randomizeR: An R Package for the Assessment and Implementation of Randomization in Clinical Trials

Diane Uschner
RWTH Aachen University

David Schindler
RWTH Aachen University

Nicole Heussen
RWTH Aachen University

Ralf-Dieter Hilgers
RWTH Aachen University

Abstract

この「初めての R package **randomizeR**」は，Uschner, Schindler, Heussen, and Hilgers (2016) の修正版である．

臨床試験におけるランダム化は，治療群の比較可能性を確保するための重要な設計手法である．ランダム化の実施を支援するソフトウェアは数多く存在するが，幅広い手順をカバーし，現実的な制約の下で手順の比較評価を可能にするツールは，先行研究では提案されていない．

R package **randomizeR** は，この必要性に対応する．

この文書は，ランダム化割り付け列の生成とランダム化手順の評価のチュートリアルとなる **randomizeR** パッケージの詳細な説明を行う．

Keywords: restricted randomization procedure, selection bias, chronological bias, R.

1. はじめに

ランダム化とは，臨床試験において治療群の比較可能性を確保するために，意図的に偶然の要素を導入する設計手法のことである．Armitage (1982) は，ランダム化によって達成されるべき 3 つの主な目的が述べた．第一に，ランダム化は既知の共変量と未知の共変量のバランスをとり，治療群の構造的な平等性を生み出す傾向がある．第二に，治験責任医師と患者の治療配分の盲検化を効果的に行うことで，患者の選択に起因するバイアスを回避することができる．最後に，ランダム化は統計的推論の基礎となる臨床試験の内部妥当性に寄与する．臨床試験におけるランダム化の重要性は，1940 年代に，ランダム化なしに治療法の盲検化を成功させることは不可能であることに気づいた Sir A. Bradford Hil (see Chalmers 1999) によって初めて指摘された．それ以来，ガイドラインにランダム化を用いるように提唱され (see for example ICH E9 1998)，先行研究ではいくつかの異なるランダム化手順が提案されている．ランダム化手順が異なると，例えば，バイアスの影響を受けやすいことや，検出力や type I エラーの確率を制御できる可能性など，挙動が異なることがわかってきた．最新の開発状況を含む概要は，Rosenberger and Lachin (2016) に掲載されている．

臨床試験において，ランダム化は重要であるにもかかわらず，過去数十年の間に開発された新技術は，ほとんど臨床現場に導入されていない．Berger, Bejleri, and Agnor (2015) によると，臨床家はランダム化手順の特性の悪さを広く議論し，より良い方法が提案されているにもかかわらず，簡単に使うことのできるランダム化手順を使い続けている．注目すべきは，最も適した

ランダム化手順は臨床的背景によって異なるということである．例えば，ブロックランダム化を使用する場合，Tamm and Hilgers (2014) は経時バイアスを制御するためには小さなブロックサイズを提唱し，一方，Kennes, Hilgers, and Heussen (2012) は選択バイアスを制御するためには大きなブロックが良いことを示している．

適切なランダム化手順の選択は，特に漸近的な仮定が成り立たないサンプルサイズが小さい場合に，検出力やタイプ I エラー率の制御などの好ましい特性を達成するために非常に重要である．R パッケージの **randomizeR** は，臨床試験担当者がこれらの基準に基づいて科学的に正しい選択をすることを支援する．このパッケージは，ランダム化手順の評価と，臨床試験用のランダム化リストの生成を組み合わせたものである．Snow (2013) の **blockrand** パッケージのような既存のソフトウェアツールでは，非常に限られた数のランダム化手順しか実装されていない．応答適応型ランダム化手続きという特殊なクラスは **randomizeR** には含まれていないが，time-to-event outcome を伴う臨床試験用に，現在，Ryeznik, Sverdlov, and Wong (2015) の MTLAB パッケージ **RARtool** で実装されている．我々の知る限りでは，ランダム化手順の実用的な特性を評価するソフトウェア・パッケージは存在せず，特に小規模な臨床試験における正確な特性を評価することはできない．

この記事では，**randomizeR** の詳細な説明を行う．構成は以下の通り．セクション 2 では，バイアスのかかりやすさなど，ランダム化手順の特性を評価するための背景を説明する．セクション 2.3 では，ランダム化手順を示す．セクション 3では，**randomizeR** が背景の章の手法をどのように実装しているかを示す．特に，セクション 3.2.2 では，**randomizeR** によるランダム化リストの生成についてのチュートリアルを，セクション 3.3.1 では，異なるタイプのバイアスやその他の基準に関してランダム化手順を評価する方法を示す．

2. ランダム化手順の特性を評価

2.1. Notation and model

ここでは，並行群設計の 2 つのアームの臨床試験の場合を考える．A と B を連続的な結果 Y に影響を与える治療法とする．*randomization procedure* $\mathcal{M}$ を，空間 $\Omega = \{0,1\}^N$ 上の確率分布とする．*randomization sequence* を， $t \in \Omega$ の要素とする．ここで，患者 i が治療法 A に割り当てられた場合は $t_i = 1$，そうでない場合は $t_i = 0$ とする．$T = (T_1, \ldots, T_N)$ は確率分布 $\mathcal{M}$ で Ω の値をとる確率変数を表すとする．本論文では，患者 $i = 1, \ldots, N$ の治療結果 y_i (訳注：式中では大文字の Y_i) を，群予測 μ_A, μ_B と，等しいが未知の分散 $\sigma^2 > 0$ を用いて，正規分布確率変数を以下に示す．

$$Y_i \sim \mathcal{N}(\mu_A \cdot T_i + \mu_B \cdot (1 - T_i), \sigma^2) \tag{1}$$

このときの結果 y_i を， 応答 と呼ぶ．応答の値が大きいほど良いとみなされる．実験的治療法 A の期待値は，対照的治療法 B の期待値と差がないという帰無仮説を，両側代替案に対して検定する．

$$H_0 : \mu_A = \mu_B \quad \text{vs.} \quad H_1 : \mu_A \neq \mu_B. \tag{2}$$

患者 i の応答 Y_i が観測されていない量 b_i に影響されるとき，b_i を i 番目の患者のバイアスと呼ぶ．バイアスには，選択バイアス，経時バイアス，あるいはその両方が含まれるが，これら

