

## **BRAT: An R-Shiny Application on Block Randomization Allocation Process for Animal Experiments in Clinical Research Area**

Shuo Zhang, Chun Yi, BeiGene Co., Ltd.

### **ABSTRACT**

Randomization is a common knowledge for statistical programmer in clinical trial, which is also needed for an animal experiment in research area. This paper illustrates the process of an application tool (BRAT, Block Randomization Allocation Tool) that is developed for automatically performing block randomization allocation. Users (Scientists) can get the following results on their own just by uploading the data per instruction and interactively choosing parameters: 1) "Automatically" generate block randomization allocation in the study-design level. 2) Simple data cleaning before analysis. 3) Simple exploration analysis. 4) Results visualization. This app could reduce time and resource needed for exploration and communication, for both statisticians and scientists.

Before this application, the biometrics team often takes time to manually calculate and communicate with scientists on different batches of sample data anytime there's need, which is kind of time-consuming. R Shiny provided an interactive way for scientists to load the sample directly and get randomization right away.

### **INTRODUCTION**

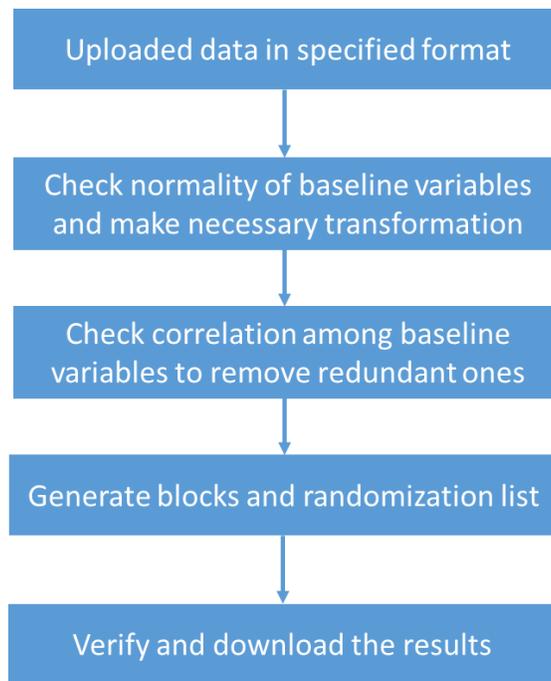
Randomized block experimental designs have been widely used in industrial research for many decades. Usually they are more powerful, have higher external validity, are less subject to bias, and produce more reproducible results than the completely randomized designs typically used in research involving laboratory animals. Previously, when performing block randomization designs for research scientists, statisticians often update their existing R programs whenever there are new comments from scientists, another batch of animal data, or covariates to add into analysis. The traditional process is kind of time-consuming and not efficient enough. A more efficient, automatically, and user-friendly way is necessary for programmers in supporting this process.

Shiny is an R package that makes it easy to build interactive web application with R, with no need to having knowledge on HTML, CSS, or JavaScript. In drug development area, more and more pharmaceutical companies have started using Shiny for various purposes along the whole drug development. Statisticians often use RStudio to perform some exploratory analysis, which makes it convenient to transform to an R Shiny app. After deploying the app, users can run the app on Shiny server, which allows much convenience as generating a web link for broader use of the app without requiring any analytical software/packages installed in the user's end. In other words, research scientists can do the randomization themselves only with a web-link, which can accordingly reduce the repetitive work of statisticians.

Therefore, this R Shiny application BRAT (Block Randomization Allocation Tool) is designed to generate a random assignment of experimental animals, while keeping the key baseline variables balanced across groups. There are four main advantages of this tool: 1) Help uses "automatically" generate block randomization allocation in the study-design level; 2) Simple data cleaning before analysis; 3) Simple exploration analysis; 4) Results visualization to help scientists to confirm the results.

### **WORKFLOW OF THE APP**

The animals are first divided into blocks, within which the animal baseline characteristics are much more similar than between blocks. Then animals are randomized to treatment groups within each block, according to pre-specified group sizes (i.e., allocation ratio). In statistical terminology, this is called block randomization / randomized block design. Such baseline variables are referred to as blocking variables in this tool. Figure 1 shows the flowchart of the R Shiny app workflow.



**Figure 1 Flowchart of the R Shiny Application workflow**

## **WORKING PROCESS DEMO BY PAGES**

Here the BRAT will be introduced by pages because the page sequence is also designed by the actual working flow. Demo as below.

