Research Journal

Kirolos A. Shahat
kshahat@vols.utk.edu

Beginning 19 May 2017

Contents

Friday, May 19 2017

1 Goals for today

- Get LaTeX up and running to begin taking notes
- Study 2007 article and use it to begin learning terminology and understand key concepts
- Study Jeremy Rogers' and Alan Dixon's previous notes and see how they began their research

2 Current Progress and Notes

- Beginning to understand layout and format of LaTeX files.
- Terminology:
  1. Codon - Sequence of three nucleotides that together form a unit of genetic code in a DNA or RNA molecule.
  2. Codon Usage Bias (CUB) - Nonuniform usage of particular synonymous codons within a genetic sequence.
  3. Genome - The genetic material of an organism.
  4. Stochastic Model - A model that allows for random variation of one or more inputs over time.
  5. Stochastic Evolutionary Model of a Protein's Production Rate (SEMPPR) - A way to link CUB and the average protein production rate mechanistically. Essentially the model makes inferences about the production rate of a gene based on its elevation on the fitness landscape of protein production costs.
  6. Polypeptide - Linear amino-acid chain which forms most, or all, of a protein.
  7. Genetic Fitness (n) - The reproductive success of a genotype.
  8. Gene - Represented as a vector of Codons.
- Concepts:
  1. A major cost of a nonsense error is the amount of energy invested into assembling the incomplete polypeptide.

2. Selection on codon usage against nonsense errors should increase with codon position along a sequence because the cost is related to the length. This leads to the prediction of increasing codon bias with codon position.
3. Adaptation of a codon sequence, within SEMPPR, refers to the state of its expected cost of producing a protein relative to the minimal possible cost.
4. The resulting output from SEMPPR is a posterior probability distribution for the protein production rate of a gene based on its observed codon sequence.
5. Incomplete proteins are the result of nonsense errors. The cost of these nonsense errors is a function of their expected number and the length of the incomplete proteins.

- Notes for next time
    1. Figure out math and vector notation for LaTeX.
    2. Continue studying 2007 article
    3. Attempt to get to Jeremy and Alan's labbooks

Monday, May 22 2017

1 Goals for today

- Short review of what I learned on May 19th 2017
- Figure out how to get mathematical notation in LaTeX documents
- After speaking with Hollis, he suggested that I look at the existing FONSE code and cross reference that code with given articles and refer to the layout of the ROC model for implementation so that is where I'll focus my time now and address questions when needed in order of relative importance.

2 Current Progress and Notes

- Terminology:
  1. Elongation - The stepwise addition of amino acids to the growing protein chain.
- Questions:
  1. Not sure what the variables bias_csp or mutation_prior_sd are in the FONSEParameter. mutation_prior_sd is the likelihood that there is a mutation where it is defaulted to 0.35. I believe that the bias_csp is the codon bias based on current position.
  2. Unsure how those values are getting updated or where they show up in the $\eta(\vec{c})$ equations.
- Current Notes:
  1. Running runFONSEmodel.R and saving the output into a file to understand what is going on
  2. Beginning scan of FONSEParameter.h and FONSEParameter.cpp and trying to document where I can from current understanding
  3. Currently looking through the constructors in FONSEParameter.cpp and following the methods that are being called. Currently in initFromRestartFile.
  4. rep(x, y) returns a vector of size y with all elements as value x in R

   5. sd = standard deviation, csp = codon specific parameter.

- Notes for next time:
   1. Ask Dr. Gilchrist what the variables in the FONSE equations represent so that I can understand how to get them/decipher them in code.
   2. Continue trying to decipher the FONSEParameter code and relating them to articles/equations. Mostly get help with initvalues methods and from there it shouldn't be too difficult to follow.

Wednesday, May 24 2017

1 Goals for today

- Continue running runFONSEmodel.R and saving the output to file
- Continue scanning through FONSEParameter.h and FONSEParameter.cpp and documenting where I can from current understanding
- If time allows, get the meanings of the variables from Dr. Gilchrist

2 Current Progress and Notes

- Ran through about 930 iterations of runFONSEmodel.R and stopped it. I have the outputs from that program to follow.
- Currently tracing through the FONSEParameter.cpp and it's leading me to initBaseValuesFromFile method so I am checking it out and documenting where I can.
- Spoke with Dr. Gilchrist and he confirmed that I should be going through the C++ code and trying to understand it. He also gave me an understanding of what the variables represent so that I can understand some of the math in the code when I come across it.
- Going through the init methods in FONSEParameter.cpp and I found that sequenceSummary is implemented using maps to emulate enumerators. I haven't seen what all it is being used for but I think enumerators are constant time when maps are log(n) time to find an element. I think it be a nice speedup if it was changed over. Just a thought for later, potentially.
- I'm adding comments and I'm noticing that the code is pretty cluttered, my impulse is to start cleaning it and making it more legible but I'm refraining from doing so for now...

