J. Biomedical Science and Engineering, 2011, 4, 734-739
doi:10.4236/jbise.2011.411090 Published Online November 2011 (http://www.SciRP.org/journal/jbise/).

Implementation of an open-source customizable minimization program for allocation of patients to parallel groups in clinical trials

Mahmoud Saghaei[1], **Sara Saghaei**[2]

[1]Department of Anesthesia, Isfahan University of Medical Sciences, Isfahan, Iran;
[2]School of Computing Science, Simon Fraser University, Vancouver, Canada.
Email: mahmood.saghaei@gmail.com

Received 23 July 2011; revised 30 August 2011; accepted 10 October 2011.

ABSTRACT

Current minimization programs do not permit full control over different aspects of minimization algorithm such as distance or probability measures and may not allow for unequal allocation ratios. This article describes the implementation of "MinimPy" an open-source minimization program in Python programming language, which provides full customizetion of minimization features. MinimPy supports naive and biased coin minimization together with various new and classic distance measures. Data syncing is provided to facilitate minimization of multi-center trial over the network. MinimPy can easily be modified to fit special needs of clinical trials and in particular change it to a pure web application, though it currently supports network syncing of data in multicenter trials using network repositories.

Keywords: Clinical Trial; Allocation Methods; Randomization; Minimization; Unequal Group Allocation; Biased Coin Minimization; Network Synchronization of Data

1. INTRODUCTION

According to the last version of Consolidated Standards of Reporting Trials (CONSORT) [1], the method of subject allocation to treatment groups in use must be clearly stated and the details of the sequence generation must be specified. Different methods of patients allocation have been advocated, of which the randomization methods have gained wide acceptance and is used in the majorities of clinical trials due to its relative simplicity and availability of software programs customizable to different randomization protocols [2]. Randomization methods although prevent selection bias and ensure the nonpredictability of treatment assignments, may result in significant imbalances in patients factors specially with smaller sample sizes [3]. Resulting imbalances may reduce the validity of data analysis or necessitate complex statistical analysis to overcome the factors imbalances. Different stratification and minimization techniques may be used to balance levels of prognostic factors among treatment groups. However minimization methods seem to provide more acceptable results compared to stratifycation [4]. Minimization is based on techniques that make the imbalance among patients' factors as low as possible and make the treatment groups more comparable to one another, with respect to basal patients' factors. In pure minimization subject allocations are completely deterministic and therefore may be predictable by the research team if the next patient's factors' levels are known, which in turn violates the principle of allocations' blindness. To overcome this pitfall, some components of randomization have been included in minimizetion algorithms, which reduces the chance of prediction, by giving higher allocation probabilities to interventions selected in favor of reducing total imbalance. Therefore the minimization algorithm can be viewed as a dynamic allocation method in which each allocation is influenced by the current state of overall treatment balances.

Initial description and methods of minimization were introduced independently by Taves [5] in 1974 and Pocock and Simon [6] in 1975. Subsequent investigators presented numerous contributions to the original algorithms and described different features of minimization methods [7-10].

Unfortunately, implementation of minimization algorithms necessitates some relatively difficult computational work which falls outside the skills and time limits of a clinical researcher. The complexity increases as the number of prognostic factors increase. In addition, prognostic factors may have different weights, which increase the computational complexity of the algorithm specially in the presence of unequal allocation ratios for different treatment groups.

In recent years some web based software have been developed to facilitate the allocation of patients in clinical trials based on minimization methods. The first minimization program "Minim" a free MS-DOS application although allows some control over minimization aspects, is a command line program; Hence its application limited to expert users [11]. Also its source is not available and there is no documentation of its algorithm. "MagMin," a Browser/Server closed source system based on the method of Pocock and Simon, was developed as a multi-center system which uses standard deviation as the distance measure [12]. However, users of this programs have limited choice over different aspects of minimization model such as distance or probability measures. It may also not allow for unequal allocation ratios. This article describes the implementation of "MinimPy" an open-source desktop minimization program written in python programming language with complete customization of minimization features.

