BayesDiagnosticTests: A program for Estimation of Disease Prevalence and the Characteristics of One, Two or Three Diagnostic Tests

(Version 3.11.1, September 2017)

1. Introduction

BayesDiagnosticTests is a software package to calculate marginal Bayesian posterior distributions via Gibbs sampling when data are available from one, two or three diagnostic tests. It is an implementation of the methods presented in

Bayesian estimation of disease prevalence and the parameters of diagnostic tests in the absence of a gold standard
Joseph L, Gyorkos T, Coupal L
Americal Journal of Epidemiology 1995; 141(3), pp. 263-272 and

Inferences for likelihood ratios in the absence of a gold standard
Joseph L, Gyorkos T
Medical Decision Making 1996; 16(4), pp. 412-417.

These papers should be read before this program is used.

Summaries of the marginal posterior densities for the prevalence of a disease and the properties of the diagnostic tests (sensitivity, specificity, positive and negative predictive values and positive and negative likelihood ratios) in the absence of a gold standard diagnostic test will be output.

BayesDiagnosticTests requires that the free software packages WinBUGS, Perl and Microsoft Windows Script be installed (see InstallInstructions.html). This program performs the same calculations as the R/S-Plus functions tt1, tt2 and tt3 that were previously available for the same problem, but adds a user-friendly graphical interface. Its use is straightforward, as shown below.

2. How to use **BayesDiagnosticTests**

The inputs required for this program are:

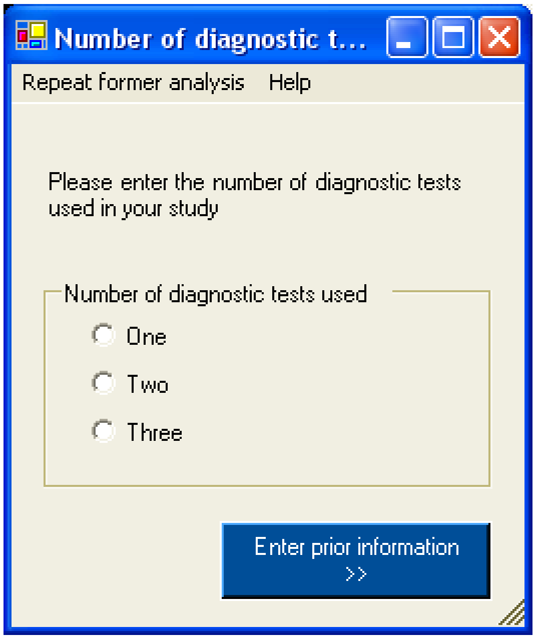
- Diagnostic test data from either one, two or three diagnostic tests, each applied to a group of subjects;

- Prior distributions for the disease prevalence and the sensitivity and specificity of each diagnostic test, in the form of a beta distribution parameters.

Once it is installed, open **BayesDiagnosticTests** by clicking on the Start button on your desktop and browse to **BayesDiagnosticTests** in the Programs menu.

The initial window is used to input the number of diagnostic tests that were used in your study.

Indicate the number of tests used by clicking on the appropriate radio button, and then click the *Enter prior information* >> button to proceed to prior information entry forms.



Number of diagnostic t...

Repeat former analysis Help

Please enter the number of diagnostic tests used in your study

Number of diagnostic tests used

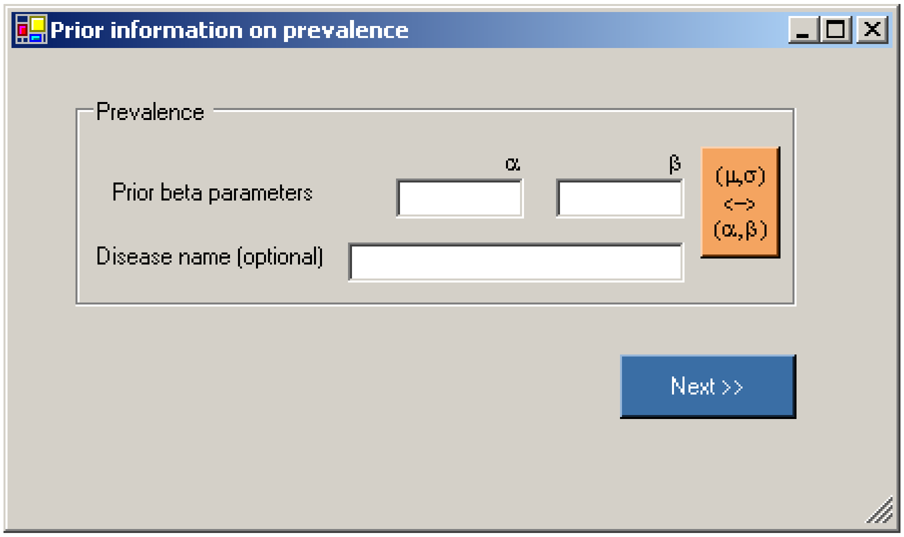
○ One

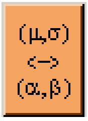
○ Two

○ Three

Enter prior information >>

The next two forms will be used to enter your prior information on the prevalence of the disease and diagnostic test(s)' sensitivity and specificity. Each prior is given a beta distribution with parameters (α,β), such that prior mean and variance are $\alpha/(\alpha+\beta)$ and $\alpha\beta/(\alpha+\beta)^2(\alpha+\beta+1)$.



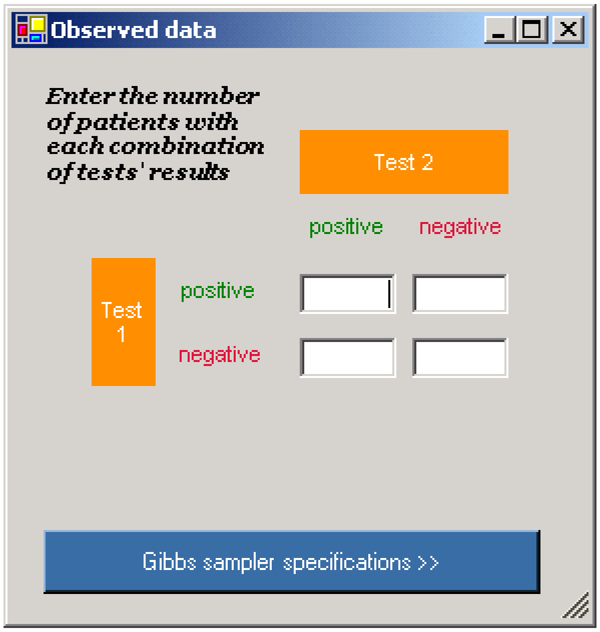


The orange button (μ,σ) <–> (α,β) allows you to specify your prior distributions in terms of prior moments (μ,σ) rather than in terms of (α,β).

Entering a label in *Disease name* or *Test label* cells will make the corresponding prior entered only one click away the next time you run this program.