Friday, May 26 2017

1 Goals for today

- Continue going through init methods in FONSEParameter.cpp and FONSEParameter.h

2 Current Progress and Notes

- Currently in initFONSEValuesFromFile in FONSEParameter.cpp. I see a print statement that just says "here" so I'm going to assume this method is not working properly for some reason, might be because eof is not doing anything differently? It's essentially breaking through the structure of the loop.
- Finished going through initFONSEValuesFromFile. I condensed two for loops into one at the end of that method. I documented what I did and left the previous code in case someone needs the previous code.
- Going through writeBasicRestartFile in Parameter.cpp. It's pretty well written, my only confusing is why a ostringstream is used, doesn't seem needed to me. Just seems to be extra memory because it's just copied to a string at the end. Can cut out the string and ostringstream and just write to the file directly and save on memory.
- Went through writeFONSERestartFile, nothing too strange there.
- Going through initParameterSet in Parameter.cpp
- Indexing vectors in R start at element 1 not 0 (Mathematicians....)
- STANDALONE represents not being run by R
- Condensed initParameterSet vector assignment loop using iterators and the push_back method but now I'm noticing a segFault in runFONSEmodel.R which is strange because I wasn't getting that before. Will have to look into that next time by re-cloning the RibModelFramework repo and seeing if re-installing that fixes the issue or not.

Tuesday, June 6 2017

1 Goals for today

- Continue going through old code

2 Current Progress and Notes

- Forked the RibModelFramework repo from Cedric's github to my own so that I can push without fear of losing work that I have done.
- initFONSEParameterSet has three for loops that, i believe, can be condensed into one. Going to try and do that.
- Successfully did the optimization and tested it.
- learned a little more about ostringstreams thanks to Hollis.
- Goal for next time: Continue going through old code starting at initAllTraces.

Wednesday, June 7 2017

1 Goals for today

- Continue going through old code

2 Current Progress and Notes

- $ is calling a method of an R object
- dM = mutation, dOmega = selection. This is with respect to FONSE
- Incase I forget: One future goal is fix some of the read restart file methods into switch statements rather than if else. It is currently very unclean.
- Goal for next time: Continue going through old code starting at the CSP functions

Friday, June 9 2017

1 Goals for today

- Continue going through old code

2 Current Progress and Notes

- The propose structures and functions are actually the 'predictions' that we are making, I believe.
- Was looking through proposeCodonSpecificParameter and found a randNorm method. I found it in parameter.cpp but it was calling rnorm which I'm not sure I know what that is. Couldn't find it anywhere. Will have to look into it soon.
- So far what I'm seeing from the propose function is that it:
  1. Loops for the total number of groupList, which I think is just the total number of different amino acids there are.
  2. Then we get the number of codons by subtracting the start from the end location of the amino acids, because the cost scales with distance or how far the production has gone.
  3. There's a vector called iidProposed, Identically independant distribution. This is basically saying that the underlying distribution is the same for all of them but it uses the randNorm function which seems to be randomly generating a different distribution for each iteration. So I'm assuming that it's constant in the sense that's it's across the board random. This is done for the total number of categories(mutation + selection) all multiplied by the number of codons.
- Next time: Continue in the proposeCodon... and start at the covaryingNums variable declaration.

Monday, June 12 2017

1 Goals for today

- Continue going through old code, starting at proposeCodonSpecificParameter in FONSEParameter.cpp

2 Current Progress and Notes

- As stated previously, the initial for loop is initializing the iidProposed to a random distribution for each element between 0 and 1 for numCodons*(mutation+selection categories) elements total.
- There were an un-optimized double for-loop chain that could be condensed in the proposeCodon... function and I optimized it and pushed it into my repo. Hoping to do a pull request soon to update the main branch.
- I'm not super sure what the covarianceMatrix stuff is used for with respect to FONSE. Whenever I talk to Dr. Gilchrist next I want to ask him about that.
- Looking at the eta analysis paper I believe that dM $=\Delta\eta$, while dOmega $= \Delta\omega$ which I believe have already been calculated and are stored within the currentCodonSpecificParameter[dM or dOmega].
- Based off of that last note: the covariance matrix is the summation term. The only thing I'm getting confused with is that I don't see any kind of difference, they are all sums. I'm assuming that it is done somewhere behind the scenes of what I currently am looking at. Should definitely make sure that I am deciphering this correctly but I think I'm getting close to a breakthrough of understanding the current code at least.
- I asked Hollis a little bit about the dM variable and he says that he doesn't believe that it is the mutation.
- Goal for next time: Try to continue comparing some of these methods with the equations in the eta analysis paper and continue to get an understanding of what is going on and try to ask Dr. Gilchrist for advice on what I can.

Monday, June 26 2017

1 Goals for today

- Get back into the flow of things. I've been gone too long.
- Start going through the R code and matching it with the C++ ones.