### **ABOUT PAGE**

Users of BRAT are research scientists who may not be trained on statistics. An introduction would be necessary to instruct users about each module of the app and how to use it correctly.

In this page, it introduces the goal, advantages, special notes of the app, and notified on the future improvement. It helps users quickly get the background before running the app.

What's more, the links of reference papers are also listed in this page, users (scientists) can just click if they'd like to know more about the block randomization process.

### Introduction

**Goal:** To generate a random assignment of experimental animals, while keeping the key baseline variables balanced across groups. The current implementation only accepts continuous baseline.

**How to achieve that goal:** The animals are first divided into blocks, within which the animal baseline characteristics are much more similar than between blocks. Then animals are randomized to treatment groups within each block, according to pre-specified group sizes (i.e. allocation ratio). In statistical terminology, this is called block randomization / randomized block design. Such baseline variables are referred to as blocking variables in this tool.

**Caution:** a) Depending on the baseline data, the algorithm may not guarantee satisfactory balance. The merit of the tool is to avoid subjective bias. b) There should not be too many baseline variables to be balanced over. The suggestion is no more than 3, including both continuous and categorical baselines. ONLY include the important ones.

**Planned future improvement** includes allowing for a) categorical baseline, b) unequal group sizes, and c) deleting selected raw data on this web interface.

### Workflow

Below is an overview. More details can be found on the corresponding page.

-  Upload data
-  Check normality of baseline variables, make necessary transformation
-  Check correlation among baseline variables to remove redundant ones
-  Generate blocks and randomization list
-  Verify and download results

**Figure 2 User Interface of the “About” Page**

## UPLOAD DATA PAGE

The second page is for users to upload data in pre-specified format.

[About](#)
[Upload Data](#)
[Check Normality](#)
[Check Correlation](#)
[Randomization](#)
[Visual Check](#)

**File loading**

1. Please upload .xlsx file.
2. The identification variable must be named as "Animal" in input data.
3. Please remove any animal with missing baseline measurement.
4. Non-numeric values are not accepted.
5. Please remove redundant data and only upload the data to be randomized.

 Example data

Browse... test\_dld

Upload complete

Animal	Weight	Score
7	32.6	0
8	40.1	0
9	45.6	0
11	40.8	0
13	46.1	0
18	47.1	0
20	44.7	0
22	36.1	0
23	45.3	0
24	40.5	0
30	45.4	0

**Figure 3 User Interface of “Upload Data” Page**

In this page, we have specific uploading instructions to make it clear of the “Dos and Don’ts” of the data. By specifying the file type, necessary variables, and whether need to include missing values in advance, we can avoid potential issues on data processing. This enables us to be more focused on the randomization process instead of repetitive communication about data issue or formatting.

What’s more, to guide the users intuitively, we give an example file (Figure 4) that user can download it to see what kind of data they need to upload.

Animal	Weight	Score	Ab
7	32.6	0	53.0303
8	40.1	0	6853.448
9	45.6	0	45707.07
11	40.8	0	0
13	46.1	0	141666.7
18	47.1	0	33160.92
20	44.7	0	407.2989

Figure 4 Example for the uploaded data

## CHECK NORMALITY PAGE

Many of the statistical procedures including correlation, regression, t-test, and ANOVA, namely parametric tests, are based on the assumption that the data follows a normal distribution. So, it’s very important to check whether the samples are normally distributed, which is this page designed for.

Normality is desired to achieve more balanced result. For each variable, a distribution plot and the Shapiro-Wilk normality test are provided. The null hypothesis is that data is from a normal distribution, while the alternative is that data is not from a normal distribution. Therefore, a large p-value is desired, suggesting no evidence of deviation from normality.

For each continuous variable, we first check normality on the original scale. If there is severe deviation, we make the log transformation (zeros are replaced by half of the minimum observed positive value).

After the Normality test and log transformation, users can create the dataset for randomization by choosing specific original variables or their log-transformed ones. After choosing the variables, data will display accordingly in the right-hand-side.

We also put this simple instruction into the pages with text, along with attentions for users.

Figure 5 User Interface of “Check Normality” Page

## CHECK CORRELATION PAGE

The blocking variables should NOT be correlated with each other. If two variables are correlated, they provide redundant information and one of them should be removed from being a blocking variable.

Three correlation measures are provided, with Pearson as the default method (which is suitable when both variables follow normal distribution). The other two methods do not assume normality in variables and could be used as a sensitivity check, when in the previous normality check the test p-value is small or distribution plot is asymmetric.