After prior parameters values for the prevalence have been entered, similar windows will appear for the sensitivity and specificity for each test use.

The next form is used for data entry. The exact form displayed will depend on the number of tests selected above, but will resemble a crosstable, to be filled with the number of subjects observed in each cell. Click the *Gibbs sampler specifications >>* button when done.

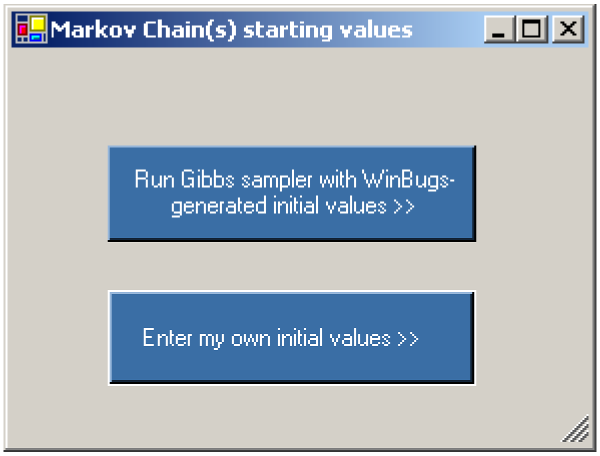


Enter the number of patients with each combination of tests' results

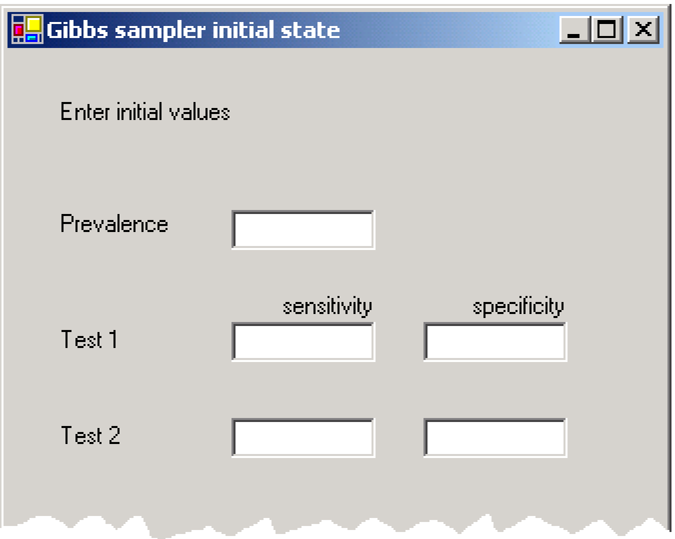
Gibbs sampler specifications form will allow you to control for the number of burn-in iterations and the number of monitored iterations. Burn-in iterations are iterations that are ignored when the summary statistics are calculated and are used to allow the Markov chain to converge; the history plots in the *.odc* file can be examined to assess convergence (of course, more formal convergence checks can be done, but this is beyond the scope of this document).

In general, this program should be run several times with different starting values to check for convergence. In our experience, this is especially important when the number of subjects is large or when two or three tests are used (as there might then be multiple local peaks in the joint posterior density). Several chains can be run simultaneously, each with different starting values by choosing the number of runs in the menu at the right of the form (not shown).

The next form allows you to run Gibbs sampler with WinBUGS-generated initial values or to choose to input your own set(s) of initial values.


Markov Chain(s) starting values
Run Gibbs sampler with WinBugs-generated initial values >>
Enter my own initial values >>

If you did not choose to use WinBUGS-generated initial values, the next form allows you to pick your own initial values for every parameter for each of the chains you selected. Fill each of the cells and click the blue button to either enter next chains' initial values or proceed to Gibbs sampling.



IMPORTANT: At the time that **BayesDiagnosticTests** was first released (October 2004), it was not possible to set your own seed for the random numbers generator in WinBUGS scripts (used by **BayesDiagnosticTests**): thus, answers will be identical from run to run if you always select the same initial values (whether you use your own or WinBUGS-generated initial values).

Depending on number of tests performed, number of patients observed and number of Gibbs iterations run, **BayesDiagnosticTests** can be slow and reduce your ability to effectively multitask on your PC (although it is much less likely to happen with version 2.0 and later). If you find the program is taking too much of your CPU, **BayesDiagnosticTests** can also be run in low-priority mode. In this mode, **BayesDiagnosticTests** will use all of the CPU power available, but will cede power to other programs. While **BayesDiagnosticTests** may run a little slower when in this mode, it will have little to no noticable effect on your running of other programs on your PC. To start **BayesDiagnosticTests** in low-priority mode, follow these steps:

1. In Windows Explorer, browse to **BayesDiagnosticTests**'s home directory.
2. Double-click on BayesDiagnosticTestsLowPriority.bat. This will launch **BayesDiagnosticTests** as usual, but every sub-program called will automatically be run with low priority.

2.1 Monte Carlo Markov Chains (MCMC) using WinBUGS interactively

You can also choose to run the analysis performed by **BayesDiagnosticTests** on your own; make a copy of the relevant file of src\tt1.txt, src\tt2.txt or src\tt3.txt, depending on the number of diagnostic tests used (make sure to work on copies of the files above, leaving the files in src\ subdirectory untouched) and load it in WinBUGS. Full instructions on how to use WinBUGS interactively can be found in WinBUGS help menu. Note that this method avoids the above-mentioned problem concerning setting seeds, as when WinBUGS is run interactively, the seed can be manually changed for each run.

In addition to cell counts, the data list (to be loaded in WinBUGS before compiling) must include values for alpha.prevalence and beta.prevalence (the prior Beta distribution parameters for prevalence) as well as vectors alpha.sensitivity, beta.sensitivity, alpha.specificity and beta.specificity, giving the α and β parameters of prior distributions for all tests' sensitivities and specificities (e.g., if sensitivity ~ beta (1, 2) and specificity ~ beta(3, 4) for test 1 while sensitivity ~ beta(5, 6) and specificity ~ beta(7, 8) for test 2, then the data list should include the following vectors: alpha.sensitivity = c(1, 5), beta.sensitivity = c(2, 6), alpha.specificity = c(3, 7), beta.specificity = c(4, 8)).

Cell counts are entered by the mean of a vector of length $2_{\text{number of tests}}$ to be called *count*: if only one test is used, count will consist of two numbers, that is, $count = c(count_{0}, count_{1})$; if two tests are used, $count = c(count_{00}, count_{01}, count_{10}, count_{11})$ while $count = c(count_{000}, count_{001}, count_{010}, count_{011}, count_{100}, count_{101}, count_{110}, count_{111})$ when three tests are used, where the 0/1 indices indicate whether test 1, 2 and 3 (in order) gave negative (0) or positive (1) results; $count_{011}$, for example, is the number of subjects with a negative result for test 1 and positive results for both tests 2 and 3.

