in the *timma* function: (a) Model selection - selecting the essential target set which consists of the most predictive targets for single-drug sensitivities using the Sequential forward floating search (SFFS) algorithm; (b) Model construction - predicting the sensitivities for all the possible target inhibition profiles in the essential target set; (c) Synergy scoring of drug combinations and visualization of target inhibition network. During the model selection stage, the progress about the selection of essential targets will be shown with its prediction error, defined as the mean absolute error (MAE) in the leave-one-out (LOO) cross-validation (Figure 1).

```
----------------Start Running TIMMA----------------------------
Number of selected targets:  1 MAE =   0.3284711
Including target # 119 MAE =   0.2870156
Excluding target # 1 MAE =   0.3064503
Number of selected targets:  1 MAE =   0.3064503
Including target # 271 MAE =   0.2606634
Excluding target # 119 MAE =   0.2978057
Number of selected targets:  1 MAE =   0.2978057
Including target # 119 MAE =   0.2606634
Number of selected targets:  2 MAE =   0.2606634
Including target # 293 MAE =   0.2396067
Number of selected targets:  3 MAE =   0.2396067
Including target # 19 MAE =   0.2196881
Number of selected targets:  4 MAE =   0.2196881
Including target # 74 MAE =   0.2053813
Number of selected targets:  5 MAE =   0.2053813
Including target # 25 MAE =   0.1970748
Number of selected targets:  6 MAE =   0.1970748
Including target # 95 MAE =   0.1829059
Number of selected targets:  7 MAE =   0.1829059
Including target # 137 MAE =   0.1781744
Number of selected targets:  8 MAE =   0.1781744
Including target # 151 MAE =   0.1714343
----------------Complete running TIMMA model--------------------
----------------Saving selectedTargets.csv----------------------
----------------Saving predictedSensitivities.csv---------------
----------------Saving predictedScoring.csv---------------------
----------------Saving targetInhibitionNetwork.pdf--------------
----------------Saving targetInhibitionNetwork.nnf--------------
Analysis finished. All the results are saved in C:/Users/localadm
```

Figure 1: The information displayed when running the *timma* function with *verbosity=T*.

6. Save the results. Five output files including three tables and two figures will be generated. The file *selectedTargets.csv* lists the selected cancer essential targets and the *timma* predictions for the single-drugs in the LOO cross-validation, as compared to the actual sensitivities (Figure 2a). The file *predictedSensitivities.csv* includes a predicted sensitivity matrix for all the inhibition profiles on the selected target set (Figure 2b). The file *predictedScoring.csv* provides synergy scores for pairwise drug combinations (Figure 2c). The file *targetInhibitionNetwork.pdf* shows the network image and the file *targetInhibitionNetwork.nnf* generates the input for Cytoscape (www.cytoscape.org).

7. Read the help documents for more functional details and example codes using the *help* (?) command:

```
>?timma
```

3 Tyner study

We will walk through the main functions of the *timma* package using an example data, which is extracted from a recent study on the kinase pathway dependence in leukemia using 65 small-molecule kinase inhibitors as chemical probes, tested on 151 leukemia patient-derived samples *ex-vivo* (Tyner *et al.*, 2013). The drug-target interaction data and the drug sensitivity data have been provided in its Supplementary materials (Supplementary Table 1 for drug-target interactions and Supplementary Table 3 for the drugs' cell proliferation IC_{50}s for the 151 patient-derived samples. We choose this study as an example case study particularly because of its translational potential, as the effective drug or drug combinations for these patient-derived samples could be readily utilized for clinical applications. Even though the focus in this tutorial is to show the analysis work flow of *timma*, we recommend that any interested researchers evaluate the clinical significance of the prediction results in a drug combination screen *ex-vivo*, patient treatment *in-vivo* or even in clinical trials.

	AKT3	AURKA	DAPK1	EGFR.L858	FES	GSK3A	RPS6KA4.I	TGFBR2	sensitivity	LOO sensitivity
ABT-869	0	1	0	0	0	0	1	0	0.670792	0.616787
AG490	0	0	0	0	0	0	0	0	0	0.069691
AKT Inhibi	1	0	0	0	0	0	0	0	0.857732	0.587584
AKT Inhibi	1	0	0	0	0	0	0	0	0.587584	0.857732
AMG-706	0	0	0	0	0	0	0	0	0	0.069691
AMPK Inh	0	0	0	0	0	0	0	0	0.601469	0.04895
AP24534	0	1	0	0	1	0	0	0	0.970923	0.818946
AST487	0	0	0	1	1	0	1	1	0.813865	0.645865
Barasertib	0	1	0	1	0	0	0	0	0	0
BIRB-796	0	0	0	0	0	0	0	0	0	0.069691
BMS-3870	0	0	0	0	0	0	0	1	0.987021	0.587484
CAL-101	0	0	0	0	0	0	0	0	0	0.069691
CHIR-258	0	0	0	0	0	0	1	0	0.746353	0.598463
CHIR-265	0	0	0	0	0	1	0	0	0.392997	0.565022
CI-1033	0	0	0	1	0	0	0	1	0.630647	0.482896
CP-690550	0	0	0	0	0	0	0	0	0	0.069691
CYC-202	0	0	0	0	0	0	0	0	0	0.069691
CYT387	0	0	0	0	1	0	0	0	0.732367	0.714156
Dasatinib	0	0	0	1	0	0	0	1	0.965791	0.315323
EGFR Inhi	0	0	0	1	0	0	0	0	0.955831	0.316719

(a) selectedTargets.csv. Drug names are shown in column A and the selected targets are shown in row 1, followed by the observed and the predicted sensitivities.

(b) predictedSensitivities.csv. The matrix shows all the possible inhibition profiles for the selected targets and their corresponding sensitivities predicted by *timma*. For example, cell 5E shows the sensitivity (0.067368) if none of the targets were inhibited, which can be considered as a residual in the model predictions; cell 6E shows the predicted sensitivity (0.387732) for the case when EGFR (L858) alone is inhibited; cell 7E shows the predicted sensitivity (0.63447) for the case if both EGFR (L858) and DAPK1 are inhibited, and etc.

