

The Bayesian Analysis Software Developed At Washington University

Karen R. Marutyan, PhD and G. Larry Bretthorst, PhD

Biomedical Magnetic Resonance Laboratory, Mallinckrodt Institute of Radiology, Washington University, St. Louis, MO 63110

Abstract. Over the last few years there has been an ongoing effort at the Biomedical Magnetic Resonance Laboratory within Washington University to develop data analysis applications using Bayesian probability theory. A few of these applications are specific to Magnetic Resonance data, however, most are general and can analyze data from a wide variety of sources. These data analysis applications are server based and they have been written in such a way as to allow them to utilize as many processors as are available. The interface to these Bayesian applications is a client based Java interface. The client, usually a Windows PC, runs the interface, sets up an analysis, sends the analysis to the server, fetches the results and displays the appropriate plots on the users client machine. Together, the client and server software can be used to solve a host of interesting problems that occur regularly in the sciences. In this paper, we describe both the client and server software and briefly discuss how to acquire, install and maintain this software.

Keywords: Bayesian analysis software, Markov chain Monte Carlo, Bayesian probability theory

PACS: 02.70.Uu

INTRODUCTION

The Bayesian Analysis software developed at the Biomedical Magnetic Resonance Laboratory within Washington University is a user friendly client/server based software system for users who are not necessarily Bayesians. The client software is a Java based interface that gathers the information needed to run the Bayesian applications and sends that information in the form of a request to the server. The server software consists of two parts: a web server and the associated software that processes server requests. Server requests can be as simple as a request for the status of a job or as complicated as requesting a Bayesian analysis be processed by the server. These Bayesian analysis include analyzing data modeled as exponentials, frequencies, two site exchange models and an application that allows users to build their own model and then analyze their data using Bayesian probability theory. The following sections give an overview of the server applications, introduce the client software, and describe how to download and install both the client and server software.

THE SERVER SOFTWARE

The server software consists of two parts, an Apache web server and the Bayesian analysis software. The Apache web server has six scripts that process requests coming to the server. These scripts do such things as get the status of a job, submit a job for

processing and send the results of a job back to the clients. Exactly how the client and the server software is not important, the important point is that the clients send jobs to the Apache server and the server processes these requests. The remaining part of this section will discuss the Bayesian analysis software. In this discussion the model used by the software will be described and, for parameter estimation, the calculations done by the Bayesian analysis applications will be given.

The Bayesian Applications

The main feature of the Bayesian applications is that they use Bayesian probability theory to analyze various common problems in data analysis. In Bayesian probability theory, states of knowledge are encoded by probabilities. For parameter estimation problems, to compute these probabilities one must relate the parameters to the available data and eventually to all of the other prior information.

In the Bayesian applications described in this paper, both the parameter estimation and model selection applications use the same general functional form to relate the parameters to the acquired data. If the data value acquired at x_i is represented by $d(x_i)$, then the data and the model are related by

$$d(x_i) = M(\mathcal{A}, \Theta, x_i) + n(x_i) \quad (1)$$

where we have split the parameters into two sets, the amplitudes represented symbolically by \mathcal{A} and all other parameters Θ . The error, the difference between the data and the model, is represented symbolically by $n(x_i)$. In the Bayesian calculations that follow, the amplitudes are marginalized analytically. Consequently, the model is written as

$$M(\mathcal{A}, \Theta, x_i) = \sum_{k=1}^n A_k G_k(\Theta, x_i) \quad (2)$$

for analytical convenience, where n is the number of amplitudes in a given data set. The model is a sum consisting of an amplitude A_k times a function $G_k(\Theta, x_i)$. The $G_k(\Theta, x_i)$ are functions of the $\Theta \equiv \{\theta_1, \theta_2, \dots, \theta_m\}$ parameters, and the amplitudes cannot appear in these functions. In this equation the sample points, x_i , the data values, $d(x_i)$ and the functions, $G_k(\Theta, x_i)$, are assumed to be vector valued. Additionally, this equation has been written as if there is only a single data set, however multiple data sets are allowed where the number of data values, the sample points and the amplitudes can be different in the different data sets.

Because this equation is written in a generalized form, it obscures the simplicity of the types of models this equation represents. To give one example, suppose there is a single data set modeled as a biexponential signal plus a constant, then Eq. (1) and Eq. (2) can be written as:

$$d(t_i) = A_1 \exp(-\theta_1 t_i) + A_2 \exp(-\theta_2 t_i) + A_3 + n(t_i) \quad (3)$$

where there are three amplitudes, $\mathcal{A} \equiv \{A_1, A_2, A_3\}$. The vector valued x_i was replaced by a scalar time value t_i . There are two Θ parameters, $\Theta \equiv \{\theta_1, \theta_2\}$, which are the decay rate constants of the two exponentials, and there are three functions G_k given by

$G_1(\Theta, t_i) \equiv \exp(-\theta_1 t_i)$, $G_2(\Theta, t_i) \equiv \exp(-\theta_2 t_i)$ and one degenerate function, $G_3(\Theta, t_i) \equiv 1$, which represents the constant offset.