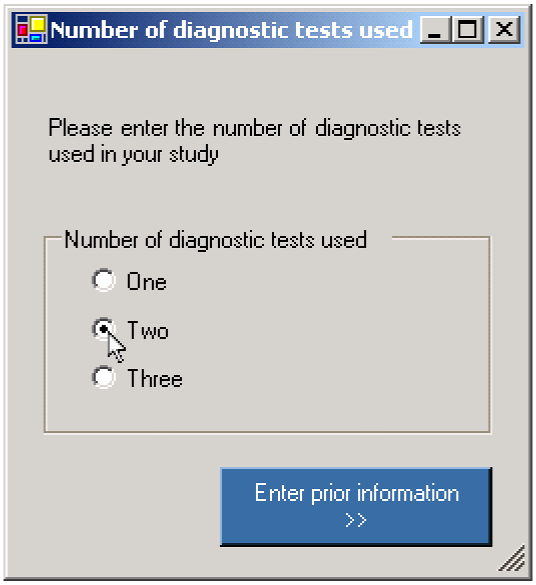
As for initial values, starting values for test sensitivities and specificities must be given in two vectors (called sensitivity and specificity respectively) while prevalence must be given its initial value in a scalar called prevalence.

3. An Example of Running **BayesDiagnosticTests**

As an illustration of this new software package, we will walk you through the example presented in Joseph L, Gyorkos T, Coupal L (1995).

If it is the first time you run this program, you will be prompted for your WinBUGS location shortly after opening **BayesDiagnosticTests**; make sure to indicate the most recent WinBUGS executable file found on your system (WinBUGS version 1.4 or later).

Click the radio button *Two* to indicate that this study involved two diagnostic tests and then click *Enter prior information >>*.



Please enter the number of diagnostic tests used in your study

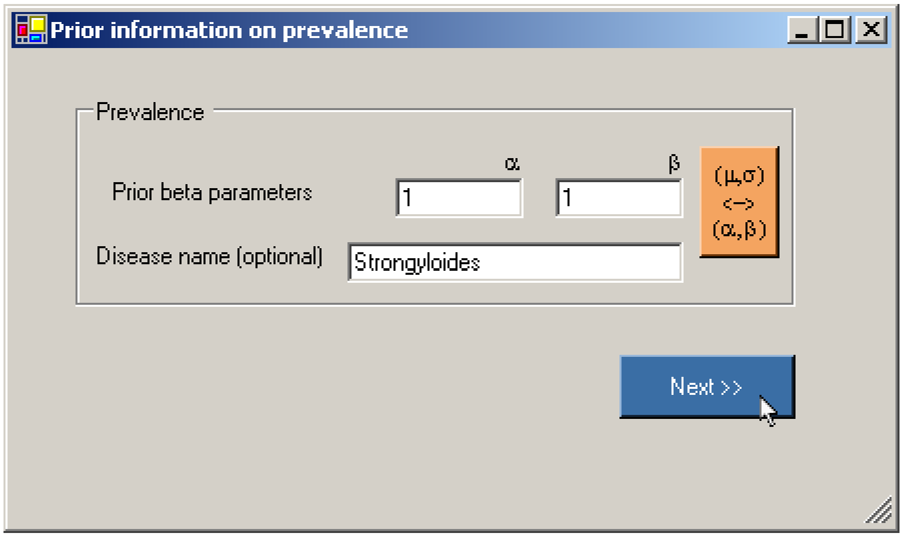
Number of diagnostic tests used

○ One

● Two

○ Three

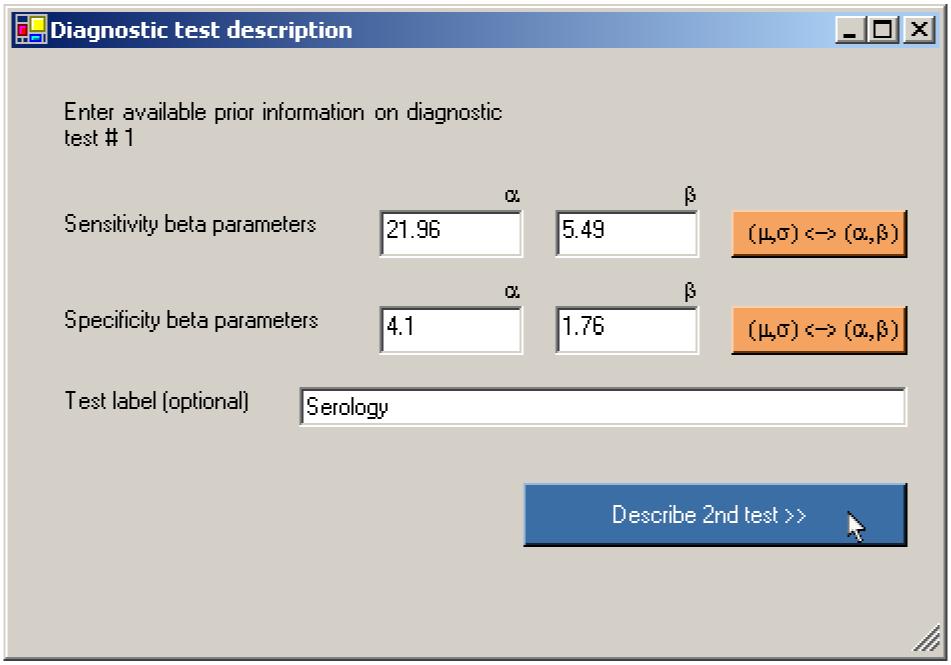
Since little prior information was available on the prevalence of the disease (Strongyloides), indicate a non-informative prior (α=β=1). Click the *Next>>* button.



Prior beta parameters: α 1, β 1

Disease name (optional): Strongyloides

We had some prior information on the characteristics of the two diagnostic tests; fill the sensitivity and specificity α and β cells with the numbers shown in the two figures below, one diagnostic test at a time.

