Structural Prediction of Proteins

Chris McCoy

29 March 2017

# 1. Introduction

In living cells, proteins are the biological machines responsible for completing most of the biological functions. Being a versatile type of macromolecule, proteins have been found to be involved in virtually every cellular/biological process including, but not limited to; replicative and transcriptional roles with DNA, cellular division and metabolism, and cellular signalling (Alberts *et al*, 2002). To understand how a cell functions , it is therefore necessary to understand explicitly the mechanisms by which the individual proteins complete their specific roles in the cell.

Structural Biology is a branch of biophysics that focuses on the characterisation of the physical structure of biological molecules. Understanding the physical structure of the way a protein, from its secondary structure, to its specific folding to build up its tertiary and quaternary is a fundamental part of understanding how and why a particular protein functions and interacts within a cell.  
Proteins are made up of amino acids, which do not remain as a straight chain. These chains fold and twist to form secondary structures which in turn can orientate themselves in a particular fashion to create recognisable motifs. Although it is easy to determine a protein amino acid sequence, currently, there is no all-encompassing methodology or code to predict how a protein folds. Therefore, to determine protein structure, to gain knowledge on how a protein functions and how alterations in their structures can affect their function, experimental methods are needed. From this we can understand how to influence protein-protein interactions and mutations which can drastically change the behaviour of proteins. This is important in many branches of science, including structure based drug design for pharmaceuticals (Scapin, 2006).

Due to biomolecules, such as proteins, being too small to visualise in any detail using light microscopy techniques many structural biology techniques have been developed including; mass spectrometry, NMR, SAXS, X-ray crystallography etc. X-ray crystallography is a powerful structural tool, resulting in almost 10-fold the number of macromolecular structures than the nearest competing structural technique (RCSB Protein Data Bank, 2010), and is less restricted in terms of molecular size than other competing techniques.

Despite this success, x-ray crystallography has a number of drawbacks associated with the technique. The first of which is that the sample needs to be present in a large enough quantity to be studied. Secondly the sample needs to be in an incredibly pure state, and also needs to be structured enough (no intrinsically disordered domains) to form protein crystals. Rough estimates put the cloning a purification procedures up to six months and the crystallisation procedures at approximately 1-12months (Drenth, 2007).

Being able to predict protein/macromolecule structures before any structural biology techniques, based on the properties of the individual amino acids, and on existing protein structures would allow us to design constructs that are likely to be well behaved, easy to purify and well structured. This would serve to save a large amount of time instead of iteratively trying a large number of different constructs to see which works best.

Although numerous server-based structure prediction servers and programs already exist, inputting and running an entire protein sequence can be time consuming. Using R and the Chou Fasman structure prediction method (Chou *et al.*, 1974), we are able to quickly screen our protein sequence for regions that might be of interest, using more basic parameters. From here smaller sequences can be submitted to such programs for more in depth analysis or used directly as boundaries for cloning/purification/crystallisation. R's graphical interface allows us to easily see regions of likely structured protein, from numerous methods such as the PDB blast or a-helix/b-sheet prediction on one output allowing us to compare different prediction results, which would have required us to input our data into numerous prediction programs, in one program allowing us to easily compare the results of different prediction criteria, giving more certainty in our predictions. Also using the "seqinr" package and the "aaindex" we can continue to add even more criteria in the future when predicting structures. The current program currently only takes advantage of the propesity values of amino acids for a-helix or b-sheet but other criteria, such as amino acid hydrophobicity for example could easily be added.

# 2. Loading and formatting information from our gene of interest

rm(list=ls(all=TRUE))  
  
###############################################################################################  
################################## Installing Packages ########################################  
###############################################################################################  
  
  
install.packages("ggplot2", repos = "http://cran.us.r-project.org")

## package 'ggplot2' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Chris McCoy\AppData\Local\Temp\RtmpOCvvZv\downloaded\_packages

install.packages("seqinr",repos = "https://cran.rstudio.com/bin/windows/contrib/3.3/seqinr\_3.3-6.zip")

## Warning: unable to access index for repository https://cran.rstudio.com/bin/windows/contrib/3.3/seqinr\_3.3-6.zip/src/contrib:  
## cannot open URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/seqinr\_3.3-6.zip/src/contrib/PACKAGES'

## Warning: package 'seqinr' is not available (for R version 3.3.3)

## Warning: unable to access index for repository https://cran.rstudio.com/bin/windows/contrib/3.3/seqinr\_3.3-6.zip/bin/windows/contrib/3.3:  
## cannot open URL 'https://cran.rstudio.com/bin/windows/contrib/3.3/seqinr\_3.3-6.zip/bin/windows/contrib/3.3/PACKAGES'

install.packages("bio3d", repos = "http://cran.us.r-project.org")

## package 'bio3d' successfully unpacked and MD5 sums checked  
##   
## The downloaded binary packages are in  
## C:\Users\Chris McCoy\AppData\Local\Temp\RtmpOCvvZv\downloaded\_packages

###############################################################################################  
#################################### Loading Packages #########################################  
###############################################################################################  
  
   
library("ggplot2")  
library("seqinr") #https://cran.r-project.org/web/packages/seqinr/seqinr.pdf  
  
sessionInfo() #show packages loaded in the current R session

## R version 3.3.3 (2017-03-06)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 10 x64 (build 14393)  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.1252   
## [2] LC\_CTYPE=English\_United Kingdom.1252   
## [3] LC\_MONETARY=English\_United Kingdom.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] seqinr\_3.3-6 ggplot2\_2.2.1  
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.10 assertthat\_0.1 digest\_0.6.12 rprojroot\_1.2   
## [5] plyr\_1.8.4 grid\_3.3.3 gtable\_0.2.0 backports\_1.0.5   
## [9] magrittr\_1.5 evaluate\_0.10 scales\_0.4.1 stringi\_1.1.3   
## [13] lazyeval\_0.2.0 rmarkdown\_1.4 tools\_3.3.3 ade4\_1.7-6   
## [17] stringr\_1.2.0 munsell\_0.4.3 yaml\_2.1.14 colorspace\_1.3-2  
## [21] htmltools\_0.3.5 knitr\_1.15.1 tibble\_1.2

###############################################################################################  
##################################### Setting the WD ##########################################  
###############################################################################################  
  
setwd("C:/Users/Chris McCoy/Documents/Calgary/Modules/R Programming/Project/Project Directory") #set the working directory  
getwd() #get the working directory

## [1] "C:/Users/Chris McCoy/Documents/Calgary/Modules/R Programming/Project/Project Directory"

dir() #show files saved in the working directory

## [1] "CAC39247.fasta" "project1.html" "project1.Rmd"

###############################################################################################  
####################################### Loading Data ##########################################  
###############################################################################################  
  
data("aaindex", package= "seqinr") #Load List of 544 physicochemical and biological properties for the 20 AA from the seqinr package  
#aa.index   
   
#http://www.ebi.ac.uk/ena/data/view/CAC39247 #location of the FASTA file used as the example in this code. Can download FASTA for any gene of interest  
Smc5 <- read.fasta("CAC39247.fasta") #Read the downloaded and saved fasta file from working directory for gene of interest  
Smc5 #Prints fasta file