In the Bayesian calculations that follow, it is assumed that a reader has a good working knowledge of Bayesian Probability theory. If not, see the excellent tutorial by Devinder Sivia [1] and for much more on probability theory see Jaynes [2]. In a typical parameter estimation problem, it is the posterior probability for each θ_j given the data and the prior informations, represented symbolically by $P(\theta_j|DI)$, that is desired. To obtain these posterior probabilities, the server applications use Markov chain Monte Carlo simulations to draw samples from the joint marginal posterior probability for the all of the Θ parameters. This probability is represented symbolically by $P(\Theta|DI)$. Having the samples from $P(\Theta|DI)$, the server applications then use Monte Carlo integration to obtain samples from each $P(\theta_j|DI)$. Consequently, the quantity that must be computed is the joint posterior probability for the Θ parameters. This probability is computed by application of Bayes' theorem [3],

$$P(\Theta|DI) = \frac{P(\Theta|I)P(D|\Theta I)}{P(D|I)} \quad (4)$$

where D symbolically represents all of the data and I represents all of the prior information. The three probabilities on the right-hand side, are the joint prior probability for the parameters, $P(\Theta|I)$, the probability for the data given the parameters, $P(D|\Theta I)$, and the probability for the data given only the prior information $P(D|I)$, which is a normalization constant. If $P(\Theta|DI)$ is normalized at the end of the calculation, this posterior probability can be written as:

$$P(\Theta|DI) \propto P(\Theta|I)P(D|\Theta I). \quad (5)$$

Assuming multiple data sets are available, $D \equiv \{D_1, \dots, D_N\}$, then the direct probability for all of the data, $P(D|\Theta I)$ can be factored using the product rule:

$$P(D|\Theta I) = \prod_{k=1}^N P(D_k|\Theta I) \quad (6)$$

where this factorization assumes that each data set is logically independent. Each of the $P(D_k|\Theta I)$ are marginal direct probabilities because the amplitudes have been marginalized using the rules of probability theory:

$$\begin{aligned} P(D_k|\Theta I) &= \int P(\mathcal{A} D_k|\Theta I) d\mathcal{A} \\ &= \int P(\mathcal{A}|I) P(D_k|\mathcal{A} \Theta I) d\mathcal{A} \end{aligned} \quad (7)$$

where it is assumed that each data set has different amplitudes \mathcal{A} . Using Eq. (6) as the symbolic representations for the direct probabilities, the posterior probability for the Θ parameters, $P(\Theta|DI)$, may be written as:

$$P(\Theta|DI) \propto \prod_{j=1}^m [P(\theta_j|I)] \left[\prod_{k=1}^N P(D_k|\Theta I) \right] \quad (8)$$

where $P(D_k|\Theta I)$ is given by Eq. (7).

For parameter estimations problems, Eq. (8) represents the calculation performed by the server software. The client software must gather the informations necessary to perform this calculation. Gathering this information includes loading the necessary data, specifying model options and making the assignments for the prior probabilities.

The prior probabilities for the θ_j parameters, $P(\theta_j|I)$, are assigned by the users. The users can select the functional form to represent these priors. These functional forms include Gaussian, uniform, Jeffreys like, exponential and a delta function. Uniform, exponential and Gaussian priors are exactly what one would expect. The delta function prior is used to constrain a parameter to a fixed value, this mechanism is used to pass constants to the model equations. These constants are typically experimental parameters needed for unit conversions. Finally, the Jeffreys like prior is given by: $P(\theta_j|I) \propto \theta_j/(M^2 + \theta_j^2)$, which has a peak at M , goes to zero at $\theta_j = 0$ and behaves like a Jeffreys' prior for large θ_j . This prior is particularly useful when estimating decay rate constants because they must be positive, one always has some vague estimate of the their value and the Jeffreys' like behavior the prior does not overly penalize large rate constants. In all cases the user must specify the functional form of prior, the parameter bounds and any other quantities associated with a given prior type. When the server receives an analysis, the server application discretized and then normalizes each prior to ensure that all numerical calculations are well behaved.

The prior probabilities for the amplitudes, $P(A_k|I)$, are assigned as Gaussians of the form:

$$P(A_k|I) = \begin{cases} N_k \exp \{ -(Mean_k - A_k)^2 / 2\sigma_k^2 \} & \text{If } Low_k \leq A_k \leq High_k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

where Low_k , $Mean_k$, $High_k$ and σ_k constitute the prior information about the k th amplitude, and N_k is the normalization constant. Because the amplitudes are marginalized, the functional form of these priors cannot be changed by the users. However, the users must specify Low_k , $Mean_k$, $High_k$ and σ_k and these quantities are used in the calculations.

Markov Chain Monte Carlo

The server software contains a series of programs which we call packages and these packages implement various Bayesian calculations. Most of these packages run multiple Markov chain Monte Carlo (McMC) simulations with simulated annealing to draw samples from the joint posterior probability for the Θ parameters, Eq. (8). Monte Carlo integration is then used to obtain samples from the posterior probabilities for each parameter. Each McMC simulation consists of three probabilities, $P(\Theta|DI)$, $P(\Theta|I)$ and $P(D|\Theta I)$, the Θ parameter samples and the amplitudes that maximize the joint posterior probability for the amplitudes given the Θ parameters, the integrand of Eq. (7). These amplitudes samples are used to approximate the marginal posterior probability for each amplitude, $P(A_k|DI)$.

