



What is JAGS?

JAGS is Just Another Gibbs Sampler. It is a program for analysis of Bayesian hierarchical models using Markov Chain Monte Carlo (MCMC) simulation. It is similar to the BUGS program (<http://www.mrc-bsu.cam.ac.uk/software/bugs/>) developed by the UK Medical Research Council.

JAGS was written with three aims in mind:

- To have a cross-platform engine for the BUGS language
- To be extensible, allowing users to write their own functions, distributions and samplers
- To be a platform for experimentation with ideas in Bayesian modelling.

JAGS is licensed under the GNU General Public License. You may freely modify and redistribute it under certain conditions.

This guide to running JAGS provides instructions on how to run a standard JAGS test case on the Supercomputing Wales systems.

Step 1 - Log in

The examples used in this guide is configured to run on the Cardiff Skylake *Hawk* cluster. Connect to *hawklogin.cf.ac.uk* with your Supercomputing user credentials using your preferred method (e.g. PuTTY from a Windows machine or ssh from any Linux terminal, thus

```
ssh -l your_username hawklogin.cf.ac.uk
```

The steps below involve typing commands (**in bold font**) in the terminal window.

Step 2 - Load a JAGS module

In common with most of the application guides in this series, we assume at the outset that the module of choice would be selected from those originally available on HPC Wales. Thus a necessary first step would be to gain access to that module set by issuing the commands

```
module purge  
module load hpcw
```

Note that the “module purge” command is required to prevent any unintentional collision with pre-existing modules. Subsequently issuing the module command

```
module avail
```

will provide visibility of and access to the entire set of HPCW modules.

A number of JAGS binary packages are available.

- List preinstalled JAGS versions:

```
module avail JAGS
```

- Load your preferred version (version 3.4.0 is used in this example):

```
module load JAGS/3.4.0
```

- Confirm the loaded modules:

```
module list
```

Note: Dependent modules, such as the compiler environment, are loaded automatically.

Step 3 - Create a directory

Create a directory to hold any user data files. For this example, a directory called JAGS should be created in your home directory:

```
cd ~  
mkdir JAGS
```

Step 4 - Obtain a test case

A test case for JAGS is provided with the installation at

```
/app/genomics/JAGS/3.4.0/example/
```

Copy the SLURM job script and input files to the JAGS directory:

```
cd ~/JAGS  
cp -r /app/genomics/JAGS/3.4.0/example/run_JAGS.SLURM.q .  
cp -rp /app/genomics/JAGS/3.4.0/example/inputFiles .
```

The test case runs one of the examples available at <http://sourceforge.net/projects/mcmc-jags/files/Examples/3.x/>. Note that the job will be run using the scratch directory `/scratch/$USER/run_JAGS_id-< JobID>`, where `<JobID>` is the ID generated by the queuing system.

Step 5 - Submit a job

JAGS is a serial application (i.e. it runs using just one core). The supplied test should run for less than one minute.

- Submit the job using:

```
sbatch run_JAGS.SLURM.q
```

- Check the job queue using:

```
squeue
```

- When completed, output can be found in a file called `JAGS.Hawk.o.<JobID>` and errors, if any, can be found in `JAGS.Hawk.e.<JobID>`. If your job is successful, its output can be found in the output file `JAGS.Hawk.out.<JobID>`.

- Compare your output that that found in the file

```
/app/genomics/JAGS/3.4.0/example/JAGS.Hawk.out.14837
```

Further info

Further information on JAGS can be found at the website <http://mcmc-jags.sourceforge.net/>