

0.1 June 6, 2016 Notes

- Discovered and fixed two bugs involving an uninitialized selection vector and Mphi values
- Said bug fixes resolved the issue of the program crashing around iteration 2500
- Despite the fixed bugs resolving that issue, log likelihood values still plummet to values eventually reaching $10e-23$ which causes the not a number error, causing the MCMC algorithm to halt.
- Due to Newton failing to execute my job request, I had to run the R script on my laptop so I was only able to run it once so more testing is needed but for the meantime it appears it takes until around iteration 42000 for the program to crash.

TODO:

Work on psuedocode

Remove redundancies

Determine if Sphi prior is being used

Improve variable names and documentation

Document the R scripts

0.2 June 7, 2016

- Removed exit statements from FONSEParameter.cpp, PANSEParameter.cpp, and RFPPParameter.cpp in order to eliminate warnings when running the check in R
- Downloaded all of the necessary files to my Gauley account.
- Running FONSE with simulated data and $b = 0.001$ led to "nan" being reached much more quickly than on my laptop, around iteration 5500
- Said issue may be a cause of R not working correctly on Gauley as compared to my laptop. Multiple errors and missing packages were encountered. These errors made it impossible to check any plots.
- Two new Items for the TODO list: Continue troubleshooting R on Gauley and alter Jeremy's .sge scripts so that they no longer email him when I run a job on Newton

0.3 June 8, 2016

- Changed the .sge scripts for Newton so that Jeremy no longer receives an email upon the completion of a job
- Changed a function in FONSEModel.cpp to match it's corresponding function in ROCModel.cpp. In particular, I changed a long series of divisions to one division and a series of multiplications because multiplication is a faster operation.
- Fixed a typo in the documentation of ROCModel.cpp

- Worked out some kink experienced while running R scripts on Gauley. In particular, finally made it so that the plots actually showed up and worked.
- Formulated the hypothesis that the reason we are experiencing a drop in log likelihood comes from a bug involving either δM , $\delta \Omega$, or ϕ after observing the trace plots.

0.4 June 13, 2016

- Fixed an error in FONSEModel.cpp involving a couple of missing semicolons
- Fixed an error in the .tex file for the notes where using the hyperref package kept the notes from compiling
- After running FONSE again, with simulated data and $b = 0.001$, it seems the errors in log likelihood are no longer deterministic. Values still plummet, and the traces suggest that both mutation and selection values may be the cause. Further testing and observation is required.
- Found a difference between FONSEModel and ROCModel where ROC would take mutation prior into account and FONSE wouldn't. This might part of the issues causing log likelihood values to plummet, but testing is required.
- Uploaded both the new .tex file and its corresponding pdf just in case there's another error that doesn't keep me from compiling on my machine but might keep another from compiling on a different machine.

0.5 June 14, 2016

- The addition of the mutation prior into FONSE seems to have stopped the rapid descent of the Log Likelihood values. After running it once with 1000 samples and 1 thinning, the log likelihood stayed somewhat stable after starting low and rising to about -20000. The values were also similar when running with 1000 samples and 10 thinning. More extensive testing is required to ensure that the plummeting problem is finally resolved.
- Although the log likelihood values are starting to come together, there is an issue observed in both the mutation and selection traces where certain codons are not being accepted.
- Conducted further comparisons on FONSE and ROC to remove redundancies and semantic oversight.

0.6 June 15, 2016

- After running FONSE overnight, I feel it's safe to conclude that the plummeting log likelihood problem is resolved although there have been issues with multiple, sometimes all but a few, codons with acceptance rates of 0 as backed up by both the output and the trace plots of both mutation and selection, although this is again non deterministic. Running it on Gauley produced far more codons with no impact than when I ran it on my laptop so the non determinism may come from memory or operating system differences.

- There was a problem with Gauley involving the machine randomly restarting without being prompted. It doesn't seem like a major problem and the effects seem to be confined to the processes being run on it at the time being prematurely terminated. I'll be keeping an eye for any unexpected behavior from Gauley in case the problem ends up being reoccurring.
- Studied the model to try and deduce how the simple inclusion of the mutation priors resolved the plummeting problem. So far it still remains fairly unclear and more studying is necessary.

0.7 June 16, 2016

- Ran FONSE with $b = 0.001$, samples = 1000, and Thinning = 10 3 separate times.
- Run 1
 - Log likelihood settled around -9600
 - AA's L and R had acceptance rates of 0
 - stdDevSynthesisRate posterior estimate settled around 1.4, .05 for the proposal width
 - Trace of the expected values of phi goes to 0.
 - All codons for L in the mutation traces stop changing and fall out around sample 100
 - R falls out around sample 300
 - It's the same story for both L and R in the selection traces except it takes about half as many samples for both to fall out
- Run 2
 - Log likelihood settled around -14000
 - AA's L and R still had acceptance rates of 0
 - stdDevSynthesisRate posterior estimate settles around 1.9, .045 for the proposal width
 - Trace of the log likelihood didn't level off as much. It seemed to still be rising.
 - expected phi trace was the same as Run 1
 - L falls out around the same place in Mutation as Run 1; R falls out twice as fast.
 - Both fall out twice as fast in the selection traces compared to Run 1.
- Run 3
 - Log likelihood settled around -20000
 - AA's L and R still had acceptance rates of 0
 - stdDevSynthesisRate posterior estimate settled around 2.5, .04 for the proposal width
 - There was a more pronounced level-off in the log likelihood trace, much like Run 1
 - expected phi trace leveled off but slightly above 0
 - Both L and R had mutation and selection traces similar to Run 1
- Also worked on documenting FONSE by taking documentation from similar parts in ROC and making the necessary alterations before putting it into FONSE