In simulated annealing, an annealing parameter, β , is introduced by raising the direct probability for the data to the β power. The calculations are done using the logarithm of the marginal posterior probability for the Θ parameters, so the McMC simulations are calculating:

$$\log[P(\Theta|DI)] = \sum_{j=1}^m \log[P(\theta_j|I)] + \beta \sum_{k=1}^N \log[P(D_k|\Theta I)]. \quad (10)$$

During the annealing phase, β is varied from zero to one according to an annealing schedule. When $\beta = 0$ the logarithm of the direct marginal probability is multiplied by zero, and the data are ignored. Consequently, the McMC simulations are exploring the joint prior probability for the Θ parameters, and because the Θ parameters are initialized by drawing samples from this joint prior, the simulations start at or near equilibrium. Here equilibrium means that as the parameter space is explored by the McMC simulations, the moments of the Θ parameters are static, i.e., unchanging to within the uncertainty in a parameter estimate in spite of the fact that each parameter is being varied in each McMC simulation.

The annealing parameter is varied from zero to one according to some schedule. Simple applications might vary β linearly from zero to one; while a more complicated applications might vary the size of the annealing step depending on the direct probability. Regardless of the schedule used, increasing the β parameter throws the McMC simulations out of equilibrium. Consequently, after β is increased the simulations are processed a number of times in an effort to bring the simulations back to equilibrium. Processing an McMC simulation consists of varying parameters, evaluating Eq. (10) and then keeping or rejecting the simulation based on a Metropolis-Hastings' acceptance criteria [8, 7].

The Outputs

After the annealing phase is completed, samples of the Θ parameters, the amplitudes and the probabilities are saved. These samples are used to compute mean and standard deviation parameter estimates and are used to produce histograms to visually represent the posterior probabilities. Figure 1 is a typical example of some of the outputs from one of the Bayesian analysis parameter estimation application. In this example a biexponential data set containing a constant offset was loaded and analyzed using the given number of exponentials package. Panel **a**, shows these biexponential data as the open circles. The solid line is a biexponential model generated from the parameters that had maximum joint marginal posterior probability. The residuals, the difference between the data and the model, are shown as the asterisks. Panel **b**, shows the 2500 McMC samples of one of the exponential decay rate constants. Panel **c**, is a smoothed histogram of these samples. This histogram serves as a visual representation of the posterior probability for this parameter and it is not used in any other way. Because this histogram is meant to represent a probability, it sums to one over the discrete bins. Experience with this histogramming technique, indicates that it does a reasonable job of representing the McMC samples and it is easy to compute. However, because it is an ad hoc technique, there is no way to tell how well the histogram represents the posterior probability. Panel **d**, contains