Drug1	Drug2	Sensitivity	Synergy.add	Synergy.multi	Synergy.highest	Target1	Target2
ABT-869	ABT-869	0.67079183	-0.07556092	0.67079183	-0.075560922	AURKA	RPS6KA4.Kin.Dom.1...N.terminal.
ABT-869	AKT Inhibi	0.865554885	-0.23027971	0.595875911	0.131049401	AURKA,RF	AKT3
ABT-869	AKT Inhibi	0.865554885	-0.23027971	0.595875911	0.131049401	AURKA,RF	AKT3
ABT-869	AP24534	0.807274374	-0.17853909	0.625072889	0.065583392	RPS6KA4.I	AURKA,FES
ABT-869	AST487	0.706250988	-0.32227061	0.481586005	-0.055634609	AURKA,RF	EGFR.L858R.,TGFBR2,RPS6KA4.Kin.Dom.1...N.termi
ABT-869	Barasertib	0.435087366	-0.32096902	0.338625793	-0.191725079	AURKA,RF	EGFR.L858R.,AURKA
ABT-869	BMS-3870	0.943731764	-0.41646544	0.575398909	-0.043289063	AURKA,RF	TGFBR2
ABT-869	CHIR-258	0.67079183	-0.07556092	0.67079183	-0.075560922	AURKA	RPS6KA4.Kin.Dom.1...N.terminal.
ABT-869	CHIR-265	0.494937871	-0.27123579	0.348280569	-0.074737147	RPS6KA4.I	GSK3A
ABT-869	CI-1033	0.630483449	-0.43006926	0.373970842	-0.14654811	AURKA,RF	EGFR.L858R.,TGFBR2
ABT-869	CYT387	0.875515645	-0.23002817	0.602213418	0.136155549	RPS6KA4.I	FES
ABT-869	Dasatinib	0.630483449	-0.43006926	0.373970842	-0.14654811	AURKA,RF	EGFR.L858R.,TGFBR2
ABT-869	EGFR Inhi	0.317235134	-0.44367307	0.172542775	-0.249807157	AURKA,RF	EGFR.L858R.
ABT-869	EKB-569	0.59637539	-0.33685062	0.387378096	-0.056825804	AURKA,RF	EGFR.L858R.,FES
ABT-869	Erlotinib	0.435087366	-0.32096902	0.338625793	-0.191725079	AURKA,RF	EGFR.L858R.,AURKA

(c) predictedScoring.csv. Sensitivity is the predicted sensitivity for a drug combination. Synergy.add, Synergy.multi and Synergy.highest are the synergy scores calculated based on the three different synergy models; Target1 and Target2 are the underlying combinatorial targets from each drug which lead to the synergistic effects.

Figure 2: Snapshots of the three output tables.

4

3.1 Data preparation

3.1.1 The drug target interaction data

The drug-target interaction data from Supplementary Table 1 in Tyner *et al.* (2013) contains 65 small-molecule kinase inhibitors and 322 targets. Each drug-target interaction has been classified into one of the six classes labelled as $[0, 1, 2, 3, 4, 5]$. Class 0 is the inactive group (i.e. no activity was found between the drug and the target). Class 1 consists of the most potent drug-target interactions, defined as those bioactivity values lower than 10-fold of the lowest Kd or IC_{50} value. Classes 2 to 5 are those intermediate interaction groups, each of which represents a 10-fold increase of bioactivity values compared to the previous class. For example, if a drug's most potent target has Kd value of 1 nM, then those targets which have Kd lower than 10 nM will be labelled as class 1 targets, and those targets with Kd between 10 nM and 100 nM will be class 2 targets and etc. To enable the use of set compare functions in *timma*, the class labels must be converted into ordinal categories, i.e. a higher class number should imply a stronger interaction. We relabel the class of the most potent drug-target interactions as class 5 and change the labels for the other classes accordingly. The corresponding target names and drug names are shown as column names and row names for the data matrix. To load the curated multi-class drug-target data frame we use the command:

```
> data(tyner_interaction_multiclass)
> # load the multi-class drug-target interaction data from the
Tyner et al.(2013) study
```

The data can be further binarized by grouping all the non-zero classes into one group, which includes both the strong and weak drug-target interactions. One can obtain such a binarized data with the command:

```
> tyner_interaction_binary <- 1*(tyner_interaction_multiclass>0)
> # binarize the multi-class data
```

A more conservative way of binarization would consider only those strongest interactions as 1s and all the other remaining weak and inactive interactions as 0s:

```
> tyner_interaction_binary_strong <- 1*(tyner_interaction_multiclass>4)
> # binarize the multi-class data considering only the class 5 targets
as true positive
```

Note that there is no standard protocol for classifying active and inactive interactions from a typical drug-target binding assay data. Many drug databases, such as DrugBank (Law *et al.*, 2014) or TTD (Zhu *et al.*, 2012), reported only a few primary targets without giving the corresponding cut-off criteria. However, analyses of many approved drugs showed that therapeutic efficacy is not necessarily associated with higher binding affinity, and thus focusing on the few primary targets might not provide sufficient information on understanding the drugs' mode of actions (Gleeson *et al.*, 2011). The ChEMBL database, on the other hand, provided a comprehensive bioactivity profile for many targeted-drugs, revealing considerably promiscuous drug-target interactions at the proteome-level (Bento *et al.*, 2014). However, the pharmacological significance of those weak or unintended interactions remain largely unknown, particularly due to the poor understanding of a drug-target interaction and its relevance on the signalling pathways in a cellular environment. Therefore, treating the drug-target interaction data without any classification might also easily introduce much noise that masks the pharmacologically important signals. We take an intermediate stand between the two extremes, i.e. aiming for collecting a drug's target profile as comprehensive as possible, and then classifying the strong and weak interactions into different potency groups to reduce noises. In section 3.3 we will evaluate the prediction performance using different classification schemes on the drug-target interaction data.

3.1.2 The drug sensitivity data

The second input is the drug sensitivities for a given patient-derived sample. The drug sensitivities in the Tyner's study were measured in a cell proliferation assay and quantified by IC_{50} values fitted from the dose-response curves. In addition, for each drug a median IC_{50} across all the 151 samples was calculated. The total drug sensitivity data thus contains 65 drugs and 152 IC_{50} columns. Note that a drug's sensitivity is inversely related to its IC_{50}. To be utilized by the *timma* function, for each sample the IC_{50} values must be

converted to sensitivities scaled in the $[0, 1]$ interval, where a higher value indicates a stronger effect. For example, a scaled sensitivity can be calculated as:

$$Sensitivity = \frac{Max_{IC_{50}} - IC_{50}}{Max_{IC_{50}} - Min_{IC_{50}}} \tag{1}$$

To load the normalized drug sensitivity data, one can use:

```
> data(tyner_sensitivity)
> # load the scaled drug sensitivity data
> median_sensitivity = tyner_sensitivity[,1]
> # the median drug sensitivity vector
```

3.2 Parameters for tuning the *timma* algorithm

Once the input data is prepared, we use the main *timma* function to run the analysis.

```
> data(tyner_interaction_binary)
> data(tyner_sensitivity)
> median_sensitivity = tyner_sensitivity[,1]
> timma(x = tyner_interaction_binary, y = median_sensitivity, sp = 1,
max_k =  8, filtering = FALSE , class = 2,  averaging = "one.sided",
weighted = FALSE)
```

The details of the parameters are described in the following:

1. x: a drug-target interaction matrix. Row names are drug names and column names are target names.

2. y: a scaled drug sensitivity vector.