2. DEVELOPMENT

2.1. Allocation Probability

The first subject is allocated randomly to one of treatment groups. Allocation of subsequent patients involves hypothetical stepwise allocation of each subject to every treatment group and computation of the imbalance score corresponding to each allocation. Produced imbalance scores will be compared and subject will be allocated to the group corresponding to the least imbalance score (preferred treatment). To include an element of randomization, usually the subject is allocated to the preferred treatment with a higher probability denoted as P_H, and to other groups (non-preferred treatment) with lower probabilities referred to as P_L. Depending on the method of minimization P_H and P_L may or may not be affected by the allocation ratios (*i.e.* unequal group sizes). In the simple form (naive minimization) the probabilities are not affected by allocation ratios and the same P_H is used for all treatment groups when they are selected as the preferred treatments. Probabilities of the non-preferred treatments are estimated equal to one another as $P_L = (1 - P_H)/(n - 1)$, and the allocation ratios are only used to correct counts of factor levels during calculation of imbalance scores. However, it sounds logical to assign higher probabilities to treatments with higher allocation ratios, which is the method proposed by Han and co-workers [13], known as the biased-coin minimization. A base P_H (P_{Hb}) is used for the group with the lowest allocation ratio when that group is selected as the preferred treatment. The P_H for other groups (P_{Hi}) when selected as the preferred treatments are calculated as a function of P_H and allocation ratios ($r_1, r_2, \cdots, r_n$) [13]:

$$P_{Hi} = 1 - \frac{\sum_1^n r_k \{k \neq i\}}{\sum_1^n r_k \{k \neq b\}} (1 - P_{Hb}) \tag{1}$$

P_L for the non-preferred group, i, when group j is the preferred treatment ($P_{Li\{j=H\}}$) and $i \neq j$ is calculated as [13]:

$$P_{Li\{j=H\}} = \frac{r_i}{\sum_1^n r_k \{k \neq j\}} (1 - P_{Hj}) \tag{2}$$

2.2. Imbalance Score

At each round of allocation, imbalance scores are calculated as a function of current allocations after the new case is hypothetically allocated to each treatment group. Different distance measures including range, variance, standard deviation have been used for calculation of imbalance scores [6]. Marginal balance proposed by Han and co-workers [13] tend to minimize more accurately when treatment groups are not equal in size. For each factor level marginal balance is calculated as a function of the adjusted number of patients at that level for each treatment group:

$$b_M = \frac{\sum_{i=1}^{n-1} \sum_{j=i+1}^{n} |a_i - a_j|}{(n-1) \sum_1^n a_i} \tag{3}$$

Other distance measures are implemented in the program as documented previously and used widely in other applications too. At the start of the trial user can choose from among different methods as explained in the implementation section.

2.3. Program Interface

Python [14] programming language was used to implement the model and the interface of the minimization program. A simple model structure was defined to hold different features of a minimization instance. These include:

- Allocations: Current state of trial allocations;
- Probability measure: simple probability assignment or biased coin minimization;
- Distance measure: Range, SD, variance or marginal balance used to calculate imbalance scores;
- PHb: High probability for the group with the lowest allocation ratio when selected as the preferred treatment;
- Variables weight: The weight assigned to each prognostic factor;
- Allocation ratio: A series of integer values, one for

each group, denoting the share of each group in the total sample size.

A minimization class was implemented to integrate different calculations of minimization algorithm. This class uses the model structure defined above for instanttiation. Various features of minimization algorithm were calculated in this class including:

- Calculation of probability assignments: a table of high and low probabilities were constructed as a function of allocation ratios, taking into account the value of PHb. Formula (1) and (2) are used for this purpose, in the case of the biased coin minimization;
- Calculation of the distance measures for the selected minimization model;
- Functions for handling tie conditions (when groups' imbalance scores are tied).

