**IMPROVING THE EFFICIENCY OF MCMC SAMPLING BY OPTIMIZING THE FREQUENCY OF PROPOSAL MECHANISMS**

OLUWADARA ELEBUTE1, OMAR KAZMI1, NICOLAS RODRIGUE1,2

1. Department of Biology, Carleton University, Ottawa, Canada,

2) Institute of Biochemistry and School of Mathematics and Statistics, Carleton University, Ottawa, Canada

**Abstract -** Efficient Monte Carlo sampling in Bayesian phylogenetics is of great interest, given the computational costs of the modelling framework.  Most of the recent developments in Markov chain Monte Carlo (MCMC) sampling for phylogenetic contexts have been focused on devising different types of parameter update mechanisms, or based on different representations of the underlying statistical models of molecular evolution.  This study focussed on the subjective settings of the MCMC algorithms, including the relative frequency and block structures of calls to the update mechanisms of tree topology: Nearest Neighbour Interchange (NNI) and Subtree Pruning and Regrafting (SPR). Specifically, we implemented a system which performed a series of short pilot runs on small trees, in order to optimise MCMC update mechanism settings, before launching a more definitive sampler that was well-suited to the context at hand. MCMC settings were adjusted on both GTR and CAT-GTR models using Phylobayes phylogenetic software on 2 varying data set sizes. It was observed that increasing the SPR tuning parameters did not necessarily improve efficiency of the MCMC sampler in both the GTR and CAT-GTR model. Conversely, efficiency was improved with increase in the NNI tuning parameter until it reached an optimal setting. A combination of SPR and NNI showed the greatest efficiency. The outcome of this study would be a more efficient Monte Carlo sampler, thereby reducing the overall required computing time for a Bayesian phylogenetic analysis.  The empirical insights gained from this work will then guide the development of an adaptive sampler.

**INTRODUCTION**

Bayesian inference is a method of statistical inference that has been adapted for modelling molecular evolution (Li *et al.*, 2000). It is similar to Maximum likelihood in that there is a search for probabilities of all possible topologies that model evolution however maximum likelihood states that the parameters are fixed constants while Bayesian inference obtains the parameters with some probability (Felsenstein, 1981; Yang & Rannala, 2012). Bayesian inference relies on a quantity called the posterior probability which is shown below.

P(T,θ | D) = P(T, θ) P(D | T,θ) [1]

P(D)

Where P(T,θ | D) is the posterior probability which is the probability of the tree topology and parameters given the data sets (branch length, substitution model, etc) . P(T, θ) is the prior probability, P(D | T,θ) is the likelihood which is the likelihood of the dataset given the topology and parameters and P(D) is a normalizing constant which integrates over all possible topologies and parameter values. (Edward, 2009)

In most practical applications the posterior probability is not analytically available. Monte Carlo Markov Chain (MCMC) is therefore an algorithm used to sample the entire parameter space from their posterior probability (Metropolis et al., 1953; Hastings, 1970). Based on a large sample using MCMC, we can approximate posterior probabilities (Metropolis et al., 1953; Hastings, 1970). A starting tree topology (T) or parameter (θ) is initially proposed randomly at the start of the chain. Another tree topology or parameter (T’, θ’) is then proposed from a modification of T and is accepted with α which represents the Metropolis Hastings ratio.

α = min 1, p(D | θ’, T’) p(θ’, T’) q(T’, θ’)

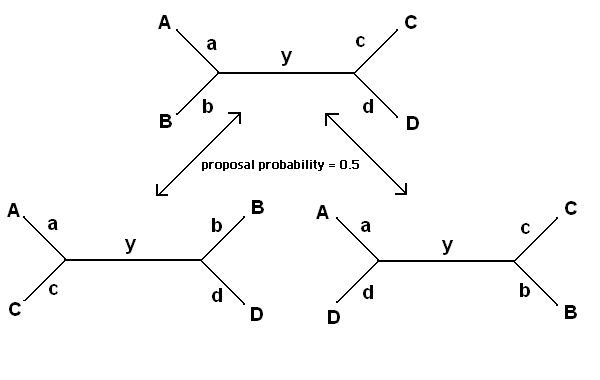
p(D | θ, T) p(θ, T) q(T, θ)

If α is greater than 1, the new proposal is always accepted; conversely, if it is less than one, the new proposal can be accepted with a certain probability represented by the 2nd part of the above equation. This is done by choosing a number between 0 and 1 and accepting if the number is less than or equal to that probability and rejecting otherwise.