## $`ENA|CAC39247|CAC39247.1`  
## [1] "a" "t" "g" "g" "c" "g" "a" "c" "t" "c" "c" "g" "a" "g" "c" "a" "a"  
## [18] "g" "a" "a" "g" "a" "c" "g" "t" "c" "a" "a" "c" "t" "c" "c" "a" "a"  
## [35] "g" "c" "c" "c" "c" "c" "a" "g" "c" "c" "t" "t" "c" "c" "a" "a" "g"  
## [52] "a" "g" "a" "g" "c" "t" "c" "t" "c" "c" "c" "g" "a" "g" "a" "g" "a"  
## [69] "c" "c" "c" "t" "t" "c" "g" "t" "c" "g" "g" "a" "g" "g" "t" "c" "c"  
## [86] "c" "g" "a" "g" "c" "a" "a" "g" "a" "g" "g" "a" "a" "g" "a" "a" "t"  
## [103] "t" "c" "g" "g" "c" "c" "c" "c" "g" "c" "a" "g" "c" "t" "g" "c" "c"  
## [120] "g" "c" "t" "g" "t" "t" "g" "c" "a" "g" "t" "c" "g" "t" "c" "c" "g"  
## [137] "g" "g" "c" "c" "t" "t" "t" "c" "g" "t" "g" "g" "a" "a" "g" "g" "c"  
## [154] "t" "c" "t" "a" "t" "c" "g" "t" "c" "c" "g" "c" "a" "t" "c" "t" "c"  
## [171] "g" "a" "t" "g" "g" "a" "g" "a" "a" "c" "t" "t" "c" "c" "t" "a" "a"  
## [188] "c" "a" "t" "a" "t" "g" "a" "t" "a" "t" "t" "t" "g" "t" "g" "a" "a"  
## [205] "g" "t" "a" "t" "c" "t" "c" "c" "t" "g" "g" "a" "c" "c" "c" "c" "a"  
## [222] "c" "t" "t" "g" "a" "a" "t" "a" "t" "g" "a" "t" "c" "g" "t" "t" "g"  
## [239] "g" "a" "g" "c" "c" "a" "a" "t" "g" "g" "a" "a" "c" "a" "g" "g" "g"  
## [256] "a" "a" "g" "t" "c" "g" "a" "g" "c" "a" "t" "t" "g" "t" "g" "t" "g"  
## [273] "t" "g" "c" "c" "a" "t" "t" "t" "g" "c" "c" "t" "t" "g" "g" "t" "t"  
## [290] "t" "a" "g" "c" "t" "g" "g" "a" "a" "a" "a" "c" "c" "t" "g" "c" "t"  
## [307] "t" "t" "c" "a" "t" "g" "g" "g" "a" "c" "g" "a" "g" "c" "a" "g" "a"  
## [324] "t" "a" "a" "g" "g" "t" "t" "g" "g" "g" "t" "t" "t" "t" "t" "t" "g"  
## [341] "t" "g" "a" "a" "g" "a" "g" "a" "g" "g" "a" "t" "g" "t" "t" "c" "t"  
## [358] "a" "g" "a" "g" "g" "c" "a" "t" "g" "g" "t" "t" "g" "a" "a" "a" "t"  
## [375] "t" "g" "a" "a" "t" "t" "g" "t" "t" "c" "a" "g" "g" "g" "c" "t" "t"  
## [392] "c" "t" "g" "g" "a" "a" "a" "t" "c" "t" "t" "g" "t" "a" "a" "t" "c"  
## [409] "a" "c" "c" "c" "g" "t" "g" "a" "g" "a" "t" "t" "g" "a" "t" "g" "t"  
## [426] "g" "g" "c" "a" "a" "a" "a" "a" "a" "t" "c" "a" "g" "t" "c" "c" "t"  
## [443] "t" "t" "t" "g" "g" "t" "t" "c" "a" "t" "c" "a" "a" "c" "a" "a" "a"  
## [460] "a" "a" "a" "t" "c" "t" "a" "c" "a" "a" "c" "c" "c" "a" "g" "a" "a"  
## [477] "a" "a" "t" "a" "g" "t" "g" "g" "a" "a" "g" "a" "g" "a" "a" "a" "g"  
## [494] "t" "t" "g" "c" "a" "g" "c" "c" "t" "t" "a" "a" "a" "t" "a" "t" "t"  
## [511] "c" "a" "a" "g" "t" "g" "g" "g" "g" "a" "a" "t" "c" "t" "t" "t" "g"  
## [528] "c" "c" "a" "g" "t" "t" "t" "c" "t" "c" "c" "c" "t" "c" "a" "g" "g"  
## [545] "a" "c" "a" "a" "a" "g" "t" "t" "g" "g" "a" "g" "a" "a" "t" "t" "t"  
## [562] "g" "c" "t" "a" "a" "a" "c" "t" "c" "a" "g" "c" "a" "a" "a" "a" "t"  
## [579] "t" "g" "a" "a" "c" "t" "c" "c" "t" "c" "g" "a" "a" "g" "c" "c" "a"  
## [596] "c" "t" "g" "a" "a" "a" "a" "g" "t" "c" "a" "a" "t" "t" "g" "g" "t"  
## [613] "c" "c" "c" "c" "c" "a" "g" "a" "a" "a" "t" "g" "c" "a" "c" "a" "a"  
## [630] "a" "t" "a" "t" "c" "a" "c" "t" "g" "t" "g" "a" "a" "c" "t" "c" "a"  
## [647] "a" "a" "a" "a" "c" "t" "t" "a" "a" "g" "g" "g" "a" "g" "a" "a" "a"  
## [664] "g" "a" "a" "a" "a" "a" "c" "a" "g" "c" "t" "c" "g" "a" "g" "a" "c"  
## [681] "c" "t" "c" "a" "t" "g" "c" "a" "a" "a" "g" "a" "g" "a" "a" "a" "a"  
## [698] "c" "t" "g" "a" "g" "t" "a" "t" "c" "t" "a" "c" "a" "g" "a" "a" "a"  
## [715] "a" "t" "g" "g" "t" "t" "c" "a" "g" "a" "g" "g" "a" "a" "t" "g" "a"  
## [732] "a" "a" "g" "a" "t" "a" "t" "a" "a" "a" "c" "a" "a" "g" "a" "t" "g"  
## [749] "t" "g" "g" "a" "g" "a" "g" "g" "t" "t" "c" "t" "a" "t" "g" "a" "a"  
## [766] "c" "g" "g" "a" "a" "g" "c" "g" "a" "c" "a" "t" "t" "t" "a" "g" "a"  
## [783] "t" "t" "t" "a" "a" "t" "t" "g" "a" "g" "a" "t" "g" "c" "t" "t" "g"  
## [800] "a" "a" "g" "c" "a" "a" "a" "a" "a" "g" "g" "c" "c" "a" "t" "g" "g"  
## [817] "g" "t" "g" "g" "a" "a" "t" "a" "t" "g" "a" "a" "a" "a" "t" "g" "t"  
## [834] "t" "c" "g" "t" "c" "a" "g" "g" "a" "a" "t" "a" "t" "g" "a" "a" "g"  
## [851] "a" "a" "g" "t" "a" "a" "a" "a" "c" "t" "a" "g" "t" "t" "c" "g" "t"  
## [868] "g" "a" "c" "c" "g" "a" "g" "t" "g" "a" "a" "g" "g" "a" "a" "g" "a"  
## [885] "g" "g" "t" "c" "a" "g" "a" "a" "a" "a" "c" "t" "t" "a" "a" "a" "g"  
## [902] "a" "a" "g" "g" "g" "c" "a" "g" "a" "t" "t" "c" "c" "t" "a" "t" "a"  
## [919] "a" "c" "a" "c" "g" "t" "c" "g" "a" "a" "t" "t" "g" "a" "a" "g" "a"  
## [936] "a" "a" "t" "g" "g" "a" "a" "a" "a" "c" "g" "a" "g" "c" "g" "t" "c"  
## [953] "a" "c" "a" "a" "t" "t" "t" "g" "g" "a" "g" "g" "c" "t" "c" "g" "a"  
## [970] "a" "t" "c" "a" "a" "a" "g" "a" "a" "a" "a" "g" "g" "c" "a" "a" "c"  
## [987] "a" "g" "a" "t" "a" "t" "t" "a" "a" "g" "g" "a" "g" "g" "c" "a" "t"  
## [1004] "c" "t" "c" "a" "a" "a" "a" "a" "t" "g" "c" "a" "a" "a" "c" "a" "g"  
## [1021] "a" "a" "g" "c" "a" "a" "g" "a" "t" "g" "t" "t" "a" "t" "a" "g" "a"  
## [1038] "a" "a" "g" "g" "a" "a" "a" "g" "a" "t" "a" "a" "a" "c" "a" "t" "a"  
## [1055] "t" "t" "g" "a" "g" "g" "a" "a" "c" "t" "t" "c" "a" "g" "c" "a" "g"  
## [1072] "g" "c" "t" "t" "t" "a" "a" "t" "a" "g" "t" "a" "a" "a" "a" "c" "a"  
## [1089] "a" "a" "a" "t" "g" "a" "a" "g" "a" "g" "c" "t" "t" "g" "a" "c" "c"  
## [1106] "g" "a" "c" "a" "g" "a" "g" "g" "a" "g" "a" "a" "t" "a" "g" "g" "t"  
## [1123] "a" "a" "t" "a" "c" "c" "c" "g" "c" "a" "a" "a" "a" "t" "g" "a" "t"  
## [1140] "a" "g" "a" "g" "g" "a" "t" "t" "t" "g" "c" "a" "a" "a" "a" "t" "g"  
## [1157] "a" "a" "c" "t" "a" "a" "a" "g" "a" "c" "c" "a" "c" "g" "g" "a" "a"  
## [1174] "a" "a" "c" "t" "g" "c" "g" "a" "g" "a" "a" "t" "c" "t" "t" "c" "a"  
## [1191] "g" "c" "c" "c" "c" "a" "g" "a" "t" "t" "g" "a" "t" "g" "c" "c" "a"  
## [1208] "t" "t" "a" "c" "a" "a" "a" "t" "g" "a" "t" "c" "t" "g" "a" "g" "a"  
## [1225] "c" "g" "g" "a" "t" "t" "c" "a" "g" "g" "a" "t" "g" "a" "a" "a" "a"  
## [1242] "g" "g" "c" "a" "t" "t" "a" "t" "g" "t" "g" "a" "a" "g" "g" "c" "g"  
## [1259] "a" "a" "a" "t" "a" "a" "t" "t" "g" "a" "t" "a" "a" "g" "c" "g" "a"  
## [1276] "a" "g" "a" "g" "a" "g" "a" "g" "g" "g" "a" "a" "a" "c" "t" "c" "t"  
## [1293] "a" "g" "a" "g" "a" "a" "g" "g" "a" "g" "a" "a" "a" "a" "a" "g" "a"  
## [1310] "g" "t" "g" "t" "g" "g" "a" "c" "g" "a" "t" "c" "a" "t" "a" "t" "t"  
## [1327] "g" "t" "a" "c" "g" "t" "t" "t" "t" "g" "a" "c" "a" "a" "t" "c" "t"  
## [1344] "t" "a" "t" "g" "a" "a" "t" "c" "a" "g" "a" "a" "g" "g" "a" "a" "g"  
## [1361] "a" "t" "a" "a" "g" "c" "t" "a" "a" "g" "a" "c" "a" "g" "a" "g" "a"  
## [1378] "t" "t" "c" "c" "g" "t" "g" "a" "c" "a" "c" "g" "t" "a" "t" "g" "a"  
## [1395] "t" "g" "c" "t" "g" "t" "t" "t" "t" "a" "t" "g" "g" "c" "t" "a" "a"  
## [1412] "g" "a" "a" "a" "t" "a" "a" "c" "a" "g" "a" "g" "a" "c" "a" "a" "a"  
## [1429] "t" "t" "t" "a" "a" "a" "c" "a" "a" "a" "g" "a" "g" "t" "c" "t" "g"  
## [1446] "t" "g" "a" "g" "c" "c" "c" "a" "t" "a" "a" "t" "g" "c" "t" "c" "a"  
## [1463] "c" "g" "a" "t" "c" "a" "a" "t" "a" "t" "g" "a" "a" "a" "g" "a" "t"  
## [1480] "a" "a" "t" "a" "a" "a" "a" "a" "t" "g" "c" "c" "a" "a" "a" "t" "a"  
## [1497] "t" "a" "t" "t" "g" "a" "a" "a" "a" "t" "c" "a" "t" "a" "t" "t" "c"  
## [1514] "c" "a" "t" "c" "a" "a" "a" "t" "g" "a" "c" "t" "t" "a" "a" "g" "a"  
## [1531] "g" "c" "c" "t" "t" "t" "g" "t" "a" "t" "t" "t" "g" "a" "a" "a" "g"  
## [1548] "t" "c" "a" "a" "g" "a" "a" "g" "a" "t" "a" "t" "g" "g" "a" "g" "g"  
## [1565] "t" "t" "t" "t" "c" "c" "t" "c" "a" "a" "a" "g" "a" "g" "g" "t" "t"  
## [1582] "c" "g" "t" "g" "a" "c" "a" "a" "t" "a" "a" "a" "a" "a" "a" "t" "t"  
## [1599] "a" "a" "g" "a" "g" "t" "a" "a" "a" "t" "g" "c" "t" "g" "t" "t" "a"  
## [1616] "t" "t" "g" "c" "t" "c" "c" "c" "a" "a" "g" "a" "g" "t" "t" "c" "a"  
## [1633] "t" "a" "t" "g" "c" "a" "g" "a" "c" "a" "a" "a" "g" "c" "a" "c" "c"  
## [1650] "t" "t" "c" "a" "a" "g" "a" "t" "c" "t" "t" "t" "g" "a" "a" "t" "g"  
## [1667] "a" "a" "c" "t" "t" "a" "a" "a" "c" "a" "a" "t" "a" "c" "g" "g" "a"  
## [1684] "t" "t" "t" "t" "t" "c" "t" "c" "t" "t" "a" "t" "t" "t" "g" "a" "g"  
## [1701] "a" "g" "a" "a" "t" "t" "a" "t" "t" "t" "g" "a" "t" "g" "c" "a" "c"  
## [1718] "c" "t" "g" "a" "t" "c" "c" "t" "g" "t" "a" "a" "t" "g" "a" "g" "t"  
## [1735] "t" "a" "c" "c" "t" "t" "t" "g" "c" "t" "g" "t" "c" "a" "g" "t" "a"  
## [1752] "t" "c" "a" "t" "a" "t" "t" "c" "a" "t" "g" "a" "a" "g" "t" "t" "c"  
## [1769] "c" "t" "g" "t" "a" "g" "g" "a" "a" "c" "t" "g" "a" "a" "a" "a" "g"  
## [1786] "a" "c" "c" "a" "g" "a" "g" "a" "a" "a" "g" "a" "a" "t" "t" "g" "a"  
## [1803] "a" "c" "g" "g" "g" "t" "a" "a" "t" "a" "c" "a" "a" "g" "a" "a" "a"  
## [1820] "c" "c" "c" "g" "a" "t" "t" "a" "a" "a" "a" "c" "a" "g" "a" "t" "t"  
## [1837] "t" "a" "t" "a" "c" "a" "g" "c" "a" "g" "a" "a" "g" "a" "a" "a" "a"  
## [1854] "g" "t" "a" "t" "g" "t" "g" "g" "t" "g" "a" "a" "a" "a" "c" "t" "t"  
## [1871] "c" "t" "t" "t" "t" "t" "a" "t" "t" "c" "a" "a" "a" "c" "a" "a" "a"  
## [1888] "g" "t" "t" "a" "t" "t" "t" "c" "t" "a" "g" "t" "a" "a" "c" "a" "c"  
## [1905] "a" "t" "c" "t" "c" "t" "a" "a" "a" "a" "g" "t" "a" "g" "c" "g" "c"  
## [1922] "a" "g" "t" "t" "t" "c" "t" "c" "a" "c" "t" "g" "t" "c" "a" "c" "t"  
## [1939] "g" "t" "g" "g" "a" "c" "c" "t" "a" "g" "a" "g" "c" "a" "g" "a" "g"  
## [1956] "a" "a" "g" "a" "c" "a" "c" "t" "t" "a" "g" "a" "a" "g" "a" "a" "c"  
## [1973] "a" "g" "c" "t" "a" "a" "a" "g" "g" "a" "a" "a" "t" "t" "c" "a" "t"  
## [1990] "a" "g" "a" "a" "a" "a" "t" "t" "g" "c" "a" "a" "g" "c" "a" "g" "t"  
## [2007] "g" "g" "a" "t" "t" "c" "a" "g" "g" "g" "t" "t" "g" "a" "t" "t" "g"  
## [2024] "c" "c" "t" "t" "a" "c" "g" "t" "g" "a" "a" "a" "c" "a" "a" "g" "c"  
## [2041] "a" "a" "a" "c" "a" "t" "c" "t" "g" "g" "a" "g" "c" "a" "c" "a" "a"  
## [2058] "a" "g" "a" "c" "a" "a" "t" "g" "a" "a" "c" "t" "t" "a" "g" "a" "c"  
## [2075] "a" "a" "a" "a" "g" "a" "a" "g" "a" "a" "g" "g" "a" "g" "c" "t" "t"  
## [2092] "c" "t" "t" "g" "a" "g" "a" "g" "a" "a" "a" "a" "a" "c" "c" "a" "a"  
## [2109] "g" "a" "a" "a" "a" "g" "a" "c" "a" "a" "c" "t" "g" "g" "a" "a" "c"  
## [2126] "a" "a" "a" "a" "a" "a" "t" "c" "a" "g" "t" "t" "c" "c" "a" "a" "a"  
## [2143] "c" "t" "a" "g" "g" "a" "a" "g" "t" "t" "t" "a" "a" "a" "g" "c" "t"  
## [2160] "g" "a" "t" "g" "g" "a" "a" "c" "a" "g" "g" "a" "t" "a" "c" "t" "t"  
## [2177] "g" "c" "a" "a" "t" "c" "t" "t" "g" "a" "a" "g" "a" "g" "g" "a" "a"  
## [2194] "g" "a" "g" "c" "g" "a" "a" "a" "a" "g" "c" "a" "a" "g" "t" "a" "c"  
## [2211] "c" "a" "a" "a" "a" "t" "c" "a" "a" "a" "g" "a" "a" "a" "t" "a" "a"  
## [2228] "a" "t" "g" "t" "t" "c" "a" "a" "a" "a" "a" "g" "c" "g" "a" "a" "a"  
## [2245] "c" "t" "t" "g" "t" "t" "a" "c" "c" "g" "a" "a" "t" "t" "a" "a" "c"  
## [2262] "a" "a" "a" "c" "c" "t" "a" "a" "t" "a" "a" "a" "g" "a" "t" "t" "t"  
## [2279] "g" "t" "a" "c" "t" "t" "c" "t" "t" "t" "g" "c" "a" "t" "a" "t" "a"  
## [2296] "c" "a" "a" "a" "a" "a" "g" "t" "a" "g" "a" "t" "t" "t" "a" "a" "t"  
## [2313] "t" "c" "t" "c" "c" "a" "a" "a" "a" "t" "a" "c" "t" "a" "c" "a" "g"  
## [2330] "t" "g" "a" "t" "c" "t" "c" "t" "g" "a" "g" "a" "a" "g" "a" "a" "c"  
## [2347] "a" "a" "a" "t" "t" "a" "g" "a" "a" "t" "c" "a" "g" "a" "t" "t" "a"  
## [2364] "t" "a" "t" "g" "g" "c" "c" "g" "c" "a" "t" "c" "t" "t" "c" "a" "c"  
## [2381] "a" "a" "c" "t" "c" "c" "g" "t" "c" "t" "t" "a" "c" "a" "g" "a" "g"  
## [2398] "c" "a" "a" "c" "a" "t" "t" "t" "c" "a" "t" "t" "g" "a" "a" "t" "t"  
## [2415] "g" "g" "a" "t" "g" "a" "a" "a" "a" "t" "a" "g" "a" "c" "a" "g" "a"  
## [2432] "g" "a" "t" "t" "a" "t" "t" "g" "c" "a" "g" "a" "a" "a" "t" "g" "c"  
## [2449] "a" "a" "g" "g" "a" "a" "c" "t" "t" "a" "t" "g" "a" "a" "a" "a" "g"  
## [2466] "a" "g" "c" "t" "a" "g" "g" "c" "a" "a" "g" "t" "a" "t" "g" "t" "a"  
## [2483] "a" "c" "c" "t" "g" "g" "g" "t" "g" "c" "a" "g" "a" "g" "c" "a" "g"  
## [2500] "a" "c" "t" "c" "t" "t" "c" "c" "t" "c" "a" "a" "g" "a" "a" "t" "a"  
## [2517] "c" "c" "a" "g" "a" "c" "a" "c" "a" "a" "g" "t" "a" "c" "c" "c" "a"  
## [2534] "c" "c" "a" "t" "t" "c" "c" "a" "a" "a" "t" "g" "g" "a" "c" "a" "c"  
## [2551] "a" "a" "c" "t" "c" "c" "t" "c" "a" "c" "t" "c" "c" "c" "c" "a" "t"  
## [2568] "g" "g" "t" "t" "t" "t" "c" "c" "a" "a" "g" "a" "c" "c" "t" "t" "c"  
## [2585] "c" "a" "a" "a" "c" "a" "c" "a" "t" "t" "g" "g" "a" "t" "g" "a" "a"  
## [2602] "a" "t" "t" "g" "a" "t" "g" "c" "t" "t" "t" "a" "t" "t" "a" "a" "c"  
## [2619] "t" "g" "a" "a" "g" "a" "a" "a" "g" "a" "t" "c" "a" "a" "g" "a" "g"  
## [2636] "c" "t" "t" "c" "c" "t" "g" "c" "t" "t" "c" "a" "c" "g" "g" "g" "a"  
## [2653] "c" "t" "g" "a" "a" "t" "c" "c" "t" "a" "c" "a" "a" "t" "t" "g" "t"  
## [2670] "t" "c" "a" "g" "g" "a" "a" "t" "a" "t" "a" "c" "a" "a" "a" "a" "a"  
## [2687] "g" "a" "g" "a" "a" "g" "a" "a" "g" "a" "a" "a" "t" "a" "g" "a" "a"  
## [2704] "c" "a" "g" "t" "t" "a" "a" "c" "t" "g" "a" "g" "g" "a" "a" "c" "t"  
## [2721] "a" "a" "a" "g" "g" "g" "a" "a" "a" "g" "a" "a" "a" "g" "t" "t" "g"  
## [2738] "a" "a" "c" "t" "a" "g" "a" "t" "c" "a" "a" "t" "a" "c" "a" "g" "g"  
## [2755] "g" "a" "a" "a" "a" "c" "a" "t" "t" "t" "c" "a" "c" "a" "g" "g" "t"  
## [2772] "a" "a" "a" "a" "g" "a" "a" "a" "g" "g" "t" "g" "g" "c" "t" "t" "a"  
## [2789] "a" "t" "c" "c" "t" "t" "t" "a" "a" "a" "a" "g" "a" "g" "c" "t" "g"  
## [2806] "g" "t" "a" "g" "a" "a" "a" "a" "a" "a" "t" "t" "a" "a" "t" "g" "a"  
## [2823] "a" "a" "a" "a" "t" "t" "c" "a" "g" "c" "a" "a" "t" "t" "t" "t" "t"  
## [2840] "t" "t" "a" "g" "t" "t" "c" "c" "a" "t" "g" "c" "a" "g" "t" "g" "t"  
## [2857] "g" "c" "t" "g" "g" "t" "g" "a" "a" "g" "t" "t" "g" "a" "t" "c" "t"  
## [2874] "c" "c" "a" "t" "a" "c" "a" "g" "a" "a" "a" "a" "t" "g" "a" "g" "g"  
## [2891] "a" "a" "g" "a" "t" "t" "a" "t" "g" "a" "t" "a" "a" "a" "t" "a" "t"  
## [2908] "g" "g" "a" "a" "t" "t" "c" "g" "a" "a" "t" "t" "a" "g" "a" "g" "t"  
## [2925] "c" "a" "a" "a" "t" "t" "t" "c" "g" "a" "a" "g" "t" "a" "g" "t" "a"  
## [2942] "c" "t" "c" "a" "a" "c" "t" "g" "c" "a" "t" "g" "a" "a" "t" "t" "a"  
## [2959] "a" "c" "t" "c" "c" "t" "c" "a" "t" "c" "a" "t" "c" "a" "a" "a" "g"  
## [2976] "t" "g" "g" "a" "g" "g" "t" "g" "a" "a" "a" "g" "a" "a" "g" "t" "g"  
## [2993] "t" "t" "t" "c" "t" "a" "c" "c" "a" "t" "g" "t" "t" "a" "t" "a" "c"  
## [3010] "t" "t" "g" "a" "t" "g" "g" "c" "a" "c" "t" "t" "c" "a" "g" "g" "a"  
## [3027] "g" "c" "t" "a" "a" "a" "t" "a" "g" "a" "t" "g" "t" "c" "c" "a" "t"  
## [3044] "t" "c" "a" "g" "a" "g" "t" "a" "g" "t" "t" "g" "a" "t" "g" "a" "a"  
## [3061] "a" "t" "c" "a" "a" "t" "c" "a" "g" "g" "g" "a" "a" "t" "g" "g" "a"  
## [3078] "c" "c" "c" "a" "a" "t" "c" "a" "a" "t" "g" "a" "a" "c" "g" "g" "a"  
## [3095] "g" "a" "g" "t" "g" "t" "t" "t" "g" "a" "a" "a" "t" "g" "g" "t" "t"  
## [3112] "g" "t" "a" "a" "a" "t" "a" "c" "t" "g" "c" "c" "t" "g" "t" "a" "a"  
## [3129] "a" "g" "a" "a" "a" "a" "t" "a" "c" "a" "t" "c" "t" "c" "a" "a" "t"  
## [3146] "a" "c" "t" "t" "t" "t" "t" "c" "a" "t" "a" "a" "c" "a" "c" "c" "a"  
## [3163] "a" "a" "g" "c" "t" "c" "c" "t" "g" "c" "a" "a" "a" "a" "t" "c" "t"  
## [3180] "t" "c" "c" "t" "t" "a" "t" "t" "c" "t" "g" "a" "a" "a" "a" "g" "a"  
## [3197] "t" "g" "a" "c" "a" "g" "t" "t" "t" "t" "g" "t" "t" "t" "g" "t" "c"  
## [3214] "t" "a" "c" "a" "a" "t" "g" "g" "c" "c" "c" "t" "c" "a" "t" "a" "t"  
## [3231] "g" "c" "t" "g" "g" "a" "a" "c" "c" "a" "a" "a" "c" "a" "c" "a" "t"  
## [3248] "g" "g" "a" "a" "t" "t" "t" "a" "a" "a" "g" "g" "c" "t" "t" "t" "c"  
## [3265] "c" "a" "a" "a" "g" "g" "c" "g" "g" "c" "g" "g" "c" "g" "c" "c" "g"  
## [3282] "t" "a" "t" "t" "a" "c" "a" "t" "t" "c" "a" "c" "t" "c" "a" "a" "c"  
## [3299] "c" "t" "t" "c" "t" "t" "a" "a"  
## attr(,"name")  
## [1] "ENA|CAC39247|CAC39247.1"  
## attr(,"Annot")  
## [1] ">ENA|CAC39247|CAC39247.1 Homo sapiens (human) SMC5 protein "  
## attr(,"class")  
## [1] "SeqFastadna"

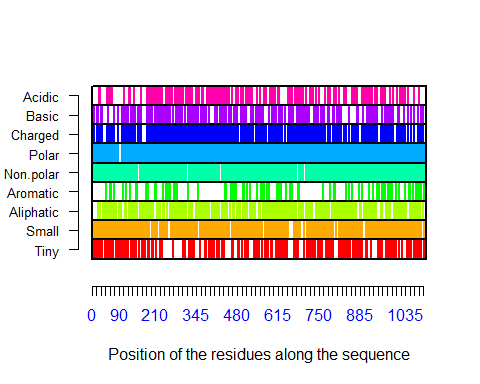
dna\_seq <- Smc5$`ENA|CAC39247|CAC39247.1` #stores the DNA sequence from the FASTA file as an object in environment  
dna\_seq <- as.vector(dna\_seq) #Takes DNA sequence from dna\_seq object and saves as a character vector  
dna\_seq #Prints dna\_seq variable

