John Miyamoto (email: jmiyamot@uw.edu)

# Installing BUGS and the R to BUGS Interface

Caveat: I am a Windows user so these notes are focused on Windows 7 installations. I will include what I know about the Mac and Linux versions of these programs but I cannot swear to the accuracy of my comments.

## Contents (Cntrl-left click on a link to jump to the corresponding section)

# End of Contents Table

## 1. Brief Overview TOC

BUGS stands for Bayesian Inference Under Gibbs Sampling[1]. The BUGS program serves two critical functions in Bayesian statistics. First, given appropriate inputs, it computes the posterior distribution over model parameters - this is critical in any Bayesian statistical analysis. Second, it allows the user to compute a Bayesian analysis without requiring extensive knowledge of the mathematical analysis and computer programming required for the analysis. The user does need to understand the statistical model or models that are being analyzed, the assumptions that are made about model parameters including their prior distributions, and the structure of the data, but the BUGS program relieves the user of the necessity of creating an algorithm to sample from the posterior distribution and the necessity of writing the computer program that computes this algorithm. Today, there are three main versions of the BUGS program: OpenBUGS, WinBUGS and JAGS. Specifically:

OpenBUGS, WinBUGS & JAGS

are three different programs that serve the same function, namely, to compute Markov-Chain-Monte-Carlo (MCMC) simulations that perform Bayesian computations. Mathematically, the MCMC simulation computes an approximation to the posterior probabilities in a Bayesian inference problem by sampling from this posterior probability distribution.

You can use R or Matlab to enter your data and manipulate it in various ways, but you need a way to transfer these data to OpenBUGS, WinBUGS or JAGS and a way to instruct the BUGS program

[1] What happened to the "I" in Bayesian Inference Under Gibbs Sampling? I have no explanation except for the point that BIUGS is awkward and not memorable.

regarding the statistical analysis that you are trying to compute. The following packages are interfaces between R and one of the BUGS programs:

**R2OpenBUGS** & **BRugs** are R packages that provide an interface between R and OpenBUGS. **R2OpenBUGS** and **BRugs** have overlapping functionality - in general they serve the same purpose, although there are situations where one or the other is easier to use or has a function that the other lacks.

**R2WinBUGS** is an R package that provides an interface between R and WinBUGS.

**rjags** & **R2jags** are R packages that provide an interface between R and JAGS. **rjags** and **R2jags** have overlapping functionality - in general they serve the same purpose, although there may be situations where one or the other is easier to use or has a function that the other lacks.

There are a variety of R packages that contain functions for analyzing BUGS output. For Psychology 548, the following two packages will be relevant.

**boa** and **coda** are R packages for analyzing the convergence of MCMC simulations, i.e., the functions in **boa** and **coda** are sets of R functions that are used to analyze the output of one of the MCMC programs, OpenBUGS, WinBUGS or JAGS. The functions in these packages appear to be highly similar - at present (3/26/2012) I do not know of a reason to prefer one of these packages over the other. **coda** is probably the more widely used.

*Comment*: It may seem daunting to imagine that you need to install all of this software and learn how to use all of the functions in these programs. This apprehension is based on a false premise - in fact, you only need a subset of this software. Here are my suggestions:

### All users must install R.

### If you are a Mac or Linux user:

- Install JAGS (this is a separate program from R).
- Install the **rjags** and **coda** packages (they run under R).
- Optionally, install the **R2jags** package.

### If you are a Windows user:

- Install OpenBUGS and JAGS (these are separate programs from R).
- Install the **BRugs**, **rjags**, and **coda** packages.
- Optionally, install the **R2OpenBUGS** and **R2jags** packages.

- Optionally, install WinBUGS (useful if someone else has computed some results with WinBUGS and then published the data and results; you may want to use WinBUGS to reproduce the results). If you follow this option, you will also want **`R2WinBUGS`**.

## 2. Installing OpenBUGS (Windows only) TOC

The OpenBUGS homepage is: http://www.openbugs.info/w/FrontPage. Windows, MacOS and Linux versions of OpenBUGS can be downloaded from: http://www.openbugs.info/w/Downloads .

(DO NOT GO TO http://mathstat.helsinki.fi/openbugs/Home.html; this is an obsolete version of the OpenBUGS website - the links to downloads from this site are broken).