The correlation test null hypothesis is that two variables have correlation of 0, while the alternative is that they are correlated. Therefore, a large p-value is desired, suggesting no evidence of correlation.

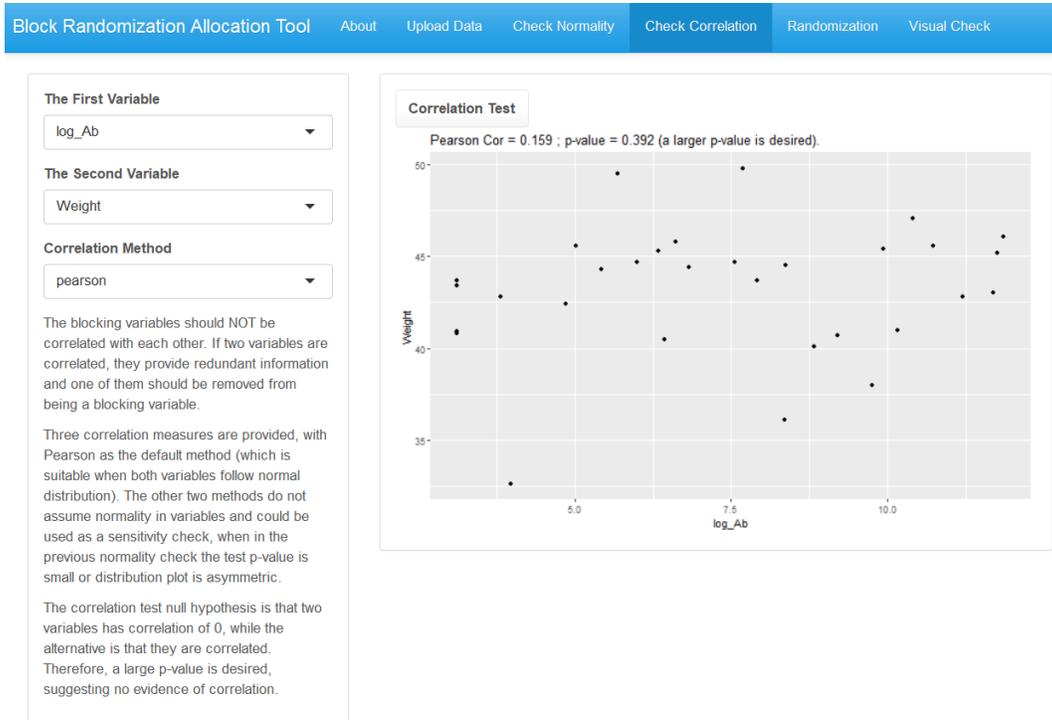


Figure 6 User Interface of “Check Correlation” Page

## RANDOMIZATION PAGE

This is the core page of BRAT. After choosing the proper variables into the model, block randomization is performed in this page. In the left-hand-side of this page, users can finally get the randomization results through 3 steps:

1. Choose the expected number of treatment groups.
2. Set a random seed. The purpose of specifying a random seed is to guarantee reproducibility of the result. In other words, you always get the same randomization result when you put same seed, and a different seed leads to another result. There is no restriction on how to specify it and users can just put an arbitrary number.
3. Afterwards, the chosen variables from previous page just display in the *Blocking Variables* text frame. Default distance measure is “Euclidean” and default agglomeration method is “Complete” according to reference paper. Distance measure and agglomeration method are input for hierarchical clustering, displayed for completeness. The default value is the expected methods and users could just keep them untouched.

Block Randomization Allocation Tool    About    Upload Data    Check Normality    Check Correlation    **Randomization**    Visual Check

**Number of treatment groups**

**Random seed**

The purpose of specifying a random seed is to guarantee reproducibility of the result. In other words, you always get the same randomization result when you put same seed, and a different seed leads to another result. There is no restriction on how to specify it, just put an arbitrary number.

**Blocking Variables**

**Distance Measure**

**Agglomeration Method**

Distance measure and agglomeration method are input for hierarchical clustering, displayed for completeness. Please keep them untouched.