This mechanism makes it possible to move through parameter space and sample less probable topologies or parameters as well those with high probabilities. In addition, the ratio makes the denominator in Bayes theorem cancel thereby avoiding its calculation altogether. The prior probability of tree topologies is always equal therefore that is cancelled out too.

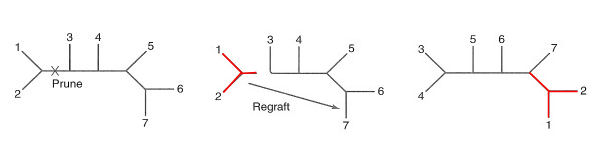
Bayesian MCMC is computationally expensive although it allows to implement much richer evolutionary models (Yang & Rannala, 2012). For tree topologies, computational expense is measured by how long it takes for 2 or more MCMC chains to converge, i.e. the independent runs produce consensus topologies that are similar across the runs (Lakner *et al*, 2008). In an MCMC chain, topology updates are made through proposal mechanisms. In this paper, we studied the effect of the frequency of two update mechanisms in particular on the convergence time of an MCMC chain which were Nearest Neighbour Interchange (NNI) and Subtree Pruning and Regrafting (SPR).

Nearest Neighbour Interchange involves swapping two neighbouring subtrees. An interior branch of a tree is chosen at random with 4 subtrees attached to it. For a chosen subtree, there are 3 different possible subtree swaps, 2 of which will result in a new topology. In figure 2A, subtree B can be swapped with A, C or D. Swapping with A will result in the same topology. Since NNI moves are often local; the moves are more modest proposals resulting in a higher acceptance rate, however tree space is generally not sampled exceptionally well.



**Figure 2A**: 2 possible topologies resulting from an NNI move of an internal node. Both have a probability of 0.5 (taken from http://www.socialdynamics.it/topics/phylogeny-and-evolution/sbix/)

Subtree Pruning and Regrafting involves pruning a subtree and reinserting it at a new random branch location as shown in Figure 2B. This method results in more global moves therefore it is considered a bold proposal. Since the subtree can be reinserted in any position, the proposed topology can be very different from the current topology resulting in a much lower acceptance rate. Using this mechanism, interesting regions of tree space tend to be visited. This mechanism also leads to a shorter burn-in period as the chain is able to leave the initial low probability space to better regions faster. A trace plot of an SPR mechanism will have many plateaus due to the low acceptance rate.



**Figure 2B**: An example of a subtree pruning and Regrafting move. Here subtree 1 & 2 is reattached at random location branch 7. (Taken from https://www.mun.ca/b-iology/scarr/Tree\_pruning.jpg)

Gibbs SPR and NNI in particular were the employed mechanisms. The Gibbs algorithm results in a guided topology proposal mechanism that reduces sampling of unnecessary regions. It increases the chances of sampling trees with the higher probability thereby increasing efficiency. This results in an increased acceptance rate and transition probability. (Hohna & Drummond, 2011).

**METHODS**

All analyses was conducted using Phylobayes MPI version (Lartillot *et al.*, 2013). The MPI implementation parallelizes and divides computational work among several slave processes to reduce computational work. An R script was written that submitted MCMC runs to a cluster at Nicolas Rodrigue’s laboratory at Carleton University.