The program uses Python's random module to provide randomization component of minimization model which uses Mersenne Twister [15] as its core generator. However the user can select alternate random number generator provided by the operating system (/dev/urandom on Unix or CryptGenRandom on Windows). Subversion (SVN) protocol [16] with its Python binding [17] is used for synchronization of data among trial centers using various plain and encrypted authenticated connection protocols (http, https, svn, svn + ssh, etc.).

3. APPLICATION

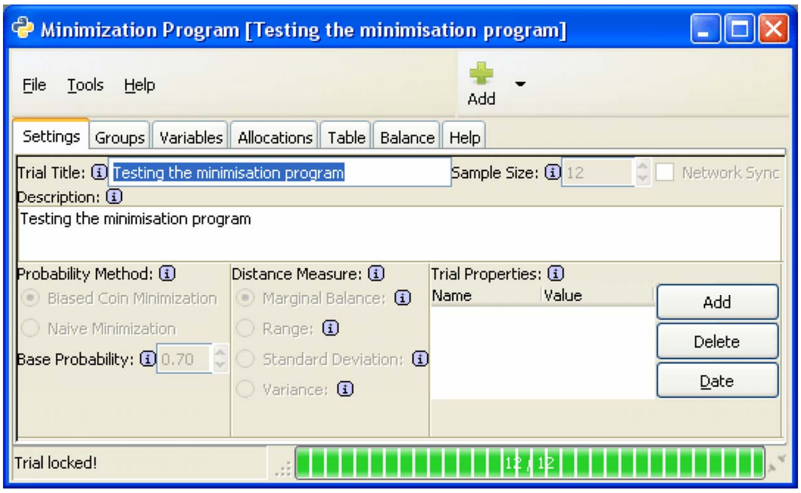
After launching MinimPy the main program window will appear (**Figure 1**). This window is composed of the following tabs:

- Settings: Different aspects of trial and minimization model are configured here.
- Groups: An interface to define each group. For each group a label and an allocation ratio is specified.
- Variables: Different prognostic factors and their levels are specified here.
- Allocations: A menu driven interface for selecting levels of different prognostic factors, to minimize the subject into trial. A table shows all currently allocated of the trial (**Figure 2**). Export and import functionalities are provided using plain text files with optional subjects. A unique random numerical identifier is assigned to each allocated subject to facilitate blinding export of SPSS syntax file to generate the data in SPSS program.
- Table: A frequency table displaying counts, at each level of the prognostic factors for each group (**Figure 3**).
- Balance: A table showing measures of group, level or total balance for the current state of the trial (**Figure 4**).
- There are buttons for adding and deleting groups and variables, and their settings can be modified directly in the displayed table. Frequency table of allocations also can be modified and saved as a pre-load of minimization at the time of initialization. This is particularly useful when importing allocations between different minimization systems.

The program flows in two phases.

3.1. Setup Phase

In this phase different trial settings can be set. These include trial title and description, sample size, method of probability assignment, imbalance score, groups, variables and extra trial settings. This phase terminates when the trial is saved or a previously saved one is loaded, which leads to the trial lock and start of the allocation phase. No further changes in trial settings is possible hereafter. Depending on the nature of the trial, the program may be used in either pure desktop mode or as a desktop application with network synchronization of data among trial centers. Registration with an SVN compatible central repository is necessary for using network


Minimization Program [Testing the minimisation program]

Trial Title: Testing the minimisation program

Sample Size: 12

□ Network Sync

Description:

Testing the minimisation program

Probability Method:

● Biased Coin Minimization

○ Naive Minimization

Base Probability: 0.70

Distance Measure:

● Marginal Balance:

○ Range:

○ Standard Deviation:

○ Variance:

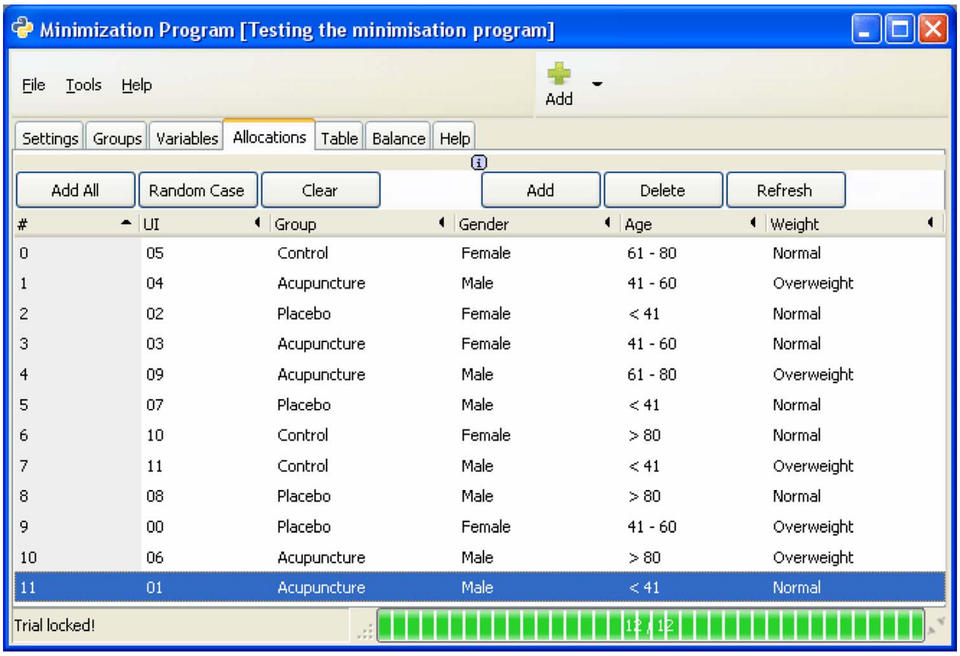
Trial Properties:

Name	Value

Trial locked!

12 / 12

Figure 1. Main window of minimization program showing Settings tab.



#	UI	Group	Gender	Age	Weight
0	05	Control	Female	61 - 80	Normal
1	04	Acupuncture	Male	41 - 60	Overweight
2	02	Placebo	Female	< 41	Normal
3	03	Acupuncture	Female	41 - 60	Normal
4	09	Acupuncture	Male	61 - 80	Overweight
5	07	Placebo	Male	< 41	Normal
6	10	Control	Female	> 80	Normal
7	11	Control	Male	< 41	Overweight
8	08	Placebo	Male	> 80	Normal
9	00	Placebo	Female	41 - 60	Overweight
10	06	Acupuncture	Male	> 80	Overweight
11	01	Acupuncture	Male	< 41	Normal

Trial locked!

12 / 12

Figure 2. Allocations tab showing subjects allocated.

synchronization mode. There are many free and payperservice SVN repositories available on the Internet.

3.2. Allocation Phase

In this phase subjects are allocated to treatment groups using the selected minimization model. No aspects of trial settings can be changed in this phase except for the title, the description and the extra properties. Last allocated patient can be deallocated if any error happened in setting factor levels. If no patient is allocated, trial can be unlocked and returned to setup phase. To facilitate close examination of the minimization results produced by MinimPy, an optional research mode can be enabled to provided mass production and minimization of simulated cases.