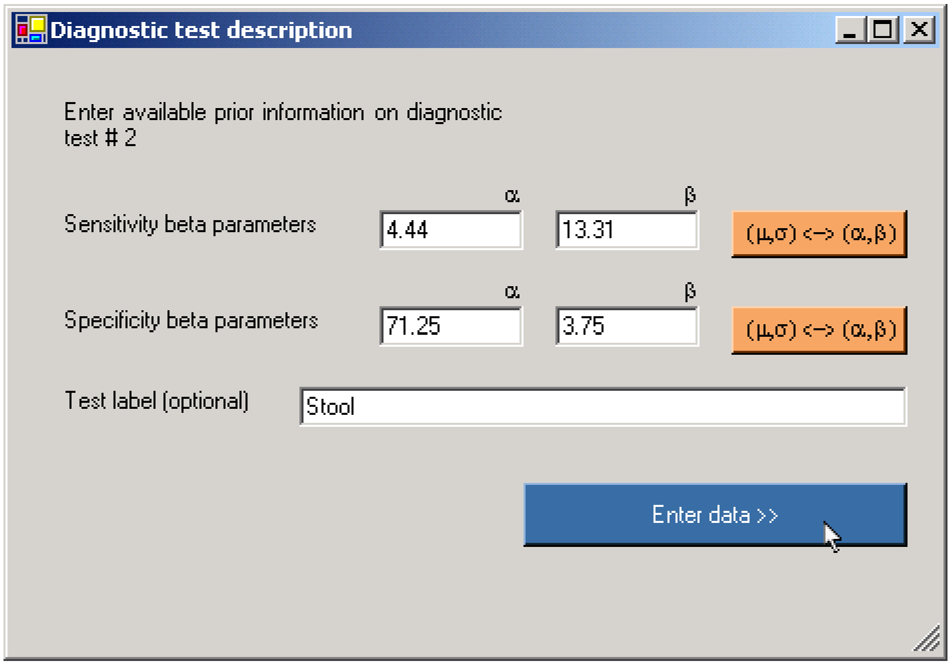


Enter available prior information on diagnostic test # 1

	α	β
Sensitivity beta parameters	21.96	5.49
Specificity beta parameters	4.1	1.76

Test label (optional): Serology

Those using a comma rather than a period as a decimal symbol should continue to use this style, assuming this concords with the setting under **Decimal symbol** in the **Customized Regional Options** form of the **Regional and Language Options** in the **Control Panel**. That is, one may have to enter 21,96 rather than 21.96 in this example.



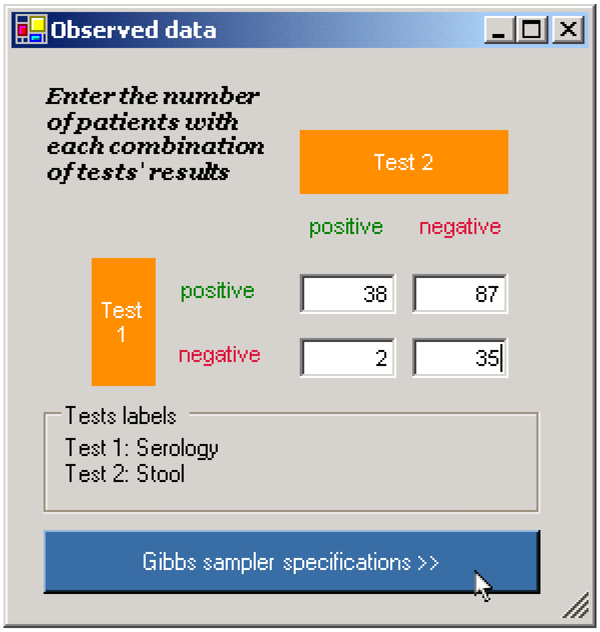
Enter available prior information on diagnostic test # 2

	α	β
Sensitivity beta parameters	4.44	13.31
Specificity beta parameters	71.25	3.75

Test label (optional): Stool

Enter the data in the crosstable cells on the next form as shown in the opposite figure. Note that tests labels entered in earlier forms are reported on the bottom half of the present form; refer to it to make sure you enter the data in the right order.

Click the *Gibbs sampler specifications >>* button.



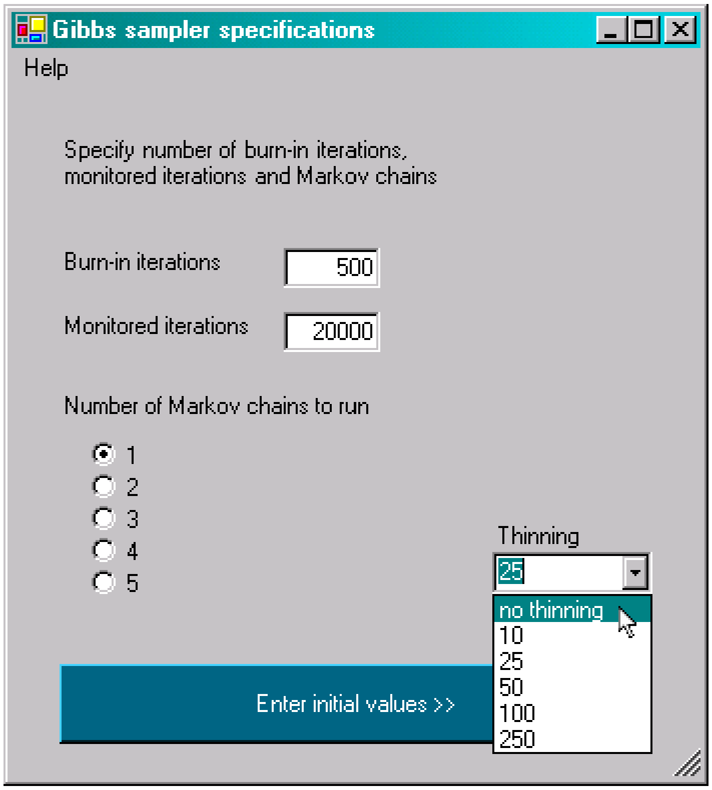
Enter the number of patients with each combination of tests' results

		Test 2	
		positive	negative
Test 1	positive	38	87
	negative	2	35

Tests labels
Test 1: Serology
Test 2: Stool

For this example, we will accept the default number of iterations (burn-in of 500 and monitoring of the next 20,000) and we will run only one Markov Chain. In addition, we will run the MCMC without thinning by selecting the *no thinning* entry in the *Thinning* list. In general, any thinning value of your choice may be entered directly in the box under *Thinning*. The chain thinning option will be especially useful when dealing with large sample sizes (e.g., several thousands of subjects), where the autocorrelation can become a major problem: autocorrelation usually fades out with thinning.
Higher thinning values are better in that regard, but larger thinning values also causes **BayesDiagnosticTests** to have longer running times.

Click the *Enter initial values>>* button.



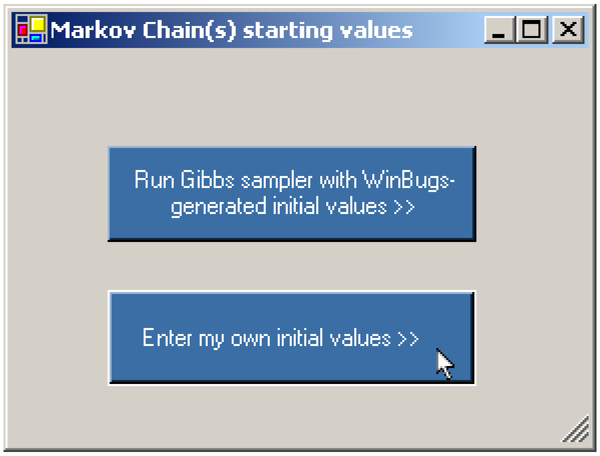
Specify number of burn-in iterations, monitored iterations and Markov chains