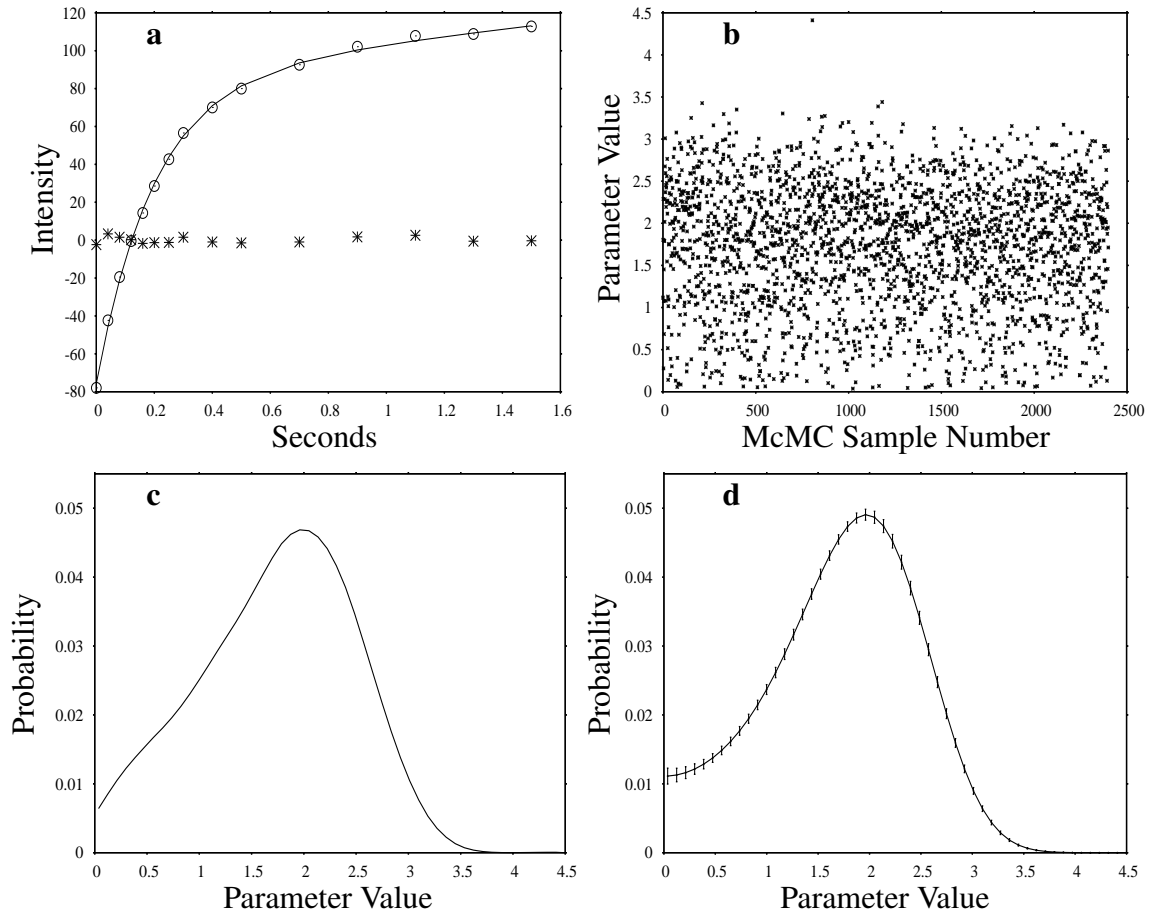


FIGURE 1. Panel **a** contains the data (open circles), a model generated from the parameters that maximized the posterior probability (solid line) and the residuals (asterisks). Panel **b** is the output McMC samples. Panel **c** is a binned smoothed histogram of the McMC samples. This histogram is the one normally generated by the server software. Panel **d** is a Maximum Entropy moment distribution with error bars, where the number of moments and the Lagrange multipliers were inferred using Bayesian probability theory.

a maximum entropy moment distribution that was inferred using Bayesian probability theory. In this Bayesian calculation, the likelihood is assigned using a maximum entropy moment distribution and the parameters are the number of moments and the associated Lagrange multipliers. The data are the output samples from the original McMC simulation, panel **b**. The output from this Bayesian calculation is a histogram with error bars. However, while this maximum entropy moment distribution is much more informative about the samples, it is not used as the default histogram because the Bayesian calculations needed to generate this distribution are usually more computationally intensive than the original parameter estimation problem.

The Applications

There are too many packages implemented within this software to discuss all of them. Here we will describe a few of these packages and the full list can be found online.

- The exponential package estimates the decay rate constants and amplitudes of signals known to be decaying exponentially. It does this when the number of the exponentials are known or unknown. The input to this package is one or more ASCII files. These files can be derived from spectra, images or loaded directly into the analysis. When multiple data sets are loaded, the package looks for exponentials having the same decay rate constants in each data set, but with amplitudes that may differ in the various data sets. Outputs from this package include the posterior probabilities for the decay rate constants, the amplitudes and the noise standard deviations. Additionally, there are listings of the mean and standard deviations of MCMC samples, plots to illustrate how well the model fits the data and plots used to illustrate whether the simulations are in equilibrium. Figure 1 is a typical example of the outputs from the given number of exponentials package. When the presence of a constant or the number of exponentials is unknown, the unknown model can be used to determine the posterior probability for the number of exponentials and whether a constant is or is not present in the data. For more on the exponential problem and how these calculations are implemented, see [4, 5, 6].
- The Enter ASCII Model package allows the user to select a Fortran or C routine that evaluates a model signal and then use Bayesian Probability theory to analyze data using that model signal. The selected model signal can be one of about 30 predefined signals or it can be a user defined model signal. An MCMC simulation is used to approximate the joint posterior probability for the parameters appearing in the model signal. When the package is run, the data and the Fortran or C model are sent to the server. The server compiles this model on the fly and creates a dynamic load library. Consequently to write user supplied signal models, the server must have access to either Fortran or C. Both Fortran and C compilers are available as freeware on both Linux PCs and Sun Solaris systems. Finally, the model signals used by the Enter ASCII Model package are the same model signals used in the analysis of images.
- The problem of estimating the frequencies, amplitudes, phases and decay rate constants of a series of complex sinusoids is extremely important, especially in an NMR laboratory. The Bayes Analyze package is a time domain frequency estimation package that is fully capable of estimating the resonance parameters in complex NMR data. NMR spectroscopic data, called free induction decays, are usually well model as a sum of exponentially decaying sinusoids having the same phases, but having differing amplitudes, frequencies and decay rate constants. This package can analyze single free induction decays, or it can analyze multiple free induction decays and look for frequencies common to these free induction decays. Input to this package can come from different sources and appropriate data conversions are carried out when the data are loaded. For more on frequency estimation, see [9, 10, 11, 12].
- The Analyze Image Pixel package uses the Fortran or C model signals mentioned

earlier and applies these models to an image on a pixel by pixel basis. The outputs from this package are maps, images, of the various parameters. For example a common magnetic resonance imaging experiment, called inversion recovery, generates a set of images where the time course from one pixel is a signal that decays exponentially to a constant. The outputs from the analysis of an inversion recovery experiment would include a map of the initial and final amplitudes, and a map of the exponential decay rate constant. As with the Enter ASCII Model package to make full use of this package the server must have access to either Fortran or C.