3. sp: the index of an initial target to be included in the essential target set in the model selection algorithm. $sp \in [1, n]$, where n is the total number of targets.

4. max_k: the number of targets that can be included in the final essential target set. The number of parameters to be estimated in the model construction will increase exponentially as max_k increases. In practice it is advised to keep $max\_k < 10$.

5. *filtering*: a logical parameter to specify whether a few targets should be filtered out before the analysis. By default, the value is FALSE, meaning that all the targets will be evaluated in the model selection. If the value is TRUE, those targets that are negatively correlated with the drug sensitivity data will be removed. The choice of this parameter depends on how strongly we assume that a target must be individually essential in a target combination.

6. *class*: the number of classes in the drug-target interaction data. For a binary data, $class = 2$.

7. *averaging*: a parameter to specify which one of the averaging algorithms will be applied in the model construction. By default, *averaging* = "one.sided", which is the original model construction algorithm utilized in Tang *et al.* (2013). When *averaging* = "two.sided", a modified averaging algorithm will be used. These two algorithms differ at the case where only one side of the set relationships is available in the training data. For example, for a queried target set if its supersets but not subsets are available, then the *one.sided* algorithm will predict the sensitivity using the Minimization rule only. The *two.sided* algorithm, however, will further average the predicted sensitivity with 0, which is the minimal sensitivity that could be obtained as a subset theoretically. Similarly, for a queried target set for which only the subsets are available in the training data, the *one.sided* algorithm will use the Maximization rule for prediction, whereas the *two.sided* algorithm will average the *one.sided* prediction result with 1, which is the maximal sensitivity for a superset can achieve.

8. *weighted*: a parameter to specify if the similarity between the queried target set and its subsets/supersets is considered as a weight factor in the averaging. This is a parameter that is defined for multi-class drug-target interaction data only. When *weighted* = TRUE, the sensitivities for those subsets/supersets which have more overlaps with the queried set will be weighted more in the final prediction.

3.3 Prediction accuracy

In this section, we will evaluate whether using multiclass drug-target interaction data will bring any significant improvement in the model prediction accuracy compared to the binarized data. We will utilize the drug sensitivities for the 151 leukemia patient samples and run the *timma* algorithm on both the multiclass and the binarized drug-target interaction data, and compare the MAEs (mean absolute error) in the LOO (leave-one-out) cross-validation. For obtaining the MAEs using the multiclass data:

```
> data(tyner_interaction_multiclass)
> data(tyner_sensitivity)
> error_multiclass <- matrix(0, nrow = 65, ncol = 151)
> # a matrix to store the prediction error
> for (i in 1:151){
    print(i)
    res <- sffsCategory(tyner_interaction_multiclass,
    tyner_sensitivity[, i+1], sp = 1, max_k = 8, loo = TRUE, class = 6)
    error_multiclass[, i] <- res$timma$error
    # class: the number of classes in the drug-target interaction data
  }
> multiclass_MAE <- colMeans(error_multiclass)
```

It takes about 2~3 hours to finish the loop analysis for all the 151 samples. For obtaining the MAEs using the binary data, we use the sffsBinary() function with the binarized data as the input:

```
> tyner_interaction_binary <- 1*(tyner_interaction_multiclass>0)
> data(tyner_sensitivity)
> error_binary <- matrix(0, nrow = 65, ncol = 151)
> # a matrix to store the prediction error
> for (i in 1:151){
   print(i)
   res <- sffsBinary(tyner_interaction_binary,
   tyner_sensitivity[, i+1], sp = 1, max_k = 8, loo = TRUE)
   error_binary[, i] <- res$timma$error
  }
> binary_MAE <- colMeans(error_binary)
```

The MAEs are compared with the following commands:

```
> redx<-c()
> redy<-c()
> bluex<-c()
> bluey<-c()
> for (i in 1:151){
   if(binary_MAE[i]>=multiclass_MAE[i]){
    redx<-c(redx, binary_MAE[i])
    redy<-c(redy,multiclass_MAE[i])
   } else{
    bluex<-c(bluex, binary_MAE[i])
    bluey<-c(bluey, multiclass_MAE[i])
   }
  }
> png(file="MAE.png", units="in", width=6, height=6, res=300)
> plot(x=c(0,0.3), y=c(0,0.3), type="n", xlab="Binary MAE", ylab
="Multiclass (c=6) MAE")
> points(redx, redy,pch=19, col="red")
> points(bluex, bluey,pch=19, col="blue")
```

```
> lines( par()$usr[1:2], par()$usr[3:4] )
> dev.off()
```


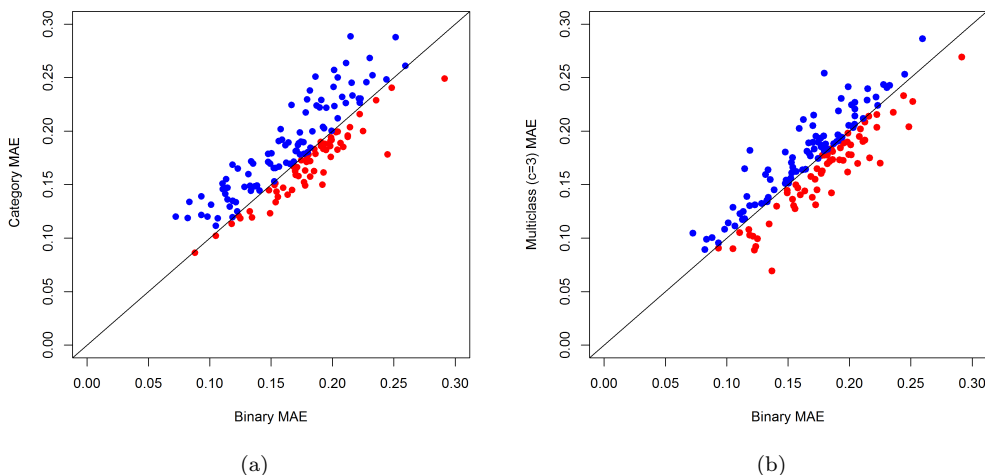(a)   (b)

Figure 3: Scatterplots showing the MAE (Mean absolute error) in the leave-one-out cross validation using either binary or multiclass drug-target interaction data for the 151 patient samples. Blue: samples where using binary data leads to lower MAE; red: samples where using multiclass data leads to lower MAE. (a) Binarization versus the original 6-class classification. (b) Binarization versus the 3-class classification.