Burn-in iterations: 500
Monitored iterations: 20000

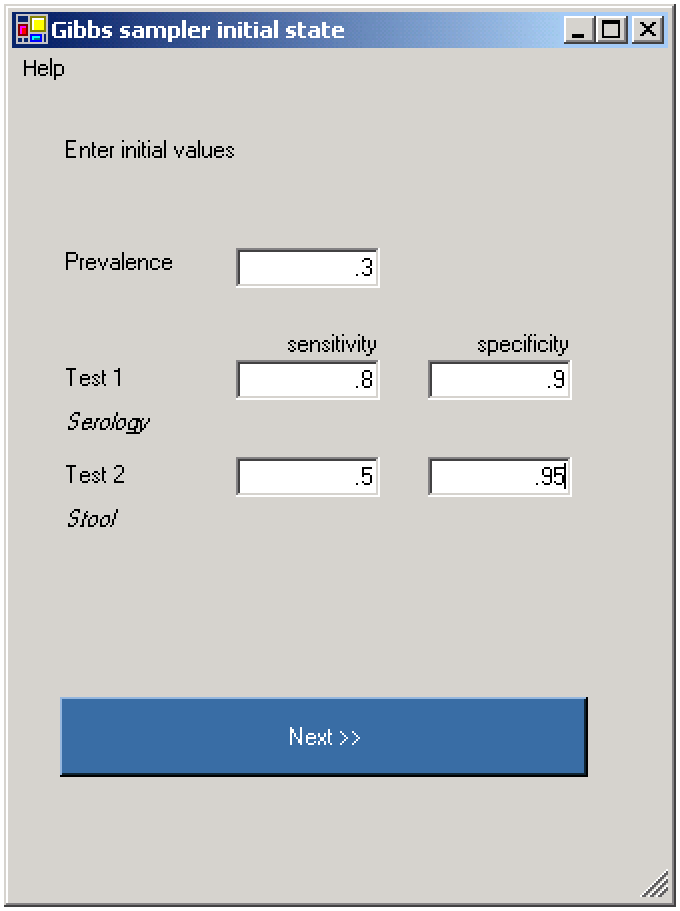
Number of Markov chains to run
● 1
○ 2
○ 3
○ 4
○ 5

Thinning: 25
no thinning
10
25
50
100
250

We will start the chain with our own initial values; click the *Enter my own initial values >>* button.


Markov Chain(s) starting values
Run Gibbs sampler with WinBugs-generated initial values >>
Enter my own initial values >>

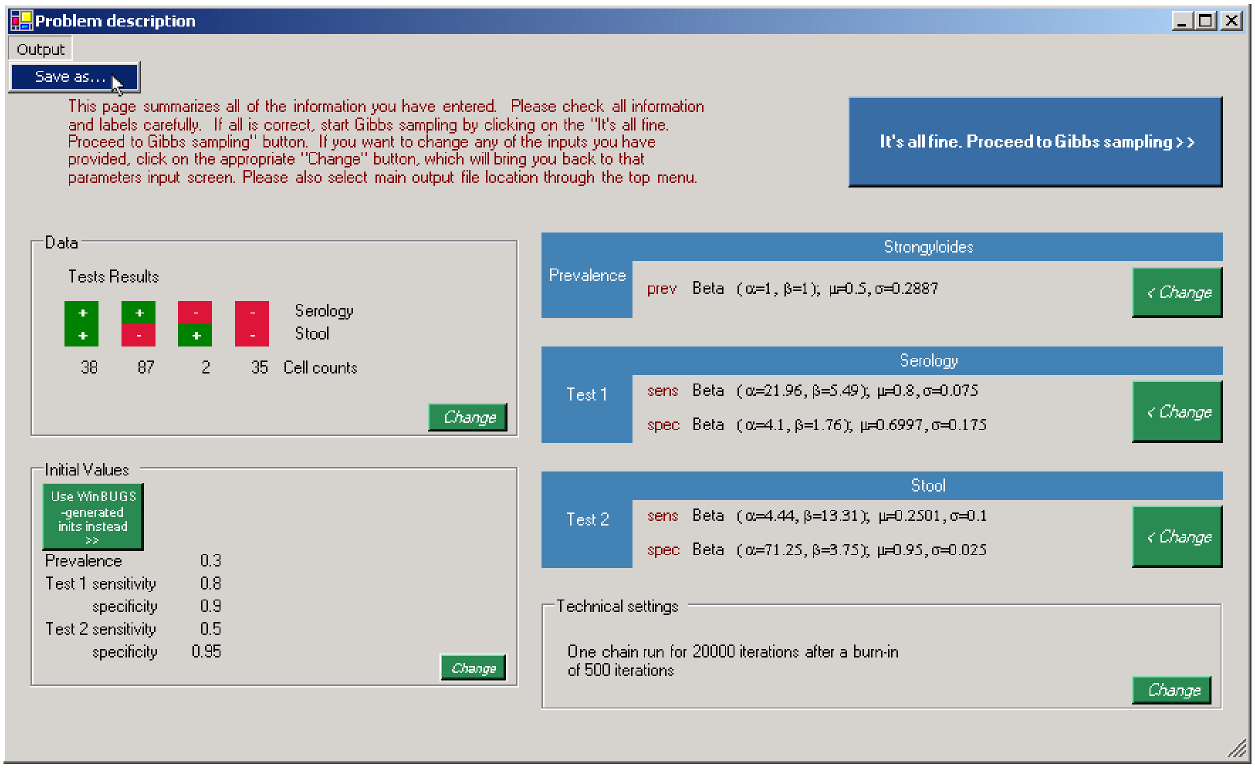
Enter the initial values as shown on the opposite figure and click *Next>>* to proceed.



Prevalence .3

	sensitivity	specificity
Test 1 *Serology*	.8	.9
Test 2 *Stool*	.5	.95

Next form (below) summarizes all of the information you have entered. If you want to change any of the inputs you have provided, click the appropriate *Change* button, which will bring you back to that parameters input screen. This form requires that you select a location for output file: click the *Save as...* item in the *Output* menu and browse to the directory of your choice when the *Save as* window opens; either enter a new file name or click a file listed in the window (which will then be overwritten if you confirm your selection). Specify a file name with the *.odc* extension [WinBUGS output extension], but note that a text file with the same name and a *.txt* extension will also be saved upon completion. The *.odc* file[1] will contain *summary statistics* for all chance nodes as well as iteration by iteration history and estimated density plots. The *.txt* file will contain the *summary statistics* as well as the observed data which were input, the prior parameters used and the initial values provided (unless you accepted the WinBUGS-generated initial values).


Problem description
Output
Save as...
It's all fine. Proceed to Gibbs sampling >>
Change
Use WinBUGS -generated inits instead >>
Change
< Change
< Change
< Change
Change

This page summarizes all of the information you have entered. Please check all information and labels carefully. If all is correct, start Gibbs sampling by clicking on the "It's all fine. Proceed to Gibbs sampling" button. If you want to change any of the inputs you have provided, click on the appropriate "Change" button, which will bring you back to that parameters input screen. Please also select main output file location through the top menu.