## [1] "a" "t" "g" "g" "c" "g" "a" "c" "t" "c" "c" "g" "a" "g" "c" "a" "a"  
## [18] "g" "a" "a" "g" "a" "c" "g" "t" "c" "a" "a" "c" "t" "c" "c" "a" "a"  
## [35] "g" "c" "c" "c" "c" "c" "a" "g" "c" "c" "t" "t" "c" "c" "a" "a" "g"  
## [52] "a" "g" "a" "g" "c" "t" "c" "t" "c" "c" "c" "g" "a" "g" "a" "g" "a"  
## [69] "c" "c" "c" "t" "t" "c" "g" "t" "c" "g" "g" "a" "g" "g" "t" "c" "c"  
## [86] "c" "g" "a" "g" "c" "a" "a" "g" "a" "g" "g" "a" "a" "g" "a" "a" "t"  
## [103] "t" "c" "g" "g" "c" "c" "c" "c" "g" "c" "a" "g" "c" "t" "g" "c" "c"  
## [120] "g" "c" "t" "g" "t" "t" "g" "c" "a" "g" "t" "c" "g" "t" "c" "c" "g"  
## [137] "g" "g" "c" "c" "t" "t" "t" "c" "g" "t" "g" "g" "a" "a" "g" "g" "c"  
## [154] "t" "c" "t" "a" "t" "c" "g" "t" "c" "c" "g" "c" "a" "t" "c" "t" "c"  
## [171] "g" "a" "t" "g" "g" "a" "g" "a" "a" "c" "t" "t" "c" "c" "t" "a" "a"  
## [188] "c" "a" "t" "a" "t" "g" "a" "t" "a" "t" "t" "t" "g" "t" "g" "a" "a"  
## [205] "g" "t" "a" "t" "c" "t" "c" "c" "t" "g" "g" "a" "c" "c" "c" "c" "a"  
## [222] "c" "t" "t" "g" "a" "a" "t" "a" "t" "g" "a" "t" "c" "g" "t" "t" "g"  
## [239] "g" "a" "g" "c" "c" "a" "a" "t" "g" "g" "a" "a" "c" "a" "g" "g" "g"  
## [256] "a" "a" "g" "t" "c" "g" "a" "g" "c" "a" "t" "t" "g" "t" "g" "t" "g"  
## [273] "t" "g" "c" "c" "a" "t" "t" "t" "g" "c" "c" "t" "t" "g" "g" "t" "t"  
## [290] "t" "a" "g" "c" "t" "g" "g" "a" "a" "a" "a" "c" "c" "t" "g" "c" "t"  
## [307] "t" "t" "c" "a" "t" "g" "g" "g" "a" "c" "g" "a" "g" "c" "a" "g" "a"  
## [324] "t" "a" "a" "g" "g" "t" "t" "g" "g" "g" "t" "t" "t" "t" "t" "t" "g"  
## [341] "t" "g" "a" "a" "g" "a" "g" "a" "g" "g" "a" "t" "g" "t" "t" "c" "t"  
## [358] "a" "g" "a" "g" "g" "c" "a" "t" "g" "g" "t" "t" "g" "a" "a" "a" "t"  
## [375] "t" "g" "a" "a" "t" "t" "g" "t" "t" "c" "a" "g" "g" "g" "c" "t" "t"  
## [392] "c" "t" "g" "g" "a" "a" "a" "t" "c" "t" "t" "g" "t" "a" "a" "t" "c"  
## [409] "a" "c" "c" "c" "g" "t" "g" "a" "g" "a" "t" "t" "g" "a" "t" "g" "t"  
## [426] "g" "g" "c" "a" "a" "a" "a" "a" "a" "t" "c" "a" "g" "t" "c" "c" "t"  
## [443] "t" "t" "t" "g" "g" "t" "t" "c" "a" "t" "c" "a" "a" "c" "a" "a" "a"  
## [460] "a" "a" "a" "t" "c" "t" "a" "c" "a" "a" "c" "c" "c" "a" "g" "a" "a"  
## [477] "a" "a" "t" "a" "g" "t" "g" "g" "a" "a" "g" "a" "g" "a" "a" "a" "g"  
## [494] "t" "t" "g" "c" "a" "g" "c" "c" "t" "t" "a" "a" "a" "t" "a" "t" "t"  
## [511] "c" "a" "a" "g" "t" "g" "g" "g" "g" "a" "a" "t" "c" "t" "t" "t" "g"  
## [528] "c" "c" "a" "g" "t" "t" "t" "c" "t" "c" "c" "c" "t" "c" "a" "g" "g"  
## [545] "a" "c" "a" "a" "a" "g" "t" "t" "g" "g" "a" "g" "a" "a" "t" "t" "t"  
## [562] "g" "c" "t" "a" "a" "a" "c" "t" "c" "a" "g" "c" "a" "a" "a" "a" "t"  
## [579] "t" "g" "a" "a" "c" "t" "c" "c" "t" "c" "g" "a" "a" "g" "c" "c" "a"  
## [596] "c" "t" "g" "a" "a" "a" "a" "g" "t" "c" "a" "a" "t" "t" "g" "g" "t"  
## [613] "c" "c" "c" "c" "c" "a" "g" "a" "a" "a" "t" "g" "c" "a" "c" "a" "a"  
## [630] "a" "t" "a" "t" "c" "a" "c" "t" "g" "t" "g" "a" "a" "c" "t" "c" "a"  
## [647] "a" "a" "a" "a" "c" "t" "t" "a" "a" "g" "g" "g" "a" "g" "a" "a" "a"  
## [664] "g" "a" "a" "a" "a" "a" "c" "a" "g" "c" "t" "c" "g" "a" "g" "a" "c"  
## [681] "c" "t" "c" "a" "t" "g" "c" "a" "a" "a" "g" "a" "g" "a" "a" "a" "a"  
## [698] "c" "t" "g" "a" "g" "t" "a" "t" "c" "t" "a" "c" "a" "g" "a" "a" "a"  
## [715] "a" "t" "g" "g" "t" "t" "c" "a" "g" "a" "g" "g" "a" "a" "t" "g" "a"  
## [732] "a" "a" "g" "a" "t" "a" "t" "a" "a" "a" "c" "a" "a" "g" "a" "t" "g"  
## [749] "t" "g" "g" "a" "g" "a" "g" "g" "t" "t" "c" "t" "a" "t" "g" "a" "a"  
## [766] "c" "g" "g" "a" "a" "g" "c" "g" "a" "c" "a" "t" "t" "t" "a" "g" "a"  
## [783] "t" "t" "t" "a" "a" "t" "t" "g" "a" "g" "a" "t" "g" "c" "t" "t" "g"  
## [800] "a" "a" "g" "c" "a" "a" "a" "a" "a" "g" "g" "c" "c" "a" "t" "g" "g"  
## [817] "g" "t" "g" "g" "a" "a" "t" "a" "t" "g" "a" "a" "a" "a" "t" "g" "t"  
## [834] "t" "c" "g" "t" "c" "a" "g" "g" "a" "a" "t" "a" "t" "g" "a" "a" "g"  
## [851] "a" "a" "g" "t" "a" "a" "a" "a" "c" "t" "a" "g" "t" "t" "c" "g" "t"  
## [868] "g" "a" "c" "c" "g" "a" "g" "t" "g" "a" "a" "g" "g" "a" "a" "g" "a"  
## [885] "g" "g" "t" "c" "a" "g" "a" "a" "a" "a" "c" "t" "t" "a" "a" "a" "g"  
## [902] "a" "a" "g" "g" "g" "c" "a" "g" "a" "t" "t" "c" "c" "t" "a" "t" "a"  
## [919] "a" "c" "a" "c" "g" "t" "c" "g" "a" "a" "t" "t" "g" "a" "a" "g" "a"  
## [936] "a" "a" "t" "g" "g" "a" "a" "a" "a" "c" "g" "a" "g" "c" "g" "t" "c"  
## [953] "a" "c" "a" "a" "t" "t" "t" "g" "g" "a" "g" "g" "c" "t" "c" "g" "a"  
## [970] "a" "t" "c" "a" "a" "a" "g" "a" "a" "a" "a" "g" "g" "c" "a" "a" "c"  
## [987] "a" "g" "a" "t" "a" "t" "t" "a" "a" "g" "g" "a" "g" "g" "c" "a" "t"  
## [1004] "c" "t" "c" "a" "a" "a" "a" "a" "t" "g" "c" "a" "a" "a" "c" "a" "g"  
## [1021] "a" "a" "g" "c" "a" "a" "g" "a" "t" "g" "t" "t" "a" "t" "a" "g" "a"  
## [1038] "a" "a" "g" "g" "a" "a" "a" "g" "a" "t" "a" "a" "a" "c" "a" "t" "a"  
## [1055] "t" "t" "g" "a" "g" "g" "a" "a" "c" "t" "t" "c" "a" "g" "c" "a" "g"  
## [1072] "g" "c" "t" "t" "t" "a" "a" "t" "a" "g" "t" "a" "a" "a" "a" "c" "a"  
## [1089] "a" "a" "a" "t" "g" "a" "a" "g" "a" "g" "c" "t" "t" "g" "a" "c" "c"  
## [1106] "g" "a" "c" "a" "g" "a" "g" "g" "a" "g" "a" "a" "t" "a" "g" "g" "t"  
## [1123] "a" "a" "t" "a" "c" "c" "c" "g" "c" "a" "a" "a" "a" "t" "g" "a" "t"  
## [1140] "a" "g" "a" "g" "g" "a" "t" "t" "t" "g" "c" "a" "a" "a" "a" "t" "g"  
## [1157] "a" "a" "c" "t" "a" "a" "a" "g" "a" "c" "c" "a" "c" "g" "g" "a" "a"  
## [1174] "a" "a" "c" "t" "g" "c" "g" "a" "g" "a" "a" "t" "c" "t" "t" "c" "a"  
## [1191] "g" "c" "c" "c" "c" "a" "g" "a" "t" "t" "g" "a" "t" "g" "c" "c" "a"  
## [1208] "t" "t" "a" "c" "a" "a" "a" "t" "g" "a" "t" "c" "t" "g" "a" "g" "a"  
## [1225] "c" "g" "g" "a" "t" "t" "c" "a" "g" "g" "a" "t" "g" "a" "a" "a" "a"  
## [1242] "g" "g" "c" "a" "t" "t" "a" "t" "g" "t" "g" "a" "a" "g" "g" "c" "g"  
## [1259] "a" "a" "a" "t" "a" "a" "t" "t" "g" "a" "t" "a" "a" "g" "c" "g" "a"  
## [1276] "a" "g" "a" "g" "a" "g" "a" "g" "g" "g" "a" "a" "a" "c" "t" "c" "t"  
## [1293] "a" "g" "a" "g" "a" "a" "g" "g" "a" "g" "a" "a" "a" "a" "a" "g" "a"  
## [1310] "g" "t" "g" "t" "g" "g" "a" "c" "g" "a" "t" "c" "a" "t" "a" "t" "t"  
## [1327] "g" "t" "a" "c" "g" "t" "t" "t" "t" "g" "a" "c" "a" "a" "t" "c" "t"  
## [1344] "t" "a" "t" "g" "a" "a" "t" "c" "a" "g" "a" "a" "g" "g" "a" "a" "g"  
## [1361] "a" "t" "a" "a" "g" "c" "t" "a" "a" "g" "a" "c" "a" "g" "a" "g" "a"  
## [1378] "t" "t" "c" "c" "g" "t" "g" "a" "c" "a" "c" "g" "t" "a" "t" "g" "a"  
## [1395] "t" "g" "c" "t" "g" "t" "t" "t" "t" "a" "t" "g" "g" "c" "t" "a" "a"  
## [1412] "g" "a" "a" "a" "t" "a" "a" "c" "a" "g" "a" "g" "a" "c" "a" "a" "a"  
## [1429] "t" "t" "t" "a" "a" "a" "c" "a" "a" "a" "g" "a" "g" "t" "c" "t" "g"  
## [1446] "t" "g" "a" "g" "c" "c" "c" "a" "t" "a" "a" "t" "g" "c" "t" "c" "a"  
## [1463] "c" "g" "a" "t" "c" "a" "a" "t" "a" "t" "g" "a" "a" "a" "g" "a" "t"  
## [1480] "a" "a" "t" "a" "a" "a" "a" "a" "t" "g" "c" "c" "a" "a" "a" "t" "a"  
## [1497] "t" "a" "t" "t" "g" "a" "a" "a" "a" "t" "c" "a" "t" "a" "t" "t" "c"  
## [1514] "c" "a" "t" "c" "a" "a" "a" "t" "g" "a" "c" "t" "t" "a" "a" "g" "a"  
## [1531] "g" "c" "c" "t" "t" "t" "g" "t" "a" "t" "t" "t" "g" "a" "a" "a" "g"  
## [1548] "t" "c" "a" "a" "g" "a" "a" "g" "a" "t" "a" "t" "g" "g" "a" "g" "g"  
## [1565] "t" "t" "t" "t" "c" "c" "t" "c" "a" "a" "a" "g" "a" "g" "g" "t" "t"  
## [1582] "c" "g" "t" "g" "a" "c" "a" "a" "t" "a" "a" "a" "a" "a" "a" "t" "t"  
## [1599] "a" "a" "g" "a" "g" "t" "a" "a" "a" "t" "g" "c" "t" "g" "t" "t" "a"  
## [1616] "t" "t" "g" "c" "t" "c" "c" "c" "a" "a" "g" "a" "g" "t" "t" "c" "a"  
## [1633] "t" "a" "t" "g" "c" "a" "g" "a" "c" "a" "a" "a" "g" "c" "a" "c" "c"  
## [1650] "t" "t" "c" "a" "a" "g" "a" "t" "c" "t" "t" "t" "g" "a" "a" "t" "g"  
## [1667] "a" "a" "c" "t" "t" "a" "a" "a" "c" "a" "a" "t" "a" "c" "g" "g" "a"  
## [1684] "t" "t" "t" "t" "t" "c" "t" "c" "t" "t" "a" "t" "t" "t" "g" "a" "g"  
## [1701] "a" "g" "a" "a" "t" "t" "a" "t" "t" "t" "g" "a" "t" "g" "c" "a" "c"  
## [1718] "c" "t" "g" "a" "t" "c" "c" "t" "g" "t" "a" "a" "t" "g" "a" "g" "t"  
## [1735] "t" "a" "c" "c" "t" "t" "t" "g" "c" "t" "g" "t" "c" "a" "g" "t" "a"  
## [1752] "t" "c" "a" "t" "a" "t" "t" "c" "a" "t" "g" "a" "a" "g" "t" "t" "c"  
## [1769] "c" "t" "g" "t" "a" "g" "g" "a" "a" "c" "t" "g" "a" "a" "a" "a" "g"  
## [1786] "a" "c" "c" "a" "g" "a" "g" "a" "a" "a" "g" "a" "a" "t" "t" "g" "a"  
## [1803] "a" "c" "g" "g" "g" "t" "a" "a" "t" "a" "c" "a" "a" "g" "a" "a" "a"  
## [1820] "c" "c" "c" "g" "a" "t" "t" "a" "a" "a" "a" "c" "a" "g" "a" "t" "t"  
## [1837] "t" "a" "t" "a" "c" "a" "g" "c" "a" "g" "a" "a" "g" "a" "a" "a" "a"  
## [1854] "g" "t" "a" "t" "g" "t" "g" "g" "t" "g" "a" "a" "a" "a" "c" "t" "t"  
## [1871] "c" "t" "t" "t" "t" "t" "a" "t" "t" "c" "a" "a" "a" "c" "a" "a" "a"  
## [1888] "g" "t" "t" "a" "t" "t" "t" "c" "t" "a" "g" "t" "a" "a" "c" "a" "c"  
## [1905] "a" "t" "c" "t" "c" "t" "a" "a" "a" "a" "g" "t" "a" "g" "c" "g" "c"  
## [1922] "a" "g" "t" "t" "t" "c" "t" "c" "a" "c" "t" "g" "t" "c" "a" "c" "t"  
## [1939] "g" "t" "g" "g" "a" "c" "c" "t" "a" "g" "a" "g" "c" "a" "g" "a" "g"  
## [1956] "a" "a" "g" "a" "c" "a" "c" "t" "t" "a" "g" "a" "a" "g" "a" "a" "c"  
## [1973] "a" "g" "c" "t" "a" "a" "a" "g" "g" "a" "a" "a" "t" "t" "c" "a" "t"  
## [1990] "a" "g" "a" "a" "a" "a" "t" "t" "g" "c" "a" "a" "g" "c" "a" "g" "t"  
## [2007] "g" "g" "a" "t" "t" "c" "a" "g" "g" "g" "t" "t" "g" "a" "t" "t" "g"  
## [2024] "c" "c" "t" "t" "a" "c" "g" "t" "g" "a" "a" "a" "c" "a" "a" "g" "c"  
## [2041] "a" "a" "a" "c" "a" "t" "c" "t" "g" "g" "a" "g" "c" "a" "c" "a" "a"  
## [2058] "a" "g" "a" "c" "a" "a" "t" "g" "a" "a" "c" "t" "t" "a" "g" "a" "c"  
## [2075] "a" "a" "a" "a" "g" "a" "a" "g" "a" "a" "g" "g" "a" "g" "c" "t" "t"  
## [2092] "c" "t" "t" "g" "a" "g" "a" "g" "a" "a" "a" "a" "a" "c" "c" "a" "a"  
## [2109] "g" "a" "a" "a" "a" "g" "a" "c" "a" "a" "c" "t" "g" "g" "a" "a" "c"  
## [2126] "a" "a" "a" "a" "a" "a" "t" "c" "a" "g" "t" "t" "c" "c" "a" "a" "a"  
## [2143] "c" "t" "a" "g" "g" "a" "a" "g" "t" "t" "t" "a" "a" "a" "g" "c" "t"  
## [2160] "g" "a" "t" "g" "g" "a" "a" "c" "a" "g" "g" "a" "t" "a" "c" "t" "t"  
## [2177] "g" "c" "a" "a" "t" "c" "t" "t" "g" "a" "a" "g" "a" "g" "g" "a" "a"  
## [2194] "g" "a" "g" "c" "g" "a" "a" "a" "a" "g" "c" "a" "a" "g" "t" "a" "c"  
## [2211] "c" "a" "a" "a" "a" "t" "c" "a" "a" "a" "g" "a" "a" "a" "t" "a" "a"  
## [2228] "a" "t" "g" "t" "t" "c" "a" "a" "a" "a" "a" "g" "c" "g" "a" "a" "a"  
## [2245] "c" "t" "t" "g" "t" "t" "a" "c" "c" "g" "a" "a" "t" "t" "a" "a" "c"  
## [2262] "a" "a" "a" "c" "c" "t" "a" "a" "t" "a" "a" "a" "g" "a" "t" "t" "t"  
## [2279] "g" "t" "a" "c" "t" "t" "c" "t" "t" "t" "g" "c" "a" "t" "a" "t" "a"  
## [2296] "c" "a" "a" "a" "a" "a" "g" "t" "a" "g" "a" "t" "t" "t" "a" "a" "t"  
## [2313] "t" "c" "t" "c" "c" "a" "a" "a" "a" "t" "a" "c" "t" "a" "c" "a" "g"  
## [2330] "t" "g" "a" "t" "c" "t" "c" "t" "g" "a" "g" "a" "a" "g" "a" "a" "c"  
## [2347] "a" "a" "a" "t" "t" "a" "g" "a" "a" "t" "c" "a" "g" "a" "t" "t" "a"  
## [2364] "t" "a" "t" "g" "g" "c" "c" "g" "c" "a" "t" "c" "t" "t" "c" "a" "c"  
## [2381] "a" "a" "c" "t" "c" "c" "g" "t" "c" "t" "t" "a" "c" "a" "g" "a" "g"  
## [2398] "c" "a" "a" "c" "a" "t" "t" "t" "c" "a" "t" "t" "g" "a" "a" "t" "t"  
## [2415] "g" "g" "a" "t" "g" "a" "a" "a" "a" "t" "a" "g" "a" "c" "a" "g" "a"  
## [2432] "g" "a" "t" "t" "a" "t" "t" "g" "c" "a" "g" "a" "a" "a" "t" "g" "c"  
## [2449] "a" "a" "g" "g" "a" "a" "c" "t" "t" "a" "t" "g" "a" "a" "a" "a" "g"  
## [2466] "a" "g" "c" "t" "a" "g" "g" "c" "a" "a" "g" "t" "a" "t" "g" "t" "a"  
## [2483] "a" "c" "c" "t" "g" "g" "g" "t" "g" "c" "a" "g" "a" "g" "c" "a" "g"  
## [2500] "a" "c" "t" "c" "t" "t" "c" "c" "t" "c" "a" "a" "g" "a" "a" "t" "a"  
## [2517] "c" "c" "a" "g" "a" "c" "a" "c" "a" "a" "g" "t" "a" "c" "c" "c" "a"  
## [2534] "c" "c" "a" "t" "t" "c" "c" "a" "a" "a" "t" "g" "g" "a" "c" "a" "c"  
## [2551] "a" "a" "c" "t" "c" "c" "t" "c" "a" "c" "t" "c" "c" "c" "c" "a" "t"  
## [2568] "g" "g" "t" "t" "t" "t" "c" "c" "a" "a" "g" "a" "c" "c" "t" "t" "c"  
## [2585] "c" "a" "a" "a" "c" "a" "c" "a" "t" "t" "g" "g" "a" "t" "g" "a" "a"  
## [2602] "a" "t" "t" "g" "a" "t" "g" "c" "t" "t" "t" "a" "t" "t" "a" "a" "c"  
## [2619] "t" "g" "a" "a" "g" "a" "a" "a" "g" "a" "t" "c" "a" "a" "g" "a" "g"  
## [2636] "c" "t" "t" "c" "c" "t" "g" "c" "t" "t" "c" "a" "c" "g" "g" "g" "a"  
## [2653] "c" "t" "g" "a" "a" "t" "c" "c" "t" "a" "c" "a" "a" "t" "t" "g" "t"  
## [2670] "t" "c" "a" "g" "g" "a" "a" "t" "a" "t" "a" "c" "a" "a" "a" "a" "a"  
## [2687] "g" "a" "g" "a" "a" "g" "a" "a" "g" "a" "a" "a" "t" "a" "g" "a" "a"  
## [2704] "c" "a" "g" "t" "t" "a" "a" "c" "t" "g" "a" "g" "g" "a" "a" "c" "t"  
## [2721] "a" "a" "a" "g" "g" "g" "a" "a" "a" "g" "a" "a" "a" "g" "t" "t" "g"  
## [2738] "a" "a" "c" "t" "a" "g" "a" "t" "c" "a" "a" "t" "a" "c" "a" "g" "g"  
## [2755] "g" "a" "a" "a" "a" "c" "a" "t" "t" "t" "c" "a" "c" "a" "g" "g" "t"  
## [2772] "a" "a" "a" "a" "g" "a" "a" "a" "g" "g" "t" "g" "g" "c" "t" "t" "a"  
## [2789] "a" "t" "c" "c" "t" "t" "t" "a" "a" "a" "a" "g" "a" "g" "c" "t" "g"  
## [2806] "g" "t" "a" "g" "a" "a" "a" "a" "a" "a" "t" "t" "a" "a" "t" "g" "a"  
## [2823] "a" "a" "a" "a" "t" "t" "c" "a" "g" "c" "a" "a" "t" "t" "t" "t" "t"  
## [2840] "t" "t" "a" "g" "t" "t" "c" "c" "a" "t" "g" "c" "a" "g" "t" "g" "t"  
## [2857] "g" "c" "t" "g" "g" "t" "g" "a" "a" "g" "t" "t" "g" "a" "t" "c" "t"  
## [2874] "c" "c" "a" "t" "a" "c" "a" "g" "a" "a" "a" "a" "t" "g" "a" "g" "g"  
## [2891] "a" "a" "g" "a" "t" "t" "a" "t" "g" "a" "t" "a" "a" "a" "t" "a" "t"  
## [2908] "g" "g" "a" "a" "t" "t" "c" "g" "a" "a" "t" "t" "a" "g" "a" "g" "t"  
## [2925] "c" "a" "a" "a" "t" "t" "t" "c" "g" "a" "a" "g" "t" "a" "g" "t" "a"  
## [2942] "c" "t" "c" "a" "a" "c" "t" "g" "c" "a" "t" "g" "a" "a" "t" "t" "a"  
## [2959] "a" "c" "t" "c" "c" "t" "c" "a" "t" "c" "a" "t" "c" "a" "a" "a" "g"  
## [2976] "t" "g" "g" "a" "g" "g" "t" "g" "a" "a" "a" "g" "a" "a" "g" "t" "g"  
## [2993] "t" "t" "t" "c" "t" "a" "c" "c" "a" "t" "g" "t" "t" "a" "t" "a" "c"  
## [3010] "t" "t" "g" "a" "t" "g" "g" "c" "a" "c" "t" "t" "c" "a" "g" "g" "a"  
## [3027] "g" "c" "t" "a" "a" "a" "t" "a" "g" "a" "t" "g" "t" "c" "c" "a" "t"  
## [3044] "t" "c" "a" "g" "a" "g" "t" "a" "g" "t" "t" "g" "a" "t" "g" "a" "a"  
## [3061] "a" "t" "c" "a" "a" "t" "c" "a" "g" "g" "g" "a" "a" "t" "g" "g" "a"  
## [3078] "c" "c" "c" "a" "a" "t" "c" "a" "a" "t" "g" "a" "a" "c" "g" "g" "a"  
## [3095] "g" "a" "g" "t" "g" "t" "t" "t" "g" "a" "a" "a" "t" "g" "g" "t" "t"  
## [3112] "g" "t" "a" "a" "a" "t" "a" "c" "t" "g" "c" "c" "t" "g" "t" "a" "a"  
## [3129] "a" "g" "a" "a" "a" "a" "t" "a" "c" "a" "t" "c" "t" "c" "a" "a" "t"  
## [3146] "a" "c" "t" "t" "t" "t" "t" "c" "a" "t" "a" "a" "c" "a" "c" "c" "a"  
## [3163] "a" "a" "g" "c" "t" "c" "c" "t" "g" "c" "a" "a" "a" "a" "t" "c" "t"  
## [3180] "t" "c" "c" "t" "t" "a" "t" "t" "c" "t" "g" "a" "a" "a" "a" "g" "a"  
## [3197] "t" "g" "a" "c" "a" "g" "t" "t" "t" "t" "g" "t" "t" "t" "g" "t" "c"  
## [3214] "t" "a" "c" "a" "a" "t" "g" "g" "c" "c" "c" "t" "c" "a" "t" "a" "t"  
## [3231] "g" "c" "t" "g" "g" "a" "a" "c" "c" "a" "a" "a" "c" "a" "c" "a" "t"  
## [3248] "g" "g" "a" "a" "t" "t" "t" "a" "a" "a" "g" "g" "c" "t" "t" "t" "c"  
## [3265] "c" "a" "a" "a" "g" "g" "c" "g" "g" "c" "g" "g" "c" "g" "c" "c" "g"  
## [3282] "t" "a" "t" "t" "a" "c" "a" "t" "t" "c" "a" "c" "t" "c" "a" "a" "c"  
## [3299] "c" "t" "t" "c" "t" "t" "a" "a"