Trend	Shape
Linear	$\tau(i,\vartheta) = i \cdot \vartheta$
Logarithmic	$\tau(i,\vartheta) = \log(\frac{i}{N}) \cdot \vartheta$
Step	$\tau(i,\vartheta) = 1_{\{i \geq n_0\}} \cdot \vartheta, n_0 \in \{1,\ldots,N\}$

Table 1: **randomizeR** に含まれている時間トレンドの 3 類型

は 2.2.1 および 2.2.2 で提案されている．ランダム化手順がモデルの誤指定をどのように管理するかを調べる．患者 i の固定バイアス b_i を用いて，

$$Y_i \sim \mathcal{N}(\mu_A \cdot T_i + \mu_B \cdot (1 - T_i) + b_i, \sigma^2) \tag{3}$$

2.2. 評価基準

本節では，ランダム化手順の選択に関する他の例示的な基準とともに，試験結果に対するバイアスの影響を抑制するランダム化手順の可能性についてまとめた．バイアスの影響の評価は，バイアスの形態が不明であったり，バイアスが観察されていない場合に重要である．

継時バイアスへの感受性

試験環境の変化（例えば，診断の可能性が高まるなど）は，時間の経過とともに治療に対する反応に影響を与える可能性がある．未観測の経時時的傾向は治療効果の推定値の偏りにつながり，この偏りを Matts and McHugh (1978) は 経時バイアス *(chronological bias)* という用語で表現している．経時バイアスは事故的バイアスの特殊なケースであり，Efron (1971) によって紹介された．Efron は，モデル内で（意図せず）無視された共変量の影響を調査した．基本的な傾向は理論的にはモデルに含まれているが，バイアスは観察されていないことも多く，正確な形が不明である．さらに，特に小さな集団では，多くの説明変数を持つモデルを使用することは困難である．Rosenkranz (2011) は，トレンド $\tau(i,\vartheta)$ の影響を受けている場合の t test の Type I エラー率の歪みにより，経時バイアスの影響を測定した．:

$$Y_i \sim \mathcal{N}(\mu_A \cdot T_i + \mu_B \cdot (1 - T_i) + \tau(i,\vartheta), \sigma^2). \tag{4}$$

Tamm and Hilgers (2014) は，表 1 にまとめられた 3 つのトレンドの形を提案した．ここで，log は自然対数を表し，$1_{\{i \geq n_0\}}$ は $i \geq n_0$ のとき 1，$i < n_0$ のとき 0 となる指示関数である．

選択バイアスへの感受性

Sir A. Bradford は，臨床試験にランダム化を採用した最初の人物である (see Chalmers 1999)．彼の目的は，効果的な盲検化を確保し，意識的または無意識的に患者を治療群に選択することによるバイアス，いわゆる *selection bias* を回避することであった．ここでは，文献で提案されている選択バイアスの 2 つの尺度，すなわち，期待される正解数と選択バイアスが試験の判定に及ぼす影響を検討する．

期待される正解数は Blackwell and Hodges (1957) によって導入された．彼らは，過去の治療の割り当てに基づいて，治験責任者が意識的または無意識的に次の治療を推測すると仮定した．患者 $i > 1$ が治療 $g(t,i) \in \{A, B\}$ を受けることを，治験責任者が過去の割り当て $(t_1,\ldots,t_{i-1})$

Imbalance	Formula
Final	D_N
Absolute final	$\lvert D_N \rvert$
Maximal	$\max_{i=1,\ldots,N} \lvert D_i \rvert$
Loss	$\frac{D_N^2}{N}$

Table 2: **randomizeR** で実装されている不均衡な手法

に基づいて推測したとする．ランダム化割り付け列の正解率は，治験責任医師が正しく推測した割り当ての数である．:

$$CG(t) = |\{i \in \{1, \ldots, N\} : g(t, i) = t_i\}|. \tag{5}$$

Blackwell and Hodges (1957) によって 2 つの推測戦略が検討された．*convergence strategy (CS)* では，実験者は次の患者がこれまでに割り当てられた回数が少ないグループに割り当てられると仮定する．また，*divergence strategy (DS)* では，実験者は次の患者がこれまでに観察された回数が多い治療法に割り当てられると仮定する．試験開始時と同点の場合，治験責任者は次の割り当てをランダムに推測する．

予想される正解数 $E(CG(t))$ は，割り当て列 $t \in \Omega$ の予測可能性を表している．ランダム化手順の全体的な予測可能性は，正解率の平均値で示される．

$$Y_i \sim \mathcal{N}(\mu_A \cdot T_i + \mu_B \cdot (1 - T_i) - \text{sign}(D_i) \cdot \eta, \sigma^2), \tag{6}$$

ここで，$D_i := D_i(T) = \sum_{j=1}^{i} T_j - \sum_{j=1}^{i}(1 - T_j)$ は，ランダム化割り付け列の 不均衡 *(imbalance)*，$\text{sign}(x)$ は符号関数, また η は選択効果を表す．Tamm, Cramer, Kennes, and Heussen (2012) は，選択効果 η の値を変えて，選択バイアスの影響を実証した．

行動のバランス化

ICH E9 (1998) によると，ランダム化手順では，予測可能性を回避しつつ，試験全体および試験終了時にグループサイズのバランスをとることが望ましいとされている．表 2 には，先行研究で提案されている不均衡の対策がまとめられている (例 Atkinson 2014)．Lachin (1988) によると，連続的なエンドポイントと相補性の場合，不均衡は統計的検定の検出力を低下させる可能性があるという．