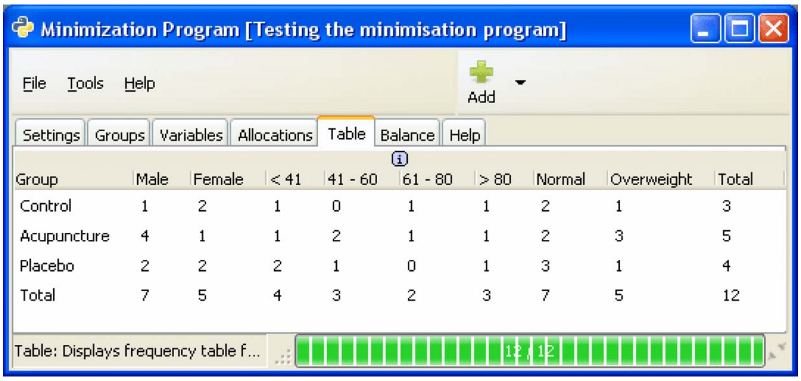
4. DISCUSSION

Minimization methods for allocation of subjects to treatments in a trial involves intense calculations which are hardly carried out without the use of computer programs. Non-predictability features of novel minimization algorithms are comparable to those seen in randomization methods with the extra advantage of balancing different prognostic factors across treatment groups. Sophisticated calculation may be used to account for conditions of unequal groups. Classic minimization methods only account for allocation ratios when calculating imbalance score. The minimization program presented in this article has the option of the biased coin minimizetion as described by Han and co-workers [13]. Computational complexities of advanced minimization algorithms necessitates close inspection and optimization of program code by developers with statistical background. Open-source development provides an environment which enables developers to examine the code and contribute to the development of an ideal and efficient minimization program. Using Python programming language which has an easy to read syntax through formated code blocks, further enhances the transparency of program logic. MinimPy has straight forward interface functions which make it easy for an ordinary user to use. All calculations were performed using python programming languages which is very strong for statistical and mathematical application. It is an interpreted language and easy to learn by non-technicals as well. Since the python can be used for both desktop and web applications, MinimPy code can easily be modified to a pure web service application suitable for multi-center trials. However MinimPy provides network synchronization of data as needed in a typical multi-center clinical trial. Although MinimPy feathers network synchronization over free or proprietary repositories, the task is not easy enough to be undertaken by non-technical users. Therefore usage of MinimPy for multi-center trials necessitates the availability of technical supports for setting a public/private repository to be used by the users of the program in different centers.

5. AVAILABILITY AND REQUIREMENT

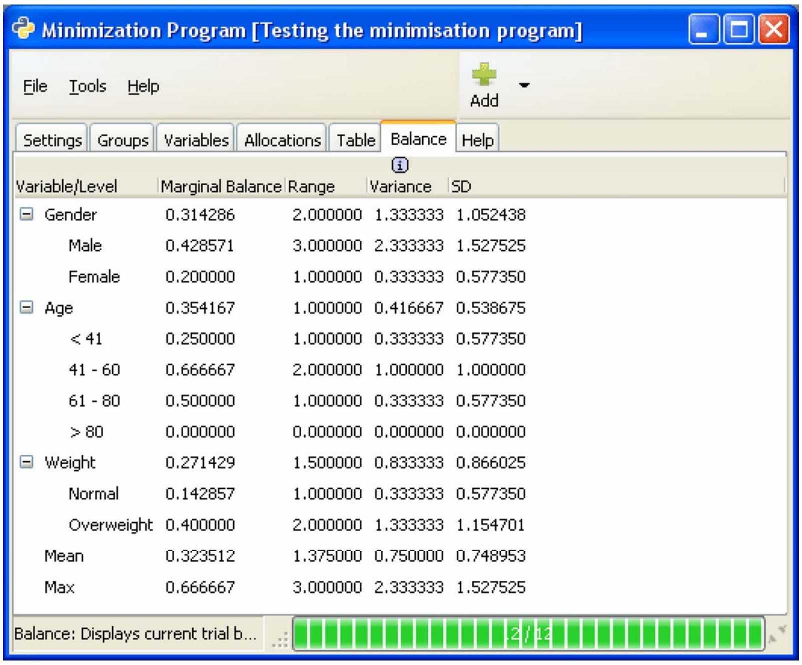
5.1. Public Access

MinimPy is distributed under the GNU GPL v3, full


Minimization Program [Testing the minimisation program]
File
Tools
Help
Add
Settings
Groups
Variables
Allocations
Table
Balance
Help
Table: Displays frequency table f...
12 / 12

