User instructions and tutorials of the *synergyfinder* package

Liye He[1], Shuyu Zheng[1], Krister Wennerberg[1], Tero Aittokallio[1,2], and Jing Tang[1,3]

[1] *Institute for Molecular Medicine Finland (FIMM), University of Helsinki, Finland*
[2] *Department of Mathematics and Statistics, University of Turku, Finland*
[3] *Institute of Biomedicine, University of Helsinki, Finland*

July 6, 2020

1 Introduction

Recently, drug combination therapies provide a promising strategy in treating cancer by inhabiting redundant pathways simultaneously [1]. Drug combination screening in the cancer cell models is often utilized as a starting point to prioritize the most potential hits for further experimental investigation and therapy optimization [2]. To facilitate the drug combination discovery, high-throughput drug combination screening has the advantage of assaying a large collection of chemical compounds, generating dynamic dose-response profiles that allow us to quantify the degree of drug-drug interactions at an unprecedented level. A drug interaction is usually classified as synergistic, antagonistic or non-interactive, based on the deviation of the observed drug combination response from the expected effect of non-interaction (the null hypothesis). To quantify the degree of drug synergy, several models have been proposed, such as those based on the Highest single agent model (HSA) [3], the Loewe additivity model (Loewe) [4] and the Bliss independence model (Bliss) [5]. However, these existing drug synergy scoring models, together with their software implementations, were proposed initially for low-throughput experiments, which are not flexible enough to analyse high throughput drug combination screening data. We have recently developed a response surface model, called Zero Interaction Potency (ZIP), which combines the advantages of the Loewe and the Bliss models, and proposed a delta score to characterize the synergy landscape over the full dose-response matrix, which is designed to analyse high throughput drug combination screening data [6]. The package *synergyfinder* provides efficient implementations for all the popular synergy scoring models, including HSA, Loewe, Bliss and ZIP. We will demonstrate how to use the *synergyfinder* package to high throughput drug combination screening data in this vignette.

2 Input Data

A data frame that describes a drug combination dataset is used as the input. We enabled our package to complicate both "camel style" and "snake style" column names. The data frame object must contain the following columns:

- block_id(BlockID): (integer) the identifier for a drug combination. If multiple drug combinations are present, e.g. in the standard 384-well plate where 6 drug combinations are fitted, then the identifiers for each of them must be unique.
- drug_col(DrugCol): (character) the name of the drug on the columns in a dose-response matrix.
- drug_row(DrugRow): (character) the name of the drug on the rows in a dose-response matrix.
- conc_c(ConcCol) and conc_r(ConcRow): (numeric) the concentrations of the column drugs and row drugs in a combination.
- conc_c_unit(ConcColUnit) and conc_r_unit(ConcRowUnit): (character) the unit of concentrations. It is typically nM or μM.
- response(Response): the effect of drug combinations at the concentrations specified by ConcCol and ConcRow. The effect must be normalized to %inhibition based on the positive and negative controls. For a well-controlled experiment, the range of the response values is expected from 0 to 100. However, missing values (The parameter *impute* in for function *ReshapeData* is designed to handle missing values) or extreme values are allowed. For input data where the drug effect is represented as %viability, the program will internally convert it to %inhibition value by 100-%viability.

We provide an example input data in the package, which is extracted from a recent drug combination screening for the treatment of diffuse large B-cell lymphoma (DLBCL) [7]. The example input data contains two representative drug combinations (ibrutinib & ispinesib and ibrutinib & canertinib) for which the %viability of a cell line TMD8 was assayed using a 6 by 6 dose matrix design. To load the example data, we use the command:

```
data("mathews_screening_data")
head(mathews_screening_data)
```

```
##   block_id  drug_row  drug_col conc_r  conc_c  response
## 1        1 ispinesib ibrutinib   2500 50.0000  7.802637
## 2        1 ispinesib ibrutinib   2500 12.5000  6.831317
## 3        1 ispinesib ibrutinib   2500  3.1250 15.089589
## 4        1 ispinesib ibrutinib   2500  0.7812 24.503885
## 5        1 ispinesib ibrutinib   2500  0.1954 38.043076
## 6        1 ispinesib ibrutinib   2500  0.0000 45.790634
##   conc_r_unit conc_c_unit
## 1          nM          nM
```

```
## 2          nM          nM
## 3          nM          nM
## 4          nM          nM
## 5          nM          nM
## 6          nM          nM
```

