

SOFTWARE METAPAPER

Quandle and Biquandle Homology Calculation in R

Roger Fenn and Ansgar Wenzel

University of Sussex, GB

Corresponding author: Roger Fenn (R.A.Fenn@sussex.ac.uk)

In knot theory several knot invariants have been found over the last decades. This paper concerns itself with invariants of several of those invariants, namely the Homology of racks, quandles, biracks and biquandles. The software described in this paper calculates the rack, quandle and degenerate homology groups of racks and biracks. It works for any rack/quandle with finite elements where there are homology coefficients in Z_k . The up and down actions can be given either as a function of the elements of Z_k or provided as a matrix. When calculating a rack, the down action should coincide with the identity map. We have provided actions for both the general dihedral quandle and the group quandle over S_3 . We also provide a second function to test if a set with a given action (or with both actions) gives rise to a quandle or biquandle. The program is provided as an R package and can be found at <https://github.com/ansgarwenzel/quhomology>.

Keywords: Homology; Quandle; Biquandle; Rack; Birack; R; Knot Theory

AMS subject classification: 57M27; 57M25

(1) Overview

Introduction

This introduction is divided into two parts: First, we are going to give some mathematical background before introducing the software itself.

Mathematical Background

Racks and Quandles were first described by John Conway and Gavin Wraith in 1959 in unpublished correspondence. In [2], David Joyce showed that racks are indeed a knot invariant. Subsequently, Fenn and Rourke introduced racks as proper knot invariants in [3]. It is instructive to consider racks as groups without their multiplicative elements. Formally, a rack is defined as a set of elements endowed with a binary operation which satisfies the following axioms (where the third axiom holds only for a quandle, not for a rack) for all a, b, c in the rack/quandle R:

1. $\forall a, b \in R \exists! c \in R$ s.t. $a^c = b$

$$\begin{matrix} & b \\ & \cdot \\ a & \end{matrix}$$
2. $(c)^{(b^c)}$
 $(c) = a^c$
3. $a^a = a$ (only quandle)

Biracks and Biquandles are defined similarly, now with two operations, satisfying the following axioms:

1. $a^{bc} = a^{cb^c}$
2. $a_{bc} = a_{cb^c}$
3. $a_b^{cb} = a_{b^c}^c$
4. $a^a = a_a$ (only biquandle)

In addition, both operations satisfy the rack axiom 1.

Here, the up and down actions allow us to introduce a switch map via $f_a(x) = x_a, f^a(x) = x^a$ as $S(a, f_a(b)) = S(b, f^b(a))$. The utility of this is clear when one observes that the axioms can be reformulated using the switch map:

- S, f^a and f_a are all bijective.
- Yang Baxter Equation: $S_1 S_2 S_1 = S_2 S_1 S_2$, where $S_1(a, b, c) = (S(a, b), c)$ and $S_2(a, b, c) = (a, S(b, c))$.

This reformulation is more amenable to computational work.

The homology of the biquandle is defined as follows:

Let X be a birack. Then $C_n^R(X)$ be the free abelian group generated by n -tuples (x_1, x_2, \dots, x_n) , $x_i \in X$, that is, $C_n^R(X) = ZX^n$.

We define a boundary homomorphism by:

$$\partial(x_1 x_2 \dots x_n) = \sum_{i=2}^n (-1)^i \left[(x_1 \dots \widehat{x}_i \dots x_n) - x_1^{x_i} \dots x_{i-1}^{x_{i-1}} (x_{i+1})_{x_i} (x_n)_{x_i} \right]$$

and $(C_n^R(X), \partial)$ is called a chain complex of X .

Furthermore, we have a subchain complex $C_n^D(X) \subset C_n^R(X)$, generated by degenerate n -tuples (x_1, x_2, \dots, x_n) , $x_i \in X$ with $x_i = x_{i+1}$ for some i . Together with the boundary homomorphism, $(C_n^D(X), \partial)$ is called the degenerate chain complex of X .

Using both of those chain complexes, we can define the biquandle chain complex via the quotient chain complex, $C_n^Q(X) = C_n^R(X) / C_n^D(X)$. This gives the following short exact sequence of chain complexes:

$$0 \rightarrow C_n^D(X) \rightarrow C_n^R(X) \rightarrow C_n^Q(X) \rightarrow 0.$$

We can then define the birack, biquandle and degenerate homology groups in the usual way. In addition, we have the following long exact sequence of homology groups:

$$\dots \rightarrow H_n^D(X) \rightarrow H_n^R(X) \rightarrow H_n^Q(X) \rightarrow H_{n-1}^D(X) \dots$$

The algorithm for the homology calculation is described in [1] for this specific software.

Software

The software, which is provided as an R package, can be accessed on github. It provides two primary functions. One of these calculates the homology of racks and biracks, whilst the other verifies if a rack or birack is induced by a given set with one or two actions.

Implementation and architecture

This software is implemented in R. It has been tested on MacOS X, CentOS/Ubuntu and Windows without any problems.

The algorithm for the homology calculation is described in the paper [1].

