Work Log for August

Logan Brown

August 27, 2014

4 Week of August 23rd-30th

4.1 Goals for the Week

1. Finish analyzing cubappr.r (especially the MCMC after line 158)
2. Look at the consequences if model is "nsef" instead of "roc"
3. REU Results
4. Data Structures in the R manual (list?)
5. **Liz Howell Code**
6. **Run CUBFITS**

4.2 Progress / Notes

4.2.1 Finish analyzing cubappr.r (especially the MCMC after line 158)

4.2.2 Look at the consequences if model is "nsef" instead of "roc"

4.2.3 REU Results

4.2.4 Data Structures in the R manual (list?)

4.2.5 Liz Howell Code

"*The code I mentioned to Liz Howell that I need you to write is very simple. Ideally it would be in a language that is more common than R, but the world is not ideal and I think doing it in R would be best for now.*

The code would take a observed DNA sequence for an given protein and output a pessimal (worst) and optimal sequence based on the delta eta values produced by cubfits. The pessimal sequence would use the codons with the longest pausing times (largest delta eta) and the optimal would use the codons with the shortest pausing times.

Note that in the current code, the table produced is always relative to the codon with the shortest pausing time, but if I would recommend not assuming that is always the

case. That is used min() and max() functions to ID the codons rather than ==0 and max().

To summarize, the code will take in a FASTA file with one or more genes and a delta eta file produced by cubfits. The output will be a FASTA file with the pessimal versions of the genes and a second FASTA file with the optimal versions of the genes.produce up to two output."

The code seems approachable. I can use basic.r to set up the reu13.df and phi.df data structures. The problem is, I need phi.obs, which is unclear. I'm going to try phi.df[,1].

~~COMPUTERS WENT DOWN.~~

3 step plan.

1. Using a given genome, generate Phi values (expression levels)
2. Using given Phi values and cubfits, generate δt values (pausing times)
3. Using δt values, generate optimal and pessimal genomes.

First priority is to generate δt values given Phi values. Then I'll work on the coding to begin and end the process.

4.2.6 Run Cubfits

~~seq.data ¡- read.seq(get.expath("seqi_200.fasta"))~~
~~phi.df ¡- read.phi.df(get.expath("phi_200.tsv"))~~
~~aa.names ¡- c("A", "C", "D")~~
~~seq.string ¡- convert.seq.data.to.string(seq.data)~~
~~reu13.df ¡- gen.reu13.df(seq.string, phi.df, aa.names)~~
~~reu13.list.new ¡- gen.reu13.list(seq.string, aa.names)~~
~~y ¡- gen.y(seq.string, aa.names)~~
~~n ¡- gen.n(seq.string, aa.names)~~
~~scuo ¡- gen.scuo(seq.string, aa.names)~~
~~phi.obs = phi.df[,2]~~

~~All except the last one came from demo/basic.r~~
~~The last one is added by me.~~

Last Error Error in .cubfitsEnv$my.stop(paste("mean(phi.Obs) =", mean(phi.Obs))) :

paste("mean(phi.Obs) =", mean(phi.Obs))) runs just fine, so it seems like an error with my.stop

R/my.stop.r has the problem. ~~It seems related to lapply, mclappaly, pdbMPI, and parallel?~~ phi.Obs must be NORMALIZED

Cedric has some scripts which generate and mold the data appropriately before a run. I've checked out the repository to cubfitsLocal/misc

4.2.7 FIXING GAULEY

1. Slow Login
2. Cannot input password at login screen
3. slow tab completion and non-commands

Fixed by

4.3 Goals for next Week

1. Future Goal