Starting from the OpenBUGS downloads page:

- Choose the flavor of OpenBUGS that matches the operating system of your computer. Then download and install OpenBUGS by the method that is standard on your operating system.
- An html manual for OpenBUGS is installed automatically when you install OpenBUGS.

## 3. Installing WinBUGs (Windows only) TOC

Go to the WinBUGS page of the BUGS project: http://www.mrc-bsu.cam.ac.uk/bugs/winbugs/contents.shtml

There are separate installation methods for the 32-bit version and the 64-bit version so make sure you know whether you are working on a 32-bit or 64-bit version of Windows.

* This Microsoft link has an explanation of how to tell whether your computer runs 32-bit or 64-bit Windows: http://windows.microsoft.com/en-US/windows-vista/32-bit-and-64-bit-Windows-frequently-asked-questions).

Comment: The **`bugs`** function in the **`R2WinBUGS`** packages assumes by default that the BUGS program has been installed to the "C:/Program Files/WinBUGS14" directory. You can install WinBUGS in a different directory but if you do so, you will need to override the default program directory when you use the **`bugs`** function. (The BUGS program directory is set by the **`bugs.directory`** argument of the **`bugs`** function.)

## 4. OpenBUGS versus WinBUGS TOC

The OpenBUGS website (http://www.openbugs.info/w/Overview) states:

> "There are two main versions of BUGS, namely WinBUGS and OpenBUGS. This site is dedicated to OpenBUGS, an open-source version of the package, on which all future development work will be focused. OpenBUGS, therefore, represents the future of the BUGS project. WinBUGS, on the other hand, is an established and stable, stand-alone version of the software, which will remain available but not further developed. "

Right now (3/26/2012), I don't know of any reasons to prefer WinBUGS over OpenBUGS. Since the programs are small by modern standards, I have installed both programs on my computer on the chance that I may want to use WinBUGS to check a computation. For my own work, I use OpenBUGS.

## 5. Installing JAGS (there are versions for Windows, Mac or Linux OS) TOC

The JAGS information page is at http://mcmc-jags.sourceforge.net/. To download JAGS,

- Go to http://sourceforge.net/projects/mcmc-jags/files/.
- Click on the link to JAGS.

- Choose the Windows, Mac or Linux version.
- Download and install the program by the method that is standard on your operating system.
- It is useful to also download the JAGS manual and examples. You don't have to download **rjags** because there is a simpler, alternative way to download and install it.

## 6. Installing R packages that are used with OpenBUGS, WinBUGS, and JAGS TOC

A "package" in R is a set of R functions and objects that have been created for some special purpose. A number of packages have been written for R that integrate R with the MCMC computational programs, **OpenBUGS**, **WinBUGS**, and **JAGS**. These packages serve two purposes: (i) to pass data and a definition of the modeling problem from R to **OpenBUGS**, **WinBUGS**, or **JAGS** where the MCMC simulation is computed; (ii) to carry out analyses in R of the output of the MCMC simulation.

**Table 1**. A Brief Description of R Packages for Bayesian Statistics

| Package | Purpose |
|---|---|
| **boa** | Bayesian Output Analysis Program (BOA) for MCMC. The **boa** package contains R functions for examining the adequacy of the MCMC simulation. "Adequacy" refers to (a) whether a burn-in sample is required, and if so, how large? (b) whether or not the samples of parameters from the posterior distribution have converged to the true distribution (this is the main question); (c) whether the samples of parameters exhibit high autocorrelation. Although there is no fool proof way to determine whether an MCMC simulation has converged, **boa** contains the standard diagnostic tools for checking failure to converge.<br>The **boa** and **coda** packages serve the same purpose, to check for convergence of MCMC chains, and at present (6/6/2012), I don't know of a reason for preferring one package over the other. |
| **BRugs** | **BRugs** contains R functions that define a Bayesian inference problem, and transfer the problem and data to OpenBUGS where the simulation is carried out. It also has functions for retrieving the output of the OpenBUGS simulation and analyzing the results. Note: **BRugs** runs well on the 32-bit version of R. It often runs well on the 64-bit version of R, but I have "succeeded" in freezing my computer while running **BRugs** with 64-bit R. See the note below re running **BRugs** on a 32-bit or 64-bit Windows computer. |
| **coda** | **coda** contains R functions for examining the adequacy of the MCMC simulation. "Adequacy" refers to (a) whether a burn-in sample is required, and if so, how large? (b) whether or not the samples of parameters from the posterior distribution have converged to the true distribution (this is the main question); (c) whether the samples of parameters exhibit high autocorrelation. Although there is no fool proof way to determine whether an MCMC simulation has converged, **coda** contains the standard diagnostic tools for checking failure to converge.<br>The **boa** and **coda** packages serve the same purpose, to check for convergence of MCMC chains, and at present (6/6/2012), I don't know of a reason for preferring one package over the other. |
| **R2jags** | **R2jags** contains R functions for defining a Bayesian inference problem, and |

