Work Log for August

Logan Brown

August 27, 2014

Contents

11 August 11th

11.1 Goals

1. Install LaTeX
2. Local CUBFits Installation
3. Preliminary Understanding of
   - Sella and Hirch paper - link biology and statistical mechanics concepts
   - GenomeGroup paper - fit the model knowing the expression levels
   - Wallace et al. paper? - MCMC stuff
   - Murray et al. paper? - framework

11.2 Progress/Notes

11.2.1 LaTeX installation

Began a LaTeX installation on Tremont, installed to the home directory. It should be usable on every computer.

I ran a full installation, which may have been a mistake. It took at least 7 hours, it was still installing when I left.

11.2.2 Local CUBFits Installation

I locally installed CUBFits, as well as the prerequisites (seqinr, VGAM, doSNOW, coda, EMCluster) to my home directory.

1. R
2. .libPaths"~/cubfitsLocal" #add local installation to library path
3. library(cubfits) #load the cubfits library
4. demo(roc.train, 'cubfits')

Doing so fixes the problem of "acceptance not in range", but introduces a new problem, "iterations terminated because half-step sizes are very small". It seems that cubfits-master is installed on Gauley, but cubfits 0.1 fixes some of those problems.

```
> .libPaths("~/cubfitsLocal/")
> .libPaths()
[1] "/home/lbrown/cubfitsLocal"     "/usr/local/lib/R/site-library"
[3] "/usr/lib/R/site-library"       "/usr/lib/R/library"
> find.package(cubfits)
Error in find.package(cubfits) : object 'cubfits' not found
```

```
> find.package("cubfits")
[1] "/home/lbrown/cubfitsLocal/cubfits"
> library(cubfits)
> demo(roc.train, 'cubfits')


        demo(roc.train)
        ---- ~~~~~~~~~

> start.time <- proc.time()

> suppressMessages(library(cubfits, quietly = TRUE))

> set.seed(1234)

> .CF.AC$renew.iter <- 3

> # .CF.CT$type.p <- "lognormal_bias"
> # .CF.CONF$scale.phi.Obs <- FALSE
> # .CF.CONF$estimate.bias.Phi <- TRUE
>  ex.train$phi.Obs <- ex.train$phi.Obs / mean(ex.train$phi.Obs)

> ret.time <- system.time({
+   ret <- cubfits(ex.train$reu13.df, ex.train$phi.Obs, ex.train$y, ex.train$n,
+                  nIter = 20,
+                  verbose = TRUE, report = 5,
+                  model = "roc", adaptive = "simple")
+ })
pid:             56149
start:           Tue Aug 12 13:29:00 2014
iter: 5          Tue Aug 12 13:29:00 2014
iter: 10         Tue Aug 12 13:29:00 2014
iter: 15         Tue Aug 12 13:29:00 2014
iter: 20         Tue Aug 12 13:29:00 2014
iter: 25         Tue Aug 12 13:29:00 2014
iter: 30         Tue Aug 12 13:29:00 2014
iter: 35         Tue Aug 12 13:29:00 2014
iter: 40         Tue Aug 12 13:29:00 2014
iter: 45         Tue Aug 12 13:29:00 2014
iter: 50         Tue Aug 12 13:29:00 2014
iter: 55         Tue Aug 12 13:29:00 2014
iter: 60         Tue Aug 12 13:29:00 2014
iter: 65         Tue Aug 12 13:29:00 2014
```

```
iter: 70         Tue Aug 12 13:29:00 2014
iter: 75         Tue Aug 12 13:29:00 2014
iter: 80         Tue Aug 12 13:29:00 2014
iter: 85         Tue Aug 12 13:29:00 2014
iter: 90         Tue Aug 12 13:29:00 2014
iter: 95         Tue Aug 12 13:29:00 2014
iter: 100        Tue Aug 12 13:29:00 2014
           user.self sys.self elapsed user.child sys.child
total.time   0.29200    8e-03 0.29900          0         0
avg.time     0.00292    8e-05 0.00299          0         0
iter: 105        Tue Aug 12 13:29:00 2014
iter: 110        Tue Aug 12 13:29:00 2014
iter: 115        Tue Aug 12 13:29:00 2014
iter: 120        Tue Aug 12 13:29:00 2014

> print(ret.time)
   user  system elapsed
  0.816   0.028   0.935

> x <- rowMeans(do.call("cbind", ret$phi.Mat)[, 11:20])

> y <- ex.train$phi.Obs

> x <- log10(x / mean(x))

> y <- log10(y / mean(y))

> print(mean(x))
[1] -0.4369554

> print(summary(lm(y ~ x))$r.squared)
[1] 0.8932789

> # warning: iterations terminated because half-step sizes are very small
>
> print(proc.time() - start.time)
   user  system elapsed
  0.864   0.028   0.986
```

11.2.3 Readings

First read of the Stella and Hirch paper. It helped develop my intuitions regarding CUB-Fits/SEMPRR, though it seems to be only tangentially related. It's good for drawing these connections, which will help with understanding and analyzing the code (indirectly)

11.3 Future Goals

1. Test LaTeX installation
2. Continue Readings
3. Continue analysing Gauley errors?

13 August 13th

13.1 Goals for the day

Goals from Last Time

1. *Test LaTeX installation*
2. *Continue Readings*
3. *Continue analysing Gauley errors?*

Additional Goals

4. Look at the code
5. Look into potential Tremont errors?
6. Talk to Cedric about errors

13.2 Progress/Notes

13.2.1 Test LaTeX installation

LaTeX installation works?