#dna\_seq\_string <- paste(dna\_seq, sep="", collapse="") #Takes our dna\_seq vector and converts into one string (can use c2s function in seqinr package.   
  
aa\_seq <- getTrans(dna\_seq) #Get the amino acid translation from our dna\_seq object  
#aa\_seq\_string <- paste(aa\_seq, sep="", collapse="") #The amino acid sequence as a string  
  
AAstat(aa\_seq, plot=FALSE) #Returns simple protein sequence information including the number of residues, the percentage physico-chemical classes and the theoretical isoelectric point. Input seq in character vector format

## $Compo  
##   
## \* A C D E F G H I K L M N P Q R S T   
## 1 45 21 47 126 36 33 22 64 106 106 26 60 41 67 81 64 58   
## V W Y   
## 66 5 27   
##   
## $Prop  
## $Prop$Tiny  
## [1] 0.2005445  
##   
## $Prop$Small  
## [1] 0.3947368  
##   
## $Prop$Aliphatic  
## [1] 0.2141561  
##   
## $Prop$Aromatic  
## [1] 0.08166969  
##   
## $Prop$Non.polar  
## [1] 0.4264973  
##   
## $Prop$Polar  
## [1] 0.5725953  
##   
## $Prop$Charged  
## [1] 0.3466425  
##   
## $Prop$Basic  
## [1] 0.1896552  
##   
## $Prop$Acidic  
## [1] 0.1569873  
##   
##   
## $Pi  
## [1] 8.686633

AAstat(aa\_seq, plot=TRUE) #if Plot=TRUE, plots the presence of residues splitted by physico-chemical classes along the sequence



## $Compo  
##   
## \* A C D E F G H I K L M N P Q R S T   
## 1 45 21 47 126 36 33 22 64 106 106 26 60 41 67 81 64 58   
## V W Y   
## 66 5 27   
##   
## $Prop  
## $Prop$Tiny  
## [1] 0.2005445  
##   
## $Prop$Small  
## [1] 0.3947368  
##   
## $Prop$Aliphatic  
## [1] 0.2141561  
##   
## $Prop$Aromatic  
## [1] 0.08166969  
##   
## $Prop$Non.polar  
## [1] 0.4264973  
##   
## $Prop$Polar  
## [1] 0.5725953  
##   
## $Prop$Charged  
## [1] 0.3466425  
##   
## $Prop$Basic  
## [1] 0.1896552  
##   
## $Prop$Acidic  
## [1] 0.1569873  
##   
##   
## $Pi  
## [1] 8.686633

#the AAstat plot illustrates the versatility of the "seqinr" package and shows a fraction of the other biochemical and biophysical properties of amino acids which could be utilised and added to the code when designing other prediction criteria.

# 3. Using the Chou-Fasman Method for Secondary Structure Prediction

Protein structure determination and prediction has long been an important part of the field of biophysics/bioinformatics because of the importance of protein structure in understanding the biological and chemical processes of cells/organisms.

The Chou-Fasman method, is a secondary structure prediction method, which largely relies on probability parameters (propensity values) (P) determimed from relative frequencies of each amino acids appearance in each type of secondary structure (Chou *et al.*, 1974). This algorithm is an empirical method that favors speed over accuracy, intended for generalizing secondary structure at a glance (Chen *et al.*, 2006).

The probability parameters (propensity values, P) for each amino acid can be loaded from the aaindex function in the "seqinr" package (Kawashima *et al*, 2000)(Tomii *et al*, 1996)(Nakai *et al*, 1988).

The aaindex function in "seqinr" returns a collection of published indices, or scales, of numerous physicochemical and biological properties of the 20 standard aminoacids

The format of the aaindex;

A list of 544 named indeces each with the following components:

H character vector: Accession number. D character vector: Data description. R character vector: LITDB entry number. A character vector: Author(s). T character vector: Title of the article. J character vector: Journal reference. C named numeric vector: Correlation coefficients of similar indeces (with coefficients of 0.8/-0.8 or more/less). The correlation coefficient is calculated with zeros filled for missing values. I named numeric vector: Amino acid index data.(Kawashima *et al*, 2000)(Tomii *et al*, 1996)(Nakai *et al*, 1988)

###############################################################################################################  
### Retrieving the relative frequencies of each amino acids appearance in each type of secondary structure ####  
###############################################################################################################  
  
  
#which(sapply(aaindex, function(x) length(grep("Chou", x$A)) !=0)) #Give alls the indices which have the author "Chou"  
  
# Aplha Helix frequencies - pulling the amino acid propensity (P) values for an Alpha helix  
  
aaindex[[38]]$D #Indicies 38 Data description. "Normalised frequency of aplha-helix"

## [1] "Normalized frequency of alpha-helix (Chou-Fasman, 1978b)"

aaindex[[38]]$I #Amino acid index(38) data. P(alpha-helix) data for each amino acid

## Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro   
## 1.42 0.98 0.67 1.01 0.70 1.11 1.51 0.57 1.00 1.08 1.21 1.16 1.45 1.13 0.57   
## Ser Thr Trp Tyr Val   
## 0.77 0.83 1.08 0.69 1.06

P\_ahelix <- aaindex[[38]]$I #Save as vector(Named number)  
P\_ahelix

## Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro   
## 1.42 0.98 0.67 1.01 0.70 1.11 1.51 0.57 1.00 1.08 1.21 1.16 1.45 1.13 0.57   
## Ser Thr Trp Tyr Val   
## 0.77 0.83 1.08 0.69 1.06

# Beta sheet frequencies - pulling the amino acid propensity (P) values for a B Sheet  
  
aaindex[[39]]$D #Indicies 39 Data description. "Normalised frequency of betasheet"

## [1] "Normalized frequency of beta-sheet (Chou-Fasman, 1978b)"

aaindex[[39]]$D #Amind acid index(39) data. P(b-sheet) data for each amino acid

## [1] "Normalized frequency of beta-sheet (Chou-Fasman, 1978b)"

P\_bsheet <- aaindex[[39]]$I #Save as a vector (Named number)  
P\_bsheet

## Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro   
## 0.83 0.93 0.89 0.54 1.19 1.10 0.37 0.75 0.87 1.60 1.30 0.74 1.05 1.38 0.55   
## Ser Thr Trp Tyr Val   
## 0.75 1.19 1.37 1.47 1.70

# Beta turn frequencies  
  
aaindex[[37]]$D #Indices 37 Data description. "Normalized frequency of beta-turn"

## [1] "Normalized frequency of beta-turn (Chou-Fasman, 1978a)"

aaindex[[37]]$I #Amino acid index(37) data. P(turn) data for each amino acid

## Ala Arg Asn Asp Cys Gln Glu Gly His Ile Leu Lys Met Phe Pro   
## 0.66 0.95 1.56 1.46 1.19 0.98 0.74 1.56 0.95 0.47 0.59 1.01 0.60 0.60 1.52   
## Ser Thr Trp Tyr Val   
## 1.43 0.96 0.96 1.14 0.50

P\_turn <- aaindex[[37]]$I   
  
# Frequencies of the first, second and third residue in a turn   
  
aaindex[[49]]$D #Indices 48 Data description. "Frequency of the 1st residue in turn"

## [1] "Frequency of the 1st residue in turn (Chou-Fasman, 1978b)"

f\_i <- aaindex[[49]]$I #Amino acid index(49) data. f(i) in Chou-Fasman method. Save as vector  
f\_i1 <- aaindex[[50]]$I #Amino acid index (50) data. f(i+1) in Chou-Fasman method.  
f\_i2 <- aaindex[[51]]$I #Amino acid index (51) data. f(i+2) in Chou-Fasman method  
f\_i3 <- aaindex[[52]]$I #Amino acid index (52) data. f(i+3) in Chou-Fasman method  
  
#Although this code will focus only on the A-helix and B-sheet prediction, the above pulled values can also be used to predict other types of secondary structure such as B-turns  
  
#Correcting the values - for ease of use  
  
P\_ahelix <- P\_ahelix \* 100  
P\_bsheet <- P\_bsheet \* 100  
P\_turn <- P\_turn \* 100  
  
#Storing all frequencies in matrix  
  
Raw\_Data <- cbind(P\_ahelix, P\_bsheet, P\_turn, f\_i, f\_i1, f\_i2) #Bind vectors of index data into a matrix, name Raw\_Data  
print(Raw\_Data)

## P\_ahelix P\_bsheet P\_turn f\_i f\_i1 f\_i2  
## Ala 142 83 66 0.060 0.076 0.035  
## Arg 98 93 95 0.070 0.106 0.099  
## Asn 67 89 156 0.161 0.083 0.191  
## Asp 101 54 146 0.147 0.110 0.179  
## Cys 70 119 119 0.149 0.053 0.117  
## Gln 111 110 98 0.074 0.098 0.037  
## Glu 151 37 74 0.056 0.060 0.077  
## Gly 57 75 156 0.102 0.085 0.190  
## His 100 87 95 0.140 0.047 0.093  
## Ile 108 160 47 0.043 0.034 0.013  
## Leu 121 130 59 0.061 0.025 0.036  
## Lys 116 74 101 0.055 0.115 0.072  
## Met 145 105 60 0.068 0.082 0.014  
## Phe 113 138 60 0.059 0.041 0.065  
## Pro 57 55 152 0.102 0.301 0.034  
## Ser 77 75 143 0.120 0.139 0.125  
## Thr 83 119 96 0.086 0.108 0.065  
## Trp 108 137 96 0.077 0.013 0.064  
## Tyr 69 147 114 0.082 0.065 0.114  
## Val 106 170 50 0.062 0.048 0.028

#Convert matrix into data frame  
  
P\_Table <- as.data.frame(Raw\_Data) #Convert Raw\_Data into a data frame  
  
#Create column for 3 letter amino acid code  
  
P\_Table$Amino\_Acid <- row.names(Raw\_Data) #Create new column from Row names (the amino acid 3letter code)  
  
#Convert 3 letter amino acid code into 1 Letter code, add as a column in data frame  
  
P\_Table$Letter\_Code <- a(P\_Table[,7]) #Create new column converting the 3ltter code in column 7 into the single ltter code for an aa acid using a() function  
print(P\_Table, row.names = FALSE)

## P\_ahelix P\_bsheet P\_turn f\_i f\_i1 f\_i2 Amino\_Acid Letter\_Code  
## 142 83 66 0.060 0.076 0.035 Ala A  
## 98 93 95 0.070 0.106 0.099 Arg R  
## 67 89 156 0.161 0.083 0.191 Asn N  
## 101 54 146 0.147 0.110 0.179 Asp D  
## 70 119 119 0.149 0.053 0.117 Cys C  
## 111 110 98 0.074 0.098 0.037 Gln Q  
## 151 37 74 0.056 0.060 0.077 Glu E  
## 57 75 156 0.102 0.085 0.190 Gly G  
## 100 87 95 0.140 0.047 0.093 His H  
## 108 160 47 0.043 0.034 0.013 Ile I  
## 121 130 59 0.061 0.025 0.036 Leu L  
## 116 74 101 0.055 0.115 0.072 Lys K  
## 145 105 60 0.068 0.082 0.014 Met M  
## 113 138 60 0.059 0.041 0.065 Phe F  
## 57 55 152 0.102 0.301 0.034 Pro P  
## 77 75 143 0.120 0.139 0.125 Ser S  
## 83 119 96 0.086 0.108 0.065 Thr T  
## 108 137 96 0.077 0.013 0.064 Trp W  
## 69 147 114 0.082 0.065 0.114 Tyr Y  
## 106 170 50 0.062 0.048 0.028 Val V

# 4. Chou-Fasman Prediction of A-helices and B-sheets

To determine the type of secondary structure the amino acid chain will form we must first calculate the a-helix/bsheet nucleation point. This is dependent on how many a-helix/b-sheet "makers" and how many a-helix/b-sheet "breakers" we have in the sequence we inspect. The propensity for an amino acid to be a "maker" or "breaker" is usually determined by the R side group. In the Chou Fasman method, it is considered;

If 6 continuous/adjacent residues have more than a third (2) of a-helix breakers (P\_ahelix < 103) then it should not form a helix

If 5 continuous/adjacent residues have more than a third (2) of b-sheet breakers (P\_bsheet < 105) it should not form a b-sheet

After we have determined a nucleation point for a particular secondary structure, we continue to extend the query sequence by one amino acid in both directions, until it is terminated, by a sequence of 4 amino acids (at the end of the query) with an average P\_ahelix<103 for an alpha-helix or 3 amino acids (at the end of the query) with an average P\_bsheet<105 for a B-sheet (Chou *et al.*, 1974).

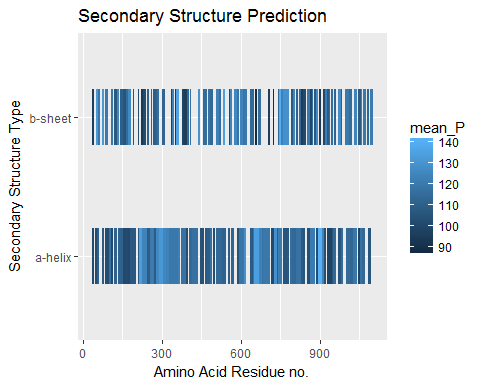
#####################################################################################################################  
######################################## Determining A-helix regions ##############################################  
#####################################################################################################################  
  
z <- NULL #Creating objects to be used in the code  
x <- 0 #x is the position along the amino acid sequence to begin scanning  
y <- x + 5 #y is the 6th (final) amino acid position in the sequence of 6 being scanned  
ahelix\_boundaries <- NULL  
  
while(y <= length(aa\_seq)-1){ #while y(the last amino acid in the sequence of 6 being analysed) is <= the length of the aa seq  
 temp <- NULL #empty vector to store the Propensity(alpha) values for each aa in the sequence of 6  
 while((length(temp[temp > 103])) < 4){ #checks that there is less than 4 values (Palpha) over the threshold of 103  
 temp <- NULL #Clears the temp value, important so values from the first iteration are not present in the 2nd  
 x <- x + 1 #the 1st amino acid in the sequence of 6 being analysed moves one along the total aa sequence.  
 y <- y + 1 #the last amino acid in the sequence of 6 being analysed moves one along the total aa sequence  
 if(y > length(aa\_seq)-1){ #if y ever reaches the end of the aa sequence, break the loop  
 break  
 }  
for(i in x:y){ #for each aa between the scanning/analysis boundaries (of 6 aa)  
 temp[i] <- P\_Table$P\_ahelix[grepl(aa\_seq[i], P\_Table$Letter\_Code)] #Use aa letter code, search for matches in Ptable, then pull the Pvalue, stores  
 temp <- temp[!is.na(temp)] #remove the NAs from temp. Na introduced after first iteration (x=2), as pos.1 in temp will be empty etc  
 if((length(temp[temp > 103])) >= 4){ #if the temp now has 4 Pa above threshold, is a nucleation point and break the while loop. else restart  
 break  
 }  
 } #if a nucleation point is found need to extend helix in both directions  
 }   
 if(y > length(aa\_seq)-1){ #if y ever reaches the end of the aa sequence break the extension loops  
 break  
 }  
 temp1 <- 104 #Set temp1 greater than 103 (threshold value), so can pass into the first iteration  
 while((mean(temp1))>103){ #while the average of temp1(the 4 left most aa) > than the threshold value. When below will terminate. x is then left boundary  
 temp1 <- NULL #clear temp1, vector where will store Palpha values  
 x <- (x-1) #takes the left nucleation boundary-1, this moves the left most amino acid being analysed 1 position back along the aa sequence  
 for(i in ((x):(x+3))){ #takes the shifted nuc. boundary, runs for loop from this position and the next 3  
 temp1[i] <- P\_Table$P\_ahelix[grepl(aa\_seq[i], P\_Table$Letter\_Code)] #Use aa letter code, search for matches in Ptable, then pull the Pvalue, stores  
 temp1 <- temp1[!is.na(temp1)] #remove the NAs from temp. Na introduced after first iteration (x=2), as pos.1 in temp will be empty etc  
 } #after for loop completes, the 4 aa Pa values are passed into the while loop.  
}  
 temp2 <- 104 #Set temp2 greater than 103 (threshold), so can pass into the first iteration  
while((mean(temp2))>103){ #while the average of temp2(the 4 right most aa) > than the threshold value. When below will terminate. y is then left boundary  
 temp2 <- NULL #clear temp1, vector where will store Palpha values  
 y <- (y+1) #takes the right nucleation boundary+1, this moves the right most amino acid being analysed 1 position along the aa sequence  
 for(i in ((y-3):y)){ #takes the shifted nuc. boundary, runs for loop for the 4 right most aa  
 temp2[i] <- P\_Table$P\_ahelix[grepl(aa\_seq[i], P\_Table$Letter\_Code)] #Use aa letter code, search for matches in Ptable, then pull the Pvalue, stores  
 temp2 <- temp2[!is.na(temp2)] #remove the NAs from temp. Na introduced after first iteration (x=2), as pos.1 in temp will be empty etc  
 } #after for loop completes, the 4 aa Pa values are passed into the while loop.  
}  
 z <- mean(temp) #mean of the original temp file. Use this as measure of likelihood when both A helix and B sheet are predicted at the same place  
 ahelix\_boundaries <- rbind(ahelix\_boundaries, c(x, y, z)) #combining the helix boundaries and mean Palpha for this iteration into a data frame  
 x <- y #make the starting point for the overall while loop equal to the position where the previous helix ended  
 y <- x + 5 #make the left most amino acid of the sequence being analysed (in the first loop) 5 residues after the first in the sequence.  
}  
   
print(ahelix\_boundaries)

## [,1] [,2] [,3]  
## [1,] 35 43 101.50000  
## [2,] 46 54 102.00000  
## [3,] 54 63 110.16667  
## [4,] 73 82 107.83333  
## [5,] 87 94 101.83333  
## [6,] 94 101 102.33333  
## [7,] 99 108 116.16667  
## [8,] 108 117 101.00000  
## [9,] 119 132 110.83333  
## [10,] 133 147 111.16667  
## [11,] 147 154 110.50000  
## [12,] 155 171 101.16667  
## [13,] 173 180 100.50000  
## [14,] 180 204 107.00000  
## [15,] 211 219 107.66667  
## [16,] 216 229 133.50000  
## [17,] 227 243 123.40000  
## [18,] 245 259 109.00000  
## [19,] 256 272 120.40000  
## [20,] 270 280 108.33333  
## [21,] 277 291 125.60000  
## [22,] 289 305 131.00000  
## [23,] 307 320 125.16667  
## [24,] 317 332 123.00000  
## [25,] 329 364 119.40000  
## [26,] 361 371 122.16667  
## [27,] 375 385 116.83333  
## [28,] 382 392 117.50000  
## [29,] 393 400 103.00000  
## [30,] 405 419 114.16667  
## [31,] 416 439 116.80000  
## [32,] 445 462 104.50000  
## [33,] 463 472 107.83333  
## [34,] 473 482 109.16667  
## [35,] 482 490 116.40000  
## [36,] 494 502 103.00000  
## [37,] 507 530 113.50000  
## [38,] 529 543 111.40000  
## [39,] 552 561 108.83333  
## [40,] 564 574 112.16667  
## [41,] 584 592 104.66667  
## [42,] 592 613 118.66667  
## [43,] 612 621 128.60000  
## [44,] 634 646 107.50000  
## [45,] 646 654 118.00000  
## [46,] 651 672 133.50000  
## [47,] 671 680 123.00000  
## [48,] 679 688 117.33333  
## [49,] 685 712 119.40000  
## [50,] 713 726 101.33333  
## [51,] 726 739 128.20000  
## [52,] 736 746 109.60000  
## [53,] 743 756 121.25000  
## [54,] 761 774 105.50000  
## [55,] 775 782 106.83333  
## [56,] 786 795 115.66667  
## [57,] 793 811 113.16667  
## [58,] 808 827 117.25000  
## [59,] 827 836 108.16667  
## [60,] 836 843 106.00000  
## [61,] 852 862 103.16667  
## [62,] 863 880 120.66667  
## [63,] 886 896 102.66667  
## [64,] 893 910 140.25000  
## [65,] 909 917 123.50000  
## [66,] 917 925 103.33333  
## [67,] 927 945 98.33333  
## [68,] 945 952 106.00000  
## [69,] 951 960 113.00000  
## [70,] 970 979 105.66667  
## [71,] 980 988 114.50000  
## [72,] 998 1013 114.00000  
## [73,] 1014 1024 112.50000  
## [74,] 1027 1040 119.50000  
## [75,] 1046 1054 98.50000  
## [76,] 1053 1061 117.25000  
## [77,] 1061 1072 113.00000  
## [78,] 1081 1092 106.16667

