SIG\_getseqs.R

koesterj

Thu Dec 11 12:26:18 2014

rm(list=ls(all=TRUE)) # clears workspace of ALL variables etc, use with caution  
  
library(seqinr)

## Warning: package 'seqinr' was built under R version 3.1.2

## Loading required package: ade4

load("/Users/koesterj/Dropbox/julie/Tps7\_2\_08\_2013/SexGenes/full.table.sig3.9724.rda")  
  
## list of 7 dataframes with order 1007, 1012, 1013, 1014, 1015, 3367, 1335  
t<- full.table.sig3.9724  
  
## get the reference seq  
ref <- paste(subset(t[[1]][,'Ref'], t[[1]][,'exon']==TRUE), collapse = '')  
ref

## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCCGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCGGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGAGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAACATCAGACTGTACTGAGTTAG"

## which strains have how many SNPs  
sapply(t, function(x){sum(x[,'snp'])})

## [1] 4 4 0 4 4 1 4

## which strains have how many SNPs in exons  
sapply(t, function(x){sum(x[,'snp'] & x[,'exon']==TRUE)})

## [1] 4 4 0 4 4 0 4

#what are the gene coordinates for SNPs in exons  
lapply(t, function(x){which(x[,'snp']==1 & x[,'exon']==TRUE)})

## [[1]]  
## [1] 238 541 589 691  
##   
## [[2]]  
## [1] 238 541 589 691  
##   
## [[3]]  
## integer(0)  
##   
## [[4]]  
## [1] 238 541 589 691  
##   
## [[5]]  
## [1] 238 541 589 691  
##   
## [[6]]  
## integer(0)  
##   
## [[7]]  
## [1] 238 541 589 691

## how long is the gene region  
length(t[[1]][,'Ref'])

## [1] 712

## how long is coding seq  
sub <- subset(t[[1]][,'Ref'],t[[1]][,'exon']==TRUE)  
length(sub)

## [1] 636

## how many aa are there  
aa <- length(sub)/3  
aa

## [1] 212

## what is the average coverage for each strain  
sapply(t, function(x) {sum(x[,'Cov'])/(length(sub))})

## [1] 47.06 89.28 89.84 39.48 86.73 77.46 170.51

## copy alternate base into refseq and place in new list of dataframes  
t2 <- lapply(t, function(x) {  
 within(x, Ref <- ifelse(snp==1 & exon==TRUE & a/Cov > 0.1 & a > 0, 'a',  
 ifelse(snp==1 & exon==TRUE & g/Cov > 0.1 & g > 0, 'g',  
 ifelse(snp==1 & exon==TRUE & c/Cov > 0.1 & c > 0, 'c',  
 ifelse(snp==1 & exon==TRUE & t/Cov > 0.1 & t > 0, 't', Ref)))))  
})  
  
  
## double check that I changed the correct bases  
lapply(t2, function(x){  
 grep('[agct]', x[,'Ref'])  
})

## [[1]]  
## [1] 238 541 589 691  
##   
## [[2]]  
## [1] 238 541 589 691  
##   
## [[3]]  
## integer(0)  
##   
## [[4]]  
## [1] 238 541 589 691  
##   
## [[5]]  
## [1] 238 541 589 691  
##   
## [[6]]  
## integer(0)  
##   
## [[7]]  
## [1] 238 541 589 691

## list of actual sequences and make fasta  
seqs <- lapply(t2, function(x) {  
 paste(subset(x[,'Ref'], x[,'exon']==TRUE), collapse = '')  
})  
seqs

## [[1]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCtGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCtGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGgGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAgCATCAGACTGTACTGAGTTAG"  
##   
## [[2]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCtGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCtGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGgGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAgCATCAGACTGTACTGAGTTAG"  
##   
## [[3]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCCGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCGGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGAGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAACATCAGACTGTACTGAGTTAG"  
##   
## [[4]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCtGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCtGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGgGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAgCATCAGACTGTACTGAGTTAG"  
##   
## [[5]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCtGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCtGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGgGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAgCATCAGACTGTACTGAGTTAG"  
##   
## [[6]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCCGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCGGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGAGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAACATCAGACTGTACTGAGTTAG"  
##   
## [[7]]  
## [1] "ATGGTCAAAGCACTCCTCGTTGTTCTCGCGACACTTGTCGCAACTGCGGGGGCTCATTGTCCAAACGGATGCAAAGGGAACGGTTCCTGTGGCATCAATGACAAGTGCACTTGCTACTTGCGTCCCAATGGAGATCCTGCATGGACCGCACACGATTGTTCtGAGAGGACATGCCCTTATGGATCTGCATGGTCGTCTGAAACGACCAACGGTGCGAACGACGCTCATCCTCACGCCGAGTGCTCTAACAAAGGAACTTGCGACCGCAACAGTGGAGAATGTGTCTGCTTCGAGAACTACGATGGAAAGGCTTGCGAACGTACCCTGTGCCCTAACGATTGCTCTGGACGTGGAATCTGCTTGACTCAAAAGGCTCTCGCGATCTTCCAAGGAGCGACATACGAGACTCCGTGGGATGCCGAGAAGCACCTCGGTTGCAAGTGCGACGTTGGATACCGTGGTCCtGATTGTTCAAGGAAGGAATGCCCTTCCGGAGAAGATATTCTTGGTGGgGATGGTGCTGTCAAAGGTCGTGAATGCTCCGGACGTGGCAACTGCAATTTCATCACGGGGTTGTGCCAATGCTTTGACGGATACTTTGGAAACAAGTGTCAgCATCAGACTGTACTGAGTTAG"

write.fasta(seqs, names = c("1007", "1012", "1013", "1014", "1015", "3367", "1335"), nbchar = 60, file.out = "Tps7\_SIG3.9724.fasta", open = "w")  
write.fasta(ref, names = "ref", nbchar = 60, file.out = "Tps7\_SIG3.9724.fasta", open = "a")