


centrifugeR: Non-Trivial Balance of Centrifuge Rotors

Duy Nghia Pham 

Plentzia Marine Station, University of the Basque Country, Spain

nghiapham@yandex.com

28 January 2020*

Abstract

Loading tubes in opposite buckets has been used universally, yet intuitively to balance centrifuge rotors. Most rotors support tube distributions with rotational symmetry of order not only 2 but also other prime divisors of the total bucket number. This potential allows rotors to be balanced by the non-trivial placement of tubes, which offers users greater flexibility and more safety in centrifuge operation. Based on linear combinations and random samples, **centrifugeR** finds the numbers of tubes that can be loaded in centrifuge rotors in a single operation and shows how to balance these tubes in cases of equal or unequal masses.

Introduction

Centrifuges are essential laboratory devices for particle fractionation. The improper use of centrifuges poses many associated hazards (Clark 2001). To reduce the risks, it is crucial to balance centrifuge rotors before the operation. A rotor is balanced when the centers of gravity of the rotor and the tube collection coincide. For rotors that employ single-tube buckets, common practices are distributing tubes in opposite buckets, adding a blank tube to counterweight if the odd numbers of tubes need to be centrifuged, placing tubes at 120 degrees to each other if the number of tubes is divisible by three (Peil and Hauryliuk 2010), and splitting tubes into smaller groups to centrifuge in turn. These practices could be time-consuming and possibly risky. For example, pure water in the blank tube which may not have the same density as the samples will alter the balance.

The balance of centrifuge rotors, however, can be achieved by the non-trivial placement of tubes. It has been proved for a rotor with n buckets that balancing k identical tubes ($0 \leq k \leq n$) is perfectly possible “if and only if both k and $n - k$ are expressible as linear combinations of prime factors of n with nonnegative coefficients” (Sivek 2010). In other words, the distribution of k loaded buckets in a balanced rotor is fundamentally a superposition of individual sets whose number of elements is a prime divisor of n and whose position in the rotor is exclusive (Peil and Hauryliuk 2010). Because rotors normally have an even number of buckets with 2 being its smallest prime divisor, tube distribution with rotational symmetry of order 2 (i.e. tubes in opposite buckets) always works. Other tube distributions with rotational symmetry of order 3, 5, 7, etc. could also be valid, which allows users more flexibility in operating centrifuges.

Notwithstanding its advantage, the non-trivial placement of tubes is difficult to be constructed and also perceived by the human mind. This issue is fully addressed by the **centrifugeR** package which yields informative results and visualizations. Given a centrifuge, the user will be informed about possible numbers of tubes that can be balanced at one time of operation, their corresponding positions in the rotor, and their required masses in case tubes are not initially identical.

*Updated 16 April 2020.

Availability

The `centrifugeR` package is written in R (R Core Team 2019) and is available on CRAN (<https://CRAN.R-project.org/package=centrifugeR>). To install and use `centrifugeR`:

```
install.packages("centrifugeR")
library(centrifugeR)
```

A web application built on `centrifugeR` and `shiny` (Chang et al. 2019) is available at <https://phamdn.shinyapps.io/centrifugeRGUI/>.

Illustrative Example

To demonstrate the functionality of `centrifugeR`, a typical centrifuge rotor with 30 buckets is considered ($n = 30$). Three prime divisors of n are 2, 3, and 5. A certain number of tubes (k) can be loaded in the rotor if and only if k and $n - k$ are linear combinations of 2, 3, and 5. Function `rotorCheck()` filters k values that meet this condition.

```
rotorCheck(30)
```

```
$valid
 [1] 0 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25
[26] 26 27 28 30
```

```
$invalid
 [1] 1 29
```

The fact that 1 or 29 tube(s) cannot be loaded in the rotor is obvious. However, some users may find the result surprising as the 30-bucket rotor supports almost all possible numbers of tubes.