The performance of the topology update mechanisms was tested on one single nucleotide sequence alignment from the Nicolas Lartillot 2012 paper (Lartillot & Delsuc, 2012). It contained a concatenation of 17 genes from 78 mammals, for a total of 15,117 sites. Two dataset sizes of 1000 and 10000 nucleotides respectively was obtained from the alignment using Nicolas Rodrigue’s SelectRandomCodon software program. The analyses of the 1000 nucleotides data set was carried out with 8 parallel processes for each chain while the 10000 nucleotide data set was carried out with 16 parallel processes.

Analyses was conducted under the GTR and CAT-GTR models of substitution which uses the infinite mixture Dirichlet process (Lartillot & Phillipe, 2004). The prior probability was the same for all topologies.

To diagnose convergence of MCMC chains, 2 chains were submitted in parallel until a bipartition support (maxdiff) of 0.1 was obtained. A bipartition support of 0.1 was low enough to assume convergence. The bipartition support was calculated using the bpcomp on Phylobayes MPI.

Test runs were carried out on dataset sizes of 300 and 1000 to test the R scripts and to obtain appropriate tuning parameter settings for the analyses. The convergence results from the test runs were not included in the result.

NNI and SPR frequency were adjusted as tuning parameters on the phylobayes command. 10 repeats were carried out for each frequency that was analysed. Every other tuning parameter apart from the topology proposal frequencies remained constant for every run. The various experiments that were carried out are shown in Table 1.

**Table 1**: Table showing the 9 separate experiments that were carried out. The shaded regions are the combinations that were used; for example the first row is analyses of the SPR frequencies using the GTR model on the smaller dataset of 1000

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Dataset | Number of processes | SPR | NNI | GTR | CAT-GTR |
| 1000 | 8 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 10000 | 16 |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
|  |  |  | 5, 10 |  |  |

**RESULTS AND DISCUSSION**

For each topology update frequency, 10 repeats were obtained. First, we started from the smaller dataset of 1000 nucleotide under the GTR substitution model and observed the effect of Subtree Pruning and Regrafting and Nearest Neighbour Interchange individually (Figure 1a and 1b). Since we started with random trees initially, the convergence was a measure of both burn-in time and mixing time. The convergence time measure was obtained from the trace file as the number of cycles of MCMC per chain and is a measure of seconds.

The SPR frequencies were measured with an interval of 1 from each other starting from 1 SPR frequency to 29 SPR frequency. SPR frequencies of 27 and 29 did not show significantly interesting results from the lower NNI frequencies therefore SPR frequencies of above 25 were not explored rigorously.

Generally no trend was observed in the boxplots displaying the SPR frequencies. The median computation times were not significantly different from each other. There were also very large differences in computation times of the repetitions in many of the SPR frequencies that resulted in outliers showing inconsistency in the result. However, the median convergence at low SPR frequencies of 1-4 were slightly larger than the rest of the SPR frequencies. In addition, it could be noticed that SPR frequencies of 17-22 had more consistent results because the outlier differences were not as substantial as the other frequencies. No valid or solid conclusion could be drawn from this result however because the results of frequencies 7-9 and 11-14 are missing due to deletion of the files by mistake.

Subtree Pruning and Regrafting in general results in more global moves and has a low acceptance rate because it samples interesting regions of tree space. Although Gibbs SPR allows for a more guided mechanism, increasing the SPR frequency is not expected to reduce convergence time because the moves will still be global without a particular trend. The significantly large outliers could be as a result of the chain getting stuck in tree space or bad luck of significantly low acceptance rates for those particular frequencies.

Due to the similarities in small frequency changes, NNI frequencies were explored using intervals of two. The most striking result in this case was that the lowest NNI frequency of 1 showed a very low convergence time as opposed to what was expected. 1 NNI frequency had one of the lowest median scores of below 300 seconds with 75% below 1000 seconds. This was not expected because NNI updates are very modest and do not do a very efficient job of mixing the MCMC chains in topology space. However, a frequency of 1 has a very high acceptance rate because of similarities in proposed topologies which could explain the observed results.