############################################################################################################################  
################################################ Predicting B-sheets #######################################################  
############################################################################################################################  
  
#The concept behind the code for predicting B-sheets is very similar to that for predicting a helices. However the threshold value is raised to 105 from 103, the range of amino acids in the sequence being analysed is 5 for B sheets as opposed to 6 for a helices. Also only 3 of the 5 need to be above the threshold to determine whether a sequence is a nucleation point or not. The propensity values the for loops pull from the P\_Table, are the P(b\_sheet) values as opposed to the P(a\_helix) values.   
  
z <- NULL  
x <- 0  
y <- x + 4  
bsheet\_boundaries <- NULL  
  
while(y <= length(aa\_seq)-1){  
 temp <- NULL  
 while((length(temp[temp > 105])) < 3){   
 temp <- NULL   
 x <- x + 1  
 y <- y + 1  
 if(y > length(aa\_seq)-1){  
 break  
 }  
for(i in x:y){   
 temp[i] <- P\_Table$P\_bsheet[grepl(aa\_seq[i], P\_Table$Letter\_Code)]   
 temp <- temp[!is.na(temp)]   
 if((length(temp[temp > 105])) >= 3){   
 break  
 }  
 }  
 }   
 if(y > length(aa\_seq)-1){  
 break  
 }  
 temp1 <- 106   
 while((mean(temp1))>105){   
 temp1 <- NULL   
 x <- (x-1)   
 for(i in ((x):(x+3))){   
 temp1[i] <- P\_Table$P\_bsheet[grepl(aa\_seq[i], P\_Table$Letter\_Code)]   
 temp1 <- temp1[!is.na(temp1)]   
 }  
}  
 temp2 <- 106  
while((mean(temp2))>105){  
 temp2 <- NULL  
 y <- (y+1)  
 for(i in ((y-3):y)){  
 temp2[i] <- P\_Table$P\_bsheet[grepl(aa\_seq[i], P\_Table$Letter\_Code)]  
 temp2 <- temp2[!is.na(temp2)]  
 }  
}  
 z <- mean(temp)  
 bsheet\_boundaries <- rbind(bsheet\_boundaries, c(x, y, z))  
 x <- y  
 y <- x + 4  
}  
   
print(bsheet\_boundaries)

## [,1] [,2] [,3]  
## [1,] 36 45 96.0000  
## [2,] 50 59 131.6000  
## [3,] 59 68 119.0000  
## [4,] 74 82 130.8000  
## [5,] 86 98 119.8000  
## [6,] 108 117 119.0000  
## [7,] 121 129 106.8000  
## [8,] 131 139 124.8000  
## [9,] 144 154 109.8000  
## [10,] 154 163 105.7500  
## [11,] 166 176 114.4000  
## [12,] 173 182 126.0000  
## [13,] 191 197 106.2000  
## [14,] 209 216 104.0000  
## [15,] 222 228 94.0000  
## [16,] 231 239 101.4000  
## [17,] 244 251 111.0000  
## [18,] 258 264 112.2000  
## [19,] 269 276 109.2000  
## [20,] 276 283 111.4000  
## [21,] 283 291 116.2000  
## [22,] 301 311 112.0000  
## [23,] 335 341 97.4000  
## [24,] 341 348 123.5000  
## [25,] 351 357 94.8000  
## [26,] 354 365 135.7500  
## [27,] 379 385 98.2000  
## [28,] 385 391 95.8000  
## [29,] 392 398 97.0000  
## [30,] 398 404 113.4000  
## [31,] 406 412 117.2000  
## [32,] 439 447 129.6000  
## [33,] 455 462 112.8000  
## [34,] 462 473 114.6000  
## [35,] 476 483 117.0000  
## [36,] 483 493 113.8000  
## [37,] 508 516 122.8000  
## [38,] 519 526 116.0000  
## [39,] 531 542 134.4000  
## [40,] 555 561 99.6000  
## [41,] 558 568 124.5000  
## [42,] 575 588 125.4000  
## [43,] 587 594 117.8000  
## [44,] 599 608 124.0000  
## [45,] 608 616 118.5000  
## [46,] 616 625 119.6000  
## [47,] 634 650 113.6000  
## [48,] 654 660 88.8000  
## [49,] 664 670 113.4000  
## [50,] 670 678 115.6000  
## [51,] 704 710 96.0000  
## [52,] 721 727 87.8000  
## [53,] 740 748 113.2000  
## [54,] 746 754 139.6667  
## [55,] 752 764 122.6000  
## [56,] 761 781 128.5000  
## [57,] 792 799 107.6000  
## [58,] 799 806 123.7500  
## [59,] 808 817 111.2000  
## [60,] 822 830 115.0000  
## [61,] 830 836 95.8000  
## [62,] 836 847 101.0000  
## [63,] 853 862 119.6000  
## [64,] 863 869 100.0000  
## [65,] 869 875 115.5000  
## [66,] 878 886 106.8000  
## [67,] 885 894 126.0000  
## [68,] 898 904 94.8000  
## [69,] 911 918 100.2000  
## [70,] 919 926 120.8000  
## [71,] 926 933 108.2000  
## [72,] 934 940 114.2000  
## [73,] 942 950 115.6000  
## [74,] 955 961 112.0000  
## [75,] 967 979 127.0000  
## [76,] 979 985 108.5000  
## [77,] 996 1009 119.8000  
## [78,] 1012 1022 115.0000  
## [79,] 1030 1042 124.0000  
## [80,] 1044 1055 108.0000  
## [81,] 1054 1061 123.3333  
## [82,] 1064 1075 119.6000  
## [83,] 1080 1087 112.8000  
## [84,] 1091 1101 120.6000

##################################################################################################################   
############################################ Plotting the CF Results #############################################  
##################################################################################################################  
  
  
colnames(ahelix\_boundaries) <-c("start", "end", "mean\_P") #Change the column names of the table, so that they are consistent with the names for the Bsheet table  
ahelix\_boundaries <- as.data.frame(ahelix\_boundaries) #Convert the table into a dataframe  
ahelix\_boundaries$type <- rep("a-helix", length(ahelix\_boundaries$mean\_P)) #Add an extra column which labels all these domains as a-helix  
  
  
colnames(bsheet\_boundaries) <- c("start", "end","mean\_P") #Change the column names of the table, so that they are consistent with the names for the a helix table  
bsheet\_boundaries <- as.data.frame(bsheet\_boundaries) #Convert the table into a dataframe  
bsheet\_boundaries$type <- rep("b-sheet", length(bsheet\_boundaries$mean\_P)) #Add an extra column which labels all these domains as b-sheet  
  
Gene\_str\_boundaries <- rbind(ahelix\_boundaries, bsheet\_boundaries) #row bind the dataframes for the a-helix boundaries and the b-sheet boundaries  
  
  
  
plot1 <- ggplot(data=Gene\_str\_boundaries, aes(x=type, ymin= start, ymax=end, colour=mean\_P))  
plotCF <- plot1 +  
 geom\_linerange(size=20)+  
 coord\_flip() +  
 labs(title="Secondary Structure Prediction", y="Amino Acid Residue no.", x="Secondary Structure Type")+  
 scale\_fill\_manual(name="Mean P")  
   
plotCF



The plotCF returns the predicted alpha helix domains and beta sheet domains and their locations along the amino acid sequence being analysed to an accuracy of between 68-80%. Application of this method assist the studying of the correlation between protein structure and biological activity, and also be an aid in x-ray crystallography when interpreting data. The data can also be useful when deciding boundaries within the protein to express and trucation product to increase the likelihood of soluble, protein which is more likely to crystallise. When designing crystallisation constructs, we need to select for well structured or folded domains. Using the plotCF we can determine where along the amino acid sequence we have secondary structure. Using the mean P value we can also quatify the likelihood of this structure existing in addition to deciphering between an alpha helix secondary structure and a beta sheet, by taking the structure which has the higher mean P value at that point. Regions where there is no prediction are likely to be coils or disordered regions and should be avoided if possible and these regions introduce flexibility into the protein, which could be problematic during the crystallisation phase. R allows us to easily visualise these regions of interest whilst also allowing the user to expand the code to add in additional variables to analyse the sequence (Chen *et al.*, 2006). For example if the user desired to determine interaction regions, the user could plot the hydrophobicity of each amino acid in the sequence and look for regions of hydrophobic patches.

# 5. Using PDB BLAST for structure prediction

BLAST, one of the most popular pairwise sequence comparison algorithms for database searching, performs gapped local alignments via a heuristic stratgey: it identifies short near exact matches or hits, bidirectionally extends non-overlapping hits resulting in ungapped extended hits or high-scoring segment pairs (HSPs), and finally extends the highest scoring HSP in both directiosn via a gappeed alignment (Altschul et al 1997). Using BLAST we can determine regions of homology between our protein sequence of interest and existing protein structures and use this as an estimate of regions that are likely to form well folded domains and likely be structured well enough for protein characterisation via X-ray crystallography.

#Load bio3d package  
  
library(bio3d)

##   
## Attaching package: 'bio3d'

## The following objects are masked from 'package:seqinr':  
##   
## consensus, read.fasta, write.fasta

#Bio3d can access several databases with blast.pdb, including the protein databank, non-redundant protein sequences and swissprot  
#As we want to define regions on our protein sequence of interest that are likely to be well structured and folded, homology will be searched for against the protein data bank, which contains the coordinates of already defined protein structures.   
#Areas of homology between our query sequence and already characterised (and therefore likely well folded/structured) proteins can be a good indication of boundaries in our sequence which are likely to follow a similar folding/structured pattern and therefore be areas that will be well well behaved during protein purification and crystallisation.  
  
#blast the sequence vs the pdb, save as an object, this may take some time, depenedent on length oF the sequence being searched, internet speed and server speed.  
  
blast\_results <- blast.pdb(aa\_seq, database = "pdb", chain.single=TRUE) #blast the aa sequence of interest, against a defined database, save results as named object