Data

Tests Results

+	+	-	-	Serology
+	-	+	-	Stool
38	87	2	35	Cell counts

Initial Values

Prevalence		0.3
Test 1	sensitivity	0.8
	specificity	0.9
Test 2	sensitivity	0.5
	specificity	0.95

Strongyloides

Prevalence: prev Beta (α=1, β=1); μ=0.5, σ=0.2887

Serology

Test 1: sens Beta (α=21.96, β=5.49); μ=0.8, σ=0.075
spec Beta (α=4.1, β=1.76); μ=0.6997, σ=0.175

Stool

Test 2: sens Beta (α=4.44, β=13.31); μ=0.2501, σ=0.1
spec Beta (α=71.25, β=3.75); μ=0.95, σ=0.025

Technical settings

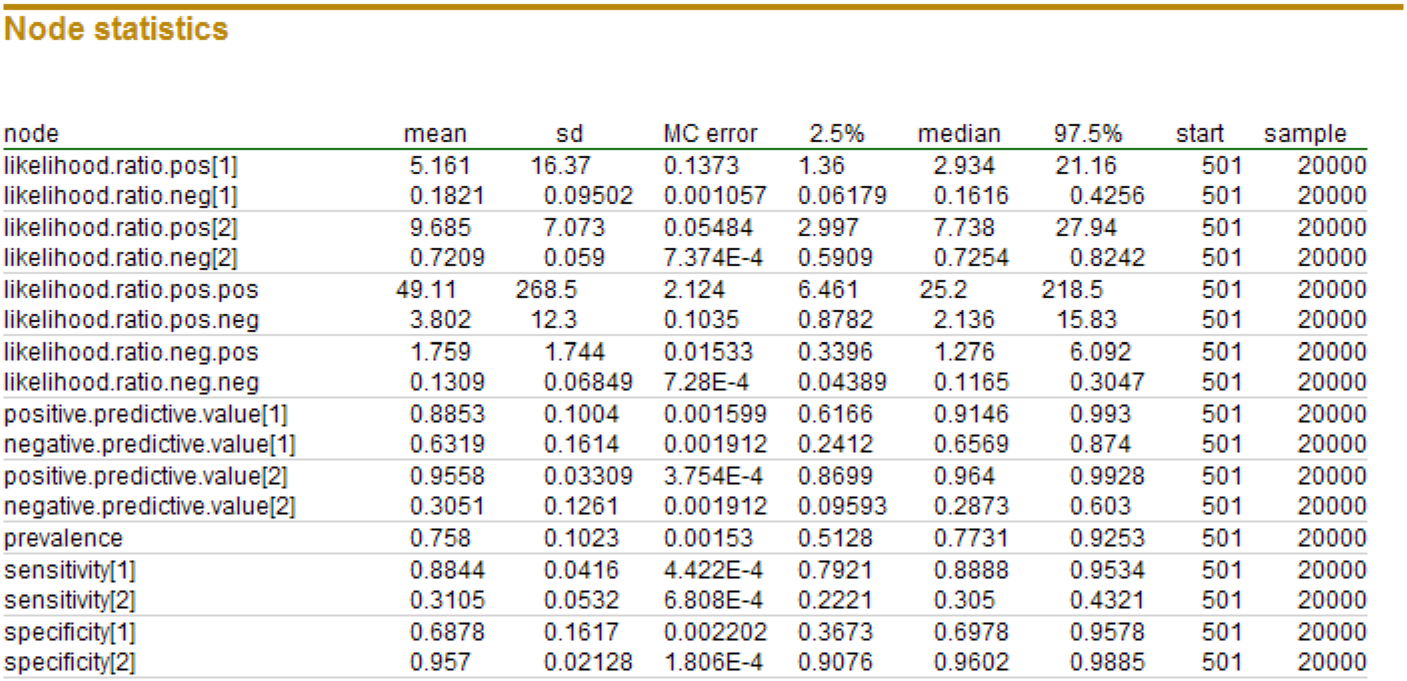
One chain run for 20000 iterations after a burn-in of 500 iterations

Click the *It's all fine. Proceed to Gibbs sampling>>* button when output file location is selected and problem description to your entire satisfaction.

BayesDiagnosticTests will now create and run a WinBUGS script, run WinBUGS itself using this script, and the output will be saved in the files selected earlier. You may want to minimize WinBUGS window while it is running (this example will typically take less than a minute to run, but others can take longer). You can compare your output files to the ones stored in doc\2tests- example-stools-serology.odc and doc\2tests-example-stools-serology.html, which were obtained with exactly the same data, priors and initial values.

[1] When BayesDiagnosticTests crashes or is stopped before completion, a log file with the same name as the expected output (but with a *.log* extension) will be written: refer to it to identify the source of the problem. Note that a copy of this log file will also be written in BayesDiagnosticTests home directory (C:\Users\user name\Documents\Bayesian Software\BayesDiagnosticTests or C:\Documents and Settings\user name\My Documents\Bayesian Software\BayesDiagnosticTests , by default, depending on your platform) under log\BayesDiagnosticTests.txt.

We reproduce below the *Node statistics* obtained in this example; numbers in brackets are the test numbers.



Node statistics

node	mean	sd	MC error	2.5%	median	97.5%	start	sample
likelihood.ratio.pos[1]	5.161	16.37	0.1373	1.36	2.934	21.16	501	20000
likelihood.ratio.neg[1]	0.1821	0.09502	0.001057	0.06179	0.1616	0.4256	501	20000
likelihood.ratio.pos[2]	9.685	7.073	0.05484	2.997	7.738	27.94	501	20000
likelihood.ratio.neg[2]	0.7209	0.059	7.374E-4	0.5909	0.7254	0.8242	501	20000
likelihood.ratio.pos.pos	49.11	268.5	2.124	6.461	25.2	218.5	501	20000
likelihood.ratio.pos.neg	3.802	12.3	0.1035	0.8782	2.136	15.83	501	20000
likelihood.ratio.neg.pos	1.759	1.744	0.01533	0.3396	1.276	6.092	501	20000
likelihood.ratio.neg.neg	0.1309	0.06849	7.28E-4	0.04389	0.1165	0.3047	501	20000
positive.predictive.value[1]	0.8853	0.1004	0.001599	0.6166	0.9146	0.993	501	20000
negative.predictive.value[1]	0.6319	0.1614	0.001912	0.2412	0.6569	0.874	501	20000
positive.predictive.value[2]	0.9558	0.03309	3.754E-4	0.8699	0.964	0.9928	501	20000
negative.predictive.value[2]	0.3051	0.1261	0.001912	0.09593	0.2873	0.603	501	20000
prevalence	0.758	0.1023	0.00153	0.5128	0.7731	0.9253	501	20000
sensitivity[1]	0.8844	0.0416	4.422E-4	0.7921	0.8888	0.9534	501	20000
sensitivity[2]	0.3105	0.0532	6.808E-4	0.2221	0.305	0.4321	501	20000
specificity[1]	0.6878	0.1617	0.002202	0.3673	0.6978	0.9578	501	20000
specificity[2]	0.957	0.02128	1.806E-4	0.9076	0.9602	0.9885	501	20000

In table above,

- likelihood.ratio.pos[i] is the posterior likelihood ratio when test *i* is positive
- likelihood.ratio.neg[i] is the posterior likelihood ratio when test *i* is negative
- likelihood.ratio.pos.pos is the posterior likelihood ratio when both tests are positive
- likelihood.ratio.pos.neg is the posterior likelihood ratio when test 1 is positive and test 2 is negative
- likelihood.ratio.neg.pos is the posterior likelihood ratio when test 1 is negative and test 2 is positive
- likelihood.ratio.neg.neg is the posterior likelihood ratio when both tests are negative.