2.3. ランダム化手順

ランダム化手順は，制限付きまたは非制限付きのランダムウォークの観点から説明することができる (see Proschan 1994)．ランダムウォークに課せられた制限は，異なるランダム化手順につながる．このセクションでは，**randomizeR** で実装されているランダム化手順について簡単に説明する．包括的な概要については，Rosenberger and Lachin (2016) を参照．

完全ランダム化 (Complete Randomization, CR) は，各患者の割り当てのために公正なコインを投げることと同等である．CR は 2^N 等確率割り付け列（N は全サンプルサイズを表す）を導く．図 1 は，CR によって生成されたランダム化割り付け列を太い黒色で示し，すべての可能な割り付け列を薄い灰色で示す．

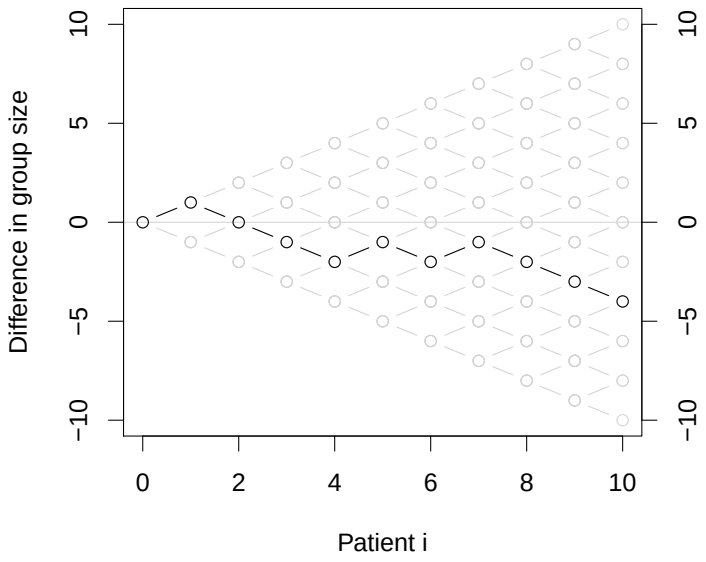


Figure 1: CR ランダム化割り付け列のランダムウォーク

RAR(Random Allocation Rule) では，各治療法ごとに $N/2$ 個のボールからなる壷から N 回，入れ替えなしに抽選して患者を割り付ける．RAR では，最終的にバランスをとることができる $\binom{N}{N/2}$ 個の等確率の列が得られる．

割り当てブロックランダム化 (PBR) は，*block constellation* $bc = (k_1, dots, k_m)$ を用いて，長さ $k_1, \ldots, k_m$ のブロック内の割り当てを均衡化するものである．各ブロック $j = 1, \ldots, m$ に対して，2 つの処理のそれぞれについて $k_j/2$ 個のボールで甕を埋め，その 甕から k_j 個のボールを入れ替えなしで引く．PBR は $\prod_{j=1}^{m} \binom{k_j}{k_j/2}$ 等確率列を導き，各ブロックの後，特に最後にバランスをとる．

random block constellation による Permuted Block Randomization (RPBR) は PBR に似ているが，ブロックコンスタレーション bc は与えられたブロック長 rb のセットからランダムにサンプリングされる．RPBR は 2 つのバリエーション (Heussen 2004; Rosenberger and Lachin 2016) を許容する．: bc のエントリ k_j は，rb から $\sum k_j \geq N$ になるまで一様に入れ替えて抽出することも，最終的なバランスをとるために $\sum k_j = N$ になるように条件付けすることもできる．

Truncated Binomial Design (TBD) は，患者の割り当てのために公正なコインを $N/2$ 個のヘッドまたはテールが発生するまで投げる．残りの患者は決定論的に反対側のグループに割り当てられる．TBD は，最終的なバランスをとる列をもたらすが，等確率な列ではない．TBD は，RAR と同じ数の列を認める．拡張として，TBD は PBR に類似したブロックで実施することも，RPBR に似たランダムブロックコンステレーション（RTBD）で実施することもできる．

Maximal Procedure (MP) は，Berger, Ivanova, and Knoll (2003) によって提案された．MP は，最終的なバランスを達成し，あらかじめ指定された最大許容不均衡 $mti \in \mathbb{N}$ を超えないすべての列に等確率を与える．不均衡の境界に達したとき，つまり $|D_i| = mti$ に達したときには，不均衡を減らすために，過小に割り当てられた群に決定論的に割り当てが行われる．

Soares and Wu (1983) によって導入された Big Stick Design (BSD) は，不均衡 $|D_i|$ が最大許容不均衡 $mti \in \mathbb{N}$ に達するまで公正なコインを投げることで構成されている．

Efron's Biased Coin design (EBC) (see Efron 1971) では，不均衡を減らすために，次の割り当てのために偏ったコインを投げる．確率 $0.5 \leq p \leq 1$ で不均衡を減らすために，次の割り当てのために偏ったコインを投げる．2 群のバランスが取れたときは，公正なコインが投げられる．

Chen (1999) は，BSD と EBC を組み合わせた，不均衡耐性のあるバイアスコインデザイン（CHEN）を提案した．グループのバランスが取れているときは，患者の割り当てのために公正なコインを投げる．それ以外の場合は，確率 $0.5 \leq p \leq 1$ の偏ったコインを，不均衡境界 $mti \in \mathbb{N}$ に達するまで投げる．不均衡境界に到達すると，不均衡を減らすために決定論的割り当てが行われる．

Antognini and Giovagnoli (2004) によって提案された Accelerated Biased Coin Design (ABCD) では，患者の割り当てに偏ったコインのトスを使用する．i 番目の患者を実験群に割り付ける確率 $p_i = p(D_{i-1}, a)$ は，不均衡 D_i に依存する．は，不均衡 D_{i-1} と 加速パラメータ $a > 0$ に依存する．加速パラメータは，不均衡を指数関数的に重み付けし，デザインのランダム性の度合いを決定する．

