BMI 651

Joshua Burkhart

February 29, 2016

HW5: APPENDIX

1

1A

Transition matrix for two-state model:

```r
AT = c(.80,.15)
GC = c(.20,.85)
df = data.frame(AT,GC)
row.names(df) <- c("AT","GC")
kable(df,row.names=TRUE)
```

	AT	GC
AT	0.80	0.20
GC	0.15	0.85

```r
TM <- matrix(c(.8, .2,
               .15,.85),ncol=2,byrow=TRUE)
```

1B

1Bi

Emission matrix for AT-rich regions:

```r
P = c(.3,.1,.2,.4)
df = data.frame(P)
row.names(df) <- c("A","C","G","T")
kable(df,row.names=TRUE)
```

	P
A	0.3
C	0.1
G	0.2
T	0.4

```
AT_EM <- matrix(c(.3,
                  .1,
                  .2,
                  .4),ncol=1,byrow=TRUE)
```

1Bii

Emission matrix for GC-rich regions:

```
P = c(.05,.30,.55,.10)
df = data.frame(P)
row.names(df) <- c("A","C","G","T")
kable(df,row.names=TRUE)
```

	P
A	0.05
C	0.30
G	0.55
T	0.10

```
GC_EM <- matrix(c(.05,
                  .3,
                  .55,
                  .1),ncol=1,byrow=TRUE)
```

1C

```
GenerateNucSeq <- function(num_nucs)
{
  #set initial state using a uniform random distribution
  state <- sample(x=c("AT","GC"),size=1)

  nuc_seq <- character()
  for(i in 1:num_nucs)
  {
    if(state == "AT"){
      state <- sample(x=c("AT","GC"),size=1,prob=TM[1,])
      nuc <- sample(x=c("A","C","G","T"),size=1,prob=AT_EM)
    }
    else if(state == "GC")
    {
      state <- sample(x=c("AT","GC"),size=1,prob=TM[2,])
      nuc <- sample(x=c("A","C","G","T"),size=1,prob=GC_EM)
    }
    nuc_seq <- c(nuc_seq,nuc)
  }
  return(nuc_seq)
}
```

```
GenerateNucSeq(100)
```

```
  [1] "A" "G" "T" "A" "A" "T" "T" "G" "G" "G" "G" "G" "G" "C" "A" "T" "T"
 [18] "T" "C" "A" "G" "G" "C" "G" "G" "C" "G" "G" "G" "T" "G" "G" "G" "G"
 [35] "G" "G" "T" "T" "T" "C" "T" "T" "T" "G" "G" "T" "G" "G" "G" "G" "G"
 [52] "G" "G" "G" "C" "A" "G" "T" "T" "G" "C" "G" "T" "A" "C" "T" "C" "C"
 [69] "A" "A" "G" "T" "G" "C" "A" "T" "T" "C" "T" "G" "G" "G" "G" "C" "C"
 [86] "C" "G" "G" "G" "G" "C" "A" "C" "G" "C" "G" "G" "C" "A" "C"
```

1D

```
AT_DICT <- list(A=.30,
                C=.10,
                G=.20,
                T=.40)

GC_DICT <- list(A=.05,
                C=.30,
                G=.55,
                T=.10)

GenerateStateSeq <- function(nuc_seq)
{
  at_state_prob <- AT_DICT[[nuc_seq[1]]] # P(nuc_seq[1]|AT)
  gc_state_prob <- GC_DICT[[nuc_seq[1]]] # P(nuc_seq[1]|GC)

  at_state_trace <- sample(x=c("AT","GC"),size=1)
  gc_state_trace <- sample(x=c("AT","GC"),size=1)

  for(i in 2:length(nuc_seq))
  {
    # AT
    at_state_prob <- c(at_state_prob,max(
      at_state_prob[i-1] * TM[1,1] * AT_DICT[[nuc_seq[i]]], # P(AT|AT) * P(qi|AT)
      gc_state_prob[i-1] * TM[1,2] * GC_DICT[[nuc_seq[i]]])) # P(GC|AT) * P(qi|GC)

    at_state_trace <- c(at_state_trace,if(at_state_prob[i-1] * TM[1,1] >
                                          gc_state_prob[i-1] * TM[1,2]) "AT" else "GC")

    # GC
    gc_state_prob <- c(gc_state_prob,max(
      gc_state_prob[i-1] * TM[2,2] * GC_DICT[[nuc_seq[i]]], # P(GC|GC) * P(qi|GC)
      at_state_prob[i-1] * TM[2,1] * AT_DICT[[nuc_seq[i]]])) # P(AT|GC) * P(qi|AT)

    gc_state_trace <- c(gc_state_trace,if(gc_state_prob[i-1] * TM[2,2] >
                                          at_state_prob[i-1] * TM[2,1]) "GC" else "AT")
  }
  return(if(prod(at_state_prob) > prod(gc_state_prob))
            at_state_trace else gc_state_trace)
}
```

```r
GenerateStateSeq(c("A", "A", "G", "C", "G", "T", "G", "G", "G", "G", "C", "C", "C", "C", "G",
                   "G", "C", "G", "A", "C", "A", "T", "G", "G", "G", "G", "T", "G", "T", "C"))
```

```
 [1] "AT" "AT" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC"
[15] "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC" "GC"
[29] "GC" "GC"
```