Group	Male	Female	< 41	41 - 60	61 - 80	> 80	Normal	Overweight	Total
Control	1	2	1	0	1	1	2	1	3
Acupuncture	4	1	1	2	1	1	2	3	5
Placebo	2	2	2	1	0	1	3	1	4
Total	7	5	4	3	2	3	7	5	12

Figure 3. Table tab showing counts of factors levels across different groups.


Minimization Program [Testing the minimisation program]
File
Tools
Help
Add
Settings
Groups
Variables
Allocations
Table
Balance
Help
Balance: Displays current trial b...
12 / 12

Variable/Level	Marginal Balance	Range	Variance	SD
⊟ Gender	0.314286	2.000000	1.333333	1.052438
Male	0.428571	3.000000	2.333333	1.527525
Female	0.200000	1.000000	0.333333	0.577350
⊟ Age	0.354167	1.000000	0.416667	0.538675
< 41	0.250000	1.000000	0.333333	0.577350
41 - 60	0.666667	2.000000	1.000000	1.000000
61 - 80	0.500000	1.000000	0.333333	0.577350
> 80	0.000000	0.000000	0.000000	0.000000
⊟ Weight	0.271429	1.500000	0.833333	0.866025
Normal	0.142857	1.000000	0.333333	0.577350
Overweight	0.400000	2.000000	1.333333	1.154701
Mean	0.323512	1.375000	0.750000	0.748953
Max	0.666667	3.000000	2.333333	1.527525

Figure 4. Balance tab showing current trial balance as four different measures for each variable, level and for the whole trial.

terms at: http://www.gnu.org/copyleft/gpl.html Latest stable version can be downloaded from sourceforge at: http://minimpy.sourceforge.net The latest developmental version can be checked out from the code repository as: svn export https://minimpy.svn.sourceforge.net/svnroot/minimpy minimpy or downloaded as: http://minimpy.svn.sourceforge.net/viewvc/minimpy/?view=tar

5.2. Programing Language

Python 2.6 or 2.7 (http://python.org/download/) Higher versions are not compatible with PyGTK. PyGTK 2.x. The version must be matched against the installed python version.

5.3. Other Requirements

Windows users need to install GTK+ libraries and GTK binding for python.as listed below:

- GTK+: http://www.gtk.org/index.php
- PyGTK 2.x: http://www.pygtk.org/
- PyCairo: http://ftp.gnome.org/pub/GNOME/binaries/win32/pycairo/
- PyGObject: http://ftp.gnome.org/pub/GNOME/binaries/win32/pygobject/
- Subversion http://subversion.tigris.org/
- PySVN, Python binding for subversion http://pysvn.tigris.org/

To facilitate the installation of these requirements for Windows users it is recommended to install python 2.7 (http://python.org/ftp/python/2.7.1/python-2.7.1.msi) first, and then the so called all-in-one package for PyGTK and the related libraries. The latter can be downloaded from: http://ftp.gnome.org/pub/GNOME/binaries/win32/pygtk/

2.22/pygtk-all-in-one-2.22.5.win32-py2.7.msi

Finally download and install SVN and PySVN which is needed for network synchronization of data. Under Windows and MAC OS X, installation of PySVN will install SVN too, so users of these operating systems do not need separate SVN package installation.

Generally Linux users do not need these libraries and bindings, because they are already included in most Linux distributions. For other platforms please consult the related documentation regarding downloading, installation and running of python and GTK application.

After installation of these requirements (Windows) simply download MinimPy as a compressed file and extract it in you hard drive. Under windows you can run the application by double clicking the "minimpy.pyw" file in the extracted folder. You can make a desktop shortcut for this file for convenience.