Antognini and Zagoraiou (2014) が「dominant biased coin design」という名前で提案した BBCD（Bayesian Biased Coin Design）は，ABCD に類似している．ここで，i 番目の患者を実験群に割り付ける確率 p_i は，加速パラメータ $a > 0$ と比 $N_A(i-1)/N_B(i-1)$ に依存する．$N_A(i-1)$ と $N_B(i-1)$ は，$i-1$ 人の患者を割り当てた後，それぞれ A 群と B 群に属する患者の数である．

Wei の Urn Design (UD, 訳注: urn は抽選用ボールを入れる「壺」の意) は，抽選のたびに組成が更新される壺を N 回抽選することで構成されている．最初の患者を割り付ける前に，壺の中には，各治療法ごとに異なる色のボールが初期数（$ini \geq 0$）個だけ入っている．患者を割り付ける際には，ボールを 1 つ引き，その色を記録し，反対の色のボールを（$add \geq 0$）個追加して入れ替える．

GBCD（Generalized Biased Coin Design）は，様々なデザインを拡張するために Smith (1984) によって開発された．偏ったコインを投げて患者を割り振り，次の患者が実験群に割り振られる確率は，$N_A(i-1)$ と $N_B(i-1)$ に加えて，バランスをとるためのパラメータである $\rho \geq 0$ である．

Bailey and Nelson (2003) によって提案された Hadamard Randomization (HADA) は，$H \in \{0,1\}^{11\times 12}$ の特殊な Hadamard 行列 H の行を患者の割り当てに使用する．H からの行は，割り当て数が計画されたサンプル数に達するまでサンプリングされる．

次章の表 3 では，紹介したランダム化手順を **randomizeR** でどのように使用するかを示す．

2.4. 実装

ランダム化手順の評価は，一連の割り当て割り付け列に基づいて行われる．評価基準の中には，例えば Type I-error や power のように応答に依存するものもあれば，例えば正解率のように応答に依存しないものもある．サンプルサイズ N に応じて割当順序のセットを生成するための 2 つの異なるオプションと，応答に基づく評価基準を計算するための 2 つの異なる方法がある．

- **Complete or simulated reference set:** $N \leq 24$ の場合，可能性のあるすべての割り当て列の集合 $\{0,1\}^N$ を生成し，割り当て列の適格性を評価し，独立して，ランダム化アルゴリズムによる割り当て列の関連確率を計算することが可能である．この結果，列の *complete set* が得られる．関数 `getAllSeq` はこの機能を提供する．セクション 3.2.3 を参照．

$N > 24$ の場合，完全なセットを合理的な時間で計算することはできない．その代わりに，特定のランダム化手順のランダム化アルゴリズムを用いて，割り当て列の数 r が生成される．関数 `genSeq` がこの機能を提供する．セクション 3.2.3 を参照．形式的には，割り当て列の相対的な頻度を用いて，割り当て列の真の確率を推定することができる．したがって，シミュレーションでは，ランダム化割り付け列の相対的な頻度を使用することに注意．この結果，列の *simulated set* が得られる．シミュレーションセットは，サンプルサイズが小さい場合や，正確なアプローチが文献に載っていない手順，すなわちランダムブロックコンステレーションを用いたブロックランダム化や Hadamard ランダム化などにも適用できる．

- **Exact or simulated response based assessment criteria:** ランダム化手順のバイアスに対する感受性は，type I または type II エラー確率の歪みとして測定できる．*exact method* は，正確な棄却確率の分布を計算する．参照セットの各ランダム化割り付け列について，Model 3 (see Langer 2014, Chapter 4) の b_i に関する知識を用いて，Student's t test の棄却確率を計算する．*simulation method* では，Model 3 にしたがって，参照集合の各配分列に対する応答ベクトルをシミュレーションし，Student's t test の検定判定を導出する．Type I または Type II エラー 確率は，誤って棄却されたテスト判定の割合として計算される．この方法は，例えば，Proschan (1994) で用いられた．セクション 3.3では，**randomizeR** を用いて正確な type I エラー確率とシミュレーションされた type II エラー確率を評価する方法を示している．

上記の方法を任意に組み合わせて使用することができる．例えば，正確な棄却確率の分布を評価するために，シミュレートされたリファレンスセットを使用することができる．文献では通常，シミュレートされたエラーレートとシミュレートされたリファレンスセットの組み合わせが使用されている．サンプルサイズが小さい場合は，正確なリファレンスセットと正確なエラー確率の分布の組み合わせが最も正確な結果を生む．

すべてのサンプリングアルゴリズムは，R の標準的な乱数生成器である Mersenne-Twister (R Core Team 2016) を使用している．

3. randomizeR パッケージ

3.1. 外観

randomizeR パッケージは，2 つの密接に関連した目的を扱っている．: すなわち，ランダム化割り付け列の生成と，前述の基準に従ったランダム化手順の評価である．前章では，この分野の基本的な用語や文献を紹介した．本章では，**randomizeR** がこれらの目的にどのように対応しているかを示す．

randomizeR の現在のバージョンは，R 3.3.0 をベースにしている．R のセッションでは:

```
R> library("randomizeR")
```

で，パッケージをロードする．

randomizeR の主要コンポーネントは全て，S4 のオブジェクト指向システムを用いて実装されている．

3.2. ランダム割り付けを生成

ランダム化列の生成には，大きく分けて 2 つの目的がある．1 つ目の目的は，臨床試験で患者を割り付けるための単一の割り付け列を生成することである．2 つ目の目的は，ランダム化手順の特性を評価するために複数の列を生成することである．いずれの目的も，ランダム化手順自体を動作の基礎とする関数とみなすことができる．そのため，ランダム化手順を表すオブジェクトを入力とするメソッドとして実装されている．

ランダム化手順を代表

randomizeR では，ランダム化手順を `randPar` クラスのサブクラスとして実装している．たとえば，サンプル・サイズ $N = 10$ の完全ランダム化を表すオブジェクトは次のように生成される．

```
R> N <- 10
R> (params <- crPar(N))


Object of class "crPar"

design = CR
N = 10
groups = A B
```