## Searching ... please wait (updates every 5 seconds) RID = EY517001013   
##

## Warning in scan(file = file, what = what, sep = sep, quote = quote, dec =  
## dec, : EOF within quoted string

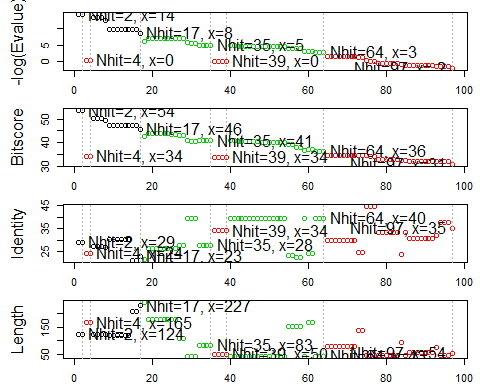
## .  
## Reporting 97 hits

blast\_results

## $hit.tbl  
## queryid subjectids identity positives  
## 1 Query\_208689 gi|444302296|pdb|4I99|A 29.032 50.00  
## 2 Query\_208689 gi|444302297|pdb|4I99|B 29.032 50.00  
## 3 Query\_208689 gi|444302296|pdb|4I99|A 24.242 43.03  
## 4 Query\_208689 gi|444302297|pdb|4I99|B 24.242 43.03  
## 5 Query\_208689 gi|299856727|pdb|3KTA|B 27.419 48.39  
## 6 Query\_208689 gi|299856729|pdb|3KTA|D 27.419 48.39  
## 7 Query\_208689 gi|58177331|pdb|1XEW|Y 27.419 48.39  
## 8 Query\_208689 gi|58177333|pdb|1XEX|B 26.613 48.39  
## 9 Query\_208689 gi|13096783|pdb|1E69|A 30.252 50.42  
## 10 Query\_208689 gi|13096784|pdb|1E69|B 30.252 50.42  
## 11 Query\_208689 gi|13096785|pdb|1E69|C 30.252 50.42  
## 12 Query\_208689 gi|13096786|pdb|1E69|D 30.252 50.42  
## 13 Query\_208689 gi|13096787|pdb|1E69|E 30.252 50.42  
## 14 Query\_208689 gi|13096788|pdb|1E69|F 30.252 50.42  
## 15 Query\_208689 gi|1143283763|pdb|5MG8|A 21.154 42.31  
## 16 Query\_208689 gi|1143283765|pdb|5MG8|C 21.154 42.31  
## 17 Query\_208689 gi|728055657|pdb|4UX3|A 22.907 43.17  
## 18 Query\_208689 gi|728055657|pdb|4UX3|A 21.429 41.18  
## 19 Query\_208689 gi|333944148|pdb|3AUX|A 26.257 44.69  
## 20 Query\_208689 gi|333944149|pdb|3AUY|A 26.257 44.69  
## 21 Query\_208689 gi|333944150|pdb|3AUY|B 26.257 44.69  
## 22 Query\_208689 gi|333944153|pdb|3AV0|B 26.257 44.69  
## 23 Query\_208689 gi|1000232403|pdb|5DNY|B 26.257 44.69  
## 24 Query\_208689 gi|1000232405|pdb|5DNY|D 26.257 44.69  
## 25 Query\_208689 gi|1002351595|pdb|5F3W|B 26.257 44.69  
## 26 Query\_208689 gi|1002351597|pdb|5F3W|D 26.257 44.69  
## 27 Query\_208689 gi|1159374129|pdb|5H68|A 27.523 47.71  
## 28 Query\_208689 gi|1159374130|pdb|5H68|B 27.523 47.71  
## 29 Query\_208689 gi|14488688|pdb|1II8|A 39.535 69.77  
## 30 Query\_208689 gi|327200667|pdb|3QKR|A 39.535 69.77  
## 31 Query\_208689 gi|327200670|pdb|3QKS|A 39.535 69.77  
## 32 Query\_208689 gi|55670515|pdb|1W1W|A 27.711 49.40  
## 33 Query\_208689 gi|55670516|pdb|1W1W|B 27.711 49.40  
## 34 Query\_208689 gi|55670517|pdb|1W1W|C 27.711 49.40  
## 35 Query\_208689 gi|55670518|pdb|1W1W|D 27.711 49.40  
## 36 Query\_208689 gi|55670515|pdb|1W1W|A 34.000 58.00  
## 37 Query\_208689 gi|55670516|pdb|1W1W|B 34.000 58.00  
## 38 Query\_208689 gi|55670517|pdb|1W1W|C 34.000 58.00  
## 39 Query\_208689 gi|55670518|pdb|1W1W|D 34.000 58.00  
## 40 Query\_208689 gi|327200677|pdb|3QKU|A 39.535 69.77  
## 41 Query\_208689 gi|327200678|pdb|3QKU|B 39.535 69.77  
## 42 Query\_208689 gi|589911123|pdb|4NCH|A 39.535 69.77  
## 43 Query\_208689 gi|589911124|pdb|4NCH|B 39.535 69.77  
## 44 Query\_208689 gi|327200673|pdb|3QKT|A 39.535 69.77  
## 45 Query\_208689 gi|327200674|pdb|3QKT|B 39.535 69.77  
## 46 Query\_208689 gi|327200675|pdb|3QKT|C 39.535 69.77  
## 47 Query\_208689 gi|327200676|pdb|3QKT|D 39.535 69.77  
## 48 Query\_208689 gi|589911125|pdb|4NCI|A 39.535 69.77  
## 49 Query\_208689 gi|589911126|pdb|4NCJ|A 39.535 69.77  
## 50 Query\_208689 gi|589911127|pdb|4NCJ|B 39.535 69.77  
## 51 Query\_208689 gi|589911128|pdb|4NCJ|C 39.535 69.77  
## 52 Query\_208689 gi|589911129|pdb|4NCJ|D 39.535 69.77  
## 53 Query\_208689 gi|589911130|pdb|4NCK|A 39.535 69.77  
## 54 Query\_208689 gi|589911131|pdb|4NCK|B 39.535 69.77  
## 55 Query\_208689 gi|444302106|pdb|3ZGX|A 23.179 46.36  
## 56 Query\_208689 gi|444302107|pdb|3ZGX|B 23.179 46.36  
## 57 Query\_208689 gi|1159374124|pdb|5H66|B 22.517 46.36  
## 58 Query\_208689 gi|1159374127|pdb|5H67|B 22.517 46.36  
## 59 Query\_208689 gi|39655001|pdb|1US8|A 39.535 69.77  
## 60 Query\_208689 gi|58177330|pdb|1XEW|X 24.242 44.24  
## 61 Query\_208689 gi|58177332|pdb|1XEX|A 24.242 44.24  
## 62 Query\_208689 gi|9954932|pdb|1F2T|A 39.535 69.77  
## 63 Query\_208689 gi|9954934|pdb|1F2U|A 39.535 69.77  
## 64 Query\_208689 gi|9954936|pdb|1F2U|C 39.535 69.77  
## 65 Query\_208689 gi|931139635|pdb|4RMO|A 29.630 55.56  
## 66 Query\_208689 gi|931139637|pdb|4RMO|C 29.630 55.56  
## 67 Query\_208689 gi|931139639|pdb|4RMO|E 29.630 55.56  
## 68 Query\_208689 gi|931139641|pdb|4RMO|G 29.630 55.56  
## 69 Query\_208689 gi|931139643|pdb|4RMO|I 29.630 55.56  
## 70 Query\_208689 gi|931139645|pdb|4RMO|K 29.630 55.56  
## 71 Query\_208689 gi|931139647|pdb|4RMO|M 29.630 55.56  
## 72 Query\_208689 gi|931139649|pdb|4RMO|O 29.630 55.56  
## 73 Query\_208689 gi|299856726|pdb|3KTA|A 24.638 44.20  
## 74 Query\_208689 gi|299856728|pdb|3KTA|C 24.638 44.20  
## 75 Query\_208689 gi|753536165|pdb|3WSX|A 44.444 55.56  
## 76 Query\_208689 gi|753536159|pdb|3WSZ|A 44.444 55.56  
## 77 Query\_208689 gi|753536162|pdb|3WSY|A 44.444 55.56  
## 78 Query\_208689 gi|329666209|pdb|3QF7|A 33.333 53.33  
## 79 Query\_208689 gi|329666210|pdb|3QF7|B 33.333 53.33  
## 80 Query\_208689 gi|747155454|pdb|4W9M|C 33.333 53.33  
## 81 Query\_208689 gi|747155456|pdb|4W9M|E 33.333 53.33  
## 82 Query\_208689 gi|747155460|pdb|4W9M|I 33.333 53.33  
## 83 Query\_208689 gi|747155462|pdb|4W9M|K 33.333 53.33  
## 84 Query\_208689 gi|13787046|pdb|1EU8|A 23.656 46.24  
## 85 Query\_208689 gi|350610724|pdb|3THO|A 33.333 53.33  
## 86 Query\_208689 gi|292659708|pdb|3KL4|A 30.508 50.85  
## 87 Query\_208689 gi|39654777|pdb|1QZW|A 30.508 50.85  
## 88 Query\_208689 gi|39654779|pdb|1QZW|C 30.508 50.85  
## 89 Query\_208689 gi|39654781|pdb|1QZW|E 30.508 50.85  
## 90 Query\_208689 gi|39654783|pdb|1QZW|G 30.508 50.85  
## 91 Query\_208689 gi|39654785|pdb|1QZX|A 30.508 50.85  
## 92 Query\_208689 gi|39654786|pdb|1QZX|B 30.508 50.85  
## 93 Query\_208689 gi|699030604|pdb|4UXV|A 32.000 54.67  
## 94 Query\_208689 gi|38492937|pdb|1QVR|A 37.500 64.58  
## 95 Query\_208689 gi|38492938|pdb|1QVR|B 37.500 64.58  
## 96 Query\_208689 gi|38492939|pdb|1QVR|C 37.500 64.58  
## 97 Query\_208689 gi|1062512851|pdb|5T33|G 35.185 57.41  
## alignmentlength mismatches gapopens q.start q.end s.start s.end  
## 1 124 82 4 930 1053 188 305  
## 2 124 82 4 930 1053 188 305  
## 3 165 100 6 47 194 2 158  
## 4 165 100 6 47 194 2 158  
## 5 124 84 3 930 1053 7 124  
## 6 124 84 3 930 1053 7 124  
## 7 124 84 3 930 1053 6 123  
## 8 124 85 3 930 1053 6 123  
## 9 119 70 4 937 1052 168 276  
## 10 119 70 4 937 1052 168 276  
## 11 119 70 4 937 1052 168 276  
## 12 119 70 4 937 1052 168 276  
## 13 119 70 4 937 1052 168 276  
## 14 119 70 4 937 1052 168 276  
## 15 208 150 6 398 592 20 226  
## 16 208 150 6 398 592 20 226  
## 17 227 152 6 53 259 4 227  
## 18 238 141 7 844 1052 267 487  
## 19 179 118 5 56 226 8 180  
## 20 179 118 5 56 226 8 180  
## 21 179 118 5 56 226 8 180  
## 22 179 118 5 56 226 8 180  
## 23 179 118 5 56 226 8 180  
## 24 179 118 5 56 226 8 180  
## 25 179 118 5 56 226 8 180  
## 26 179 118 5 56 226 8 180  
## 27 109 72 3 924 1032 182 283  
## 28 109 72 3 924 1032 182 283  
## 29 43 26 0 55 97 5 47  
## 30 43 26 0 55 97 5 47  
## 31 43 26 0 55 97 5 47  
## 32 83 55 2 952 1033 298 376  
## 33 83 55 2 952 1033 298 376  
## 34 83 55 2 952 1033 298 376  
## 35 83 55 2 952 1033 298 376  
## 36 50 31 2 51 98 2 51  
## 37 50 31 2 51 98 2 51  
## 38 50 31 2 51 98 2 51  
## 39 50 31 2 51 98 2 51  
## 40 43 26 0 55 97 5 47  
## 41 43 26 0 55 97 5 47  
## 42 43 26 0 55 97 5 47  
## 43 43 26 0 55 97 5 47  
## 44 43 26 0 55 97 5 47  
## 45 43 26 0 55 97 5 47  
## 46 43 26 0 55 97 5 47  
## 47 43 26 0 55 97 5 47  
## 48 43 26 0 55 97 5 47  
## 49 43 26 0 55 97 5 47  
## 50 43 26 0 55 97 5 47  
## 51 43 26 0 55 97 5 47  
## 52 43 26 0 55 97 5 47  
## 53 43 26 0 55 97 5 47  
## 54 43 26 0 55 97 5 47  
## 55 151 99 5 914 1053 244 388  
## 56 151 99 5 914 1053 244 388  
## 57 151 100 5 914 1053 6 150  
## 58 151 100 5 914 1053 6 150  
## 59 43 26 0 55 97 5 47  
## 60 165 100 6 47 194 2 158  
## 61 165 100 6 47 194 2 158  
## 62 43 26 0 55 97 5 47  
## 63 43 26 0 55 97 5 47  
## 64 43 26 0 55 97 5 47  
## 65 81 49 3 748 821 59 138  
## 66 81 49 3 748 821 59 138  
## 67 81 49 3 748 821 59 138  
## 68 81 49 3 748 821 59 138  
## 69 81 49 3 748 821 59 138  
## 70 81 49 3 748 821 59 138  
## 71 81 49 3 748 821 59 138  
## 72 81 49 3 748 821 59 138  
## 73 138 81 5 47 167 2 133  
## 74 138 81 5 47 167 2 133  
## 75 45 21 1 489 529 227 271  
## 76 45 21 1 489 529 171 215  
## 77 45 21 1 489 529 171 215  
## 78 45 30 0 55 99 5 49  
## 79 45 30 0 55 99 5 49  
## 80 45 30 0 55 99 5 49  
## 81 45 30 0 55 99 5 49  
## 82 45 30 0 55 99 5 49  
## 83 45 30 0 55 99 5 49  
## 84 93 64 2 862 950 14 103  
## 85 45 30 0 55 99 22 66  
## 86 59 25 2 73 131 97 139  
## 87 59 25 2 73 131 104 146  
## 88 59 25 2 73 131 104 146  
## 89 59 25 2 73 131 104 146  
## 90 59 25 2 73 131 104 146  
## 91 59 25 2 73 131 104 146  
## 92 59 25 2 73 131 104 146  
## 93 75 46 3 292 361 409 483  
## 94 48 28 2 892 937 496 543  
## 95 48 28 2 892 937 496 543  
## 96 48 28 2 892 937 496 543  
## 97 54 29 2 913 962 192 243  
## evalue bitscore mlog.evalue pdb.id acc  
## 1 6.55e-07 53.5 14.23863060 4I99\_A 444302296  
## 2 6.55e-07 53.5 14.23863060 4I99\_B 444302297  
## 3 6.90e-01 34.3 0.37106368 4I99\_A 444302296  
## 4 6.90e-01 34.3 0.37106368 4I99\_B 444302297  
## 5 1.39e-06 50.4 13.48620681 3KTA\_B 299856727  
## 6 1.39e-06 50.4 13.48620681 3KTA\_D 299856729  
## 7 1.44e-06 50.1 13.45086744 1XEW\_Y 58177331  
## 8 3.39e-06 49.3 12.59468064 1XEX\_B 58177333  
## 9 5.48e-05 47.4 9.81182036 1E69\_A 13096783  
## 10 5.48e-05 47.4 9.81182036 1E69\_B 13096784  
## 11 5.48e-05 47.4 9.81182036 1E69\_C 13096785  
## 12 5.48e-05 47.4 9.81182036 1E69\_D 13096786  
## 13 5.48e-05 47.4 9.81182036 1E69\_E 13096787  
## 14 5.48e-05 47.4 9.81182036 1E69\_F 13096788  
## 15 6.18e-05 47.4 9.69160719 5MG8\_A 1143283763  
## 16 6.18e-05 47.4 9.69160719 5MG8\_C 1143283765  
## 17 2.14e-04 45.8 8.44953454 4UX3\_A 728055657  
## 18 2.00e-03 42.7 6.21460810 4UX3\_A 728055657  
## 19 7.81e-04 43.9 7.15493541 3AUX\_A 333944148  
## 20 7.81e-04 43.9 7.15493541 3AUY\_A 333944149  
## 21 7.81e-04 43.9 7.15493541 3AUY\_B 333944150  
## 22 7.81e-04 43.9 7.15493541 3AV0\_B 333944153  
## 23 7.81e-04 43.9 7.15493541 5DNY\_B 1000232403  
## 24 7.81e-04 43.9 7.15493541 5DNY\_D 1000232405  
## 25 8.53e-04 43.5 7.06675101 5F3W\_B 1002351595  
## 26 8.53e-04 43.5 7.06675101 5F3W\_D 1002351597  
## 27 1.00e-03 43.1 6.90775528 5H68\_A 1159374129  
## 28 1.00e-03 43.1 6.90775528 5H68\_B 1159374130  
## 29 3.00e-03 40.8 5.80914299 1II8\_A 14488688  
## 30 4.00e-03 40.4 5.52146092 3QKR\_A 327200667  
## 31 4.00e-03 40.4 5.52146092 3QKS\_A 327200670  
## 32 7.00e-03 40.8 4.96184513 1W1W\_A 55670515  
## 33 7.00e-03 40.8 4.96184513 1W1W\_B 55670516  
## 34 7.00e-03 40.8 4.96184513 1W1W\_C 55670517  
## 35 7.00e-03 40.8 4.96184513 1W1W\_D 55670518  
## 36 1.10e+00 33.9 -0.09531018 1W1W\_A 55670515  
## 37 1.10e+00 33.9 -0.09531018 1W1W\_B 55670516  
## 38 1.10e+00 33.9 -0.09531018 1W1W\_C 55670517  
## 39 1.10e+00 33.9 -0.09531018 1W1W\_D 55670518  
## 40 7.00e-03 40.8 4.96184513 3QKU\_A 327200677  
## 41 7.00e-03 40.8 4.96184513 3QKU\_B 327200678  
## 42 9.00e-03 40.4 4.71053070 4NCH\_A 589911123  
## 43 9.00e-03 40.4 4.71053070 4NCH\_B 589911124  
## 44 9.00e-03 40.4 4.71053070 3QKT\_A 327200673  
## 45 9.00e-03 40.4 4.71053070 3QKT\_B 327200674  
## 46 9.00e-03 40.4 4.71053070 3QKT\_C 327200675  
## 47 9.00e-03 40.4 4.71053070 3QKT\_D 327200676  
## 48 1.00e-02 40.0 4.60517019 4NCI\_A 589911125  
## 49 1.00e-02 40.0 4.60517019 4NCJ\_A 589911126  
## 50 1.00e-02 40.0 4.60517019 4NCJ\_B 589911127  
## 51 1.00e-02 40.0 4.60517019 4NCJ\_C 589911128  
## 52 1.00e-02 40.0 4.60517019 4NCJ\_D 589911129  
## 53 1.10e-02 40.0 4.50986001 4NCK\_A 589911130  
## 54 1.10e-02 40.0 4.50986001 4NCK\_B 589911131  
## 55 2.00e-02 39.3 3.91202301 3ZGX\_A 444302106  
## 56 2.00e-02 39.3 3.91202301 3ZGX\_B 444302107  
## 57 2.20e-02 38.1 3.81671283 5H66\_B 1159374124  
## 58 2.20e-02 38.1 3.81671283 5H67\_B 1159374127  
## 59 4.70e-02 36.6 3.05760768 1US8\_A 39655001  
## 60 5.50e-02 37.0 2.90042209 1XEW\_X 58177330  
## 61 5.50e-02 37.0 2.90042209 1XEX\_A 58177332  
## 62 5.70e-02 36.2 2.86470401 1F2T\_A 9954932  
## 63 5.70e-02 36.2 2.86470401 1F2U\_A 9954934  
## 64 5.70e-02 36.2 2.86470401 1F2U\_C 9954936  
## 65 2.50e-01 34.7 1.38629436 4RMO\_A 931139635  
## 66 2.50e-01 34.7 1.38629436 4RMO\_C 931139637  
## 67 2.50e-01 34.7 1.38629436 4RMO\_E 931139639  
## 68 2.50e-01 34.7 1.38629436 4RMO\_G 931139641  
## 69 2.50e-01 34.7 1.38629436 4RMO\_I 931139643  
## 70 2.50e-01 34.7 1.38629436 4RMO\_K 931139645  
## 71 2.50e-01 34.7 1.38629436 4RMO\_M 931139647  
## 72 2.50e-01 34.7 1.38629436 4RMO\_O 931139649  
## 73 3.10e-01 34.7 1.17118298 3KTA\_A 299856726  
## 74 3.10e-01 34.7 1.17118298 3KTA\_C 299856728  
## 75 7.70e-01 34.7 0.26136476 3WSX\_A 753536165  
## 76 8.90e-01 34.3 0.11653382 3WSZ\_A 753536159  
## 77 8.90e-01 34.3 0.11653382 3WSY\_A 753536162  
## 78 1.80e+00 33.1 -0.58778666 3QF7\_A 329666209  
## 79 1.80e+00 33.1 -0.58778666 3QF7\_B 329666210  
## 80 1.80e+00 33.1 -0.58778666 4W9M\_C 747155454  
## 81 1.80e+00 33.1 -0.58778666 4W9M\_E 747155456  
## 82 1.80e+00 33.1 -0.58778666 4W9M\_I 747155460  
## 83 1.80e+00 33.1 -0.58778666 4W9M\_K 747155462  
## 84 2.50e+00 32.7 -0.91629073 1EU8\_A 13787046  
## 85 3.00e+00 32.3 -1.09861229 3THO\_A 350610724  
## 86 3.60e+00 32.0 -1.28093385 3KL4\_A 292659708  
## 87 3.70e+00 32.0 -1.30833282 1QZW\_A 39654777  
## 88 3.70e+00 32.0 -1.30833282 1QZW\_C 39654779  
## 89 3.70e+00 32.0 -1.30833282 1QZW\_E 39654781  
## 90 3.70e+00 32.0 -1.30833282 1QZW\_G 39654783  
## 91 3.70e+00 32.0 -1.30833282 1QZX\_A 39654785  
## 92 3.70e+00 32.0 -1.30833282 1QZX\_B 39654786  
## 93 4.30e+00 32.0 -1.45861502 4UXV\_A 699030604  
## 94 4.90e+00 32.0 -1.58923521 1QVR\_A 38492937  
## 95 4.90e+00 32.0 -1.58923521 1QVR\_B 38492938  
## 96 4.90e+00 32.0 -1.58923521 1QVR\_C 38492939  
## 97 8.60e+00 30.8 -2.15176220 5T33\_G 1062512851  
##   
## $raw  
## queryid  
## 1 Query\_208689  
## 2 Query\_208689  
## 3 Query\_208689  
## 4 Query\_208689  
## 5 Query\_208689  
## 6 Query\_208689  
## 7 Query\_208689  
## 8 Query\_208689  
## 9 Query\_208689  
## 10 Query\_208689  
## 11 Query\_208689  
## 12 Query\_208689  
## 13 Query\_208689  
## 14 Query\_208689  
## 15 Query\_208689  
## 16 Query\_208689  
## 17 Query\_208689  
## 18 Query\_208689  
## 19 Query\_208689  
## 20 Query\_208689  
## 21 Query\_208689  
## 22 Query\_208689  
## 23 Query\_208689  
## 24 Query\_208689  
## 25 Query\_208689  
## 26 Query\_208689  
## 27 Query\_208689  
## 28 Query\_208689  
## 29 Query\_208689  
## 30 Query\_208689  
## 31 Query\_208689  
## 32 Query\_208689  
## 33 Query\_208689  
## 34 Query\_208689  
## 35 Query\_208689  
## 36 Query\_208689  
## 37 Query\_208689  
## 38 Query\_208689  
## subjectids  
## 1 gi|444302296|pdb|4I99|A;gi|444302297|pdb|4I99|B  
## 2 gi|444302296|pdb|4I99|A;gi|444302297|pdb|4I99|B  
## 3 gi|299856727|pdb|3KTA|B;gi|299856729|pdb|3KTA|D  
## 4 gi|58177331|pdb|1XEW|Y  
## 5 gi|58177333|pdb|1XEX|B  
## 6 gi|13096783|pdb|1E69|A;gi|13096784|pdb|1E69|B;gi|13096785|pdb|1E69|C;gi|13096786|pdb|1E69|D;gi|13096787|pdb|1E69|E;gi|13096788|pdb|1E69|F  
## 7 gi|1143283763|pdb|5MG8|A;gi|1143283765|pdb|5MG8|C  
## 8 gi|728055657|pdb|4UX3|A  
## 9 gi|728055657|pdb|4UX3|A  
## 10 gi|333944148|pdb|3AUX|A;gi|333944149|pdb|3AUY|A;gi|333944150|pdb|3AUY|B;gi|333944153|pdb|3AV0|B;gi|1000232403|pdb|5DNY|B;gi|1000232405|pdb|5DNY|D  
## 11 gi|1002351595|pdb|5F3W|B;gi|1002351597|pdb|5F3W|D  
## 12 gi|1159374129|pdb|5H68|A;gi|1159374130|pdb|5H68|B  
## 13 gi|14488688|pdb|1II8|A  
## 14 gi|327200667|pdb|3QKR|A;gi|327200670|pdb|3QKS|A  
## 15 gi|55670515|pdb|1W1W|A;gi|55670516|pdb|1W1W|B;gi|55670517|pdb|1W1W|C;gi|55670518|pdb|1W1W|D  
## 16 gi|55670515|pdb|1W1W|A;gi|55670516|pdb|1W1W|B;gi|55670517|pdb|1W1W|C;gi|55670518|pdb|1W1W|D  
## 17 gi|327200677|pdb|3QKU|A;gi|327200678|pdb|3QKU|B  
## 18 gi|589911123|pdb|4NCH|A;gi|589911124|pdb|4NCH|B  
## 19 gi|327200673|pdb|3QKT|A;gi|327200674|pdb|3QKT|B;gi|327200675|pdb|3QKT|C;gi|327200676|pdb|3QKT|D  
## 20 gi|589911125|pdb|4NCI|A;gi|589911126|pdb|4NCJ|A;gi|589911127|pdb|4NCJ|B;gi|589911128|pdb|4NCJ|C;gi|589911129|pdb|4NCJ|D  
## 21 gi|589911130|pdb|4NCK|A;gi|589911131|pdb|4NCK|B  
## 22 gi|444302106|pdb|3ZGX|A;gi|444302107|pdb|3ZGX|B  
## 23 gi|1159374124|pdb|5H66|B;gi|1159374127|pdb|5H67|B  
## 24 gi|39655001|pdb|1US8|A  
## 25 gi|58177330|pdb|1XEW|X;gi|58177332|pdb|1XEX|A  
## 26 gi|9954932|pdb|1F2T|A;gi|9954934|pdb|1F2U|A;gi|9954936|pdb|1F2U|C  
## 27 gi|931139635|pdb|4RMO|A;gi|931139637|pdb|4RMO|C;gi|931139639|pdb|4RMO|E;gi|931139641|pdb|4RMO|G;gi|931139643|pdb|4RMO|I;gi|931139645|pdb|4RMO|K;gi|931139647|pdb|4RMO|M;gi|931139649|pdb|4RMO|O  
## 28 gi|299856726|pdb|3KTA|A;gi|299856728|pdb|3KTA|C  
## 29 gi|753536165|pdb|3WSX|A  
## 30 gi|753536159|pdb|3WSZ|A;gi|753536162|pdb|3WSY|A  
## 31 gi|329666209|pdb|3QF7|A;gi|329666210|pdb|3QF7|B;gi|747155454|pdb|4W9M|C;gi|747155456|pdb|4W9M|E;gi|747155460|pdb|4W9M|I;gi|747155462|pdb|4W9M|K  
## 32 gi|13787046|pdb|1EU8|A  
## 33 gi|350610724|pdb|3THO|A  
## 34 gi|292659708|pdb|3KL4|A  
## 35 gi|39654777|pdb|1QZW|A;gi|39654779|pdb|1QZW|C;gi|39654781|pdb|1QZW|E;gi|39654783|pdb|1QZW|G;gi|39654785|pdb|1QZX|A;gi|39654786|pdb|1QZX|B  
## 36 gi|699030604|pdb|4UXV|A  
## 37 gi|38492937|pdb|1QVR|A;gi|38492938|pdb|1QVR|B;gi|38492939|pdb|1QVR|C  
## 38 gi|1062512851|pdb|5T33|G  
## identity positives alignmentlength mismatches gapopens q.start q.end  
## 1 29.032 50.00 124 82 4 930 1053  
## 2 24.242 43.03 165 100 6 47 194  
## 3 27.419 48.39 124 84 3 930 1053  
## 4 27.419 48.39 124 84 3 930 1053  
## 5 26.613 48.39 124 85 3 930 1053  
## 6 30.252 50.42 119 70 4 937 1052  
## 7 21.154 42.31 208 150 6 398 592  
## 8 22.907 43.17 227 152 6 53 259  
## 9 21.429 41.18 238 141 7 844 1052  
## 10 26.257 44.69 179 118 5 56 226  
## 11 26.257 44.69 179 118 5 56 226  
## 12 27.523 47.71 109 72 3 924 1032  
## 13 39.535 69.77 43 26 0 55 97  
## 14 39.535 69.77 43 26 0 55 97  
## 15 27.711 49.40 83 55 2 952 1033  
## 16 34.000 58.00 50 31 2 51 98  
## 17 39.535 69.77 43 26 0 55 97  
## 18 39.535 69.77 43 26 0 55 97  
## 19 39.535 69.77 43 26 0 55 97  
## 20 39.535 69.77 43 26 0 55 97  
## 21 39.535 69.77 43 26 0 55 97  
## 22 23.179 46.36 151 99 5 914 1053  
## 23 22.517 46.36 151 100 5 914 1053  
## 24 39.535 69.77 43 26 0 55 97  
## 25 24.242 44.24 165 100 6 47 194  
## 26 39.535 69.77 43 26 0 55 97  
## 27 29.630 55.56 81 49 3 748 821  
## 28 24.638 44.20 138 81 5 47 167  
## 29 44.444 55.56 45 21 1 489 529  
## 30 44.444 55.56 45 21 1 489 529  
## 31 33.333 53.33 45 30 0 55 99  
## 32 23.656 46.24 93 64 2 862 950  
## 33 33.333 53.33 45 30 0 55 99  
## 34 30.508 50.85 59 25 2 73 131  
## 35 30.508 50.85 59 25 2 73 131  
## 36 32.000 54.67 75 46 3 292 361  
## 37 37.500 64.58 48 28 2 892 937  
## 38 35.185 57.41 54 29 2 913 962  
## s.start s.end evalue bitscore  
## 1 188 305 6.55e-07 53.5  
## 2 2 158 6.90e-01 34.3  
## 3 7 124 1.39e-06 50.4  
## 4 6 123 1.44e-06 50.1  
## 5 6 123 3.39e-06 49.3  
## 6 168 276 5.48e-05 47.4  
## 7 20 226 6.18e-05 47.4  
## 8 4 227 2.14e-04 45.8  
## 9 267 487 2.00e-03 42.7  
## 10 8 180 7.81e-04 43.9  
## 11 8 180 8.53e-04 43.5  
## 12 182 283 1.00e-03 43.1  
## 13 5 47 3.00e-03 40.8  
## 14 5 47 4.00e-03 40.4  
## 15 298 376 7.00e-03 40.8  
## 16 2 51 1.10e+00 33.9  
## 17 5 47 7.00e-03 40.8  
## 18 5 47 9.00e-03 40.4  
## 19 5 47 9.00e-03 40.4  
## 20 5 47 1.00e-02 40.0  
## 21 5 47 1.10e-02 40.0  
## 22 244 388 2.00e-02 39.3  
## 23 6 150 2.20e-02 38.1  
## 24 5 47 4.70e-02 36.6  
## 25 2 158 5.50e-02 37.0  
## 26 5 47 5.70e-02 36.2  
## 27 59 138 2.50e-01 34.7  
## 28 2 133 3.10e-01 34.7  
## 29 227 271 7.70e-01 34.7  
## 30 171 215 8.90e-01 34.3  
## 31 5 49 1.80e+00 33.1  
## 32 14 103 2.50e+00 32.7  
## 33 22 66 3.00e+00 32.3  
## 34 97 139 3.60e+00 32.0  
## 35 104 146 3.70e+00 32.0  
## 36 409 483 4.30e+00 32.0  
## 37 496 543 4.90e+00 32.0  
## 38 192 243 8.60e+00 30.8  
##   
## $url  
## EY517001013   
## "https://blast.ncbi.nlm.nih.gov/Blast.cgi?CMD=Get&FORMAT\_OBJECT=Alignment&ALIGNMENT\_VIEW=Tabular&RESULTS\_FILE=on&FORMAT\_TYPE=CSV&ALIGNMENTS=20000&RID=EY517001013"   
##   
## attr(,"class")  
## [1] "blast"