REFERENCES

[1] Schulz, K.F., Altman, D.G. and Moher, D. (2010) CONSORT 2010 Statement: Updated guidelines for reporting parallel group randomized trials. *Annals of Internal Medicine*, **152**, 1-8.

[2] Directory of randomization software and service. (2011). http://www-users.york.ac.uk/~mb55/guide/randserv.htm.

[3] Lachin, J.M. (1988) Statistical properties of randomization in clinical trials. *Controlled Clinical Trials*, **9**, 289-311. doi:10.1016/0197-2456(88)90045-1

[4] Kernan, W.N., Viscoli, C.M., Makuch, R.W., Brass, L.M. and Horwitz, R.I. (1999) Stratified randomization for clinical trials. *Journal of Clinical Epidemiology*, **52**, 19-26. doi:10.1016/S0895-4356(98)00138-3

[5] Taves, D.R. (1974) Minimization: A new method of assigning patients to treatment and control groups. *Clinical Pharmacology & Therapeutics*, **15**, 443-453.

[6] Pocock, S.J. and Simon, R. (1975) Sequential treatment assignment with balancing for prognostic factors in the controlled clinical trial. *Biometrics*, **31**, 103-115. doi:10.2307/2529712

[7] Scott, N.W., McPherson, G.C., Ramsay, C.R., Campbell MK. (2002) The method of minimization for allocation to clinical trials: A review. *Controlled Clinical Trials*, **23**, 662-674. doi:10.1016/S0197-2456(02)00242-8

[8] McEntegart, D.J. (2003) The pursuit of balance using stratified and dynamic randomization techniques: An overview. *Drug Information Journal*, **37**, 293-308.

[9] Halpern, J. and Brown, B.W.Jr. (1986) Sequential treatment allocation procedures in clinical trials with particular attention to the analysis of results for the biased coin design. *Statistics in Medicine*, **5**, 211-229. doi:10.1002/sim.4780050303

[10] Lachin, J.M., Matts, J.P. and Wei, L.J. (1988) Randomization in clinical trials: Conclusions and recommendations. *Controlled Clinical Trials*, **9**, 365-374. doi:10.1016/0197-2456(88)90049-9

[11] Minim: Allocation by minimisation in clinical trials, 2004. http://www-users.york.ac.uk/~mb55/guide/minim.htm.

[12] Cai, H.W., Xia, J.L., Gao, D.H. and Cao, X.M. (2010) Implementation and experience of a web-based allocation system with Pocock and Simon's minimization methods. *Contemporary Clinical Trials*, Vol. 31, No. 6, 2010, pp. 510-513. doi:10.1016/j.cct.2010.07.009

[13] Han, B., Enas, N.H. and McEntegart, D.J. (2009) Randomization by minimization for unbalanced treatment allocation. *Statistics in Medicine*, **28**, 3329-3346. doi:10.1002/sim.3710

[14] Python Programming Language Official Website. (2011). http://www.python.org/.

[15] Matsumoto, M. and Nishimura, T. (1998) Mersenne twister: A 623-dimensionally equidistributed uniform pseudo- random number generator. *ACM Transactions on Modeling and Computer Simulation*, **8**, 3-30. doi:10.1145/272991.272995

[16] Subversion.tigris.org. (2011). http://subversion.tigris.org/

[17] Pysvn.tigris.org. (2011). http://pysvn.tigris.org/

Abbreviations and Acronyms

P_H = preferred treatment;
P_L = non-preferred treatment;
P_{Hb} = base P_H used for the group with the lowest allocation ratio as the preferred treatment;
P_{Hi} = P_H for groups with higher allocation ratios as the preferred treatment;
$r_1, r_2, \cdots, r_n$ = Allocation ratios for groups 1 to n;
b_M = Marginal balance;
a_i = adjusted number of patients present in a factor level for each treatment group.