2 Current Progress and Notes

- Went through genome::readFasta and reorganized it and made it a bit cleaner.
- Goal for next time: Start going through genome::readObservedPhiValues

Friday, June 30 2017

1 Goals for today

- Continue running through runFONSEmodel.R with initializeParameterObject function call

2 Current Progress and Notes

- Steps in runFONSEmodel.R
    1. init a genome object
    2. init a FONSE parameter object, goes to initParameter set in parameter.cpp and then initFONSEparameterset and then calculates SCUO(syn. codon usage order)
    3. init MCMC object does nothing special
    4. runMCMC which calls run and then run calls varyinitialconditions in MCMCAlgorithm.cpp, that is where I will pick up next time.

Monday, July 10 2017

1 Goals for today

- Continue going through MCMC code

2 Current Progress and Notes

- (x-mean)/sd = standard score
- got to calculateLogLikelihoodRatioPerGroupingPerCategory method in FONSEModel.cpp, start there next time.

Tuesday, July 11 2017

1 Goals for today

- Begin analysis of calculateLogLikelihoodRatioPerGroupingPerCategory method in FONSEModel.cpp
- Talk to Dr. Gilchrist if both of us have time.

2 Current Progress and Notes

- Essentially the calculateLogLikelihoodRatioPerGroupingPerCategory method just does variable assignments and then calls calculateLogLikelihoodRatioPerAA which does the calculations. Will begin analysis. Note: Selection gets updated from getParameterForCategory by reference.
- calculateLogLikelihoodRatioPerAA begins by getting the index of the smallest selection. getCodonPositions seems to be a recursive definition and is called within a loop. I can definitely see why this program is computationally loaded.
- Didn't get to meet with Dr. Gilchrist, he didn't show up and I have to leave to do reading before class. Try to meet with him tomorrow as well as continuing to go through calculateLogLikelihoodRatioPerAA beginning at the second for loop in the function.

Thursday, July 13 2017

1 Goals for today

- Continue analysis of calculateLogLikelihoodRatioPerAA method in FONSEModel.cpp
- Talk to Dr. Gilchrist if both of us have time.

2 Current Progress and Notes

- calculating the loglikelihood requires that we first get the start and stop location for the specific grouping, getting the position per aa location, iterating through those positions, and then calling calculateLogCodonProbabilityVector. This is seems like a rabbit hole but I feel like I'm getting close to the base.
- calculateLogCodonProbabilityVector has equations in the comments and one thing that is confusing about it is that the left hand side has a log while the right hand side does not.
- Spoke with Dr. Gilchrist, he explained the logarithmic approach used and why it works, some probability and it's relation to likelihood, and clarified some of my goals.
- Asked me to figure out how much I will be able to show up in the fall and to figure that own within the next couple of weeks.
- Goal for next time: Continue going through code in calculateLogCodonProbabilityVector.

Wednesday, July 26 2017

1 Goals for today

- Go through calculateLogCodonProbabilityVector

2 Current Progress and Notes

- For some reason the calculation in this method is mostly going into calculating the denominator which is being subtracted off of the original calculation to get the codon probability. Not sure I fully get that but that's fine. The method updates the codonProb vector with the proper values.
- Returning to calculateLogLikelihoodRatioPerAA, the sum of the codon probabilities is the loglikelihood.
- Returning to calculateLogLikelihoodRatioPerGroupingPerCategory, calculateLogLikelihoodRatioPerAA is called twice, once for the likelihood and another for the proposed likelihood. After going through all of the genes, the acceptance ratio is the difference of the likelihood from the proposed likelihood.
- Going back to acceptRejectCodonSpecificParameter in MCMCAlgorithm.cpp, the method updates posterior, likelihoodTrace and posteriorTrace with elements from the acceptanceRatioForAllMixtures vector but it seems that it is currently just updating them with '0.0'. I'm assuming that in further iterations this vector changes to where it's not always '0.0' but we shall see.
- Going back to the run method in MCMCAlgorithm.cpp, the method sometimes calls adaptCodonSpecificParameterProposalWidth from Parameter.cpp which is very math heavy. Will begin analysis of this method.
- Spoke with Dr. Gilchrist and he suggested that my goal be shifted to explaining the parameters that are given and then focus on fitting simulated and real genome data and comparing them to the values that I am supposed to be getting to see if the model is doing well. That will be my goal for the remainder of my time. He also suggested that documentation of the current code for future ease of understanding would be very good and beneficial.

- Goal for next time: generate simulated data will be the main goal but afterwards I will continue to do the tasks given to me by Dr. Gilchrist.

Thursday, July 27 2017

1 Goals for today

- Run simulations and compare outputs

2 Current Progress and Notes

- Ran runNSE2000.R and finished the MCMC loop. Got an error that I didn't have the Hmisc package. Currently getting that. Seems to be a big package that includes a lot.
- Successfully installed Hmisc.
- After running the program again I get a strange error: No window was given, zooming in at last 10% of trace. Error in genome$getGenomeSize() : could not find valid method when the method is defined with a defaulting boolean argument.
- I placed the parameter FALSE inside of the getGenomeSize method which removed part of the error but generated a different set of errors. Focusing on these errors will be my goal for next time.