We provide a function *ReshapeData* to reshape and pre-process the input data to a dose-response matrix format for further analysis:

```
# Set the random number seed for generating noises.
set.seed(1)
dose.response.mat <- ReshapeData(mathews_screening_data,
                                 data.type = "viability",
                                 impute = TRUE,
                                 noise = TRUE,
                                 correction = "non")
```

We provide 3 processes to adjust data:

- impute: Impute missing values in response data by calling function *ImputeNA*. As down stream functions can not handle the matrix with *NA* values, we recommend to set it as *TRUE*, if input matrix contains missing values.
- noise: Add small random number to response data to avoid the error caused by exact same response values. It is done by calling function *AddNoise*. We recommend to set it as *TRUE*.
- correction: Adjust base line of response matrix to make it closer to 0 by calling function *CorrectBaseLine*. We provide three options: *non*, do not correct base line; *part*, correct base line but only adjust negative response values in matrix; *all*, correct base line with adjusting all values in matrix.

The output *dose.response.mat* is a list containing the following components:

- dose.response.mats: the dose-reponse matrix reshaped from the original input.
- adjusted.response.mats: the dose-response matrix reshaped from the original input and adjusted by *impute*, *noise*, or *correction*. If no process was chosen (set as *impute=FALSE, noise=FALSE, correction="non"*), the output will not contain this list.
- drug.pairs: the information of the drug combination such as drug names and drug concentrations.

```
str(dose.response.mat)

## List of 3
##  $ dose.response.mats    :List of 2
##   ..$ 1: num [1:6, 1:6] -23 59.2 60.2 60.8 60.6 ...
```

```
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##   ..$ 2: num [1:6, 1:6] -15 -53.9 -87.2 -81.2 -51.5 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##  $ adjusted.response.mats:List of 2
##   ..$ 1: num [1:6, 1:6] -23 59.2 60.2 60.8 60.6 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##   ..$ 2: num [1:6, 1:6] -15 -53.9 -87.2 -81.2 -51.5 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##  $ drug.pairs            :'data.frame': 2 obs. of  5 variables:
##   ..$ block_id   : int [1:2] 1 2
##   ..$ drug_row   : chr [1:2] "ispinesib" "canertinib"
##   ..$ drug_col   : chr [1:2] "ibrutinib" "ibrutinib"
##   ..$ conc_r_unit: chr [1:2] "nM" "nM"
##   ..$ conc_c_unit: chr [1:2] "nM" "nM"
```

3 Dose response matrix visualization

The function *PlotDoseResponse* fits a four-parameter log-logistic model to generate the dose-response curves for the single drugs based on the first row and first column of the dose-response matrix. The drug combination responses are also plotted as heatmaps, from which one can assess the therapeutic significance of the combination, e.g. by identifying the concentrations at which the drug combination can lead to a maximal effect on cancer inhibition (Fig. 1).

```
  PlotDoseResponse(dose.response.mat)

## NULL
```

The PlotDoseResponse function also provides a parameter *save.file*. It will save the plots of each drug combination as high-resolution pdf files:

```
  PlotDoseResponse(dose.response.mat, save.file = TRUE)

## NULL
```

The pdf files will be saved under the current work directory with the filename: "DrugRow.DrugCol.adjusted.dose.response.BlockID.pdf" or "DrugRow.DrugCol.dose.response.BlockID.pdf"(if original input data is plotted).
The PlotDoseResponse function also provides other parameters for users to adjust the output. For instance, the parameter *adjusted = FALSE* allows user to