As shown in Figure 3a, we do not find a significant difference between data binarization and multiclass classification in terms of the prediction accuracy. The average differences in MAE across the 151 samples is 0.0091, which is quite minimal compared to the actual MAEs. On the other hand, it shows that introducing more potency classes may not always lead to a better prediction. The reasons are two-fold. First, adding more classes increases model search space from 2^n to p^n, where p is the number of classes and n is the number of targets. Estimating the increasing number of model parameters would need a larger amount of training data. If the number of drugs or the number of targets does not increase significantly, treating the drug-target interactions as multiple classes will create more sparseness in the training data and result in higher uncertainty in the model predictions. Secondly, the Tyner *et al.* scheme to classify drug-target interactions is quite optimistic, as all the weak interactions (i.e. those with Kd or IC_{50} close to 10 μM) are labelled as one of the active classes. Such a classification might be sub-optimal for understanding the response of patient-derived samples given that the majority of the drugs' *in vivo* or *ex vivo* efficacy is expected to be elicited via its targets with nanomolar potency.

To further explore the impact of classification on the prediction accuracy, we test a simpler scheme by considering only 3 classes: class 0 being the inactive group; class 4 and 5 as the strong interaction group while class 1, 2 and 3 together as the weak interaction group:

```
> tyner_interaction_multiclass_2 <- tyner_interaction_multiclass
> tyner_interaction_multiclass_2
[tyner_interaction_multiclass>0 & tyner_interaction_multiclass<4] = 1
> tyner_interaction_multiclass_2[tyner_interaction_multiclass>=4] = 2
> error_multiclass_2 <- matrix(0, nrow = 65, ncol = 151)
> for (i in 1:151){
    print(i)
    res <- sffsCategory(tyner_interaction_multiclass_2,
    tyner_sensitivity[, i+1], sp = 1, max_k = 8, loo = TRUE, class = 3)
    error_multiclass_2[, i] <- res$timma$error
    # class: the number of classes in the drug-target interaction data
```

```
    # now we have only 3 classes
  }
> multiclass_2_MAE <- colMeans(error_multiclass_2)
```

We found that such a simplified classification indeed improved the average MAE from 0.1792 to 0.1754, with increased number of cases (from $n = 61$ to $n = 74$) where it outperformed the binarization (Figure 3b). However, as the overall difference between binarization and multi-classification remained minimal, we conclude that as long as the drug-target interaction data is of high-quality (e.g. manually curated from experimental results), the classification schemes seem to have little impact on the prediction accuracy of *timma*.

3.4 Synergy scores

A synergistic drug combination should treat the cancer by blocking multiple survival pathways, with an effect which cannot be achieved by any of the single-drugs alone. In TIMMA, a drug combination is modelled as the combinatorial effect of its target inhibitions. We will first evaluate the synergistic effect for its underlying target combinations and then derive a summarized score for the drug combination. The synergy score for a target pair $\{i, j\}$ is based on one of the three different reference models:

1. Multiplicative score, defined as $S_m(i,j) = y(i,j) - y(i) * y(j)$;

2. Additive score, defined as $S_a(i,j) = y(i,j) - (y(i) + y(j))$;

3. Highest agent score, defined as $S_l(i,j) = y(i,j) - max(y(i), y(j))$,

where $y(i,j)$ is the predicted sensitivities for the target combination and $y(i), y(j)$ are the predicted sensitivities for the single-target inhibitions. For a drug combination $(d1, d2)$, we take an average synergy score over all the underlying pairwise target combinations:

$$S(d1, d2) = \frac{1}{n} \sum_{i \in d1, j \in d2} S(i, j) \tag{2}$$

The synergy scores should be interpreted as interval scales rather than ratio scales. For example, a synergy score of 0 does not necessarily imply no interaction, and a score of 1 cannot be interpreted to be twice as synergistic as a score of 0.5. This is similar to the case when comparing temperatures in the Celcius scale. Rather, the synergy scores should be used for ranking the drug combinations so that the top ones can be prioritized for experimental validation. Under different assumptions, the rankings of drug combinations using the three reference models do not always match each other. If there is no preferred synergy model we may select those combinations that are commonly enriched in all of the three rankings. On the other hand, synergy should not be confused with sensitivity: synergy is an *extra* effect beyond the expectation under the non-interaction model, while sensitivity is an end-point response of a drug combination by adding the expectation and the synergy together:

$$sensitivity = expectation + synergy \tag{3}$$

A drug combination which shows a strong synergy might be much less effective than a drug combination which shows zero synergy, as their reference sensitivity may differ significantly. It is advised to consider both the synergy scores and the predicted sensitivity scores for the most promising drug combinations that show clinically-relevant synergies.

3.5 Target inhibition network

In this section, we will describe how to visualize the target inhibition network for the selected target set. The selected target set can be obtained by:

```
> data(tyner_interaction_binary)
> data(tyner_sensitivity)
```

```
> median_sensitivity <- tyner_sensitivity[, 1]
> # get the median drug sensitivity data
> res <- sffsBinary(tyner_interaction_binary, median_sensitivity,
sp = 1, max_k = 6, loo = TRUE, verbosity = T)
> selected_targets <- res$k_sel
> # get the index of the selected targets
> tyner_interaction_binary<-data.frame(tyner_interaction_binary)
> profile <- tyner_interaction_binary[, selected_targets]
> # get the drug target profiles for the selected target set
```

To visualize the interactions between the target inhibitions, we consider the drug-target profiles for the selected target set as the explanatory data, and the drug sensitivities as the response variable. For facilitating the boolean network reconstruction the drug sensitivities are classified as either effective or ineffective, using the threshold of 0.5. We take a Boolean expression approach to estimate the minimal configurations in the drug-target profiles that lead to effective drug sensitivities. The resulting network is a two-terminal graph with multiple components, each of which represents a minimal Boolean expressions determined using the Quine-McCluskey algorithm, implemented in the QCA package in R (Dusa, 2010) (Figure 4).

```
> one_index <- which(median_sensitivity > 0.5)
> # get the index of the sensitive drugs
> zero_index <- which(median_sensitivity <= 0.5)
> # get index of the insensitive drugs
> median_sensitivity_binary <- median_sensitivity
> median_sensitivity_binary[one_index] <- 1
> median_sensitivity_binary[zero_index] <- 0
> # binarize the drug sensitivity data
> draw_data <- cbind(profile, median_sensitivity_binary)
> drawGraph(draw_data)
> # plot the target inhibition network
```


Figure 4: An example target inhibition network. The network can be interpreted as data-driven hypotheses on the pharmacologically actionable target interactions which block the parallel cancer survival pathways. For example, a drug combination that collectively inhibits TGFBR2 and RPS6KA4 is expected to block the whole pathway. Alternatively, simultaneous inhibition of RPS6KA4, FES, DAPK1 and AURKA can be also utilized as an effective combinatorial strategy.

The target inhibition network is a binary representation about the potent target combinations based on the model predictions. From the network topology itself one can pinpoint those target combinations that pass a sensitivity threshold, but their actual sensitivity values need to be compared in the predictedSensitivities.csv file. For example, the combined inhibition of RPS6KA4 and DAPK1 has a predicted sensitivity of 0.57 while the combined inhibition of RPS6KA4, FES, DAPK1 and AURKA results in a higher sensitivity of 0.90; the single inhibition of AKT3, on the other hand, can reach a sensitivity of 0.72. In the target inhibition

network, all of the three target sets are listed as potent combinations as their sensitivities are higher than the default sensitivity threshold 0.5.