A general trend could be observed in the box plots of the NNI frequency. Excluding the NNI frequencies 9 and 11, there was a general decrease in the median convergence time of the plots of frequencies from 3 to about 21. The lowest median convergence scores could be observed in 17, 19 and 21. At NNI frequency of 23, the median convergence time began to increase gradually with larger outlier ranges. Although the median was low, the upper quartile was much larger meaning that results was not replicable with each repetition.

This results supported the given hypothesis because larger NNI values resulted in decreasing convergence times. The major disadvantage of NNI proposal mechanisms is the very long burn-in period as described by Lakner in 2008. Increasing NNI values results in a more intermediate proposal mechanism resulting in a shorter burn-in period as well as a high acceptance rate. The observed increase at 23 could be as a result of increasing boldness of the proposals. Larger NNI values of 23 and onward could have caused the newly proposed trees to be significantly different and reduce the acceptance rate mirroring bold mechanisms like SPR.

Contrary to Lakner’s results in 2008, the NNI convergence times and outliers on average were much larger than SPR (Lakner *et al.*, 2008). These results suggested that SPR is a more efficient mechanism than NNI. It is possible that the initial trees were very bad starting trees. According to Lakner, NNI performed terribly with very bad starting trees because of extended burn-in periods (Lakner *et al.*, 2008). This could further attribute the decreased convergence time with increase in NNI frequency to be as a result of decreased burn-in periods.

The large datasets signify a more practical size of datasets analysed normally using Bayesian phylogenetics. The same computations were repeated using both the GTR model and the CAT-GTR model. These were done using frequency intervals of 5 and the results were clustered to result in more runs i.e. the first boxplot signified 1-5, and the second one 6-10, etc. Despite the increased intervals, the same trends observed in the smaller datasets could be seen for both the SPR and NNI of the larger datasets. The SPR showed no trend in particular but it could be noticed that in accordance with the smaller datasets, the interval of 16-20 had the lowest convergence times in general. The NNI trend of decreasing convergence with increasing frequency was also obvious from frequency 6 till frequency 20 with the interval of 16-20 yielding the lowest convergence times. It was also noticed that the frequency interval of 1-5 also yielded a very low median although the upper quartile was high. This result was also in accordance with the smaller dataset which showed an NNI frequency of 1 yielding a low convergence time.

An interesting observation was that NNI had an equal or even better performance than SPR generally with the larger datasets contrary to the smaller datasets that showed significantly lower convergence time for SPR than NNI. With the smaller datasets, the variance in results for NNI was high yielding convergence scores in some cases that were almost 10 times that of SPR. However, the larger datasets showed average lower convergence times for NNI. NNI convergence times were all under 400000 while SPR was under 550000. It could be possible that the performance of NNI is improved with larger datasets, however this will have to be examined in detail.

The results from the CAT-GTR model using the dataset of 1000 showed similarities to the GTR model for both NNI and SPR. Due to the computational expense of the CAT-GTR model, run times were prolonged and could only be repeated 5 times for time constraint purposes (Lartillot & Phillipe, 2004). For this reason, box plots could not be constructed for these. The results and average times however are given in tabular form in the appendix.

**Figure 1a**: Boxplot of SPR frequency with convergence time on 1000 nucleotide dataset with the GTR substitution model showing a randomness and inconsistency of convergence times with increasing frequencies. Convergence times include burn-in period and mixing time. Results missing due to deletion error.

**Figure 1b**: Boxplot of NNI frequency with convergence time on 1000 nucleotide dataset with the GTR substitution model showing a general trend of decreasing convergence time with increasing frequencies. Convergence times include burn-in period and mixing time.

**Figure 2a**: Boxplot of SPR frequency with convergence time on 10000 nucleotide dataset with the GTR substitution model showing randomness and inconsistency of convergence times with increasing frequencies. Convergence times include burn-in period and mixing time.