When three tests are used, output will contain the following variables:

· likelihood.ratio.pos[i] is the posterior likelihood ratio when test *i* is positive
· likelihood.ratio.neg[i] is the posterior likelihood ratio when test *i* is negative

· likelihood.ratio.pos.pos[i,j] is the posterior likelihood ratio when tests *i* and *j* are both positive
· likelihood.ratio.pos.neg[i,j] is the posterior likelihood ratio when test *i* is positive and test *j* is negative
· likelihood.ratio.neg.pos[i,j] is the posterior likelihood ratio when test *i* is negative and test *j* is positive
· likelihood.ratio.neg.neg[i,j] is the posterior likelihood ratio when tests *i* and *j* are both negative

· likelihood.ratio.pos.pos.pos is the posterior likelihood ratio when all three tests are positive
· likelihood.ratio.pos.pos.neg is the posterior likelihood ratio when test 1 is positive, test 2 is positive and test 3 is negative
· likelihood.ratio.pos.neg.pos is the posterior likelihood ratio when test 1 is positive, test 2 is negative and test 3 is positive
· likelihood.ratio.pos.neg.neg is the posterior likelihood ratio when test 1 is positive, test 2 is negative and test 3 is negative
· likelihood.ratio.neg.pos.pos is the posterior likelihood ratio when test 1 is negative, test 2 is positive and test 3 is positive
· likelihood.ratio.neg.pos.neg is the posterior likelihood ratio when test 1 is negative, test 2 is positive and test 3 is negative
· likelihood.ratio.neg.neg.pos is the posterior likelihood ratio when test 1 is negative, test 2 is negative and test 3 is positive
· likelihood.ratio.neg.neg.neg is the posterior likelihood ratio when all three tests are negative.

4. Quick Re-runs with Small Changes and Sensitivity Analyses

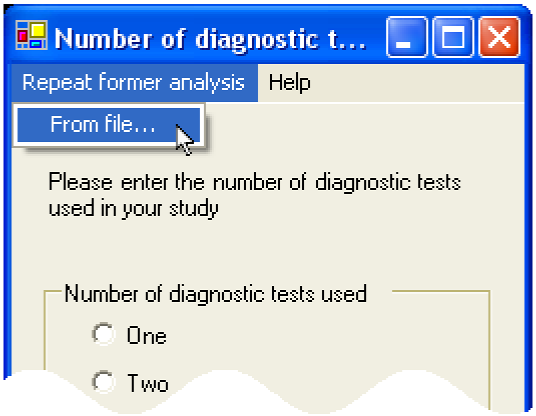
Sometimes after running **BayesDiagnosticTests,** one wants to re-run either the identical analysis with different starting values (for example, to ensure good convergence if only one chain was used in the previous run), or re-run essentially the same analysis with just a small change. This can occur, for example, when one wants to slightly change the data set (perhaps adding in another subset of patients) or a different prior density (i.e., a sensitivity analysis on the choice of prior density, important especially if one or two tests are used, as these non-identifiable models depend on the prior density used). Regardless of the specific reason for re-running the program, the general idea is to check the degree of change in the posterior inferences across a reasonable range of prior distributions, initial values, small changes to the data set, and so on. If the conclusions derived from each run are similar, then the inferences can be considered as robust. If not, then the choice of prior distributions/initial values/data set used needs to be more carefully assessed, and/or conclusions need to be drawn with some question as to their robustness.

See the references at the start of this article for more details about sensitivity analysis and Gibbs sampling in this application. We will now show how easy it is to re-run an analysis with small changes to the inputs.

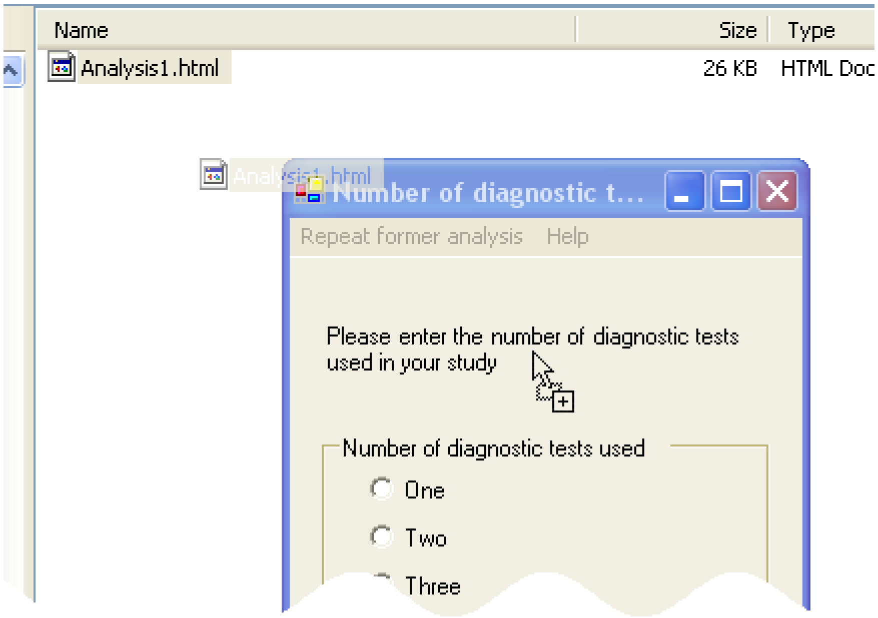
The re-analysis of data already analyzed by **BayesDiagnosticTests** is made easy through the top menu item *Repeat former analysis/From file...* (top right) from which you will be able to browse to a **BayesDiagnosticTests** html output file.

You can alternatively drag-and-drop a **BayesDiagnosticTests** html output file onto the label *Please enter the number of diagnostic tests used in your study* (bottom right).

BayesDiagnosticTests will then read the parameters used in the former analysis and bring you directly to the final *Problem description* form from which you can easily modify one or more inputs, as desired. Once this is complete, you can run the program as usual.