関数 `crPar` は，いわゆるコンストラクタ関数，すなわち，クラス `crPar` のオブジェクトを生成し，使用に備える関数である．このオブジェクト `param` には，ランダム化手順に関するすべての情報が含まれている．表 3 には，セクション 2.3 で説明されているランダム化手順のコンストラクタ関数がまとめられている．**randomizeR** では，実装されているランダム化手順の概要を `?randPar` で示す．

一つの割付列を生成

関数 `genSeq` を使って，特定の臨床試験のための単一のランダム化割り付け列を生成することができる．この関数は，ランダム化手順を表すオブジェクトを入力とし，この手順を使用してランダムに列を生成する．例えば，以下のコードは，上記のようにサンプルサイズ $N = 10$ の完全ランダム化を用いてランダム化割り付け列を生成する．

```
R> params <- crPar(N)
R> (R <- genSeq(params))


Object of class "rCrSeq"

design = CR
seed = 1617816586
N = 10
groups = A B
```

Randomization procedure	Constructor function	Parameters
Complete Randomization (CR)	`crPar(N)`	`N` sample size
Random Allocation Rule (RAR)	`rarPar(N)`	`N` sample size
Permuted Block Randomization (PBR)	`pbrPar(bc)`	`bc` block constellation
Rand. Permuted Block Randomization (RPBR)	`rpbrPar(N, rb)`	`N` sample size `rb` random block lengths
Truncated Binomial Design (TBD)	`tbdPar(bc)`	`bc` block constellation
Rand. Truncated Binomial Design (RTBD)	`rtbdPar(N, rb)`	`N` sample size `rb` random block lengths
Maximal Procedure (MP)	`mpPar(N, mti)`	`N` sample size `mti` max. tolerated imbalance
Big Stick Design (BSD)	`bsdPar(N, mti)`	`N` sample size `mti` max. tolerated imbalance
Efrons Biased Coin Design (EBC)	`ebcPar(N, p)`	`N` sample size `p` biased coin probability
Chen's Design (CHEN)	`chenPar(N, mti, p)`	`N` sample size `mti` max. tolerated imbalance `p` biased coin probability
Generalized Biased Coin Design (GBCD)	`gbcdPar(N, rho)`	`N` sample size `rho` balance factor
Adjustable Biased Coin Design (ABCD)	`abcdPar(N, a)`	`N` sample size `a` balance factor
Bayesian Biased Coin Design (BBCD)	`bbcdPar(N, a)`	`N` sample size `a` balance factor
Wei's Urn Design (UD)	`udPar(N, ini, add)`	`N` sample size `ini` initial urn composition `add` adjustment in each step
Hadamard Randomization (HADA)	`hadaPar(N)`	`N` sample size

Table 3: Randomization procedures included in **randomizeR**.

```
The sequence M:

1 B B B B B B A A B A
```

結果の再現性を確保し，ランダム化手順の報告を充実させるため，関数 genSeq は，ランダム化割り付け列の生成に使用したすべての情報を，ランダム化割り付け列自体とともにオブジェクト R に保存する．

オブジェクト R に保存されたランダム化割り付け列を取得するためには，関数 getRandList を使用することができる．

```
R> getRandList(R)

     [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] "B"  "B"  "B"  "B"  "B"  "B"  "A"  "A"  "B"  "A"
```

オブジェクト R に格納されているランダム化割り付け列とその他の情報は，saveRand 関数を使用して，.csv ファイルに保存できる．

```
R> saveRand(R, file="myRandList.csv")
```

セクション 2.3 の図 1 は，ランダム化割り付け列 R のランダムウォークを示している．この図は，**randomizeR** に含まれる関数 plotSeq を用いて生成することができる．

```
R> plotSeq(R, plotAllSeq = T)
```

関数 genSeq は，表 3 からすべてのランダム化手順のランダム化割り付け列を生成することができる．関数 genSeq は，各乱数化手順に対応したメソッドを持っている．すべての乱数化手順について，その出力はクラス randSeq を拡張したクラスのオブジェクトである．

割り付け列セットを生成

乱数化手順は，生成する割り付け列のセットに基づいて評価することができる（セクション 2.4 参照）．**randomizeR** では，特定の手順から複数のランダム化列を生成する 2 つの方法を提供している．

小さなサンプルサイズ $N \leq 24$ の場合は，関数 getAllSeq を使ってランダム化割り付け列の完全なセットを生成することができる．この関数は，ランダム化手順を表すオブジェクトを入力として受け取り，その手順のランダム化割り付け列の完全なセットを計算する．例えば，上記のサンプルサイズ $N = 10$ の完全ランダム化の場合，以下のコードをを実行して

```
R> (allSeqs <- getAllSeq(params))


Object of class "crSeq"

design = CR
N = 10
```

```
groups = A B

The first 3 of 1024 sequences of M:

1 A A A A A A A A A A
2 B A A A A A A A A A
3 A B A A A A A A A A
...
```

$2^{10} = 1024$ の割り付け列の完全なセットを得ることができる．なお，関数 `getAllSeq` は，アルゴリズムが確立されていない，ランダムブロック設計の RPBR および RTBD，あるいは Hadamard Randomization をサポートしていない．

列の完全なセットを列挙することが計算上負荷が強かったり，アルゴリズム的に実行不可能であったりする場合には，関数 `genSeq` を用いて，シミュレーションされた参照セットを生成することができる．例えば，サンプルサイズ $N = 50$ で Complete Randomization の場合，次のようにサイズ $r = 10,000$ のシミュレーション対照集合が生成される．

```
R> N <- 50
R> params <- crPar(N)
R> (randomSeqs <- genSeq(params, r = 10000))


Object of class "rCrSeq"

design = CR
seed = 727679323
N = 50
groups = A B

The first 3 of 10000 sequences of M:

1 A   B   A   A   A   B   B   B   A   B   ...
2 A   A   A   A   A   A   A   A   A   A   ...
3 A   A   A   B   A   B   B   B   B   B   ...
...
```