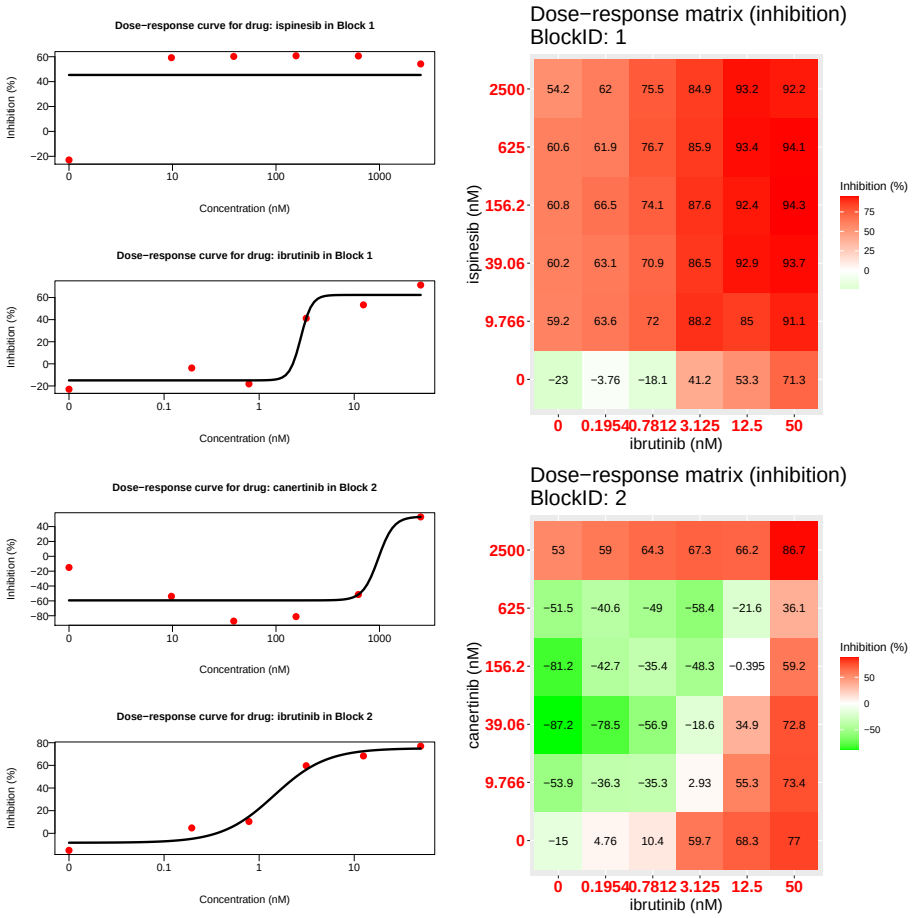

Figure 1: **Plots for single drug dose-response curves and drug combination dose-response matrices.** (A) The ibrutinib and ispinesib combination. (B) The ibrutinib and canertinib combination. Left panel: single drug dose-response curves fitted with the commonly-used 4-parameter log-logistic (4PL) function. Right panel: Heatmap of the dose-response matrix.

plot the original input data (if there is no missing values in the original data), the parameter *pair.index* allows users to choose which pair of drugs (by block IDs) to visualize and the parameters ... are further graphical parameters from the default plot function in R.

4 Drug synergy scoring

The current synergyfinder package provides the synergy scores of four major reference models, including 'HSA', 'Loewe', 'Bliss' and 'ZIP'. Let's consider a drug combination experiment where drug 1 at dose x_1 is combined with drug 2 at dose x_2. The effect of such a combination is y_c as compared to the monotherapy effect $y_1(x_1)$ and $y_2(x_2)$. To be able to quantify the degree of drug interactions,

one needs to determine the deviation of y_c from the expected effect y_e of non-interaction, which is calculated in different ways with the reference models.

- HSA: y_e is the effect of the highest monotherapy effect, i.e. $y_e = max(y_1, y_2)$.
- Loewe: y_e is the effect as if a drug is combined with itself, i.e. $y_e = y_1(x_1 + x_2) = y_2(x_1 + x_2)$.
- Bliss: y_e is the effect as if the two drugs are acting independently on the phenotype, i.e. $y_2 = y_1 + y_2 - y_1 y_2$.
- ZIP: y_e is the effect as if the two drugs do not potentiate each other, i.e. both the assumptions of the Loewe model and the Bliss model are met.

Once y_e can be determined, the synergy score can be calculated as the difference between the observed effect and the expected effect. According to whether score is negative or positive, the drug combination can be classified as synergistic or antagonist, respectively. Furthermore, as the input data has been normalized as %inhibition values then the synergy score can be directly interpreted as the proportion of cellular responses that can be attributed to the drug interactions.