4 Miller study

The Tyner study provided large-scale drug-target interaction data which make the *timma* analysis quite straightforward. However, for many drug combination studies one is being provided with few drug-target interaction data. Many drugs, especially those experimental compounds, are often annotated with their primary targets only. These primary targets are useful for explaining the intended therapeutical actions of the compounds, but provide limited information on the polypharmacological effects of a drug combination. In this section we will show a complete pipeline analysis following the work flow in Figure 1 in the manuscript, starting from collecting the quantitative drug-target interaction data from public databases or literature, analyzing the dose-response curves from drug screen, and then utilizing *timma* for making predictions about the drug combinations. We will illustrate the whole process on the data provided by Miller *et al.* (2013), where 14 drugs have been tested as a pairwise combination matrix in a Dedifferentiated liposarcoma (DDLS)-derived cancer cell line.

Table 1: The drug information in the Miller study

Drug	Abbreviation	Primary Targets	Company	Company Product#	ChEMBL ID
AG538	AG	IGF1R	EMD Millipore	658403	NA
AKT 1/2 Inh	AK	AKT1/2	EMD Millipore	124018	CHEMBL258844
FR180204	ER	ERK1/2	EMD Millipore	328007	CHEMBL259551
Gefitinib	GF	EGFR	Tocris	3000	CHEMBL939
HNHA	HN	HDAC	Cayman	13295	CHEMBL2419279
PDGFR TKI III	PD	PDGFR	EMD Millipore	521232	CHEMBL102346
PI3Ka Inh IV	PI	PI3K (PIK3CA/B/C/D)	EMD Millipore	528111	NA
Rapamycin		mTOR	Selleckchem	1039	CHEMBL413
Rottlerin	RT	PKC/CaMKIII	Tocris	1610	CHEMBL34241
Ryuvidine	RY	CDK4/6	Tocris	2609	NA
SL327	SL	MEK1/2	EMD Millipore	444939	CHEMBL261237
SRC Inh I	SR	SRC	EMD Millipore	567805	CHEMBL97771
Stattic	ST	STAT3	EMD Millipore	573099	CHEMBL1337170
SU11247	SU	MET	EMD Millipore	448101	CHEMBL2218936

4.1 The drug-target interaction data

The 14 drugs used in the Miller's study are shown in Table 1. The listed primary targets are important indicators about the drugs' mode of actions, but for understanding their polypharmacological interactions we still need to obtain the quantitative drug-target interaction profiles at the proteome level. We find that 9 of these 14 drugs are from the EMD Millipore company. Recently, EMD Millipore has profiled 158 drugs using a broad activity screen (Gao *et al.*, 2013)(PMID:23398362). We start by processing the drug-target interaction data from Gao *et al.* (2013), referred hereafter as the Millipore data, which can be downloaded from http://www.biochemj.org/bj/451/bj4510313add05.xls:

```
> # make sure that the Java version matches your R version!
> # install.packages("rJava")
> # install.packages("xlsx")
> # install.packages("plyr")
> library(xlsx)
> library(plyr)
> # load the Millipore kinome screen data
> # from Supplementary Table S5
> download.file("http://www.biochemj.org/bj/451/bj4510313add05.xls",
"bj4510313add05.xls", mode="wb")
> millipore_raw = xlsx::read.xlsx(file.path(getwd(),'bj4510313add05.xls'),1,header=F)
>
```

```
> # a bit of cleaning
> millipore_raw = apply(millipore_raw,2,as.character)
> millipore_gene = millipore_raw[c(8:241),2]
> millipore_cat_id = millipore_raw[1,-c(1:6)]
> millipore_profile = as.data.frame(millipore_raw[c(8:241),c(7:166)])
> colnames(millipore_profile) = millipore_cat_id
> millipore_profile = apply(millipore_profile,2,as.numeric)
>
> # Gene symbol for row 140 should be PKA
> millipore_gene[140] = "PKA"
> # Gene symbol for row 187 should be C-RAF rather than RAF1
> millipore_gene[187] = "C-RAF"
> millipore_gene[220] = "TIE2" # change from TEK to TIE2
>
> # average the duplicates
> # 224 targets and 160 compounds
> tmp = as.data.frame(cbind(millipore_gene,millipore_profile))
> tmp[,-1] = apply(tmp[,-1], 2, as.numeric)
> tmp2 = ddply(tmp[,1:161],'millipore_gene',numcolwise(function(x) median(x)))
> millipore_profile_clean = tmp2[,-1]
> rownames(millipore_profile_clean) = tmp2[,1]
```

The curated Millipore drug-target data contain 160 compounds and 224 targets and each drug-target interaction was assigned into one of four potency categories: inactive (0), weakly active (1), active (2) or very active (3). Using the product IDs in Table 1 we manage to find 6 Miller compounds (AKT 1/2 Inh, FR180204, PDGFR TKI III, SL327, SRC Inh I and SU11247) from the Millipore data:

```
> # load the miller drug list, i.e. Table 1 of the tutorial
> data(miller_drugs)
> # create an empty drug-target matrix, 14 drugs and 224 targets
> miller_targets = as.data.frame(matrix(data=NA,nrow=14,ncol=224))
> colnames(miller_targets) = millipore_profile_clean[,1]
> rownames(miller_targets) = miller_drugs$Drug
> # search in the Millipore data
> index = match(miller_drugs$"Company.Product#", colnames(millipore_profile_clean))
> # 6 drugs can be found
> miller_targets[which(!is.na(index)),] = millipore_profile_clean[index[which(!is.na(index))],]
```

For the drug AG538 (abbr. AG), we go to the PubMed reference PMID 12869569 and find the following targets (with IC50s): IGF1R (61 nM), INSR (113 nM), EGFR (370 nM) and SRC (2000 nM). This target profile is loaded into *miller_targets* as below:

```
> # the target symbols and UniProt IDs
> AG.targets = c('IGF1R','INSR','EGFR','SRC')
> # AG.targets = c('P08069','P06213','P00533','P12931')
> index = match(AG.targets,colnames(miller_targets))
> # miller_targets['AG538',index] = log10(c(61,113,370,2000))
> miller_targets['AG538',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug Gefitinib (abbr. GF), we go the Tyner drug-target interaction data and find the following targets (n=16) within the 224-target list:

```
> index = grep("Gefitinib",rownames(tyner_interaction_binary))
> GF.targets = colnames(tyner_interaction_binary)[which(tyner_interaction_binary[index,]==1)]
> index = match(GF.targets,colnames(miller_targets))
> miller_targets['Gefitinib',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug HNHA (abbr. HN), we go to the PubMed reference PMID 17353008 and find its single target as HDAC (100 nM). However, HDAC is not in the 224-target list, we will add it as the 225th target:

```
> # HN.targets = c('Q13547')
> HN.targets = 'HDAC'
> miller_targets = cbind(miller_targets,NA)
> colnames(miller_targets)[225] = 'HDAC'
> index = match(HN.targets,colnames(miller_targets))
> miller_targets['HNHA',index] = 1
```

For the drug PI3Ka Inh IV (abbr. PI), we find the target IC50s from the product page of EMD Millipore as well as in the PubMed reference PMID 16837202, as PIK3CA (2 nM), PIK3CB (16 nM), PIK3C2B (220nM) and PIK3CG (660 nM). Again these targets are not covered in the current target list and thereafter added as new targets in *miller_targets*:

```
> # PI.targets = c('P42336','P42338','O00750','P48736')
> PI.targets = c('PIK3CA','PIK3CB','PIK3C2B','PIK3CG')
> miller_targets = cbind(miller_targets,NA,NA,NA,NA)
> colnames(miller_targets)[226:229] = PI.targets
> index = match(PI.targets,colnames(miller_targets))
> miller_targets['PI3Ka Inh IV',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug Rapamycin, we determine its target profile from the KiBA bioactivity data, which provides a summary scoring of drug-target interactions by integrating replicated data from multiple studies and assay conditions (Tang *et al.*, 2014):

```
> data(kiba) # load the KiBA drug-target interaction data
> # Rapamycin.targets = c('P00533', 'P04626', 'P06239', 'P06241', 'P17252',
'P17948', 'P27361', 'P28482', 'P42345', 'Q16539')
> Rapamycin.targets = c('EGFR', 'ERBB2', 'LCK', 'FYN', 'PRKCA', 'FLT1', 'MAPK3',
'MAPK1', 'MTOR', 'MAPK14')
> index = match(Rapamycin.targets,colnames(miller_targets))
> miller_targets['Rapamycin',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug Rottlerin (abbr. RT), the following targets can be also obtained from the KiBA data:

```
> # RT.targets = c('Q8IW41','Q05655','P49137')
> RT.targets = c('MAPKAPK5','PRKCD','MAPKAPK2')
> index = match(RT.targets,colnames(miller_targets))
> miller_targets['Rottlerin',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug Ryuvidine (abbr. RY), we find the bioactivity values from the PubMed reference PMID 24902048: CDK4 (IC50=600 nM) and CDC7 (IC50<10,000 nM).

```
> # RY.targets = c('P11802','O00311')
> RY.targets = c('CDK4','CDC7')
> miller_targets = cbind(miller_targets,NA,NA)
> colnames(miller_targets)[230:231] = RY.targets
> index = match(RY.targets,colnames(miller_targets))
> miller_targets['Ryuvidine',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

For the drug Stattic (abbr. ST), we find the target IC50s from the PubMed reference PMID 17114005 including STAT3 (5100 nM), STAT1 (<10,000 nM), STAT5 (<10,000 nM). These targets were not found previously and thus will be added.

```
> # ST.targets = c('P40763','P42224','P42229')
> ST.targets = c('STAT3','STAT1','STAT5')
> miller_targets = cbind(miller_targets,NA,NA,NA)
```

```
> colnames(miller_targets)[232:234] = ST.targets
> index = match(ST.targets,colnames(miller_targets))
> miller_targets['Stattic',index[which(!is.na(index))]] = rep(1,sum(!is.na(index)))
```

Finally, we finalize the drug-target matrix by the same binarization as done for the Tyner study, e.g. 0 stands for inactive target and 1 stands for active target. The final drug-target interaction matrix contains 14 drugs and 234 targets.

```
> # binarization
> miller_targets[which(is.na(miller_targets),arr.ind=T)] = 0
> miller_targets[which(miller_targets=="NA",arr.ind=T)] = 0
> miller_targets[which(miller_targets > 0, arr.ind=T)] = 1
```

The number of targets for each drug after the binarization is shown in Table 2. Note that for most of the drugs the literature review process revealed many targets showing either strong or weak binding affinities. Although most of them were not listed as the intended primary targets, they might still contribute to the drugs' mechanisms of actions. The data obtained from a global drug-target profiling (e.g. Gao *et al.* (2013) and Tyner *et al.* (2013)), or from the integration of multiple studies (e.g. Tang *et al.* (2014)) is an essential step to enable the prediction of drug combinations based on their polypharmacolgical interactions.

Table 2: The number of drug targets identified from the literature

Drug	Abbreviation	Number of targets	PubMed reference
AG538	AG	4	12869569
AKT 1/2 Inh	AK	64	23398362
FR180204	ER	58	23398362
Gefitinib	GF	16	24065146
HNHA	HN	1	17353008
PDGFR TKI III	PD	126	23398362
PI3Ka Inh IV	PI	4	16837202
Rapamycin		9	24521231
Rottlerin	RT	3	24521231
Ryuvidine	RY	2	24902048
SL327	SL	26	17114005
SRC Inh I	SR	75	23398362
Stattic	ST	3	23398362
SU11247	SU	79	23398362

4.2 The drug sensitivity data

The dose-response values of single and paired drug combinations on the tumor-derived cell line DDLS8817 can be downloaded from Table S2 of the Miller study. However, from Table S2 we can find only 13 drugs, whereas Rapamycin sensitivities were missing. Each of the 13 drugs were tested on 7 doses, where the responses were measured as the relative cell metabolic activities. We will first use the logistic curve-fitting to derive the cell viability IC50s using the *drc* package in R (Ritz and Streibig, 2005).

```
> install.packages("drc")
> install.packages("plyr")
> library(drc)
> library(plyr)
> data(miller_drug_response)
> # calcuate the IC50s based on dose-response curves
> miller_ic50 = ddply(miller_drug_response,"Drug",
function(x) {tmp=drm(Effect ~ Dose.uM, data = x, fct = LL.4(fixed = c(NA,0,1,NA),
names = c("SLOPE","MIN","MAX","IC50")),na.action=na.omit); summary(tmp)$coefficients[2,]})
```

The derived IC50s for the 13 drugs are listed in Table 3. The cancer cell is most sensitive to PI3Ka Inh IV (PI), Stattic (ST) and Ryuvidine (RY), while being resistant to AG538(AG), FR180204(ER) and SRC Inh I(SR).