- Finally, there are two packages concerned with producing absorption mode images. In NMR spectroscopic applications the Fourier transform of a free induction decay is usually displayed in absorption mode, the real part of the discrete Fourier transform after applying a zero and first order phase. The reasons for displaying the spectrum in this fashion are numerous, but one of the the main reasons is that it preserves the sign of the a resonances amplitude. In magnetic resonance imaging, absorption mode images are almost never displayed because of the difficulty in determining the zero and first order phases parameters. The image phasing packages produces absorption mode images as their output. There are two of these packages: one that assumes the image can be phased by three parameters, two first order phases and one zero order phase. The second package, assumes the phases vary in a nonlinear fashion. The output from both packages are a set of phased images that can serve as input to the image processing packages. For more on how to phase magnetic resonance images, see [13, 14].

In this section, five of the approximately 20 packages in the Bayesian analysis software have been described. The remaining packages solve such problems as the Behrens-Fisher problem [15, 16, 17], errors-in-variables [18, 19], magnetization transfer problems [20, 21, 22] which are also known as two-site exchange problems, as well as many others.

THE BAYESIAN ANALYSIS INTERFACE

When the Bayesian analysis software is installed or updated, it comes with the most recent version of both the server and the client software. The client software is written in Java and, consequently, runs on any platform having Java 5.0 or higher; it has been verified that the interface runs on Sun, Linux, Windows and Apple.

The interface as well as the installation kit run using the Java web start program. This program, “javaws,” reads a set of files from an HTTP server that tells javaws how to download and install the software. Whenever the software is run again, javaws will check for updates and, if available, download and install the updated software. By using javaws it is possible to distribute and maintain the client interface on a host of different architectures all of which are keep up-to-date automatically.

To run the client software interface, log onto the client machine and run the following command:

```
javaws http://your.server.org:port/Bayes/launch.jnlp
```


where “your.server.org” should be replaced by the name of your the Bayesian analysis server and “port” by the port number used in the installation procedure, see below. This startup command can be initiated on a command line, as a short cut or the HTTP address can be entered into some web browsers.

When the Java client interface first starts, it displays a default panel. This panel has three main functions: First, it can restore the previous analysis. Second, it can change to a new work area and restore that analysis, and third, it allows one to select a new package. Consequently, the user can configure one work area to run something like an exponential analysis. He can then submit that analysis and change to a new work area. Finally, when the user returns to the original work area, that analysis is restored to its previous status and the user can pick up where they left off.

The panel shown in Fig. 2 is a typical client interface panel. The panel consists of three general sections, the menu bar on the top border is general and exist on all panels, including the default panel. Below these are the widgets used to configure and control a package. Some of these widgets are general and appear on all packages, for example the widgets titled “Submit Job to Server” and “Server” are general. These widgets are responsible for selecting a server, sending a job to a server and getting the results from the server. The widgets labeled “Load and Build Model” are package specific. In this case, they allow the user to select a predefined model from either a “User” or “System” directory. Finally, at the bottom of this interface is a display/graphics area. This area is used to view various types of data. The type of data being viewed, is controlled by the tab widgets along the top of the plotting area. In Fig. 2, the “Plot Results” tab has been activated. When activated the “Plot Results” tab displays a panel appropriate for viewing output plots for this package; other tabs display other panels appropriate for viewing other types of data. For example, the “Text Results” tab brings up a viewer appropriate for looking at text files.

INSTALLING THE BAYESIAN ANALYSIS SOFTWARE

In the previous sections we have described some of the general features of this software. The software consists of a client and server software system. The client is Java based and runs on a local user machine. The major requirement to run the client software is that one have Java 5.0 or higher, javaws and at least 1Gb of memory. The servers software requires a more substantial machine. We use Dell multicore Linux PCs running CentOS 4.7 and these machines have at least 8 processors. However, the software was developed on a Sun Solaris system having only 2 processors and such machines can run most applications. However, machines with more than 2 processors are highly desirable for many applications. Installation kits are available for both 32 and 64 bit Linux CentOS 4.7 machines and for Sun Solaris 9.0 or higher.

The first step in the installation procedure is to choose a machine that is to act as the Bayesian analysis server. For purpose of this discussion, this machine will be called “your.server.org” with the understanding that references to “your.server.org” should be replaced by the appropriate host name.

The second step is to create a user account on “your.server.org.” This user will run the Bayesian analysis software as well as the web server that the clients communicate with.