**Figure 2b**: Boxplot of NNI frequencies with convergence time on 10000 nucleotide dataset with the GTR substitution model showing a general trend of decreasing convergence time with increasing frequencies. Convergence times include burn-in period and mixing time.

**Combination of SPR and NNI**

The most interesting results were obtained when SPR was combined with some NNI frequencies using the larger datasets and the CAT-GTR model. Figure 3 shows a scatter plot of the average convergence times at each frequencies observed with 5 repeats instead of 10 for frequency intervals of 5. As was shown in figure 3, SPR on its own had much larger convergence times and more variance among the frequencies. The NNI values were more consistent with lower average convergence times than the SPR. When the SPR frequencies were combined with a constant NNI frequency of 5, it was observed that on average, the convergence times were lower than the individual SPR and NNI times although there was still no significant trend. This was also repeated for an NNI frequency of 10 which showed reduced convergence times for all frequencies but was not significantly different from the combination of SPR and the NNI frequency of 5. This result suggested that the combinations of SPR and NNI improves efficiency a lot more than their independent counterparts. The explanation for this could be because SPR and NNI independently are on opposite ends of the spectrum i.e. bold and modest consecutively (Lakner *et al.*, 2008). Therefore a combination of both results in an intermediate proposal mechanism which allows for good mixing subsequently a short burn-in period and a high acceptance rate. This made transition of the MCMC chains more efficient and resulted in better sampling. The lowest significant average convergence time was seen with an SPR frequency of 1 and an NNI frequency of 5.

**Figure 3**: Scatter plot showing the relative speeds of SPR alone, NNI alone and a combination of SPR and 5 NNI

**CONCLUSION AND FUTURE DIRECTIONS**

Markov Chain Monte Carlo algorithm has made the impossible calculation of the true posterior probability possible from very long chains (Yang & Rannala, 2012). The MCMC sampling is however computationally expensive and leaves Bayesian phylogenetics at a disadvantage (Yang & Rannala, 2012; Lakner *et al*., 2008). An MCMC chain is limited by its mixing time, burn-in time and the time it takes separate chains to converge. A lot of research has gone into convergence diagnostics looking at tree topology update mechanisms and their modifications as well as tuning parameters of Bayesian phylogenetic calculations such as the extension mechanisms, tree branch lengths and substitution models. This paper in particular looked at the effect of the frequency of the topology update mechanisms Nearest Neighbour Interchange (NNI) and Subtree Pruning and Regrafting (SPR); this is a tuning parameter of Phylobayes calculations. The frequency determines how many of these updates are made with each cycle of the MCMC chain. For the purpose of this study, other tuning parameters were set to the default Phylobayes settings consistently so that only the effect of the frequencies will be studied. Our hypothesis was that there will be a significant decrease in convergence time with increase in the frequency of NNI only.

The results support the hypothesis and suggest that when using NNI, frequencies of 16-21 generally results in a greater efficiency. MCMC efficiency on the other hand will not be affected by changing the tuning parameter settings of SPR. The most valuable result from this experiment is that combining SPR and NNI results in a much greater efficiency than the independent topology frequencies using the CAT-GTR model because it allows for intermediary proposals.

The short-term effect of this result could possibly reflect on the Phylobayes MPI software default settings of the topology update frequency. In fact, the default setting currently is 5 SPR and no NNI. Although for the smaller datasets, the results showed SPR having a better performance, the larger dataset which is a better representation of datasets that are normally analysed shows otherwise. Subsequently, the results suggest that that default setting may not yield optimal performance and time efficiency compared to a setting of NNI and SPR combined. From the result, the optimal frequencies that yielded the lowest convergence time was 1 SPR and 5 NNI.