#Plot summary statistics of the pdb blast  
#plotting the blast results gives a graphical output for the E-value, bitscore identity and length for each hit. The E value, is used to determine the significance of the blast results. A higher -log(E) indicates a higher significance. (Unlike the raw score, this score is independent og the substitution matrix and the query database lengths, and thus is comparable between BLASTP searches).  
  
plot(blast\_results)

## \* Possible cutoff values: 14 0 8 4 -1 2 -3   
## Yielding Nhits: 2 4 17 35 39 64 97   
##   
## \* Chosen cutoff value of: 8   
## Yielding Nhits: 15



#remove duplicates  
  
UniqueBlasts <- blast\_results$hit.tbl[!duplicated(blast\_results$hit.tbl$subjectids),] #remove duplicate entries based on the subjectids  
  
UniqueBlasts

## queryid subjectids identity positives  
## 1 Query\_208689 gi|444302296|pdb|4I99|A 29.032 50.00  
## 2 Query\_208689 gi|444302297|pdb|4I99|B 29.032 50.00  
## 5 Query\_208689 gi|299856727|pdb|3KTA|B 27.419 48.39  
## 6 Query\_208689 gi|299856729|pdb|3KTA|D 27.419 48.39  
## 7 Query\_208689 gi|58177331|pdb|1XEW|Y 27.419 48.39  
## 8 Query\_208689 gi|58177333|pdb|1XEX|B 26.613 48.39  
## 9 Query\_208689 gi|13096783|pdb|1E69|A 30.252 50.42  
## 10 Query\_208689 gi|13096784|pdb|1E69|B 30.252 50.42  
## 11 Query\_208689 gi|13096785|pdb|1E69|C 30.252 50.42  
## 12 Query\_208689 gi|13096786|pdb|1E69|D 30.252 50.42  
## 13 Query\_208689 gi|13096787|pdb|1E69|E 30.252 50.42  
## 14 Query\_208689 gi|13096788|pdb|1E69|F 30.252 50.42  
## 15 Query\_208689 gi|1143283763|pdb|5MG8|A 21.154 42.31  
## 16 Query\_208689 gi|1143283765|pdb|5MG8|C 21.154 42.31  
## 17 Query\_208689 gi|728055657|pdb|4UX3|A 22.907 43.17  
## 19 Query\_208689 gi|333944148|pdb|3AUX|A 26.257 44.69  
## 20 Query\_208689 gi|333944149|pdb|3AUY|A 26.257 44.69  
## 21 Query\_208689 gi|333944150|pdb|3AUY|B 26.257 44.69  
## 22 Query\_208689 gi|333944153|pdb|3AV0|B 26.257 44.69  
## 23 Query\_208689 gi|1000232403|pdb|5DNY|B 26.257 44.69  
## 24 Query\_208689 gi|1000232405|pdb|5DNY|D 26.257 44.69  
## 25 Query\_208689 gi|1002351595|pdb|5F3W|B 26.257 44.69  
## 26 Query\_208689 gi|1002351597|pdb|5F3W|D 26.257 44.69  
## 27 Query\_208689 gi|1159374129|pdb|5H68|A 27.523 47.71  
## 28 Query\_208689 gi|1159374130|pdb|5H68|B 27.523 47.71  
## 29 Query\_208689 gi|14488688|pdb|1II8|A 39.535 69.77  
## 30 Query\_208689 gi|327200667|pdb|3QKR|A 39.535 69.77  
## 31 Query\_208689 gi|327200670|pdb|3QKS|A 39.535 69.77  
## 32 Query\_208689 gi|55670515|pdb|1W1W|A 27.711 49.40  
## 33 Query\_208689 gi|55670516|pdb|1W1W|B 27.711 49.40  
## 34 Query\_208689 gi|55670517|pdb|1W1W|C 27.711 49.40  
## 35 Query\_208689 gi|55670518|pdb|1W1W|D 27.711 49.40  
## 40 Query\_208689 gi|327200677|pdb|3QKU|A 39.535 69.77  
## 41 Query\_208689 gi|327200678|pdb|3QKU|B 39.535 69.77  
## 42 Query\_208689 gi|589911123|pdb|4NCH|A 39.535 69.77  
## 43 Query\_208689 gi|589911124|pdb|4NCH|B 39.535 69.77  
## 44 Query\_208689 gi|327200673|pdb|3QKT|A 39.535 69.77  
## 45 Query\_208689 gi|327200674|pdb|3QKT|B 39.535 69.77  
## 46 Query\_208689 gi|327200675|pdb|3QKT|C 39.535 69.77  
## 47 Query\_208689 gi|327200676|pdb|3QKT|D 39.535 69.77  
## 48 Query\_208689 gi|589911125|pdb|4NCI|A 39.535 69.77  
## 49 Query\_208689 gi|589911126|pdb|4NCJ|A 39.535 69.77  
## 50 Query\_208689 gi|589911127|pdb|4NCJ|B 39.535 69.77  
## 51 Query\_208689 gi|589911128|pdb|4NCJ|C 39.535 69.77  
## 52 Query\_208689 gi|589911129|pdb|4NCJ|D 39.535 69.77  
## 53 Query\_208689 gi|589911130|pdb|4NCK|A 39.535 69.77  
## 54 Query\_208689 gi|589911131|pdb|4NCK|B 39.535 69.77  
## 55 Query\_208689 gi|444302106|pdb|3ZGX|A 23.179 46.36  
## 56 Query\_208689 gi|444302107|pdb|3ZGX|B 23.179 46.36  
## 57 Query\_208689 gi|1159374124|pdb|5H66|B 22.517 46.36  
## 58 Query\_208689 gi|1159374127|pdb|5H67|B 22.517 46.36  
## 59 Query\_208689 gi|39655001|pdb|1US8|A 39.535 69.77  
## 60 Query\_208689 gi|58177330|pdb|1XEW|X 24.242 44.24  
## 61 Query\_208689 gi|58177332|pdb|1XEX|A 24.242 44.24  
## 62 Query\_208689 gi|9954932|pdb|1F2T|A 39.535 69.77  
## 63 Query\_208689 gi|9954934|pdb|1F2U|A 39.535 69.77  
## 64 Query\_208689 gi|9954936|pdb|1F2U|C 39.535 69.77  
## 65 Query\_208689 gi|931139635|pdb|4RMO|A 29.630 55.56  
## 66 Query\_208689 gi|931139637|pdb|4RMO|C 29.630 55.56  
## 67 Query\_208689 gi|931139639|pdb|4RMO|E 29.630 55.56  
## 68 Query\_208689 gi|931139641|pdb|4RMO|G 29.630 55.56  
## 69 Query\_208689 gi|931139643|pdb|4RMO|I 29.630 55.56  
## 70 Query\_208689 gi|931139645|pdb|4RMO|K 29.630 55.56  
## 71 Query\_208689 gi|931139647|pdb|4RMO|M 29.630 55.56  
## 72 Query\_208689 gi|931139649|pdb|4RMO|O 29.630 55.56  
## 73 Query\_208689 gi|299856726|pdb|3KTA|A 24.638 44.20  
## 74 Query\_208689 gi|299856728|pdb|3KTA|C 24.638 44.20  
## 75 Query\_208689 gi|753536165|pdb|3WSX|A 44.444 55.56  
## 76 Query\_208689 gi|753536159|pdb|3WSZ|A 44.444 55.56  
## 77 Query\_208689 gi|753536162|pdb|3WSY|A 44.444 55.56  
## 78 Query\_208689 gi|329666209|pdb|3QF7|A 33.333 53.33  
## 79 Query\_208689 gi|329666210|pdb|3QF7|B 33.333 53.33  
## 80 Query\_208689 gi|747155454|pdb|4W9M|C 33.333 53.33  
## 81 Query\_208689 gi|747155456|pdb|4W9M|E 33.333 53.33  
## 82 Query\_208689 gi|747155460|pdb|4W9M|I 33.333 53.33  
## 83 Query\_208689 gi|747155462|pdb|4W9M|K 33.333 53.33  
## 84 Query\_208689 gi|13787046|pdb|1EU8|A 23.656 46.24  
## 85 Query\_208689 gi|350610724|pdb|3THO|A 33.333 53.33  
## 86 Query\_208689 gi|292659708|pdb|3KL4|A 30.508 50.85  
## 87 Query\_208689 gi|39654777|pdb|1QZW|A 30.508 50.85  
## 88 Query\_208689 gi|39654779|pdb|1QZW|C 30.508 50.85  
## 89 Query\_208689 gi|39654781|pdb|1QZW|E 30.508 50.85  
## 90 Query\_208689 gi|39654783|pdb|1QZW|G 30.508 50.85  
## 91 Query\_208689 gi|39654785|pdb|1QZX|A 30.508 50.85  
## 92 Query\_208689 gi|39654786|pdb|1QZX|B 30.508 50.85  
## 93 Query\_208689 gi|699030604|pdb|4UXV|A 32.000 54.67  
## 94 Query\_208689 gi|38492937|pdb|1QVR|A 37.500 64.58  
## 95 Query\_208689 gi|38492938|pdb|1QVR|B 37.500 64.58  
## 96 Query\_208689 gi|38492939|pdb|1QVR|C 37.500 64.58  
## 97 Query\_208689 gi|1062512851|pdb|5T33|G 35.185 57.41  
## alignmentlength mismatches gapopens q.start q.end s.start s.end  
## 1 124 82 4 930 1053 188 305  
## 2 124 82 4 930 1053 188 305  
## 5 124 84 3 930 1053 7 124  
## 6 124 84 3 930 1053 7 124  
## 7 124 84 3 930 1053 6 123  
## 8 124 85 3 930 1053 6 123  
## 9 119 70 4 937 1052 168 276  
## 10 119 70 4 937 1052 168 276  
## 11 119 70 4 937 1052 168 276  
## 12 119 70 4 937 1052 168 276  
## 13 119 70 4 937 1052 168 276  
## 14 119 70 4 937 1052 168 276  
## 15 208 150 6 398 592 20 226  
## 16 208 150 6 398 592 20 226  
## 17 227 152 6 53 259 4 227  
## 19 179 118 5 56 226 8 180  
## 20 179 118 5 56 226 8 180  
## 21 179 118 5 56 226 8 180  
## 22 179 118 5 56 226 8 180  
## 23 179 118 5 56 226 8 180  
## 24 179 118 5 56 226 8 180  
## 25 179 118 5 56 226 8 180  
## 26 179 118 5 56 226 8 180  
## 27 109 72 3 924 1032 182 283  
## 28 109 72 3 924 1032 182 283  
## 29 43 26 0 55 97 5 47  
## 30 43 26 0 55 97 5 47  
## 31 43 26 0 55 97 5 47  
## 32 83 55 2 952 1033 298 376  
## 33 83 55 2 952 1033 298 376  
## 34 83 55 2 952 1033 298 376  
## 35 83 55 2 952 1033 298 376  
## 40 43 26 0 55 97 5 47  
## 41 43 26 0 55 97 5 47  
## 42 43 26 0 55 97 5 47  
## 43 43 26 0 55 97 5 47  
## 44 43 26 0 55 97 5 47  
## 45 43 26 0 55 97 5 47  
## 46 43 26 0 55 97 5 47  
## 47 43 26 0 55 97 5 47  
## 48 43 26 0 55 97 5 47  
## 49 43 26 0 55 97 5 47  
## 50 43 26 0 55 97 5 47  
## 51 43 26 0 55 97 5 47  
## 52 43 26 0 55 97 5 47  
## 53 43 26 0 55 97 5 47  
## 54 43 26 0 55 97 5 47  
## 55 151 99 5 914 1053 244 388  
## 56 151 99 5 914 1053 244 388  
## 57 151 100 5 914 1053 6 150  
## 58 151 100 5 914 1053 6 150  
## 59 43 26 0 55 97 5 47  
## 60 165 100 6 47 194 2 158  
## 61 165 100 6 47 194 2 158  
## 62 43 26 0 55 97 5 47  
## 63 43 26 0 55 97 5 47  
## 64 43 26 0 55 97 5 47  
## 65 81 49 3 748 821 59 138  
## 66 81 49 3 748 821 59 138  
## 67 81 49 3 748 821 59 138  
## 68 81 49 3 748 821 59 138  
## 69 81 49 3 748 821 59 138  
## 70 81 49 3 748 821 59 138  
## 71 81 49 3 748 821 59 138  
## 72 81 49 3 748 821 59 138  
## 73 138 81 5 47 167 2 133  
## 74 138 81 5 47 167 2 133  
## 75 45 21 1 489 529 227 271  
## 76 45 21 1 489 529 171 215  
## 77 45 21 1 489 529 171 215  
## 78 45 30 0 55 99 5 49  
## 79 45 30 0 55 99 5 49  
## 80 45 30 0 55 99 5 49  
## 81 45 30 0 55 99 5 49  
## 82 45 30 0 55 99 5 49  
## 83 45 30 0 55 99 5 49  
## 84 93 64 2 862 950 14 103  
## 85 45 30 0 55 99 22 66  
## 86 59 25 2 73 131 97 139  
## 87 59 25 2 73 131 104 146  
## 88 59 25 2 73 131 104 146  
## 89 59 25 2 73 131 104 146  
## 90 59 25 2 73 131 104 146  
## 91 59 25 2 73 131 104 146  
## 92 59 25 2 73 131 104 146  
## 93 75 46 3 292 361 409 483  
## 94 48 28 2 892 937 496 543  
## 95 48 28 2 892 937 496 543  
## 96 48 28 2 892 937 496 543  
## 97 54 29 2 913 962 192 243  
## evalue bitscore mlog.evalue pdb.id acc  
## 1 6.55e-07 53.5 14.2386306 4I99\_A 444302296  
## 2 6.55e-07 53.5 14.2386306 4I99\_B 444302297  
## 5 1.39e-06 50.4 13.4862068 3KTA\_B 299856727  
## 6 1.39e-06 50.4 13.4862068 3KTA\_D 299856729  
## 7 1.44e-06 50.1 13.4508674 1XEW\_Y 58177331  
## 8 3.39e-06 49.3 12.5946806 1XEX\_B 58177333  
## 9 5.48e-05 47.4 9.8118204 1E69\_A 13096783  
## 10 5.48e-05 47.4 9.8118204 1E69\_B 13096784  
## 11 5.48e-05 47.4 9.8118204 1E69\_C 13096785  
## 12 5.48e-05 47.4 9.8118204 1E69\_D 13096786  
## 13 5.48e-05 47.4 9.8118204 1E69\_E 13096787  
## 14 5.48e-05 47.4 9.8118204 1E69\_F 13096788  
## 15 6.18e-05 47.4 9.6916072 5MG8\_A 1143283763  
## 16 6.18e-05 47.4 9.6916072 5MG8\_C 1143283765  
## 17 2.14e-04 45.8 8.4495345 4UX3\_A 728055657  
## 19 7.81e-04 43.9 7.1549354 3AUX\_A 333944148  
## 20 7.81e-04 43.9 7.1549354 3AUY\_A 333944149  
## 21 7.81e-04 43.9 7.1549354 3AUY\_B 333944150  
## 22 7.81e-04 43.9 7.1549354 3AV0\_B 333944153  
## 23 7.81e-04 43.9 7.1549354 5DNY\_B 1000232403  
## 24 7.81e-04 43.9 7.1549354 5DNY\_D 1000232405  
## 25 8.53e-04 43.5 7.0667510 5F3W\_B 1002351595  
## 26 8.53e-04 43.5 7.0667510 5F3W\_D 1002351597  
## 27 1.00e-03 43.1 6.9077553 5H68\_A 1159374129  
## 28 1.00e-03 43.1 6.9077553 5H68\_B 1159374130  
## 29 3.00e-03 40.8 5.8091430 1II8\_A 14488688  
## 30 4.00e-03 40.4 5.5214609 3QKR\_A 327200667  
## 31 4.00e-03 40.4 5.5214609 3QKS\_A 327200670  
## 32 7.00e-03 40.8 4.9618451 1W1W\_A 55670515  
## 33 7.00e-03 40.8 4.9618451 1W1W\_B 55670516  
## 34 7.00e-03 40.8 4.9618451 1W1W\_C 55670517  
## 35 7.00e-03 40.8 4.9618451 1W1W\_D 55670518  
## 40 7.00e-03 40.8 4.9618451 3QKU\_A 327200677  
## 41 7.00e-03 40.8 4.9618451 3QKU\_B 327200678  
## 42 9.00e-03 40.4 4.7105307 4NCH\_A 589911123  
## 43 9.00e-03 40.4 4.7105307 4NCH\_B 589911124  
## 44 9.00e-03 40.4 4.7105307 3QKT\_A 327200673  
## 45 9.00e-03 40.4 4.7105307 3QKT\_B 327200674  
## 46 9.00e-03 40.4 4.7105307 3QKT\_C 327200675  
## 47 9.00e-03 40.4 4.7105307 3QKT\_D 327200676  
## 48 1.00e-02 40.0 4.6051702 4NCI\_A 589911125  
## 49 1.00e-02 40.0 4.6051702 4NCJ\_A 589911126  
## 50 1.00e-02 40.0 4.6051702 4NCJ\_B 589911127  
## 51 1.00e-02 40.0 4.6051702 4NCJ\_C 589911128  
## 52 1.00e-02 40.0 4.6051702 4NCJ\_D 589911129  
## 53 1.10e-02 40.0 4.5098600 4NCK\_A 589911130  
## 54 1.10e-02 40.0 4.5098600 4NCK\_B 589911131  
## 55 2.00e-02 39.3 3.9120230 3ZGX\_A 444302106  
## 56 2.00e-02 39.3 3.9120230 3ZGX\_B 444302107  
## 57 2.20e-02 38.1 3.8167128 5H66\_B 1159374124  
## 58 2.20e-02 38.1 3.8167128 5H67\_B 1159374127  
## 59 4.70e-02 36.6 3.0576077 1US8\_A 39655001  
## 60 5.50e-02 37.0 2.9004221 1XEW\_X 58177330  
## 61 5.50e-02 37.0 2.9004221 1XEX\_A 58177332  
## 62 5.70e-02 36.2 2.8647040 1F2T\_A 9954932  
## 63 5.70e-02 36.2 2.8647040 1F2U\_A 9954934  
## 64 5.70e-02 36.2 2.8647040 1F2U\_C 9954936  
## 65 2.50e-01 34.7 1.3862944 4RMO\_A 931139635  
## 66 2.50e-01 34.7 1.3862944 4RMO\_C 931139637  
## 67 2.50e-01 34.7 1.3862944 4RMO\_E 931139639  
## 68 2.50e-01 34.7 1.3862944 4RMO\_G 931139641  
## 69 2.50e-01 34.7 1.3862944 4RMO\_I 931139643  
## 70 2.50e-01 34.7 1.3862944 4RMO\_K 931139645  
## 71 2.50e-01 34.7 1.3862944 4RMO\_M 931139647  
## 72 2.50e-01 34.7 1.3862944 4RMO\_O 931139649  
## 73 3.10e-01 34.7 1.1711830 3KTA\_A 299856726  
## 74 3.10e-01 34.7 1.1711830 3KTA\_C 299856728  
## 75 7.70e-01 34.7 0.2613648 3WSX\_A 753536165  
## 76 8.90e-01 34.3 0.1165338 3WSZ\_A 753536159  
## 77 8.90e-01 34.3 0.1165338 3WSY\_A 753536162  
## 78 1.80e+00 33.1 -0.5877867 3QF7\_A 329666209  
## 79 1.80e+00 33.1 -0.5877867 3QF7\_B 329666210  
## 80 1.80e+00 33.1 -0.5877867 4W9M\_C 747155454  
## 81 1.80e+00 33.1 -0.5877867 4W9M\_E 747155456  
## 82 1.80e+00 33.1 -0.5877867 4W9M\_I 747155460  
## 83 1.80e+00 33.1 -0.5877867 4W9M\_K 747155462  
## 84 2.50e+00 32.7 -0.9162907 1EU8\_A 13787046  
## 85 3.00e+00 32.3 -1.0986123 3THO\_A 350610724  
## 86 3.60e+00 32.0 -1.2809338 3KL4\_A 292659708  
## 87 3.70e+00 32.0 -1.3083328 1QZW\_A 39654777  
## 88 3.70e+00 32.0 -1.3083328 1QZW\_C 39654779  
## 89 3.70e+00 32.0 -1.3083328 1QZW\_E 39654781  
## 90 3.70e+00 32.0 -1.3083328 1QZW\_G 39654783  
## 91 3.70e+00 32.0 -1.3083328 1QZX\_A 39654785  
## 92 3.70e+00 32.0 -1.3083328 1QZX\_B 39654786  
## 93 4.30e+00 32.0 -1.4586150 4UXV\_A 699030604  
## 94 4.90e+00 32.0 -1.5892352 1QVR\_A 38492937  
## 95 4.90e+00 32.0 -1.5892352 1QVR\_B 38492938  
## 96 4.90e+00 32.0 -1.5892352 1QVR\_C 38492939  
## 97 8.60e+00 30.8 -2.1517622 5T33\_G 1062512851