Compiles, works just fine. Was a slight adjustment from the windows MikTex GUI, but largely the same. Developed the format used here.

13.2.2 Readings

13.2.3 Gauley Errors

Cedric says that the Gauley errors AREN'T ERRORS. It's actually just a regular part of the code output. Also the "#warning:iterations terminated because half-step sizes are very small" isn't a problem. Copasetic.

13.2.4 Code Examination

It seems like I should examing roc.appr.r first. any *appr* is for approximating the codon bias without knowing the expression levels, which is my goal.

What does roc.appr.r do?

1. Get initial ϕ value from data/ex.test.rda
2. Change renew.iter variable from 100 to 3 in .CF.AC, which is in data/control.r
3. Calls cubappr to do the actual approximations. Times this process. cubappr is exported to mycubappr in mycubappr.r

(a) Setup model and check data, including data structures and storage

(b) Initialize ϕ using my.pInit in my.init_param.r. Uses EMCluster iff estimate.bias.Phi (false by default)

(c) Initialize β, a vector of amino acids

(d) Runs MCMC, starting at line 124

i. my.drawBConditionalAll Updates β. Seems quite important. uses VGAM? Further study

ii. Other parameters: nu.Phi and sigma.Phi

iii.

13.2.5 Tremont Errors?

Likely because I was using the cubfits-0.1.0, the version on CRAN, while I should be using cubfits-0.1.1, the version in cubfits-master.tar.gz Moving to Gauley?

13.3 Future Goals

1. Continue analyzing cubfits-master/R files, especially mycubappr.r
   (a) Especially study the MCMC from mycubappr.r 124-196
   (b) nu.Phi and sigma.Phi?
2. Read REU Paper on Nonsense Model

15 August 15th

15.1 Goals for the day

Goals from Last Time

1. *Continue analyzing cubfits-master/R files, especially mycubappr.r*
   (a) *Especially study the MCMC from mycubappr.r 124-196*
   (b) *nu.Phi and sigma.Phi?*
2. *Read REU Paper on Nonsense Model*

Additional Goals

15.2 Progress/Notes

15.2.1 Continue analyzing cubfits-master/R files, especially mycubappr.r

I want to analyze the MCMC that happens in my.cubappr.r, which is only about 50 lines of code in my.cubappr.r, but it expands more.

This function calls my.drawBConditionalAll... but that function is gotten from .cubfitsEnv. It could be .current, .random, or .RW_Norm. Only .random uses VGAM, so that's relevant. I can't tell from my.cubfitsappr.r or roc.appr.r which is being called. The best next step is to probably put in print statements for each of these splits and see which is called.

my.drawPhiConditionalAll is from .cubfitsEnv as well, but it looks like there is only one .drawPhiConditionalAll. Will put in a print statement just to double check.

Understanding the workflow is key to understanding the code.

15.2.2 Read REU Paper on Nonsense Model

Equation 2 needed explanation.

$$\eta(\vec{c}) = \frac{\sum_{i=1}^{n+1} \beta_{i-1} p_i \sigma_{i-1}}{\sigma_n}$$

Where

β_i = cost of a codon = $a_1 + a_2 i$

p_i = probability of an error at codon i

σ_i = probablity of succesfulling reaching codon $i = \prod_{j=1}^{i}(1 - p_j)$

We peel off the n^{th} term.

$$\eta(\vec{c}) = \beta_n + \frac{\sum_{i=1}^{n} \beta_{i-1} p_i \sigma_{i-1}}{\sigma_n}$$

Substitute σ

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} p_i \frac{\prod_{k=1}^{i-1} 1 - p_i}{\prod_{k=1}^{n} 1 - p_k}$$

Divide out the $\prod$ term

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} p_i \frac{1}{\prod_{k=i}^{n} 1 - p_k}$$

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} \frac{p_i}{1 - p_i} \prod_{k=i+1}^{n} \frac{1}{1 - p_k}$$

A simplification on the $\prod$

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} \frac{p_i}{1-p_i} \prod_{k=i+1}^{n} \frac{1-p_k+p_k}{1-p_k}$$

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} \frac{p_i}{1-p_i} \prod_{k=i+1}^{n} (1+\frac{p_k}{1-p_k})$$

The version in the REU paper, expanding β

$$\eta(\vec{c}) = a_1 + a_2 n + \sum_{i=1}^{n} (a_1 + a_2(i-1)) \frac{p_i}{1-p_i} \prod_{k=i+1}^{n} (1+\frac{p_k}{1-p_k})$$

Easy to read version using $\beta_i = a_1 + a_2 i$ and $\omega_i = \frac{p_i}{1-p_k}$

$$\eta(\vec{c}) = \beta_n + \sum_{i=1}^{n} \beta_{i-1} \omega_i \prod_{k=i+1}^{n} (1+\omega_k)$$

Take the first order taylor approximation, about $p_i = 0$, then apply that to the fitness function ((1) in the paper), and we get the main function

$$Pr(\vec{c}|\phi) \propto \prod_{i=1}^{n} \exp[\ln\mu_i + \omega_i(a_1+a_2)y_1 + \omega_i a_2 y_1 i]$$

The rest of the paper has their results, which might be important to look at, in the future. For now, the important part is to understand how the equations work, and then how the code works.

15.3 Future Goals

1. Analyze my.cubappr.r
   (a) Add in print statements, then rerun the example
   (b) my.drawBConditionalAll.??????
2. (Optional) Study R user manual more?