Table 3: IC50 estimation using the four-parameter logistic regression in the *drc* package

Index	Drug	Estimate	Std. Error	t-value	p-value
1	AG	46.44124133	NA	NA	NA
2	AK	7.572383482	0.493068449	15.35767194	2.12E-05
3	ER	50.37797632	10	5.037797632	0.003974337
4	GF	12.76729578	0.358251586	35.63779277	3.27E-07
5	HN	7.131568813	0.462386355	15.42339806	2.08E-05
6	PD	4.288026375	0.197069574	21.75894678	3.80E-06
7	PI	0.879652961	0.0561139	15.67620433	1.92E-05
8	RT	1.884818728	0.155908377	12.0892717	6.84E-05
9	RY	1.117143932	0.16069572	6.951920883	0.000946274
10	SL	9.173322269	NA	NA	NA
11	SR	18.60253895	228.8209795	0.081297349	0.938359423
12	ST	0.863373171	0.086393759	9.993466865	0.000171484
13	SU	10.1542157	1.540481569	6.591585322	0.001207286

The drug sensitivity values are inversely scaled from the IC50s according to (1):

```
> miller_sensitivity = (max(miller_ic50$Estimate)-miller_ic50$Estimate)/
(max(miller_ic50$Estimate)-min(miller_ic50$Estimate))
> miller_sensitivity = as.data.frame(miller_sensitivity)
> rownames(miller_sensitivity) = miller_ic50$Drug
```

4.3   Drug combination analysis

We run the *timma* analysis on the 13 drugs for which the target profiles and drug sensitivities are available (Rapamycin was shown in the drug list but not included in the drug combination testing):
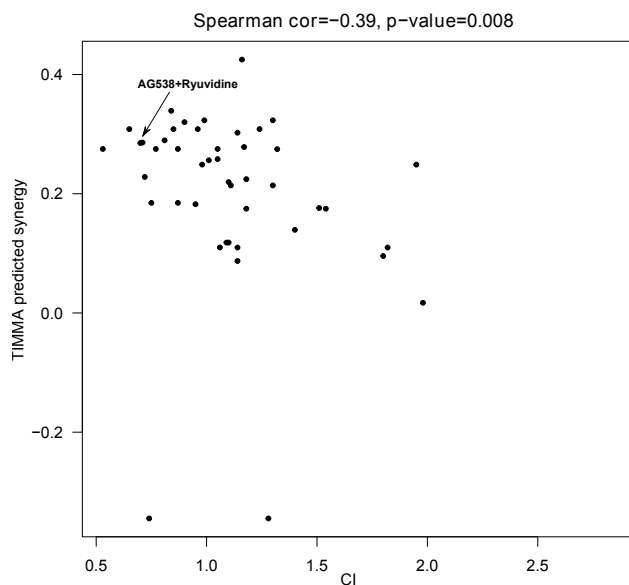

Figure 5: The predicted synergy scores as compared to the combination index (CI) in the Miller study. The negative correlation was expected since low CI indicates stronger synergistic effect.

```
> data(miller_interaction_binary)
> data(miller_sensitivity)
> timma(x = miller_interaction_binary,
y = as.matrix(miller_sensitivity[,1]), sp = 1, max_k = 8,verbosity = T)
```

The predicted synergy scores can be found in the result file *predictedScoring.csv*:

```
> predicted =
 read.table(file="predictedScoring.csv",header=T,sep=",") # read the predicted score data
```

Now we load the combination index data extracted from Figure 1 of Miller *et al.* (2013), and compare it with the multiplicative synergy score predicted by *timma*:

```
> # install.packages("reshape2")
> library(reshape2)
> data(ci) # load the combination index data of Miller study
> ci.pair = melt(as.matrix(ci),id=rownames(ci))
> colnames(ci.pair) = c("drug1","drug2","ci")
> ci.pair = ci.pair[-which(is.na(ci)==T),]
> drug.comb.ci = apply(ci.pair,1,function(x)
paste(sort(c(as.character(x[1]),as.character(x[2]))),collapse=" "))
> ci.pair = cbind(ci.pair,drug.comb.ci)
> drug.comb.timma = apply(predicted[,c("Drug1","Drug2")],1,
function(x)paste(sort(c(as.character(x[1]),as.character(x[2]))),collapse=" "))
> cor.test(ci.pair$ci,predicted$Synergy.mult[match(drug.comb.ci,drug.comb.timma)],method="spearman")
> plot(ci.pair$ci,predicted$Synergy.multi[match(drug.comb.ci,
drug.comb.timma)],xlab="CI",ylab="TIMMA synergy (multiplicative)")
```
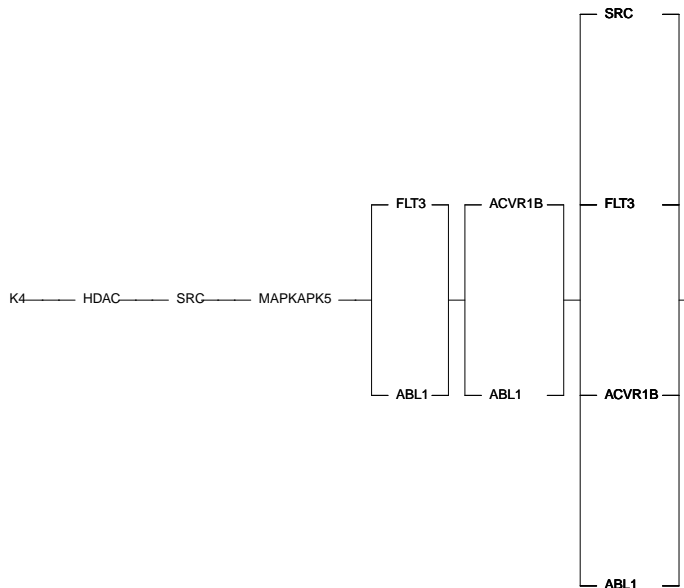

Figure 6: The target inhibition network for the DDLS-derived cancer cell.

The spearman correlation between the predicted TIMMA multiplicative synergy scores and the combination index scores is -0.39 with a p-value of 0.008 (Figure 5). Notably, TIMMA predicted that AG538 is synergistic with Ryuvidine with a synergy score of 0.285. Further, their targets IGF1R and CDK4 in combination was also predicted a sensitivity of 0.997, which is in line with the major finding of Miller *et al.* (2013). On the

other hand, the derived target inhibition network showes alternative target combinations, such as inhibitions of ABL1 and FLT3, or ABL1 and ACVR1B, that might be also synergistically blocking the cancer survival pathway (Figure 6).

5   Other data preprocessing functions

In this section we briefly describe two data preprocessing functions for binarizing drug-target interactions from quantitative binding affinity assays and for converting drug response IC50 values into drug sensitivity scores separately.