This study is just a small subset of several experiments that need to be done. Improvements on this study will require a larger number of starting trees and several repeats of to improve reliability of the results. Most convergence diagnostic studies repeat calculations up to a 100 times, however due to the time constraint of this study, only 10 repetitions were done with 5 repetitions in some cases. Secondly, this study needs to be performed on other nucleotide sequence alignments to confirm the universality of the results. It is possible that the results of this study only applies to the sequence alignment that was used. In addition, other amino acid and codon substitution models need to be studied to observe if the results here apply to those as well. Lastly, several other topology update mechanisms exist that were not studied in this paper. One of such is the extension Tree Bisection and Reconnection mechanism which according to several studies has one of the best performances in reducing convergence times thus it will be interesting to observe its effect in combination with NNI or SPR. Further research in this area could be a step towards less expensive and predictable MCMC calculations.

**ACKNOWLEDGEMENTS**

I would like to express my sincere gratitude to Nicolas Rodrigue for monitoring the progress of the project and for inputting his time and knowledge to help in the successful completion of this project. I would also like to thank Omar Kazmi for assisting with the computational difficulties and issues that were encountered.

**REFERENCES**

Edwards, S. V., Liu, L. & Pearl, D. K. High-resolution species trees without concatenation. Proc. Natl Acad. Sci. USA 104, 5936–5941 (2007)

Felsenstein J. 1981. Evolutionary trees from DNA sequences: a maximum likelihood approach. J. Mol. Evol. 17:368–376.

Felsenstein, J. 2004. Inferring phylogenies. Sinauer Associates, Sunderland, Massachusetts.

Hastings, W. K. 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57:97–109.

Hohna, S. & Drummond, A. (2011). Guided Tree Topology Proposals for Bayesian Phylogenetic Inference. *Systematic Biology*, *61*(1), 1-11.

Lakner C, van der Mark P, Huelsenbeck JP, Larget B, Ronquist F. Efficiency of Markov chain Monte Carlo tree proposals in Bayesian phylogenetics. Syst. Biol. 2008;57:86-103.

Lartillot N., Philippe H. 2004. A Bayesian mixture model for across-site heterogeneities in the amino-acid replacement process. Mol. Biol. Evol. 21:1095–1109

Lartillot & Delsuc, Evolution, 66:1773-87, 2012

Lartillot, N., Rodrigue, N., Stubbs, D., & Richer, J. (2013). PhyloBayes MPI: Phylogenetic Reconstruction with Infinite Mixtures of Profiles in a Parallel Environment. *Systematic Biology*, *62*(4), 611-615. http://dx.doi.org/10.1093/sysbio/syt022

Metropolis, N., A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. 1953. Equation of state calculations by fast computing machines. J. Chem. Phys. 21:1087–1092

**APPENDIX**

**Table 2:** Table showing the average convergence times of the 1000 nucleotide datasets under the CAT model. In this case, the SPR had a worse performance than NNI.

|  |  |  |
| --- | --- | --- |
| **FREQUENCY** | **NNI AVERAGE CONV-ERGENCE TIME (Sec)** | **SPR AVERAGE CONV-ERGENCE TIME (Sec)** |
| **1-5** | **12527.8** | **196557.5** |
| **6-10** | **37336** | **122487** |
| **11-15** | **15972** | **62775.9** |
| **16-20** | **13387.9** | **119058** |
| **21-25** | **23946.2** | **93046.2** |

**TEST SCRIPT**

#!/bin/bash  
  
i=$1  
      
while [ $i -lt  21 ]; do  
  
    spr=$i  
    nni=$2  
    fname1=$3'-'$i  
    fname2=$4'-'$i  
    direcname=$5'-'$i  
    tree=$6  
  