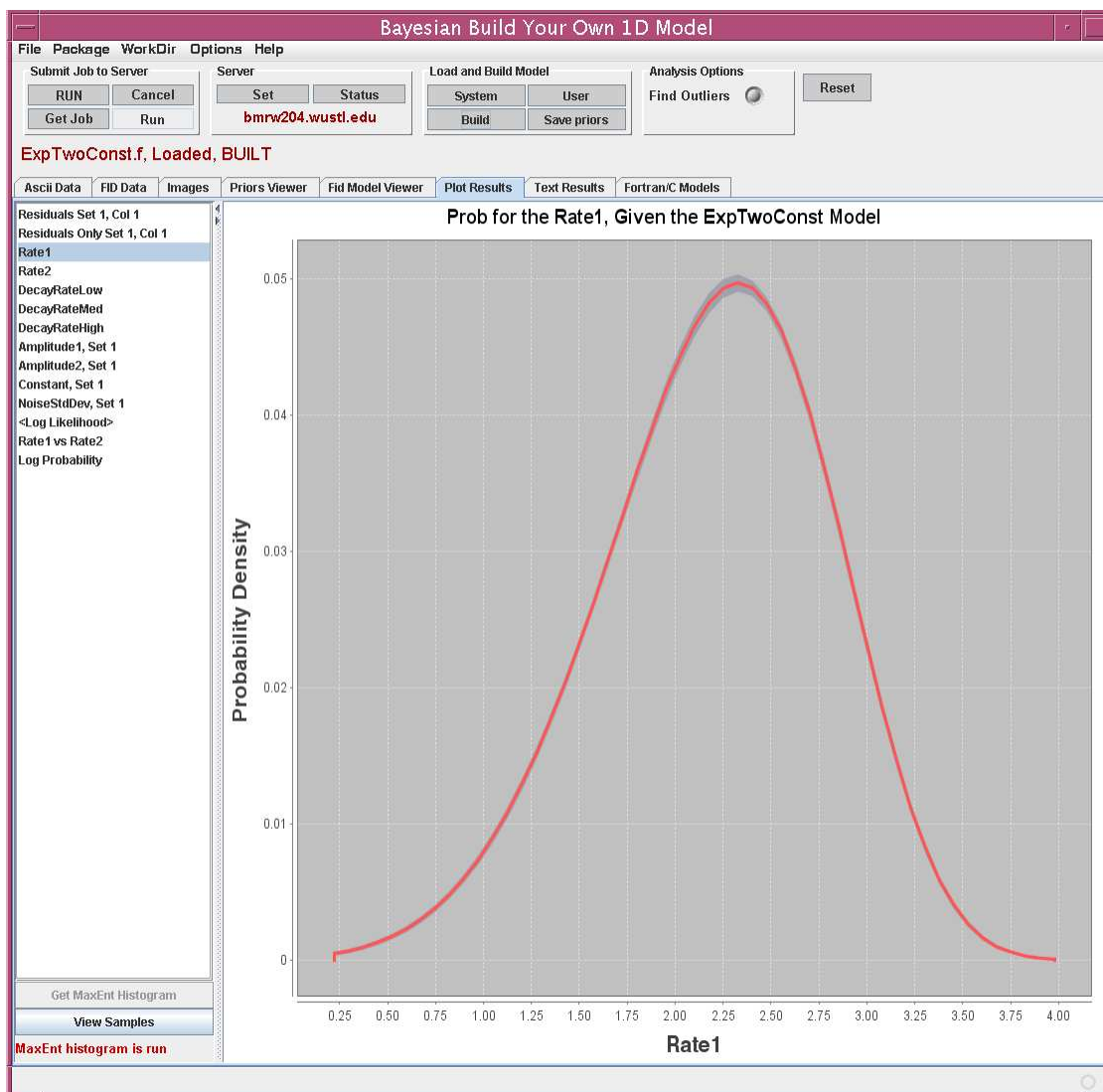
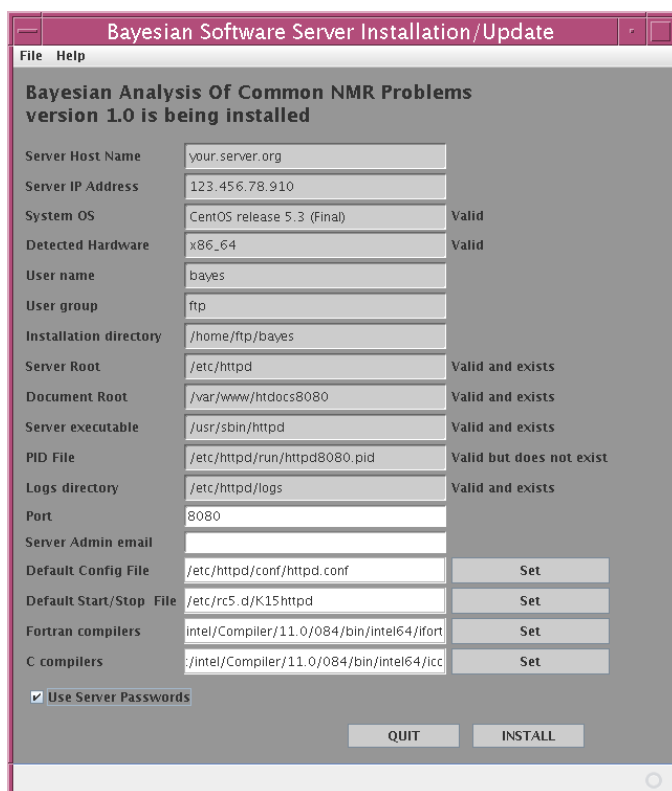


FIGURE 2. This display shows the interface to a typical package, in this case the Enter ASCII model package. The menu bar on the top border takes care of loading data, changing the analysis package, changing which analysis is being viewed, and setting options. The widgets in the box labeled “Submit Job to Server” handle sending and retrieving jobs from the server. The widgets labeled “Server” allow one to select which server is to be used and they allow one to get a snap shot of the activity on this server. The “Load and Build Model” widgets are package specific and are used by the Enter ASCII model package to load and build models written in either Fortran or C. The entire bottom part of this panel is a display area used to display various types of information. The current panel displays the graphic outputs from an analysis and is activated by the “Plot Results” tab.