For binarizing drug-target interactions, we implemented a function *binarizeDrugTargets* and provided an example binding affinity data downloaded from a kinome-wide screening study (Davis *et al.*, 2011). The *davis* data include the dissociation constant (Kd) results for 72 drugs and 442 kinases from binding affinity assays.

```
> data(davis)
> davis_binary <- binarizeDrugTargets(davis, method="drug-specific", threshold="100fold")
```

We provided two methods for binarization. The method *drug-specific* utilizes a threshold which is n-fold greater than the minimal Kd value for each drug ($n = 10, 50, 100$). Alternatively, we may use a fixed threshold (e.g. 1,000 nM) across all the drugs, such as

```
> davis_binary <- binarizeDrugTargets(davis, method="universal", threshold="1000nM")
```

For converting the drug response IC50 values into a sensitivity score in a $[0, 1]$ scale, we implemented three methods including *minmax*, *logistic* and *hyperbolic*. The *minmax* method scales the IC50 values according to Eq (1) defined in section 3.1.2. The *logistic* method normalizes the IC50 values according to

$$y = \frac{1}{1 + e^{(-x)}} \tag{4}$$

,where $x = 1/IC50$. The *hyperbolic* method is

$$y = tanh(x) \tag{5}$$

These methods were implemented in a function *normalizeSensitivity* with the syntax:

```
> y <- normalizeSensitivity(IC50, method = c("minmax","logistic","hyperbolic))
```

6   Conclusion

In this tutorial, we showed the work flow of the drug combination analysis using the *timma* R package. The data from two real case studies, i.e. the Tyner study and the Miller study were shown as running examples. The Tyner study provided manually-curated drug-target interactions as well as well-characterized drug sensitivity data for 151 patient-derived samples, which make it an ideal case study for illustrating the analytical pipeline of TIMMA. Further, this study is highly translational, as the samples were derived from leukemia patients, so that the identified drug combinations could be readily tested in clinical settings. On the other hand, the Miller study provided the sensitivities for both single small-molecule drugs and drug combinations, which makes it an ideal case study to evaluate the validity of the predictions. However, the Miller study listed only one or two putative targets for each drug, whereas the polypharmacological connectivity patterns between the drugs at the proteome scale is largely missing. To maximize the prediction power of TIMMA, our first effort is to define the drugs polypharmacological profiles considering both strong and weak interactions, so that the effect of a drug combination can be modelled as the effects of their (multiple) target combinations. We queried these drugs in ChEMBL, as well as did a thorough search in literature for the bioactivity data from multiple studies. Finally, we have compiled a drug-target interaction profile including 234 targets. For the single-drug sensitivities on the cell line, we calculated the IC50s based on the dose-response data provided in Table S2 of Miller et al. (2013). The TIMMA analysis was then carried

out the same way as done for the Tyner study. The predicted drug synergy scores were compared with the Combination Index scores (CI). We found that the multiplicative synergy scores correlate the best with the CI scores. For both of the two case studies, a rich set of data preprocessing functions for the drug-target interactions and the drug sensitivities are covered. The R scripts provided in the supplementary material include downloading the literature raw data, binarization/multi-classification and drug sensitivity analysis using the drc R package and in addition to other data preprocessing functions. These case studies should provide very helpful guidelines on the drug-target interaction data processing, the drug sensitivity scoring and the drug combination predictions.

References

Bento, A. P., Gaulton, A., Hersey, A., Bellis, L. J., Chambers, J., Davies, M., KrÃijger, F. A., Light, Y., Mak, L., McGlinchey, S., Nowotka, M., Papadatos, G., Santos, R., and Overington, J. P. (2014). The chembl bioactivity database: an update. *Nucleic Acids Research*, **42**(D1), D1083–D1090.

Davis, M. I., Hunt, J. P., Herrgard, S., Ciceri, P., Wodicka, L. M., Pallares, G., Hocker, M., Treiber, D. K., and Zarrinkar, P. P. (2011). Comprehensive analysis of kinase inhibitor selectivity. **29**(11), 1046–1051.

Dusa, A. (2010). A mathematical approach to the boolean minimization problem. *Quality and Quantity*, **44**(1), 99–113.

Gao, Y., Davies, S. P., Augustin, M., Woodward, A., Patel, U. A., Kovelman, R., and Harvey, K. J. (2013). A broad activity screen in support of a chemogenomic map for kinase signalling research and drug discovery. *Biochemical Journal*, **451**(2), 313–328.

Gleeson, M. P., Hersey, A., Montanari, D., and Overington, J. (2011). Probing the links between in vitro potency, ADMET and physicochemical parameters. *Nature Reviews. Drug Discovery*, **10**(3), 197–208.

Law, V., Knox, C., Djoumbou, Y., Jewison, T., Guo, A. C., Liu, Y., Maciejewski, A., Arndt, D., Wilson, M., Neveu, V., Tang, A., Gabriel, G., Ly, C., Adamjee, S., Dame, Z. T., Han, B., Zhou, Y., and Wishart, D. S. (2014). Drugbank 4.0: shedding new light on drug metabolism. *Nucleic Acids Research*, **42**(D1), D1091–D1097.

Miller, M. L., Molinelli, E. J., Nair, J. S., Sheikh, T., Samy, R., Jing, X., He, Q., Korkut, A., Crago, A. M., Singer, S., Schwartz, G. K., and Sander, C. (2013). Drug synergy screen and network modeling in dedifferentiated liposarcoma identifies CDK4 and IGF1r as synergistic drug targets. *Science signaling*, **6**(294), ra85.

Ritz, C. and Streibig, J. C. (2005). Bioassay analysis using r. *Journal of Statistical Software*, **12**.

Tang, J., Karhinen, L., Xu, T., Szwajda, A., Yadav, B., Wennerberg, K., and Aittokallio, T. (2013). Target inhibition networks: Predicting selective combinations of druggable targets to block cancer survival pathways. *PLoS Computational Biology*, **9**(9), e1003226.

Tang, J., Szwajda, A., Shakyawar, S., Xu, T., Hintsanen, P., Wennerberg, K., and Aittokallio, T. (2014). Making sense of large-scale kinase inhibitor bioactivity data sets: A comparative and integrative analysis. *Journal of Chemical Information and Modeling*, **54**(3), 735–743.

Tyner, J. W., Yang, W. F., Bankhead, A., Fan, G., Fletcher, L. B., Bryant, J., Glover, J. M., Chang, B. H., Spurgeon, S. E., Fleming, W. H., Kovacsovics, T., Gotlib, J. R., Oh, S. T., Deininger, M. W., Zwaan, C. M., Den Boer, M. L., van den Heuvel-Eibrink, M. M., O'Hare, T., Druker, B. J., and Loriaux, M. M. (2013). Kinase pathway dependence in primary human leukemias determined by rapid inhibitor screening. *Cancer Research*, **73**(1), 285–296.

Zhu, F., Shi, Z., Qin, C., Tao, L., Liu, X., Xu, F., Zhang, L., Song, Y., Liu, X., Zhang, J., Han, B., Zhang, P., and Chen, Y. (2012). Therapeutic target database update 2012: a resource for facilitating target-oriented drug discovery. *Nucleic Acids Research*, **40**(D1), D1128–D1136.