パラメータ `seed` を `genSeq` に渡すことで，結果の再現性を確保することができる．

`genSeq` はランダム化割り付け列を置換しながら，`params` で表されるランダム化手順の真の発生確率でサンプリングするので，結果には重複が含まれることがある．より確率の高い列はより頻繁に発生する．

randomizeR は，列のランダム化の真の発生確率を計算するための関数 `getProb` を提供している．`getProb` は `randSeq` を継承した任意のオブジェクトを受け取る．

```
R> p <- getProb(allSeqs)
R> head(data.frame(Sequences = myPaste(getRandList(allSeqs)),
+                   Probability = round(p, 6)))
```

```
   Sequences Probability
1 AAAAAAAAAA    0.000977
2 BAAAAAAAAA    0.000977
3 ABAAAAAAAA    0.000977
4 BBAAAAAAAA    0.000977
5 AABAAAAAAA    0.000977
6 BABAAAAAAA    0.000977
```

getAllSeq から得られたオブジェクトに適用された場合，結果として得られる確率の合計は常に 1 になる．逆に，genSeq に適用した場合，確率の合計は 1 になりません．これは，通常，サンプリングされた列のセットは，列の完全なセットのサブセットにすぎないからである．したがって，シミュレーションでは，ランダム化割り付け列の真の発生確率ではなく，サンプリングされた頻度が使用されることに注意してください．サンプリングアルゴリズムの性質上，サンプルが十分に大きければ，サンプルされた頻度は真の確率に収束する．

3.3. ランダム化手順を評価

セクション 2.2 の評価基準のほとんどは，(1) に従った正規分布の応答の仮定に依存している．両方の処理が等しい期待値 $\mu_A = \mu_B = 0$ と分散 $\sigma_A = \sigma_B = 1$ であると仮定する．すると，正規エンドポイントを表す関数 normEndp を使って，これらを **randomizeR** に渡すことができる．

```
R> muA <- muB <- 0
R> sigmaA <- sigmaB <- 1
R> normalEndpoint <- normEndp(mu = c(muA, muB), sigma = c(sigmaA, sigmaB))
```

endpoint は，他のエンドポイントへの拡張に柔軟に対応することができる．現在は通常のエンドポイントのみ利用可能である．

randomizeR では，ランダム化手順の評価基準をクラス issue のサブクラスとして実装している．例えば，強度が $\vartheta = 1$ である線形時間傾向による経時偏りがある場合の正確な棄却確率を表すオブジェクトは，

```
R> (cb <- chronBias(type = "linT", theta = 1, method = "exact"))


 Object of class "chronBias"

         TYPE = linT
         THETA = 1
         METHOD = exact
         ALPHA = 0.05
```

パラメータ method は，Type I エラー率の正確な分布を計算するか，あるいはトレンドの影響を受けた回答を生成してテスト判定をシミュレートするかを示す（method = "sim"）．関数 chronBias は，クラス chronBias のオブジェクトのコンストラクタ関数である．オブジェクト cb には，バイアスに関するすべての情報が格納されている．表 4 には，セクション 2.2 で示されたすべての評価基準に対するコンストラクタ関数がまとめられている．**randomizeR** では，実装された評価基準の概要を ?issues で表示する．

Criterion	Constructor Function	Parameters
Chronological Bias	`chronBias(type,theta, method)`	`type` of trend `theta` strength of trend `method` of assessment
Selection Bias	`selBias(type,eta, method)`	`type` of guessing strategy `eta` selection effect `method` of assessment
Correct Guesses	`corGuess(type)`	`type` of guessing strategy
Imbalance	`imbal(type)`	`type` of imbalance
Combined Bias	`combineBias(selBias, chronBias)`	`selBias` object `chronBias` object
Power loss due to imbalance	`setPower(d, method)`	`d` detectable effect `method` of assessment

Table 4: Criteria for the assessment.

ランダム化手順の評価

randomizeR パッケージには，セクション 2.2 の 1 つ以上の基準に関して，ランダム化手順の挙動を評価するための関数 `assess` が含まれている．

例えば，標本サイズ $N = 12$，不均衡許容値 $mti = 2$ の Big Stick Design の挙動を，グループ平均値の差 $d = 1.796$ を与えたときの経時バイアス，選択バイアス，パワーロスに関して評価したい場合，次のように呼び出す．

```
R> N <- 12
R> mti <- 2
R> bsdSeq <- getAllSeq(bsdPar(N, mti))
R>
R> d <- 1.796
R> sb <- selBias("CS", eta = d/4, method = "exact")
R> cb <- chronBias("linT", theta = 1/N, method = "exact")
R> pw <- setPower(d, method = "exact")
```

関数 `assess` は，各基準と各ランダム化列に対する真の棄却確率を求める．

```
R> (A <-assess(bsdSeq, sb, cb, pw, endp = normalEndpoint))


Assessment of a randomization procedure

design = BSD(2)
N = 12
K = 2
groups = A B
```

```
The first 3 rows of 972 rows of D:

       Sequence Probability P(rej)(CS) P(rej)(linT) power(exact)
1 BBABABABA ...       0.004      0.045         0.05        0.789
2 BABBABABA ...       0.002      0.042         0.05        0.789
3 ABBBABABA ...       0.002      0.039         0.05        0.789
...
```

サンプルサイズ $N = 12$，最大許容不均衡 $mti = 2$ の Big Stick Design の場合，972 個の可能な列がある．評価の最初の列はランダム化列に対応し，2 番目の列はその列の発生確率に対応する．次の列は，`assess` に渡された基準 `sb`，`cb`，`pw` に対応している．`P(rej)(type)` という表記は，与えられた `type` の偏りがある場合の棄却確率を意味する．列 `power(exact)` は各ランダム化列の正確なパワーを示す．`assess` には，任意の数の評価基準を渡すことができる．評価基準 `imbal` および `corGuess` については，エンドポイント `endp` は関係ないので省略可能である．