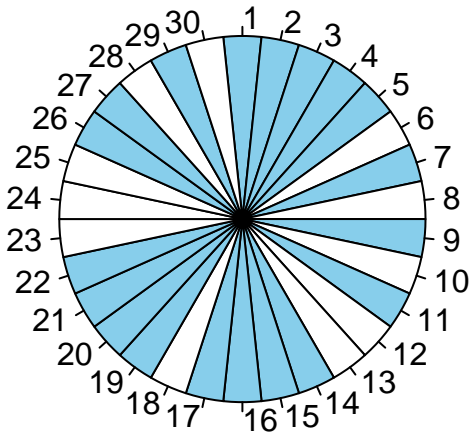
Users dealing with 19 tubes, for instance, tend to add a blank tube (increase k from 19 to 20) or split tubes into two turns of centrifugation (break k into 16 and 3). Function `rotorEqual()` can find a non-trivial way to balance 19 tubes. There are more than one possibility to express 19 as a linear combination of 2, 3, and 5 (e.g. $19 = 8 \times 2 + 1 \times 3 + 0 \times 5$ and $19 = 2 \times 2 + 0 \times 3 + 3 \times 5$). Not all of them work since the position of individual sets in the rotor must be exclusive. `rotorEqual()` performs random sampling to provide users with a solution (Figure 1).

```
rotorEqual(30, 19)
```

```
$loaded
 [1] 1 2 3 4 5 7 9 11 14 15 16 17 19 20 21 22 26 27 29
```

```
$empty
 [1] 6 8 10 12 13 18 23 24 25 28 30
```

Rotor with 30 buckets



19 loaded buckets in blue

Figure 1: The distribution of 19 tubes of equal mass in a 30-bucket rotor given by `rotorEqual()`.

In the case of 19 tubes of unequal initial mass, it is not necessary to increase the mass of 18 tubes to the mass of the heaviest tube. The only requirement is that tubes in the same set must have the same mass. Function `rotorUnequal()` returns the required masses and the positions of tubes (Figure 2).

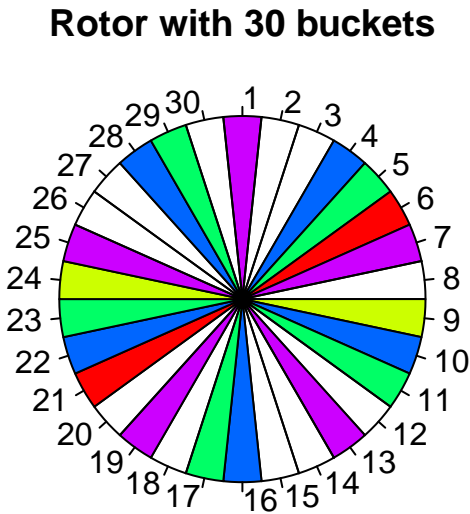
```
samples #mass values of 19 tubes
```

```
[1] 10.01  9.91  9.31  9.70 10.15 10.19  9.40  9.82  9.19  9.87 10.55 10.38
[13]  9.88 10.49 10.37 10.04  9.52  9.90 10.46
```

```
rotorUnequal(30, samples)
```

	initial	required	position
S9	9.19	9.31	6
S3	9.31	9.31	21
S7	9.40	9.52	9
S17	9.52	9.52	24
S4	9.70	9.90	5
S8	9.82	9.90	11
S10	9.87	9.90	17
S13	9.88	9.90	23
S18	9.90	9.90	29
S2	9.91	10.19	4
S1	10.01	10.19	10
S16	10.04	10.19	16
S5	10.15	10.19	22
S6	10.19	10.19	28
S15	10.37	10.55	1

S12	10.38	10.55	7
S19	10.46	10.55	13
S14	10.49	10.55	19
S11	10.55	10.55	25



[2 sets of 2 tubes][3 sets of 5 tubes]

Figure 2: The distribution of 19 tubes of unequal mass in a 30-bucket rotor given by `rotorUnequal()`.

Due to random sampling, both `rotorEqual()` and `rotorUnequal()` can give different results each time given the same input. Users who find it undesirable may use `set.seed()` before calling the functions.

Acknowledgments

The author would like to thank Ms. Martina Schmirler and Dr. Svetlana Herbrandt from the CRAN team for their helpful comments on the package. Thanks to Mr. Tuan-Anh Nguyen for proofreading the package documentation and the article.

References

- Chang, Winston, Joe Cheng, JJ Allaire, Yihui Xie, and Jonathan McPherson. 2019. *Shiny: Web Application Framework for R*. <https://CRAN.R-project.org/package=shiny>.
- Clark, Donald E. 2001. "Safety and the Laboratory Centrifuge." *Chemical Health and Safety* 8 (6): 7–13.
- Peil, Oleg, and Vasili Haurlyuk. 2010. "A New Spin on Spinning Your Samples: Balancing Rotors in a Non-Trivial Manner." *arXiv Preprint arXiv:1004.3671*.
- R Core Team. 2019. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: R Foundation for Statistical Computing. <https://www.R-project.org/>.
- Sivek, Gary. 2010. "On Vanishing Sums of Distinct Roots of Unity." *Integers* 10 (3): 365–68.