

User Manual for BACH

Quan Zhou (serotonin.z.q@gmail.com) and Yongtao Guan

Last update: 7 Dec 2015

BACH implements Bausch's algorithm to evaluate distribution function of chi-square weighted sum [1]; one important application is to compute p-values of Bayes factors [2]. There are two ways to run BACH : simple or in batch. Note in the following example, we use “./bach” as the program name; in the download there are “bach-mac” (bach for Mac OS X) and “bach-linux” (bach for Linux). You may either rename the executable or modify the command lines below.

1 Simple run

Denoted by χ_1^2 a chi-squared random variable (r.v.) with 1 degree of freedom (d.f). Let $Y = X_1 + 0.8X_2 + 0.6X_3 + 0.4X_4$, where $X_i \sim \chi_1^2$, and we want to compute $P(Y > 10)$. The command line is:

```
./bach 1 0.8 0.6 0.4 10
```

A p-value will be printed to the screen:

```
0.00914343
```

Rules of input: Numbers are separated by space, the last number is statistic and numbers before are coefficients.

Suppose we also want to compute $P(Y > 20)$. Of course we may repeat the above command line, substituting 10 with 20. But a more efficient way is to compute $P(Y > 20)$ and $P(Y > 10)$ simultaneously because we may reuse many intermediate results. The command line is:

```
./bach 1 0.8 0.6 0.4 10,20
```

Two p-values will be printed to the screen:

```
0.00914343 4.31004e-5
```

Rules of input: If we want to perform more computations involving the same coefficients, just append more test statistics in the end using ',' to delimit.

2 Batch run

Suppose $Z = 5X_5 + 0.2X_6$, where $X_i \sim \chi_1^2$, and we want to compute $P(Z > 34)$ in a batch with $P(Y > 10)$ and $P(Y > 20)$. We can create the following file (named “text.input”):

```
1 0.8 0.6 0.4 10,20
5 0.2 34
```

Each row in the input file is what you would type using command line save the “./bach”.
To run BACH in batch:

```
./bach -i test.input
```

Three p-values will then be printed on the screen:

```
0.00914343 4.31004e-5
0.00932733
```

Within each row, multiple p-values are delimited by a tab ('\t'); and rows in printout correspond to rows in the input file.

3 Output file

When ‘-o’ argument is invoked for batch mode, an output file will be generated.

```
./bach -i test.input -o test.output
```

This command will produce a ‘test.output’ which reads

p-value	error-bound	coefficients	statistic
0.00914343	3.02826e-15	0.4 0.6 0.8 1	10
4.31004e-05	5.37358e-11	0.4 0.6 0.8 1	20
0.00932733	3.55176e-08	0.2 5	34

The output file contains 4 columns: p-value; error bound, which is the upper error bound of the p-value; coefficients, which are coefficients of χ_1^2 used in computation; and statistic. Between field the delimit is the tab, and within the filed of coefficient the delimit is the blank.

4 Options

- -i [string=]: the input filename. This is required for batch mode.
- -o [string=]: the output filename for the batch mode
- -c [int=6]: the precision of all output (number of significant digits, default is 6)
- -h : stop the program and print the HELP

There are additional options to control error bounds, some of which concern GMP library used in our implementation; one may send an email to inquire if on the off chance one is interested.

References

- [1] Johannes Bausch. On the efficient calculation of a linear combination of chi-square random variables with an application in counting string vacua. *Journal of Physics A: Mathematical and Theoretical*, 46(50):505202, 2013.
- [2] Quan Zhou and Yongtao Guan. On the null distribution of bayes factors in linear regressin. *submitted*, 2015.