For a given dose-response matrix, one need to first choose the reference model and then apply the *CalculateSynergy* function to calculate the corresponding synergy scores at each dose combination. For example, the ZIP-based synergy score for the example data can be obtained by calling:

```
  synergy.score <- CalculateSynergy(data = dose.response.mat,
                                    method = "ZIP")

## Warning:  Convergence failed.  The model was not fitted!
## Warning:  Convergence failed.  The model was not fitted!
## Warning:  Convergence failed.  The model was not fitted!
## Warning:  Convergence failed.  The model was not fitted!
```

Other reference models can be chosen by setting the *method* parameter as 'HSA', 'Loewe' or 'Bliss'.

The output *synergy.score* contains the following components:

- dose.response.mats: the dose-reponse matrix reshaped from the original input.
- adjusted.response.mats: the dose-response matrix reshaped from the original input and adjusted by *impute*, *noise*, or *correction*. (Might not exist)
- drug.pairs: the information of the drug combination such as drug names and drug concentrations.
- scores: a score matrix of the same size to facilitate a dose-level evaluation of drug synergy as well as a direct comparison of the synergy scores between two reference models.

- method: which method is used to generate the synergy scores.

```
  str(synergy.score)
```

```
## List of 5
##  $ dose.response.mats    :List of 2
##   ..$ 1: num [1:6, 1:6] -23 59.2 60.2 60.8 60.6 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##   ..$ 2: num [1:6, 1:6] -15 -53.9 -87.2 -81.2 -51.5 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##  $ adjusted.response.mats:List of 2
##   ..$ 1: num [1:6, 1:6] -23 59.2 60.2 60.8 60.6 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##   ..$ 2: num [1:6, 1:6] -15 -53.9 -87.2 -81.2 -51.5 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##  $ drug.pairs            :'data.frame': 2 obs. of  5 variables:
##   ..$ block_id   : int [1:2] 1 2
##   ..$ drug_row   : chr [1:2] "ispinesib" "canertinib"
##   ..$ drug_col   : chr [1:2] "ibrutinib" "ibrutinib"
##   ..$ conc_r_unit: chr [1:2] "nM" "nM"
##   ..$ conc_c_unit: chr [1:2] "nM" "nM"
##  $ scores                :List of 2
##   ..$ 1: num [1:6, 1:6] 0 0 0 0 0 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##   ..$ 2: num [1:6, 1:6] 0 0 0 0 0 ...
##   .. ..- attr(*, "dimnames")=List of 2
##   .. .. ..$ : chr [1:6] "0" "9.7656" "39.0626" "156.25" ...
##   .. .. ..$ : chr [1:6] "0" "0.1954" "0.7812" "3.125" ...
##  $ method                : chr "ZIP"
```

5 Visualization of synergy scores

The synergy scores are calculated across all the tested concentration combinations, which can be straightforwardly visualized as either a two-dimensional or a three-dimensional interaction surface over the dose matrix. The landscape of such a drug interaction scoring is very informative when identifying the specific dose regions where a synergistic or antagonistic drug interaction occurs. The height of the 3D drug interaction landscape is normalized as the % inhibition

effect to facilitate a direct comparison of the degrees of interaction among multiple drug combinations. In addition, a summarized synergy score is provided by averaging over the whole dose-response matrix. To visualize the drug interaction landscape, one can utilize the *PlotSynergy* function as below (Fig.2):

```
  PlotSynergy(synergy.score, type = "all", save.file = TRUE)

## [1] 1.147
```

The *type* parameter specifies the visualization type of the interaction surface as 2D, 3D or both.