| | |
|---|---|
| | transfering the problem and data to JAGS where the simulation is carried out. |
| **R2OpenBUGS** | **R2OpenBUGS** contains R functions for defining a Bayesian inference problem, and transfering the problem and data to OpenBUGS where the simulation is carried out. |
| **R2WinBUGS** | **R2WinBUGS** contains R functions for defining a Bayesian inference problem, and transfering the problem and data to WinBUGS where the simulation is carried out. |
| **rjags** | **rjags** contains R functions for defining a Bayesian inference problem, and transfering the problem and data to JAGS where the simulation is carried out. |

Installing packages in R is extremely easy. If your computer is currently connected to the internet, you can download and install an R package as follows. Run R; then type the R command:

```
install.packages(c(<package names>))
```

where **`<package names>`** has been replaced with a vector of package names, like: **`c("boa", "BRugs")`**. The packages that are listed in Table 1 can be installed in two easy steps:

(i) Run R.

(ii) Run the R command (this command assumes that you are connected to the internet):

```
install.packages(pkgs = c("boa", "BRugs", "coda", "R2jags", "R2OpenBUGS",
        "R2WinBUGS", "rjags", ), repos = "http://cran.fhcrc.org/")
```

- You can copy and paste this command into R - it should work without modification.
- Some authors have stated that you must install JAGS *before* you install the JAGS-related packages. I haven't tested whether this is really true, but it can't hurt to do things in this order. I have also seen it stated that you should install **`rjags`** last among the packages. Again, I haven't tested what happens if you install packages in different orders.
- The final clause, **`repos = http://cran.fhcrc.org/`**, simply indicates that you want to download from the R mirror site at the Fred Hutchison Cancer Center (local to Seattle). You can omit this clause and you will be given a choice of other sites.

In the future, you can use a command like the one shown in (ii), if you need to install more packages. In addition, if you want to update the version of your packages, e.g., if some time has passed and you are concerned that you don't have the latest version, then give the R command:

```
update.packages(c(<package names>))
```

*WARNING*: Under Windows 7, you may need to establish that R has "administrator privileges" before R is allowed to add new packages to the R installation. I don't know the principles behind how to do this, but what I do is to right-click on the R program icon (on the Desktop) and choose "Properties". I then go through every tab of the dialog box, click on every button that is labeled "Advanced", and if there is any option to "run as administrator" I choose that option. I find that this generally works, but I don't really understand what is happening here.

## 7. Running `BRugs` on a 32-bit or 64-bit Windows computer TOC

* This section is irrelevant to a Mac or Linux user.

The current Windows computers have either a 32-bit or 64-bit processor (never both). A 64-bit processor can usually (not invariably) run software that was written for a 32-bit processor, but the opposite is not true. The standard installation of R gives the user both a 32-bit version of R and a 64-bit version of R. If you are using a computer with a 32-bit processor, you can only use the 32-bit version of

R, but if you are using a computer with a 64-bit processor, you can use either version of R. To find out whether your computer has a 32-bit processor or a 64-bit processor, go to:

> http://windows.microsoft.com/en-us/windows7/32-bit-and-64-bit-Windows-frequently-asked-questions

This webpage will instruct you as to how to decide which processor you have.

**BRugs** is fully compatible with the 32-bit version but not the 64-bit version. The BRugs description on CRAN (http://cran.r-project.org/web/packages/BRugs/index.html) states:

> "[r]uns natively and stably in 32-bit R under Windows. Versions running on Linux and on 64-bit R under Windows are in "beta" status and less efficient."