Please enter the number of diagnostic tests used in your study

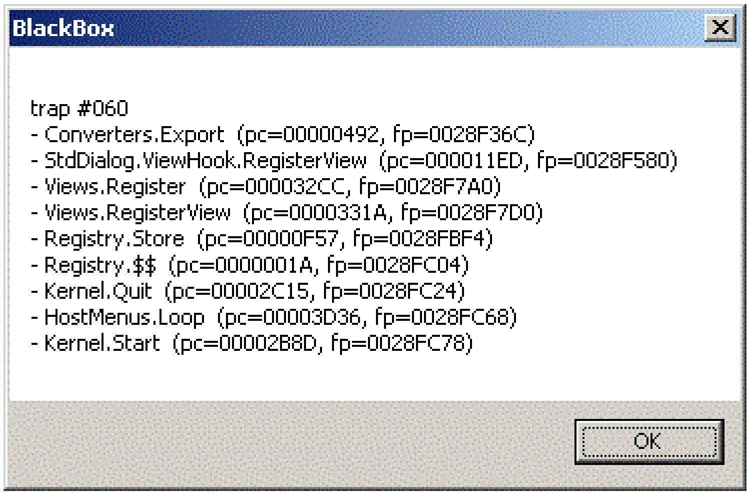


Please enter the number of diagnostic tests used in your study

5. Avoiding trap errors from permission settings on Windows 7 and Windows Vista platforms

If you are working on a Windows 7 or Windows Vista platform and have run WinBUGS before, you may have already run into the cryptic **Trap #060** error message illustrated to the right. This is due to restricted write permissions in c:\Program Files, where you may have installed WinBUGS.

WinBUGS **must** be installed in a directory where you have write permissions (e.g. C:\Users\user name\Documents) for **BayesDiagnosticTests** to run smoothly.



```
trap #060
- Converters.Export  (pc=00000492, fp=0028F36C)
- StdDialog.ViewHook.RegisterView  (pc=000011ED, fp=0028F580)
- Views.Register  (pc=000032CC, fp=0028F7A0)
- Views.RegisterView  (pc=0000331A, fp=0028F7D0)
- Registry.Store  (pc=00000F57, fp=0028FBF4)
- Registry.$$  (pc=0000001A, fp=0028FC04)
- Kernel.Quit  (pc=00002C15, fp=0028FC24)
- HostMenus.Loop  (pc=00003D36, fp=0028FC68)
- Kernel.Start  (pc=00002B8D, fp=0028FC78)
```

6. Change log

Version 1.1 (February 3rd, 2005). A minor technical change from version 1.0: earlier versions of WinBUGS required specification of a prior on 1-specificity (called one.minus.specificity) rather than on the specificity parameter itself, the latter leading to the inability of WinBUGS to draw samples from the marginal posterior density of this parameter. Since this barrier has been removed in more recent versions of WinBUGS, we have changed the WinBUGS code that is generated by **BayesDiagnosticTests** to be slightly easier to follow (since it is now written in terms of specificity rather than one.minus.specificity). These changes are invisible to the user unless they search for the WinBUGS program that **BayesDiagnosticTests** uses internally to derive estimates, and does not change any results produced by the program.

Version 1.2 (July 2005). **BayesDiagnosticTests** now includes a Problem Description window, which allows the user to review/change each component of the problem, from the disease prevalence and diagnostic tests priors or labels to the observed data.

Version 1.3 (September 2005). A minor bug fix update: previous versions would cause a WinBUGS trap message to appear if the path where files were saved or input from were too long (longer than 119 characters). Version 1.3 avoids this trap by detecting troublesome long path names, and saving some files in temporary locations at WinBUGS exit (for output files) or prior to running WinBUGS (for input files), and removing input file copies and moving output files to their proper locations after WinBUGS closes. This change is invisible and has no consequences for the user.

Version 2.0 (February 2006). Internal WinBUGS code was changed, making **BayesDiagnosticTests** run faster compared to earlier versions (especially for large sample sizes).

Version 2.1 (March 2006). The thinning option was added (see example on page 9 of this document for details). This option will sometimes prove useful for data sets with large sample sizes, where fast mixing of the Gibbs sampler can be a problem.

Version 2.2 (March 2007). When the program has finished, a GUI will appear announcing program completion and giving you the opportunity to view the output immediately.

Version 2.3 (September 2007). Three-tests data entry form was redesigned to make it more user-friendly.

Version 2.4 (April 2008). In earlier versions, default path to WinBUGS may not have been defined properly on Windows x64 platforms. This has now been corrected.

Version 2.5 and 2.5.1 (June 2008). Earlier versions accepted commas in numeric inputs, which, depending on their placement, could have led to unintended inputs being used. If a comma is found, a pop-up box now asks you to remove it, eliminating all ambiguity.

Version 2.6 (January 2009). The program now accepts either commas or periods as decimal symbols, depending on Regional Settings. Both commas and periods cannot be used at the same time, you must use the option chosen for your computer. See section 3 for full details.

Version 3.0 (June 2009). Quick re-runs, for example, to run sensitivity analyses, is now made very easy (see Section 4). Also, the main output file is now an easier to read html document.

Version 3.1 and 3.2 (November 2009). A title can now be added to the top of the first page of the output. Test labels were also added to the Node statistics section for better readability.

Versions 3.3 and 3.3.1 (December 2011). The previous default application folder (c:\Program Files) caused write permission problems for some Windows 7 and Vista users. Default application folder now changed to C:\Users\user name\Documents.

Versions 3.4 and 3.4.1 (February 2012). Minor technical problem solved from previous version.

Versions 3.5 and 3.5.1 (April 2012). We suggest a solution to prevent Trap errors for Windows 7 and Windows Vista users.

Version 3.6 (July 2012). Minor update: cmd.exe now closes automatically when program terminates.

Version 3.7 (August 2012). The path to the sub-directory where temporary files are stored was added to the Help menu of the initial form. While you can usually ignore these files, they can sometimes be helpful in troubleshooting when there are problems.

Versions 3.8--3.8.4 (September/October 2012). **BayesDiagnosticTests** must limit the length of temporary paths, since it uses WinBUGS scripts, which limits file paths to a maximum of 119 characters. Longer names will cause WinBUGS to freeze. Therefore, if the default temp directory path is too long, **BayesDiagnosticTests** will ask the user to enter a path with a shorter name.

Version 3.9 (January 2015). Minor update.

Version 3.9.1 (April 2015). Minor bug fix update: a potential installation problem was solved.

Versions 3.10--3.10.2 (January 2016). Minor update.

Versions 3.11 and 3.11.1 (September 2017). **BayesDiagnosticTests** now works on Windows 8 & 10. Windows 7 users do not need to reinstall or upgrade.

Questions? Please send email to: Lawrence.Joseph@McGill.ca

Other Bayesian software packages are available at
http://www.medicine.mcgill.ca/epidemiology/Joseph