20 August 20th

20.1 Goals for the day

Goals from Last Time

1. *Continue analyzing my.cubappr.r*
2. *Look for Wei-Chen code on the NSE model*
3. *Look at REU results (esp. a1 discussion?)*
4. *Look into workflow programs for R*
5. *Read about Data Structures in R user manual*

Additional Goals

6. Automate parts of the .tex process

20.2 Progress/Notes

20.2.1 Continue analyzing my.cubappr.r

Wei-Chen included a breakdown of all the arguments to the cubappr.r in cubfits.pdf page 14. He also explains the data formats on page 19.

b is apparently... ”A named list A contains amino acids. Each element of the list A[[i]] is a list of elements coefficients (coefficients of log(mu) and Delta.t), coef.mat (matrix format of coefficients), and R (covariance matrix of coefficients). Note that coefficients and R are typically as in the output of vglm() of VGAM package. Also, coef.mat and R may miss in some cases. e.g. A[[i]]$coef.mat is the regression beta matrix of i-th amino acid.”

What is logL? Log likelihood

Here’s the values for the variables, instead of using that confusing .CF.CONF$ values

reu13.df.obs	=	some huge matrix that looks like codon usage levels.
phi.pred.Init	=	matrix containing phi guesses
y	=	a vector that has codon counts
nIter	=	20
b.Init	=	NULL
init.b.Scale	=	1
b.drawScale	=	1
b.RInit	=	NULL
p.Init	=	NULL
p.nclass	=	2
p.DrawScale	=	0.1
phi.pred.DrawScale	=	1
model	=	“roc”
adaptive	=	“logonormal”
verbose	=	TRUE
iterThin	=	1
report	=	5

It’s notable that the code frequently calls things as “variable[1]”, instead of just calling ”variable”, even if the variable only has one item in it. Ex. model[1] is totally unnecessary, model is just “roc”.

b.Mat is a 10 by 21 matrix (nBparams by nSave or (# of regression parameters) by (space for iterations)). Looks like it will contain log(mu) and Delta.t

p.Mat is a 2 by 21 matrix (nPrior (# of prior params) by nSave)

phi.pred.Mat is a 100 by 21 matrix (n.G (number of Genes) by nSave)

logL.Mat is a 1 by 21 matrix (1 by nSave)

Mostly everything before the MCMC is just initializing variables, moving things around in memory.

my.DrawBConditionalAll is my.drawBConditionalAll.RW_NORM

I got to line 161 (including my comments 158 without them): ”p.Curr <- .cubfitsEnv$my.pPropTypeNoObs(n.G, phi.Curr, p.Curr, hp.param)”

20.2.2 Look for Wei-Chen code on the NSE model

I found the code in the CRAN cubfits installation, in demo. It’s nearly identical to the roc.appr demo, but model is set to ”nsef” instead of ”roc”.

20.2.3 Look at REU results

20.2.4 Look into workflow programs for R

So far, I’ve replaced this with R’s build in Debug program.

1. library(cubfits)
2. debug(cubappr)
3. demo(roc.appr, 'cubfits')

20.2.5 Read about Data Structures in R user manual

20.2.6 Automate parts of the .tex process

wrote writeGoals.cpp, which should save me an immense amount of time spent formatting.

20.3 Future Goals

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?)