The program is divided into the following two main parts: The calculation of the boundary matrix and the subsequent calculation of the Homology. For a graphical description see the following **Figure 1**, which was created with the following code:

```
library(proftools)
Rprof(tmp <- tempfile(), line.profiling = T); homology
(4,5,F);Rprof(append=F); pd <- readProfileData(tmp)
plotProfileCallGraph(pd,style=google.style,score="total",nodeSizeScore = "none",layout="dot",rankDir = "LR")
```

The boundary matrices are computed using the functions *boundary_matrix* (for quandle and rack boundary matrices) and *boundary_matrix_degenerate* (for degenerate boundary matrices), respectively. The methodology of both functions is similar, differing only in the manner in which degenerate or non-degenerate entries are removed where required. In particular, after creating a right-sized matrix, they call *boundary_names* or *boundary_names_degenerate* to create the row and column names of the boundary matrix. After this, they loop through the column names to calculate their boundaries and construct the matrix (for details see [1]).

These boundary matrices represent the boundary maps of the simplicial complex of the rack/birack.

After both boundary matrices have been calculated, they are returned to the *homology* and *degenerate_homology* functions, respectively. As is the case for the boundary matrices functions, these two functions only differ in the boundary functions called and in their respective output texts.

As an aside, those two functions should be the only ones that would have to be called by the user in order to calculate a homology.

Following on with the algorithm described in [1], those two functions calculate the image and kernel of the boundary map representations (the boundary matrices) before finding a representation of the homology group. For this, they call various functions, namely *findX*, which “finds X ” (this is defined in [1]), *row_space*, which calculates a basis of the row space of a matrix, *matrix_rank*, which calculates the rank of a matrix and *GaussianElimination*, a function written by Prof John Fox (see [4] for a source) which does a Gaussian Elimination on a matrix and returns its reduced row echelon form.

Using the function *smith*, the smith normal form is obtained from the representation matrix of the homology. This is done via repeated calculation of the hermite normal form of the matrix and its transpose, using the *hermiteNF* function from the *numbers* package, [8]. In addition, this function checks if the diagonal of the matrix is in the correct form via the function *check_more_push* and, if required, will call *push_down* to do what the name implies.

Finally, the homology group is obtained using the diagonal of the smith normal form.

The second function of this package, the testing if a give operation or operations give rise to a quandle or biquandle is done via the function *S_test*. This function receives as input the order of the underlying set and then proceeds to test the two requirements for a quandle/biquandle as described before.

Quality control

The results of the program have been compared to known results and in addition, *R CMD check* has been used to quality check the code itself.

Additionally, a few tests have been provided in *test/testthat.R*.

(2) Availability

Operating system

Any OS that can install and run R (at least version 3.0.0).

Programming language

R 3.0.0+.

Additional system requirements

The more RAM, the higher the homology groups that can be calculated.

Presently, output is on screen, but can be changed to a file, if necessary.

Required input devices: keyboard only.

Date published: 04/01/2017

Version: 1.1.0

Code Repository

Names: GitHub

Identifier: <https://github.com/ansgarwenzel/quhomology>

License: GNU GPL v3.0

Date published: 05/05/2016

Version: 1.1.0

Language

English

Support

Support is currently provided via Github issues.

(3) Reuse potential

This software can be used to calculate the homology groups of most racks and biracks. It is very easy to adapt for application to other racks/biracks. Furthermore we believe that it can easily be extended to the calculation of Cohomology groups. Finally, the possibility of quickly identifying if a given action/set of actions gives rise to a rack/birack, is very useful.

Acknowledgements

We would like to thank all contributors and developers of the R software, in particular the authors of the packages used in this program as well as of RStudio.

Competing Interests

The authors have no competing interests to declare.

References

1. **Fenn, R** 2014 *How to calculate Homology*. <http://www.maths.sussex.ac.uk/Staff/RAF/Maths/>.
2. **Joyce, D** 1982 A classifying invariant of knots: the knot quandle. *Journal of Pure and Applied Algebra* 23: 37–65. DOI: [https://doi.org/10.1016/0022-4049\(82\)90077-9](https://doi.org/10.1016/0022-4049(82)90077-9)
3. **Fenn, R** and **Rourke, C** 1992 Racks and links in codimension 2. In *Journal of Knot Theory and its Ramifications* 1: 343–406.
4. **Fox, J** 2014 *Gaussian Elimination and rref function*. <http://socserv.mcmaster.ca/jfox/Courses/R-course-Berkeley/>.
5. **R Core Team** 2014 R: A language and environment for statistical computing. *R Foundation for Statistical Computing*. Vienna, Austria. <http://www.R-project.org/>.
6. **Venables, W N** and **Ripley, B D** 2002 *Modern Applied Statistics with S*. Fourth Edition. Springer, New York. ISBN 0-387-95457-0. DOI: <https://doi.org/10.1007/978-0-387-21706-2>
7. **Bates, D** and **Maechler, M** 2014 *Matrix: Sparse and Dense Matrix Classes and Methods*. R package version 1.1–4. <http://CRAN.R-project.org/package=Matrix>.
8. **Borchers, H W** 2014 *numbers: Number-theoretic Functions*. R package version 0.4–5, <http://CRAN.R-project.org/package=numbers>.

How to cite this article: Fenn, R and Wenzel, A 2018 Quandle and Biquandle Homology Calculation in R. *Journal of Open Research Software*, 6: 3. DOI: <https://doi.org/10.5334/jors.53>

Submitted: 16 July 2014 **Accepted:** 31 August 2017 **Published:** 16 January 2018

Copyright: © 2018 The Author(s). This is an open-access article distributed under the terms of the Creative Commons Attribution 4.0 International License (CC-BY 4.0), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited. See <http://creativecommons.org/licenses/by/4.0/>.

 *Journal of Open Research Software* is a peer-reviewed open access journal published by Ubiquity Press

OPEN ACCESS 