評価の `summary` には，各基準の平均値，標準偏差，最小値，最大値，定量値など，割付割り付け列の分布の重要な特性が示されている．:

```
R> summary(A)

     P(rej)(CS) P(rej)(linT) power(exact)
mean      0.056         0.05        0.795
sd        0.013         0.00        0.006
max       0.109         0.05        0.800
min       0.034         0.05        0.789
x05       0.037         0.05        0.789
x25       0.048         0.05        0.789
x50       0.054         0.05        0.789
x75       0.062         0.05        0.800
x95       0.079         0.05        0.800
```

例えば，線形時間トレンド `linT` がある棄却確率の 5 パーセントの四分位 `x05` は，0.042 である．ただし，今まで，先行研究では各基準の平均値のみが研究されてきたことに注意．

ランダム化手順の比較

randomizeR では，いくつかのランダム化手順をセクション 2.2 の基準の一つに関して比較するための関数 `compare` を提供している．例えば，上記と同じ設定（$N = 12$）で，Big Stick Design，$mti = 2$ の Maximal Procedure，ブロックサイズ 4 の Permuted Block Randomization を，選択バイアスの影響を受けやすいかどうかで比較したいとする．前のコードを一部流用して，比較したい他のランダム化手順のパラメータを設定する．

```
R> mpSeq <- getAllSeq(mpPar(N, mti))
R> bc <- rep(4, N/4)
R> pbrSeq <- getAllSeq(pbrPar(bc))
```

そして，以下のコードは上で述べた，選択バイアスについての手順を比較している．

```
R> (C <- compare(sb, bsdSeq, mpSeq, pbrSeq, endp = normalEndpoint))
```

```
Comparison for P(rej)(CS)

     BSD.2. MP.2. PBR.4.
mean  0.056 0.072  0.082
sd    0.013 0.015  0.015
max   0.109 0.109  0.109
min   0.034 0.040  0.050
x05   0.037 0.050  0.061
x25   0.048 0.061  0.072
x50   0.054 0.072  0.079
x75   0.062 0.079  0.099
x95   0.079 0.100  0.103
```

特定のランダム化手順に対する基準値の分布は，箱ひげ図（図 2b）やバイオリン図（図 2a）によって視覚化され，異なるランダム化手順間で比較することができる．

```
R> plot(C)
R> plot(C, y = "boxplot")
```

箱ひげ図とバイオリン図の生成には，Wickham (2009) によって提案された R パッケージ **ggplot2** の関数 `geom_boxplot` と `geom_violin` を使用した．

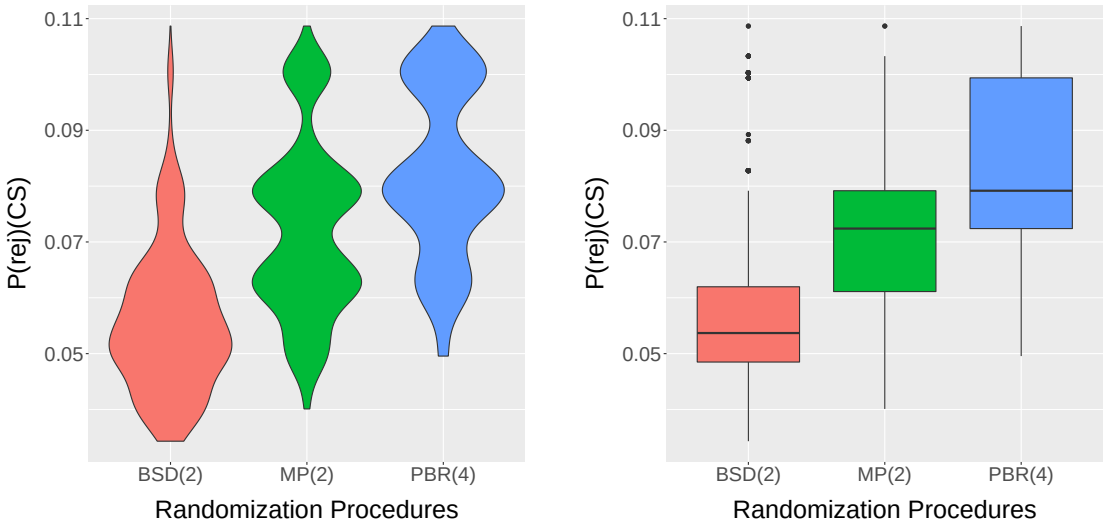
4. 要約と今後の研究

randomizeR は，多数のランダム化手順に対するランダム化リストの生成を容易にし，様々な基準に基づいたランダム化手順の評価を可能にする R パッケージである．このパッケージは現在，15 のランダム化手順と，その手順を評価するための 6 つの基準を実装している．臨床試験のデザイン段階で，ランダム化手順の選択とデザインの実施を両立させることで，研究者を支援する．

我々は **randomizeR** を様々な方向に拡張することを目指している．オブジェクト指向のアプローチにより，新しいランダム化手順や評価基準を簡単に追加することができる．様々な基準に基づいてランダム化手順の適合性を統一的に判断するための，統一的な評価基準を盛り込むこともできる．モデルは，time-to-event データのような他のエンドポイントや，2 つ以上の治療群に拡張することができる．最後に，パラメトリックな仮定に依存しないランダム化ベースの推論を可能にするために，ランダム化テストを実装することができる．

謝辞

本研究は，欧州連合（EU）の第 7 次研究・技術開発・実証計画である IDeAl FP7 プロジェクトの一環として，Grant Agreement no 602552 により資金提供を受けている．



(a) Violin plot (b) Boxplot

Figure 2: Comparative visualization of the distribution of the Type I-error probability under the influence of unobserved selection bias for BSD(2), MP(2) and PBR(4).

翻訳者

馬場 美彦 yoshihiko.baba@uclmail.net

References

Antognini AB, Giovagnoli A (2004). "A new 'biased coin design' for the sequential allocation of two treatments." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, **53**(4), 651–664. ISSN 1467-9876. URL http://dx.doi.org/10.1111/j.1467-9876.2004.00436.x.