The default user name is “bayer,” but any valid Unix user name may be used. In the discussions that follows, the default name “bayer” will be used with the understanding that “bayer” should be replaced by your user name. The “bayer” user and its home directory should be dedicated to running the Bayesian analysis software and it should be on a disk directly mounted on “your.server.org;” it should not be remote mounted.



When the Bayesian analysis server installation software is activated, it looks over the hardware and software environment and then reports what it finds in this window. The items at the top are informational and cannot be changed. The items at the bottom can be changed or set. If you hold the cursor over a field, including the informational items, a tool tip will appear that tells you more about what is needed in any given field. If any of the informational items are incorrect, abort the installation immediately and take corrective action.

FIGURE 3. The Server Installation/Update Window

Remote mounts will work, but they will slow things down because of the extra network IO.

Third, log into the “bayes” user on “your.server.org.” Configure enough of a path to be able to find javaws and issue:

```
javaws http://bayes.wustl.edu/ServerSoftware/launch.jnlp
```

This Java application will look over the hardware and software environment and then display the panel shown in Fig. 3. The top part of this panel is not editable, while the bottom part can be set and or edited. Some of the noneditable items are just to verify that your operating system and hardware have been correctly identified; others are things that were configured in your operating system, like the host name and the IP address. The remaining items are the default locations for various Apache server files. The default locations for these files are not changed, but the default names were changed by appending the port number to the names. Changing the port number will change these names. If you wish to use nonstandard locations or names, you will have to make these changes manually to the generated files after this installation procedure finishes. There are seven items that can be set by the user: the port, the server administrator email address, the name of the Apache server prototype configuration file, the name of the prototype script that starts or stops the Apache server, the location of the Fortran and C compilers and whether or not passwords are to be implemented.

Web servers read messages from ports. The default port for an HTTP web server is port 80 and this port is often available to any web users. When designing this software, it was decided to restrict access to the Bayesian analysis server in two ways: the server uses passwords and a nonstandard port. Both of these configuration options make it much harder for a nonauthorized user to gain access to the server. Whether or not passwords are activated is a configuration option. However, use of a nonstandard port is not. The default port used by the Bayesian analysis server is port 8080. However, the Apache server can be configured to listen to any port, except port 80; this can be useful if you wish to monitor the CPU usage for various projects or users. However, if you are running only a single Bayesian analysis server on “your.server.org,” there is no need to change the default port.

The default configuration file is the name of the default Apache server configuration file on your server. The contents of this file is hardware and software dependent. This file is used a prototype and its contents are read by the installation kit and an updated version of this prototype is written into the “bayes” user’s “system” directory. Similarly the Start/Stop is the default file used to start and stop the Apache server whenever Unix starts or stops. Like the configuration file, this file is read and modified and the resulting updated file is placed in the “system” directory. When the installation procedure finishes, these files must be copied by the root user to the appropriate locations for your server. The installation kit will indicate where to place these files when the installation is completed.

The Fortran and C compilers should be set to the fully qualified name of the Fortran and C compilers that the Bayesian analysis server are to use. Use the “Set” button to navigate to and set the name of these compilers. If you do not have any compilers, leave these set to “None.” However, without the Fortran compilers several of the most useful packages will have limited usefulness. On Linux machines the Intel compilers are only few hundred dollars and you should consider installing them. If you cannot afford these compilers, the Sun freeware compilers are available for both Linux and Sun systems and, while they do not handle all Fortran 95 features, they handle enough to make them useful and testing here at Washington University indicates that these compilers work with the Bayesian analysis software.

The server password check box is used to indicate that passwords are required to send jobs to “your.server.org.” These passwords are used to ensure that only authorized users can send jobs to this server. If you check the password box, your server administrator must set a password for each authorized user using the “htpasswd” command.

SUMMARY AND CONCLUSIONS

The Bayesian analysis software described in this paper is available online from bayes.wustl.edu. The software is a client/server based software package that solves many problems in NMR and the sciences in general. The software can be installed on Linux PCs running CentOS 4.7 or higher and Sun systems running Solaris 9 or higher. The client software is Java based and will run on Linux, Apple, Sun and Windows systems having Java 5.0 or higher. The installation software will automatically install both the client and server software. Client software acts as a user friendly interface

to gather data, setup an analysis, send the analysis to the server, retrieve the analysis from the server and finally display the results of the analysis. The server software accepts requests from the clients and the runs the appropriate applications. For Bayesian analysis servers having Fortran or C, user defined models can be written and then Bayesian Probability theory can be used to analyze data with these models. Finally, because the interface and the installation procedure are written in Java, installing and maintaining the software is not much more complicated than running the appropriate javaws commands.