#Can get different isoforms of the same protein. These are essentially duplicate entries, but however will not be removed with the above code as they have a different letter at the end of their subject id. As these hits will have the same statistical scores so can remove duplicates by removing the duplicates based on the evalue  
  
UniqueBlasts <- UniqueBlasts[!duplicated(UniqueBlasts$evalue),] #remove duplicate entries based on the evalue  
  
UniqueBlasts

## queryid subjectids identity positives  
## 1 Query\_208689 gi|444302296|pdb|4I99|A 29.032 50.00  
## 5 Query\_208689 gi|299856727|pdb|3KTA|B 27.419 48.39  
## 7 Query\_208689 gi|58177331|pdb|1XEW|Y 27.419 48.39  
## 8 Query\_208689 gi|58177333|pdb|1XEX|B 26.613 48.39  
## 9 Query\_208689 gi|13096783|pdb|1E69|A 30.252 50.42  
## 15 Query\_208689 gi|1143283763|pdb|5MG8|A 21.154 42.31  
## 17 Query\_208689 gi|728055657|pdb|4UX3|A 22.907 43.17  
## 19 Query\_208689 gi|333944148|pdb|3AUX|A 26.257 44.69  
## 25 Query\_208689 gi|1002351595|pdb|5F3W|B 26.257 44.69  
## 27 Query\_208689 gi|1159374129|pdb|5H68|A 27.523 47.71  
## 29 Query\_208689 gi|14488688|pdb|1II8|A 39.535 69.77  
## 30 Query\_208689 gi|327200667|pdb|3QKR|A 39.535 69.77  
## 32 Query\_208689 gi|55670515|pdb|1W1W|A 27.711 49.40  
## 42 Query\_208689 gi|589911123|pdb|4NCH|A 39.535 69.77  
## 48 Query\_208689 gi|589911125|pdb|4NCI|A 39.535 69.77  
## 53 Query\_208689 gi|589911130|pdb|4NCK|A 39.535 69.77  
## 55 Query\_208689 gi|444302106|pdb|3ZGX|A 23.179 46.36  
## 57 Query\_208689 gi|1159374124|pdb|5H66|B 22.517 46.36  
## 59 Query\_208689 gi|39655001|pdb|1US8|A 39.535 69.77  
## 60 Query\_208689 gi|58177330|pdb|1XEW|X 24.242 44.24  
## 62 Query\_208689 gi|9954932|pdb|1F2T|A 39.535 69.77  
## 65 Query\_208689 gi|931139635|pdb|4RMO|A 29.630 55.56  
## 73 Query\_208689 gi|299856726|pdb|3KTA|A 24.638 44.20  
## 75 Query\_208689 gi|753536165|pdb|3WSX|A 44.444 55.56  
## 76 Query\_208689 gi|753536159|pdb|3WSZ|A 44.444 55.56  
## 78 Query\_208689 gi|329666209|pdb|3QF7|A 33.333 53.33  
## 84 Query\_208689 gi|13787046|pdb|1EU8|A 23.656 46.24  
## 85 Query\_208689 gi|350610724|pdb|3THO|A 33.333 53.33  
## 86 Query\_208689 gi|292659708|pdb|3KL4|A 30.508 50.85  
## 87 Query\_208689 gi|39654777|pdb|1QZW|A 30.508 50.85  
## 93 Query\_208689 gi|699030604|pdb|4UXV|A 32.000 54.67  
## 94 Query\_208689 gi|38492937|pdb|1QVR|A 37.500 64.58  
## 97 Query\_208689 gi|1062512851|pdb|5T33|G 35.185 57.41  
## alignmentlength mismatches gapopens q.start q.end s.start s.end  
## 1 124 82 4 930 1053 188 305  
## 5 124 84 3 930 1053 7 124  
## 7 124 84 3 930 1053 6 123  
## 8 124 85 3 930 1053 6 123  
## 9 119 70 4 937 1052 168 276  
## 15 208 150 6 398 592 20 226  
## 17 227 152 6 53 259 4 227  
## 19 179 118 5 56 226 8 180  
## 25 179 118 5 56 226 8 180  
## 27 109 72 3 924 1032 182 283  
## 29 43 26 0 55 97 5 47  
## 30 43 26 0 55 97 5 47  
## 32 83 55 2 952 1033 298 376  
## 42 43 26 0 55 97 5 47  
## 48 43 26 0 55 97 5 47  
## 53 43 26 0 55 97 5 47  
## 55 151 99 5 914 1053 244 388  
## 57 151 100 5 914 1053 6 150  
## 59 43 26 0 55 97 5 47  
## 60 165 100 6 47 194 2 158  
## 62 43 26 0 55 97 5 47  
## 65 81 49 3 748 821 59 138  
## 73 138 81 5 47 167 2 133  
## 75 45 21 1 489 529 227 271  
## 76 45 21 1 489 529 171 215  
## 78 45 30 0 55 99 5 49  
## 84 93 64 2 862 950 14 103  
## 85 45 30 0 55 99 22 66  
## 86 59 25 2 73 131 97 139  
## 87 59 25 2 73 131 104 146  
## 93 75 46 3 292 361 409 483  
## 94 48 28 2 892 937 496 543  
## 97 54 29 2 913 962 192 243  
## evalue bitscore mlog.evalue pdb.id acc  
## 1 6.55e-07 53.5 14.2386306 4I99\_A 444302296  
## 5 1.39e-06 50.4 13.4862068 3KTA\_B 299856727  
## 7 1.44e-06 50.1 13.4508674 1XEW\_Y 58177331  
## 8 3.39e-06 49.3 12.5946806 1XEX\_B 58177333  
## 9 5.48e-05 47.4 9.8118204 1E69\_A 13096783  
## 15 6.18e-05 47.4 9.6916072 5MG8\_A 1143283763  
## 17 2.14e-04 45.8 8.4495345 4UX3\_A 728055657  
## 19 7.81e-04 43.9 7.1549354 3AUX\_A 333944148  
## 25 8.53e-04 43.5 7.0667510 5F3W\_B 1002351595  
## 27 1.00e-03 43.1 6.9077553 5H68\_A 1159374129  
## 29 3.00e-03 40.8 5.8091430 1II8\_A 14488688  
## 30 4.00e-03 40.4 5.5214609 3QKR\_A 327200667  
## 32 7.00e-03 40.8 4.9618451 1W1W\_A 55670515  
## 42 9.00e-03 40.4 4.7105307 4NCH\_A 589911123  
## 48 1.00e-02 40.0 4.6051702 4NCI\_A 589911125  
## 53 1.10e-02 40.0 4.5098600 4NCK\_A 589911130  
## 55 2.00e-02 39.3 3.9120230 3ZGX\_A 444302106  
## 57 2.20e-02 38.1 3.8167128 5H66\_B 1159374124  
## 59 4.70e-02 36.6 3.0576077 1US8\_A 39655001  
## 60 5.50e-02 37.0 2.9004221 1XEW\_X 58177330  
## 62 5.70e-02 36.2 2.8647040 1F2T\_A 9954932  
## 65 2.50e-01 34.7 1.3862944 4RMO\_A 931139635  
## 73 3.10e-01 34.7 1.1711830 3KTA\_A 299856726  
## 75 7.70e-01 34.7 0.2613648 3WSX\_A 753536165  
## 76 8.90e-01 34.3 0.1165338 3WSZ\_A 753536159  
## 78 1.80e+00 33.1 -0.5877867 3QF7\_A 329666209  
## 84 2.50e+00 32.7 -0.9162907 1EU8\_A 13787046  
## 85 3.00e+00 32.3 -1.0986123 3THO\_A 350610724  
## 86 3.60e+00 32.0 -1.2809338 3KL4\_A 292659708  
## 87 3.70e+00 32.0 -1.3083328 1QZW\_A 39654777  
## 93 4.30e+00 32.0 -1.4586150 4UXV\_A 699030604  
## 94 4.90e+00 32.0 -1.5892352 1QVR\_A 38492937  
## 97 8.60e+00 30.8 -2.1517622 5T33\_G 1062512851

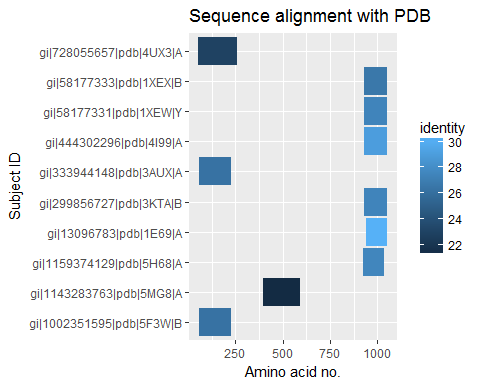
#Dont want really small fragments, when expressing proteins, want to express sequences which have biological interest, so want the sequence to be long enough to form some sort of folded domain or structure or functional domain. Also small sequences are less likely to be of significant interest, as the sequence length increases, it is more likely that the homology of the amino acid sequence will correlate with homologous structure. So next will subset by fragments longer than a particular length >100  
  
UniqueBlasts <- UniqueBlasts[(UniqueBlasts$alignmentlength > 100), ] #subset UniqueBlasts, all columns, and only rows where the alignment length >100  
  
UniqueBlasts

## queryid subjectids identity positives  
## 1 Query\_208689 gi|444302296|pdb|4I99|A 29.032 50.00  
## 5 Query\_208689 gi|299856727|pdb|3KTA|B 27.419 48.39  
## 7 Query\_208689 gi|58177331|pdb|1XEW|Y 27.419 48.39  
## 8 Query\_208689 gi|58177333|pdb|1XEX|B 26.613 48.39  
## 9 Query\_208689 gi|13096783|pdb|1E69|A 30.252 50.42  
## 15 Query\_208689 gi|1143283763|pdb|5MG8|A 21.154 42.31  
## 17 Query\_208689 gi|728055657|pdb|4UX3|A 22.907 43.17  
## 19 Query\_208689 gi|333944148|pdb|3AUX|A 26.257 44.69  
## 25 Query\_208689 gi|1002351595|pdb|5F3W|B 26.257 44.69  
## 27 Query\_208689 gi|1159374129|pdb|5H68|A 27.523 47.71  
## 55 Query\_208689 gi|444302106|pdb|3ZGX|A 23.179 46.36  
## 57 Query\_208689 gi|1159374124|pdb|5H66|B 22.517 46.36  
## 60 Query\_208689 gi|58177330|pdb|1XEW|X 24.242 44.24  
## 73 Query\_208689 gi|299856726|pdb|3KTA|A 24.638 44.20  
## alignmentlength mismatches gapopens q.start q.end s.start s.end  
## 1 124 82 4 930 1053 188 305  
## 5 124 84 3 930 1053 7 124  
## 7 124 84 3 930 1053 6 123  
## 8 124 85 3 930 1053 6 123  
## 9 119 70 4 937 1052 168 276  
## 15 208 150 6 398 592 20 226  
## 17 227 152 6 53 259 4 227  
## 19 179 118 5 56 226 8 180  
## 25 179 118 5 56 226 8 180  
## 27 109 72 3 924 1032 182 283  
## 55 151 99 5 914 1053 244 388  
## 57 151 100 5 914 1053 6 150  
## 60 165 100 6 47 194 2 158  
## 73 138 81 5 47 167 2 133  
## evalue bitscore mlog.evalue pdb.id acc  
## 1 6.55e-07 53.5 14.238631 4I99\_A 444302296  
## 5 1.39e-06 50.4 13.486207 3KTA\_B 299856727  
## 7 1.44e-06 50.1 13.450867 1XEW\_Y 58177331  
## 8 3.39e-06 49.3 12.594681 1XEX\_B 58177333  
## 9 5.48e-05 47.4 9.811820 1E69\_A 13096783  
## 15 6.18e-05 47.4 9.691607 5MG8\_A 1143283763  
## 17 2.14e-04 45.8 8.449535 4UX3\_A 728055657  
## 19 7.81e-04 43.9 7.154935 3AUX\_A 333944148  
## 25 8.53e-04 43.5 7.066751 5F3W\_B 1002351595  
## 27 1.00e-03 43.1 6.907755 5H68\_A 1159374129  
## 55 2.00e-02 39.3 3.912023 3ZGX\_A 444302106  
## 57 2.20e-02 38.1 3.816713 5H66\_B 1159374124  
## 60 5.50e-02 37.0 2.900422 1XEW\_X 58177330  
## 73 3.10e-01 34.7 1.171183 3KTA\_A 299856726

#Ordering the results by -logE (Significance)  
  
UniqueBlasts <- UniqueBlasts[order(UniqueBlasts$mlog.evalue)] #order the remaining hits by the significance  
  
  
#Saving the top 10 signifcant blast results  
  
UniqueBlasts <-UniqueBlasts[1:10, ] #subset, all columns, rows 1 to 10. As ordered by significance already, these will be the 10 most significant hits, matching the other subsetting criteria.  
  
UniqueBlasts

## mlog.evalue bitscore evalue s.end s.start q.end q.start gapopens  
## 1 14.238631 53.5 6.55e-07 305 188 1053 930 4  
## 5 13.486207 50.4 1.39e-06 124 7 1053 930 3  
## 7 13.450867 50.1 1.44e-06 123 6 1053 930 3  
## 8 12.594681 49.3 3.39e-06 123 6 1053 930 3  
## 9 9.811820 47.4 5.48e-05 276 168 1052 937 4  
## 15 9.691607 47.4 6.18e-05 226 20 592 398 6  
## 17 8.449535 45.8 2.14e-04 227 4 259 53 6  
## 19 7.154935 43.9 7.81e-04 180 8 226 56 5  
## 25 7.066751 43.5 8.53e-04 180 8 226 56 5  
## 27 6.907755 43.1 1.00e-03 283 182 1032 924 3  
## mismatches alignmentlength positives identity subjectids  
## 1 82 124 50.00 29.032 gi|444302296|pdb|4I99|A  
## 5 84 124 48.39 27.419 gi|299856727|pdb|3KTA|B  
## 7 84 124 48.39 27.419 gi|58177331|pdb|1XEW|Y  
## 8 85 124 48.39 26.613 gi|58177333|pdb|1XEX|B  
## 9 70 119 50.42 30.252 gi|13096783|pdb|1E69|A  
## 15 150 208 42.31 21.154 gi|1143283763|pdb|5MG8|A  
## 17 152 227 43.17 22.907 gi|728055657|pdb|4UX3|A  
## 19 118 179 44.69 26.257 gi|333944148|pdb|3AUX|A  
## 25 118 179 44.69 26.257 gi|1002351595|pdb|5F3W|B  
## 27 72 109 47.71 27.523 gi|1159374129|pdb|5H68|A  
## queryid  
## 1 Query\_208689  
## 5 Query\_208689  
## 7 Query\_208689  
## 8 Query\_208689  
## 9 Query\_208689  
## 15 Query\_208689  
## 17 Query\_208689  
## 19 Query\_208689  
## 25 Query\_208689  
## 27 Query\_208689

############################################################################################################################  
########################################### Plotting the BLAST results #####################################################  
############################################################################################################################  
  
plot2 <- ggplot(data=UniqueBlasts, aes(x=UniqueBlasts$subjectids, ymin= UniqueBlasts$q.start, ymax=UniqueBlasts$q.end, colour=identity))  
  
plotBlasts <- plot2 +  
 geom\_linerange(size=10)+  
 coord\_flip() +  
 labs(y="Amino acid no.", x="Subject ID", title= "Sequence alignment with PDB")  
plotBlasts



The plotBlasts returns the regions of homology of the top ten significant hits from the BLAST against the PDB along the amino acid sequence of our protein of interest. In a similar way to the plotCF this data can assist the studying of the correlation between protein structure and biological activity (by comparing our protein of interest with the biological activity of the PDB hits) and also be an aid in deciding boundaries to be used for protein trucations for structural biology applications, such as x-ray crystallography. The regions of homology represent locations on our protein which show homology with proteins that have already been structurally characterised. Therefore we can determine regions on our protein which are more likely to be characterisable by structural methods, and also have an idea of the conditions in which our regions of homology can be crystallised (if the hit is an x-ray crystallography structure). This allows us to speed up the protein purification/characterisation pipeline by assuming our regions of homology will act in a similar way to previous successful structural characterisation.

# 6. References

Chou P.Y. & Fasman G.D. (1974). Prediction of Protein conformation. *Biochemistry*. 13 (2), 222-245.

Chen H, Gu F. & Huang, Z. (2006). Improved Chou-Fasman method for protein secondary structure prediction. *BMC Bioinformatics*. 7 (S4).

Alberts B, Johnson A, Lewis J, et al (2002).Analyzing Protein Structure and Function. 4th edition. New York: Garland Science.*Molecular Biology of the Cell*.

Kawashima, S. and Kanehisa, M. (2000) AAindex: amino acid index database. *Nucleic Acids Res.*, 28:374.

Tomii, K. and Kanehisa, M. (1996) Analysis of amino acid indices and mutation matrices for sequence comparison and structure prediction of proteins. *Protein Eng.*, 9:27-36.

Nakai, K., Kidera, A., and Kanehisa, M. (1988) Cluster analysis of amino acid indices for prediction of protein structure and function. *Protein Eng.* 2:93-100.

"PDB Statistics". RCSB Protein Data Bank. Retrieved 2010-02-09.

Scapin G. (2006). Structural biology and drug discovery. *Curr. Pharm.* Des. 12 (17): 2087-97.

Altschul S.F., Madden T.L., Schäffer A.A., Zhang J., Zhang Z., Miller W. & Lipman D.J. (1997). Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17), 3389-3402.

Drenth J.(2007) *Principles of Protein-Xray Crystallography* 3rd ed.

sessionInfo()

## R version 3.3.3 (2017-03-06)  
## Platform: x86\_64-w64-mingw32/x64 (64-bit)  
## Running under: Windows 10 x64 (build 14393)  
##   
## locale:  
## [1] LC\_COLLATE=English\_United Kingdom.1252   
## [2] LC\_CTYPE=English\_United Kingdom.1252   
## [3] LC\_MONETARY=English\_United Kingdom.1252  
## [4] LC\_NUMERIC=C   
## [5] LC\_TIME=English\_United Kingdom.1252   
##   
## attached base packages:  
## [1] stats graphics grDevices utils datasets methods base   
##   
## other attached packages:  
## [1] bio3d\_2.3-1 seqinr\_3.3-6 ggplot2\_2.2.1  
##   
## loaded via a namespace (and not attached):  
## [1] Rcpp\_0.12.10 knitr\_1.15.1 magrittr\_1.5 munsell\_0.4.3   
## [5] colorspace\_1.3-2 stringr\_1.2.0 plyr\_1.8.4 tools\_3.3.3   
## [9] parallel\_3.3.3 grid\_3.3.3 gtable\_0.2.0 htmltools\_0.3.5   
## [13] yaml\_2.1.14 ade4\_1.7-6 lazyeval\_0.2.0 rprojroot\_1.2   
## [17] digest\_0.6.12 assertthat\_0.1 tibble\_1.2 evaluate\_0.10   
## [21] rmarkdown\_1.4 labeling\_0.3 stringi\_1.1.3 scales\_0.4.1   
## [25] backports\_1.0.5