Antognini AB, Zagoraiou M (2014). "Balance and randomness in sequential clinical trials: the dominant biased coin design." *Pharmaceutical Statistics*, **13**(2), 119–127. ISSN 1539-1612. URL http://dx.doi.org/10.1002/pst.1607.

Armitage P (1982). "The role of randomization in clinical trials." *Statistics in Medicine*, **1**(4), 345–352. ISSN 1097-0258. URL http://dx.doi.org/10.1002/sim.4780010412.

Atkinson AC (2014). "Selecting a Biased-Coin Design." *Statist. Sci.*, **29**(1), 144–163. URL http://dx.doi.org/10.1214/13-STS449.

Bailey RA, Nelson P (2003). "Hadamard Randomization: a valid restriction of random permuted blocks." *Biometrical Journal*, **45**, 554–560.

Berger VW, Bejleri K, Agnor R (2015). "Comparing MTI randomization procedures to blocked randomization." *Statistics in Medicine*, pp. n/a–n/a. ISSN 1097-0258. URL http://dx.doi.org/10.1002/sim.6637.

Berger VW, Ivanova A, Knoll DM (2003). "Minimizing predictability while retaining balance through the use of less restrictive randomization procedures." *Statistics in Medicine*, **22**(19), 3017–3028. ISSN 1097-0258. URL http://dx.doi.org/10.1002/sim.1538.

Blackwell D, Hodges J (1957). "Design for the control of selection bias." *Annals of Mathematical Statistics*, **25**, 449–460.

Chalmers I (1999). "Why transition from alternation to randomisation in clinical trials was made." *BMJ*, **319**(7221), 1372. ISSN 0959-8138. URL http://www.bmj.com/content/319/7221/1372.

Chen YP (1999). "Biased coin design with imbalance tolerance." *Communications in Statistics. Stochastic Models*, **15**(5), 953–975. URL http://dx.doi.org/10.1080/15326349908807570.

Efron B (1971). "Forcing a sequential experiment to be balanced." *Biometrika*, **58**(3), 403–417. URL http://biomet.oxfordjournals.org/content/58/3/403.abstract.

Heussen N (2004). *Der Einfluss der Randomisierung in Blöcken zufälliger Länge auf die Auswertung klinischer Studien mittels Randomisationstest.* Ph.D. thesis, RWTH Aachen University.

ICH E9 (1998). "Statistical principles for clinical trials." *Current version dated 5 Februrary 1998. Last access in September 2014. Available from:* http://www.ich.org.

Kennes LN, Hilgers RD, Heussen N (2012). "Choice of the Reference Set in a Randomization Test Based on Linear Ranks in the Presence of Missing Values." *Communications in Statistics - Simulation and Computation*, **41**, 1051–1061.

Lachin JM (1988). "Statistical Properties of Randomization in Clinical Trials." *Elsevier Sdence Publishing Co., Inc.*

Langer S (2014). *The modified distribution of the t-test statistic under the influence of selection bias based on random allocation rule.* Master's thesis, RWTH Aachen.

Matts JP, McHugh RB (1978). "Analysis of accrual randomized clinical trials with balanced groups in strata." *Journal of Chronic Diseases*, **31**(12), 725 – 740. ISSN 0021-9681. URL http://www.sciencedirect.com/science/article/pii/0021968178900577.

Proschan M (1994). "Influence of selection bias on type 1 error rate under random permuted block designs." *Statistica Sinica*, **4**, 219–231.

R Core Team (2016). *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/.

Rosenberger W, Lachin J (2016). *Randomization in Clinical Trials: Theory and Practice.* Wiley Series in Probability and Statistics. Wiley. ISBN 9781118742242. URL https://books.google.de/books?id=ZJEvCgAAQBAJ.

Rosenkranz GK (2011). "The Impact of Randomization on the Analysis of Clinical Trials." *Statistics in Medicine*, **30**, 3475–3487.

Ryeznik Y, Sverdlov O, Wong WK (2015). "RARtool: A MATLAB Software Package for Designing Response-Adaptive Randomized Clinical Trials with Time-to-Event Outcomes." *Journal of Statistical Software*, **66**(1), 1–22. ISSN 1548-7660. doi:10.18637/jss.v066.i01. URL https://www.jstatsoft.org/index.php/jss/article/view/v066i01.

Smith RL (1984). "Sequential Treatment Allocation Using Biased Coin Designs." *Journal of the Royal Statistical Society. Series B (Methodological)*, **46**(3), 519–543. URL http://www.jstor.org/stable/2345691.

Snow G (2013). *blockrand: Randomization for block random clinical trials*. R package version 1.3, URL https://CRAN.R-project.org/package=blockrand.

Soares JF, Wu C (1983). "Some Restricted randomization rules in sequential designs." *Communications in Statistics - Theory and Methods*, **12**(17), 2017–2034. http://dx.doi.org/10.1080/03610928308828586, URL http://dx.doi.org/10.1080/03610928308828586.

Tamm M, Cramer E, Kennes LN, Heussen N (2012). "Influence of selection bias on the test decision - a simulation study." *Methods of Information in Medicine*, **51**, 138–143.

Tamm M, Hilgers RD (2014). "Chronolgical bias in randomized clinical trials under different types of unobserved time trends." *Meth. Inf. Med.*, **53**(6), 501–510.

Uschner D, Schindler D, Heussen N, Hilgers RD (2016). "randomizeR: An R Package for the Assessment and Implementation of Randomization in Clinical Trials." *submitted to the Journal of Statistical Software*.

Wickham H (2009). *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York. ISBN 978-0-387-98140-6. URL http://ggplot2.org.

Affiliation:

Diane Uschner
Department of Medical Statistics
RWTH Aachen University
Pauwelsstraße 30
52068 Aachen, Germany
E-mail: duschner@ukaachen.de
URL: http://www.ideal.rwth-aachen.de/