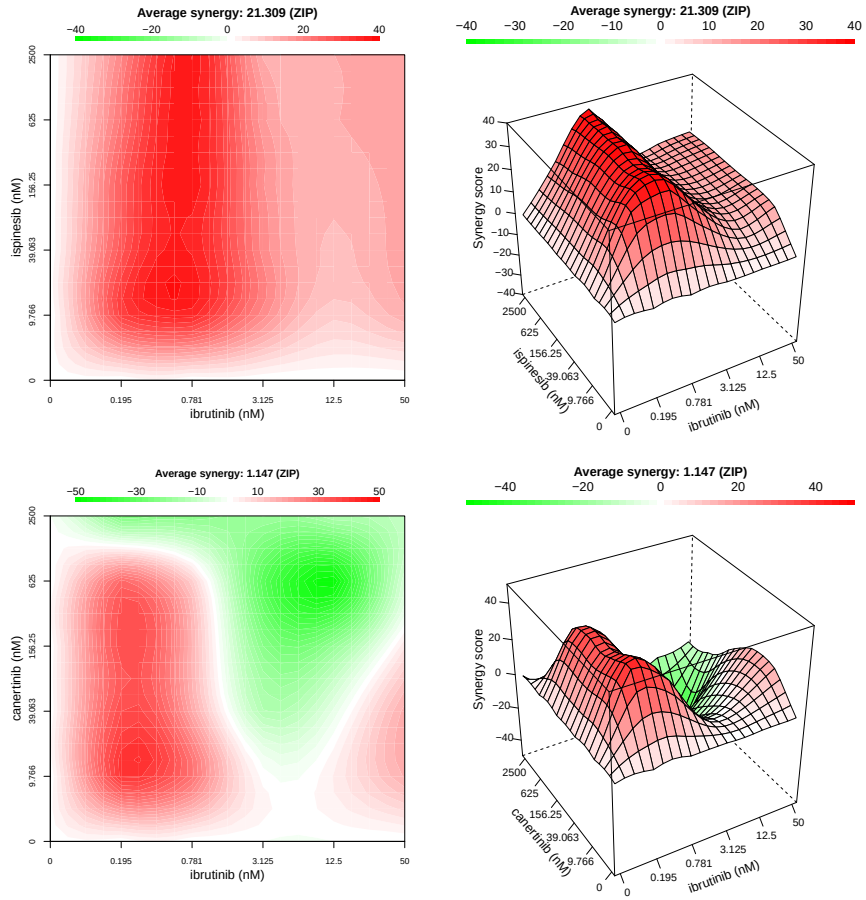

Figure 2: **The drug interaction landscapes based on the ZIP model.** (A) The ibrutinib and ispinesib combination. (B) The ibrutinib and canertinib combination.

6 Summary

In this vignette, we demonstrated how to use the package *synergyfinder* with an example high-throughput drug combination screening data from [7]. We followed the procedure: illustrating input data, visualising input data, scoring input data and visualising synergy scores. Please go to references [2, 6] for more information about the scoring methods.

References

[1] Hu Y, Gupta-Ostermann D, Bajorath J (2014) Exploring compound promiscuity patterns and multi-target activity spaces. Comput Struct Biotechnol J 9: 1-11

[2] He LY, Kulesskiy E, Saarela J et al (2018) Methods for High-throughput Drug Combination Screening and Synergy Scoring. Methods Mol Biol. 1711: 351-398

[3] Berenbaum MC (1989) What is synergy. Pharmacol Rev 41: 93-141

[4] Loewe S (1953) The problem of synergism and antagonism of combined drugs. Arzneimittelforschung 3: 285-290

[5] Bliss CI (1939) The toxicity of poisons applied jointly. Ann Appl Biol 26: 585-615

[6] Yadav B, Wennerberg K, Aittokallio T et al (2015) Searching for drug synergy in complex dose-response landscapes using an interaction potency model. Comput Struct Biotechnol J 13: 504-513

[7] Mathews Griner LA, Guha R, Shinn P et al (2014) High-throughput combinatorial screening identifies drugs that cooperate with ibrutinib to kill activated B-cell-like diffuse large B-cell lymphoma cells. Proc Natl Acad Sci USA 111: 2349-54

[8] Alina Malyutina, Muntasir Mamun Majumder, Wenyu Wang et al (2019) Drug combination sensitivity scoring facilitates the discovery of synergistic and efficacious drug combinations in cancer. PLOS Computational Biology 15(5): e1006752.

[9] Zagidullin Bulat, Jehad Aldahdooh, Shuyu Zheng et al (2019) DrugComb: An Integrative Cancer Drug Combination Data Portal. Nucleic Acids Res doi:10.1093/nar/gkz337