The OpenBUGS website (http://openbugs.info/w/UserContributedCode) states:

> "BRugs works natively on 32-bit R under Windows by calling the 32-bit OpenBUGS shared library. Under Linux or 64-bit R Windows, BRugs uses a proxy program to call the shared library - therefore on these platforms BRugs is less efficient and stable."

From my perspective, the main issue is not whether **BRugs** is efficient on 64-bit R, but whether it is stable, i.e., does it hang up? I have encountered cases where an analysis can be performed with **BRugs** on a 32-bit R but not on a 64-bit R. My advice is:

(a) if you work on a Windows computer with a 32-bit CPU, you must use the 32-bit version of R so you have no choices on this issue;

(b) if you work on a Windows computer with a 64-bit CPU, you should use the 32-bit version of R whenever you plan to use **BRugs** to control your work with OpenBUGS; if your work does not involve **BRugs**, you can use the 64-bit version of R because it is slightly faster. If the models you study are relatively simple and if the dataset that you work on are relatively small, you will not the difference in speed of 32-bit R and 64-bit R.

(c) if you do not plan to use **BRugs**, e.g., because you are a Mac user (the **BRugs**/OpenBUGS interface will not run on a Mac) or because you plan to use **R2OpenBUGS** to interface with OpenBUGS, then you can use the 64-bit version of R.

I generally use the 32-bit version of R because I use **BRugs** quite often and because the datasets that I analyze are so small that the difference in processing time between the 32-bit and 64-bit versions of R is negligible.

## 8. Hints for Mac and Linux Users TOC

I am not a Mac user, and I am a novice Linux user, so all I can really do is to pass along information that I have encountered over the internet. In general, assume that I don't really know what I am talking about when it comes to Mac or Linux issues.

- Kruschke has a note about installing JAGS on a Mac:

  http://doingbayesiandataanalysis.blogspot.com/2011/06/brugs-for-mac-users.html.
  He references: http://enfascination.com/wiki/Weblog:Bayesian_data_analysis_on_a_Mac
  (this looks like a good source for installing JAGS on a Mac.)
  See also the following for more suggestions regarding the installation of JAGS on a Mac.
  http://doingbayesiandataanalysis.blogspot.com/2012/01/complete-steps-for-installing-software.html

- This is a careful explanation of how to install JAGS and **`R2jags`** on a Mac (http://jkarreth.myweb.uga.edu/bayes/jags.tutorial.pdf). If you can't find this pdf on the internet, I have a pdf copy.
- According to **`R2jags`** creator, it can be tricky to install **`JAGS`** and **`R2jags`** on a Mac or Linux system. The following two blog posts give advice as to how to do this.
  - http://andrewgelman.com/2009/02/our_new_r_packa/
  - http://yusung.blogspot.com/2009/01/install-jags-and-rjags-in-fedora.html

  Frankly, I don't understand these notes, but I'm passing them along in case they are useful to Mac or Linux users.
- David Huber has a discussion of how to run BUGS on a Mac. See http://psy2.ucsd.edu/~dhuber/rjags.pdf
- The OpenBUGS website has a discussion of how to install OpenBUGS on a Mac or Linux machine.
- There is a post at:

  http://cddesjardins.wordpress.com/2009/06/15/openbugs-and-winbugs-on-mac-os-x/

  for installing JAGS on a Mac. The following are the recommended steps.
  1) Download the Mac OS X JAGS 2.1.0 image by clicking here (http://ubuntuone.com/p/AqO/).
  2) Install the .dmg like you would any other Application (e.g. Firefox)
  3) Install rjags or R2jags
  3a) Open R
  3b) then run the following:

  ```
  install.packages("rjags")
  install.packages("R2jags")
  ```

  - JAGS has a news site where Mac OSx issues are discussed. http://martynplummer.wordpress.com/category/mac-os-x/

## 9. References TOC

Gilks, W. R., Thomas, A., & Spiegelhalter, D. J. (1994). A language and program for complex Bayesian modelling. *The Statistician*, *43*(1), 169–177.

Lunn, D., Spiegelhalter, D., Thomas, A., & Best, N. (2009). The BUGS project: Evolution, critique and future directions. Statistics in Medicine, 28, 3049-3967.