**Study ID**

Go!    Download data    Download report

Randomization list    Summary by treatment group

**Animals excluded from randomization**

Group	Animal	Cluster	Score	log_Ab
G1	11	Cluster_1	0	3.12
G1	57	Cluster_2	0	7.56
G1	77	Cluster_3	0	7.91
G1	84	Cluster_4	0	11.7
G2	20	Cluster_2	0	6
G2	30	Cluster_3	0	9.94
G2	82	Cluster_1	0	3.12
G2	9	Cluster_4	0	10.73
G3	23	Cluster_2	0	6.33
G3	45	Cluster_3	0	8.38
G3	66	Cluster_1	0	3.12
G3	79	Cluster_4	0	10.17
G4	33	Cluster_4	0	11.76
G4	53	Cluster_3	0	9.2
G4	61	Cluster_1	0	4.86

**Figure 7 User Interface of “Randomization” Page**

After the necessary parameters input for the model, users can click the “Go!” button and get the randomization list. Besides, a report will also be provided, including both the results and other information including study ID, run date etc., which is generated by inserted R Markdown language.

In the right-hand-side, final randomization list displays along with expected animal groups and blocks (Cluster variable), which scientists can use to assign animal treatment groups.

In the *Randomization List* tab, we also provide the animals excluded from randomization. Now the application is designed only for balanced design, so there are same number of animals in each block and redundant animals are excluded accordingly. For example, if there are 31 animals to perform block randomization and 7 groups are needed, then there will be at least 4 blocks, with 4 animals in each group and 7 animals for each block, then  $(31-4*7) = 3$  animals will be excluded. List is provided for scientists if they need to check which animals are excluded in the analysis.

In the *Summary by treatment group* tab, means and SDs are showed to see whether it’s balanced across groups, which is showed in Figure 8.





Figure 10 User Interface of the “Visual Check” Page

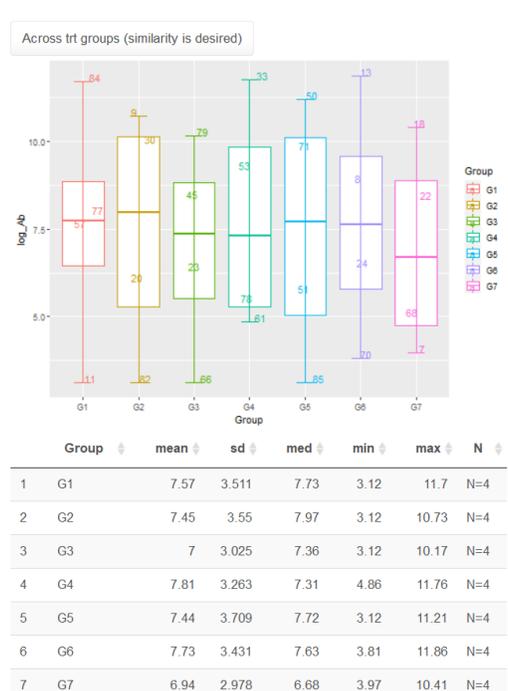


Figure 11 Example of the descriptive analysis among blocks and among groups

## CONCLUSION

Besides the advantages mentioned before, BRAT has its own specialties, including,

1. Inform scientists the interpretation of the results in favor of multiple exploratory analysis.
2. Integrate R Shiny, statistical models, R plotting and R Markdown language effectively.
3. Enough guidance allowing new app users to learn from scratch, such as the template upload data.

4. Generate a report showing the information including dates, usernames, animal batches etc., which is benefit for recording and tracing back.

For statistical programmers, knowledge about statistics and ideas on how to release the repetitive work with tools may benefits a lot in our daily tasks, besides, finding proper tools for specific application scenarios is more important. R Shiny is really fit some exploration analysis with simple data manipulation, and more friendly for new learners of R to build interactive tools.

## REFERENCES

1. Shiny Home Page: <http://shiny.rstudio.com/>
2. Iturria SJ. A method for obtaining randomized block designs in preclinical studies with multiple quantitative blocking variables. Pharm Stat. 2011;10(2):169-174.  
<https://pubmed.ncbi.nlm.nih.gov/20572260/>
3. Keene ON. The log transformation is special. Stat Med. 1995;14(8):811-819.  
<https://pubmed.ncbi.nlm.nih.gov/7644861/>
4. Festing MF. Randomized block experimental designs can increase the power and reproducibility of laboratory animal experiments. ILAR J. 2014;55(3):472-476.  
<https://academic.oup.com/ilarjournal/article/55/3/472/645707>

## ACKNOWLEDGMENTS

Many thanks to our statistician colleagues Xikun Wu and Xiaolei Xun for their guidance and supporting in building the R Shiny application together.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

<Shuo Zhang>  
<shuo1.zhang@beigene.com>

<Chun Yi>  
<chun.yi@beigene.com>