REFERENCES

1. Sivia, Devinder and John Skilling, "Data Analysis: A Bayesian Tutorial," Oxford University press, USA, 2006, ISBN: 978-0198568322.
2. E. T. Jaynes, "Probability Theory—The Logic of Science," G. L. Bretthorst, Ed., Cambridge University Press, Cambridge, UK, 2003, ISBN: 978-0521592710.
3. Bayes, Rev. T. (1763), "An Essay towards solving a Problem in the Doctrine of Chances," *Philos. Trans. R. Soc. London* **53**, 370; reprinted in *Biometrika* **45**, 293 (1958), and "Facsimiles of Two Papers by Bayes," with commentary by W. Edwards Deming Hafner, New York, 1963.
4. G. Larry Bretthorst, William C. Hutton, Joel R. Garbow, Joseph J.H. Ackerman, "Exponential parameter estimation (in NMR) using Bayesian probability theory," *Concepts in Magnetic Resonance*, **27A**, Issue 2, pp. 55-63, 2005.
5. G. Larry Bretthorst, William C. Hutton, Joel R. Garbow, Joseph J.H. Ackerman, "Exponential model selection (in NMR) using Bayesian probability theory," *Concepts in Magnetic Resonance*, **27A**, Issue 2, pp. 64-72, 2005.
6. G. Larry Bretthorst, "How accurately can parameters from exponential models be estimated? A Bayesian view," *Concepts in Magnetic Resonance*, **27A**, Issue 2, pp. 73-83, 2005.
7. Hastings, W. K., "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, **57**, pp. 97-109, 1970.
8. Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., and Teller, E., "Equation of state calculations by fast computing machines," *Journal of Chemical Physics*, **21**, pp. 1087-1092, 1953.
9. Bretthorst, G. Larry, "Bayesian Analysis I, Parameter Estimation" *J. Magn. Reson.*, **88**, pp. 533-551, 1990.
10. Bretthorst, G. Larry, "Bayesian Analysis II, Model Selection" *J. Magn. Reson.*, **88**, pp. 552-570, 1990.
11. Bretthorst, G. Larry, "Bayesian Analysis III, Examples Relevant to NMR" *J. Magn. Reson.*, **88**, pp. 571-595, 1990.
12. Bretthorst, G. Larry, "Bayesian Spectrum Analysis and Parameter Estimation," in *Lecture Notes in Statistics*, **48**, J. Berger, S. Fienberg, J. Gani, K. Krickenberg, and B. Singer (eds), Springer-Verlag, New York, New York 1988, ISBN: 978-0387968711.
13. Bretthorst, G. Larry, "Automatic phasing of MR Images. Part I: Linearly varying phase," *J. Mag. Res.*, **191**, 184-192, 2008.
14. Bretthorst, G. Larry, "Automatic phasing of MR Images. Part II: Voxel-wise phase estimation," *J. Mag. Res.*, **191**, 192-201, 2008.
15. Behrens, W. V., "Ein Beitrag zur Fehlerberechnung bei wenigen Beobachtungen," *Landwirtschaftliche Jahrbücher*, **68**, pp. 807-837, 1929.
16. Fisher, R. A., "The fiducial argument in statistical inference," *Ann. Eugenics*, **6**, pp. 391-398, 1935.
17. Bretthorst, G. Larry, "On The Difference In Means," in *Physics and Probability Essays in honor of Edwin T. Jaynes*, W. T. Grandy and P. W. Milonni Eds., Cambridge University Press, England, 1993.
18. Jaynes, E. T., 'Straight Line Fitting - A Bayesian Solution,' an unpublished and incomplete manuscript available at <http://bayes.wustl.edu/etj/articles/leapz.pdf> 1991.
19. Gull, Stephen F., "Bayesian Data Analysis: Straight-line Fitting," in *Maximum Entropy and Bayesian Methods*, J. Skilling (ed), Kluwer Academic Publishers, Dordrecht the Netherlands, pp. 511-518, 1989.

20. d'Avignon, D. André, G. Larry Bretthorst, Marilyn Emerson Holtzer, and Alfred Holtzer, "Site-Specific Thermodynamics and Kinetics of a Coiled-Coil Transition by Spin Inversion Transfer NMR," *Biophysical Journal*, **74**, pp. 3190-3197, 1998.
21. d'Avignon, D. André, G. Larry Bretthorst, Marilyn Emerson Holtzer, and Alfred Holtzer, "Thermodynamics and Kinetics of a Folded-Folded' Transition at Valine-9 of a GCN4-Like Leucine Zipper," *Biophysical Journal*, **76**, pp. 2752-2759, 1999.
22. Holtzer, Marilyn Emerson, G. Larry Bretthorst, D. André d'Avignon, Ruth Hogue Angelette, Lisa Mints, and Alfred Holtzer, "Temperature Dependence of the Folding and Unfolding Kinetics of the GCN4 Leucine Zipper via ^{13}C alpha-NMR," *Biophysical Journal*, **80**, pp. 939-951, 2001.