    > $fname1  
          
    > $fname2

// Creating the script to run Phylobayes MPI  
      
    echo 'Creating Phylobayes Script'  
          
        printf '#!/bin/bash' >> $fname1  
  
        printf '#!/bin/bash' >> $fname2  
      
     printf '' >> $fname1  
  
        printf '' >> $fname2  
          
        printf '\nmpirun -np 16 ../pb\_mpi -d '$tree' -gtr -ncat 1 -spr '$spr' -nni '$nni' run1\_'$i'\n' >> $fname1  
          
        printf '\nmpirun -np 16 ../pb\_mpi -d '$tree' -gtr -ncat 1 -spr '$spr' -nni '$nni' run2\_'$i'\n' >> $fname2  
  
    echo 'creating executable'  
         
        chmod "a+x" $fname1  
          
        chmod "a+x" $fname2  
  
     echo 'passing to the queue'  
          
        qsub -q all.q@compute-0-9.local -cwd -pe orte 16 $fname1  
          
        qsub -q all.q@compute-0-9.local -pe orte 16 -cwd $fname2  
  
    // Storing results in result1 file  
  
    printf '\n'$spr'\t'$nni >> result1  
  
    sleep 100  
  
    echo 'calling toptest.sh'  
      
    one='run1\_'$i  
    two='run2\_'$i  
  
    ./toptest.sh 0.1 $one $two   
  
    echo 'starting avgcompt.sh'  
  
    ./avgcompt 5 $one $two

//copy files into a new directory  
  
    mkdir $direcname  
      
      cp $fname1.\* $direcname  
  
      cp $fname2.\* $direcname  
  
    sleep 100  
  
    mv run\* $direcname  
  
    i=$((i+3))  
  
done

**Test Script** (Convergence Testing)

#!/bin/bash  
  
req=$1  
  
ffile=$2  
  
sfile=$3

// obtaining the value of run file  
  
var1=`cat $ffile.run`  
var2=`cat $sfile.run`  
  
while [ $var1 == 1 ] && [ $var2 == 1 ]; do

// starting bpcomp  
  
     ./bpsesh $ffile $sfile

// collecting value of maxdiff  
     
        val=$(awk '{ print $3 }' bpcomp.bpdiff | head -2| tail -1)  
            
             if (( $(bc <<< "$val <= $req") == 1 )); then  
                echo 'in the if statement'  
                       printf "0" > $ffile.run  
                        printf "0" > $sfile.run  
                        var1=`cat $ffile.run`  
                        var2=`cat $sfile.run`  
                fi  
done

**Test Script** (Computing convergence time)

#!/bin/bash  
  
echo 'last step'  
  
tracker=$1  
ttracker="-$tracker"  
ffile="$2.trace"  
sfile="$3.trace"  
  
next=1  
  
val1=$(awk '{ print $2 }' $ffile | head $ttracker | tail -1)  
sleep 3  
  
cycnum1=$(awk '{ print $1 }' $ffile | tail -1)  
accycnum1=$(($cycnum1 - ($tracker-1)))  
  
  
while [ $tracker -gt 0 ]; do  
    if [ $tracker -eq $cycnum1 ]; then  
        tracker=0  
    else  
    tracker=$(($tracker+1))  
    ttracker="-$tracker"  
    next=$(awk '{ print $2 }' $ffile | head $ttracker | tail -1)  
    val1=`echo $val1 $next | awk '{print $1 + $2}'`  
    fi  
      
done  
  
tracker=$1  
ttracker="-$tracker"  
  
val2=$(awk '{ print $2 }' $sfile | head $ttracker | tail -1)  
cycnum2=$(awk '{ print $1 }' $sfile | tail -1)  
accycnum2=$(($cycnum2 - ($tracker-1)))  
  
while [ $tracker -gt 0 ]; do  
    if [ $tracker == $cycnum2 ]; then  
        tracker=0  
    else  
    tracker=$(($tracker+1))  
    ttracker="-$tracker"  
    next=$(awk '{ print $2 }' $sfile | head $ttracker | tail -1)  
    val1=`echo $val1 $next | awk '{print $1 + $2}'`  
    fi  
      
done  
  
echo 'computing average'  
  
accycnum=$(( $accycnum1 + $accycnum2 ))  
  
val=`echo $val1 $val2 | awk '{print $1 + $2}'`  
  
avg=`echo "$val / $accycnum" | bc -l`  
  
printf '\t'$val'\t'$avg >> Lartillot10000res  
  
echo 'done'