Rapport TP R

By: Aubin, Tengfei

1. Introduction

Autosomal dominant polycystic kidney disease (PKD) is a genetic disease characterized by the progressive development of fluid cysts in the kidneys. In this project, we will work with DNA sequences of patients suffering from PKD.

2. Reminder on R

Exercise 2.1

The function `compte` return the number of occurences of a `letter` in a `sequence`.

```
compte <- function(letter, seq){
  occur = seq == letter
  letter.occur <- c(letter, sum(occur))
  print(letter.occur) # to print letter before the number for a better view
  # return(sum(occur))
}
```

We first use this function to compte how many "a" there is in the given squence :

```
sequence <- c("a","a","t","g","a","g","c","t","a","g","c","t","g")
> compte("a", sequence)
[1] "a" "4"
```

In order to count the number of occurences of each letter in the sequence, we apply a `for` loop :

```
> for (elt in unique(sequence)){
+   compte(elt, sequence)
+ }
[1] "a" "4"
[1] "t" "3"
[1] "g" "4"
[1] "c" "2"
```

3. Dealing with sequences with seqinr

Exercise 3.1

Options explaination

- The `seqtype` option indicates the nature of the sequence (DNA or AA), by default `seqtype = "DNA"`.
- The `forceDNAtolower` option is a boolean that indicates wether the character of a `seqtype = "DNA"` sequence are lower case letters or not.

We can print the first 50 nucleotide of the sequence with the command `head` :

```
> head(sequences$seq0, n = 50)
 [1] "g" "c" "g" "c" "c" "g" "g" "g" "a" "a" "g" "a" "a" "a" "g" "g" "a" "a"
[19] "c" "a" "t" "g" "g" "c" "t" "c" "c" "t" "g" "a" "g" "g" "c" "g" "c" "a"
[37] "c" "a" "g" "c" "g" "c" "c" "g" "a" "g" "c" "g" "c" "g"
```

We can print the length of the sequence with the command `length` :

```
> length(sequences$seq0)
[1] 5080
```

We can print the number of occurences of the nucleotide in the sequence using the `table` command or a `for` loop like in Exercise 2.1 :

```
> table(sequences$seq0)
   a    c    g    t
1391 1079 1202 1408
```

Exercise 3.2

As we known from the molecular biology, the nucleotides G and C are always connected in the DNA sequence, so the numbers of G and C are equal. Therefore, when computing the GC-content, we can add up the proportions of G and C in one cDNA to get the proportion of GC-content for the complete DNA.

As we do below:

```
> proportions = table(sequences$seq0)/length(sequences$seq0)
> proportions[[2]] + proportions[[3]]
[1] 0.4490157
```

Exercise 3.3

By convention, we write DNA in the direction 5' → 3'.

- The 5' extremity is the last phosphate group linked to the 5' carbon of the pentose.
- The 3' extremity is the OH linked to the 3' carbon of the pentose.

The function `complementary` below returns the complementary sequence:

```
complementary <- function(s){
  com.seq <- chartr("A", "t", chartr("t", "a",
                          chartr("a", "A", chartr("C", "g",
                                             chartr("g", "c", chartr("c", "C", seq))))))
  return(com.seq)
}
```

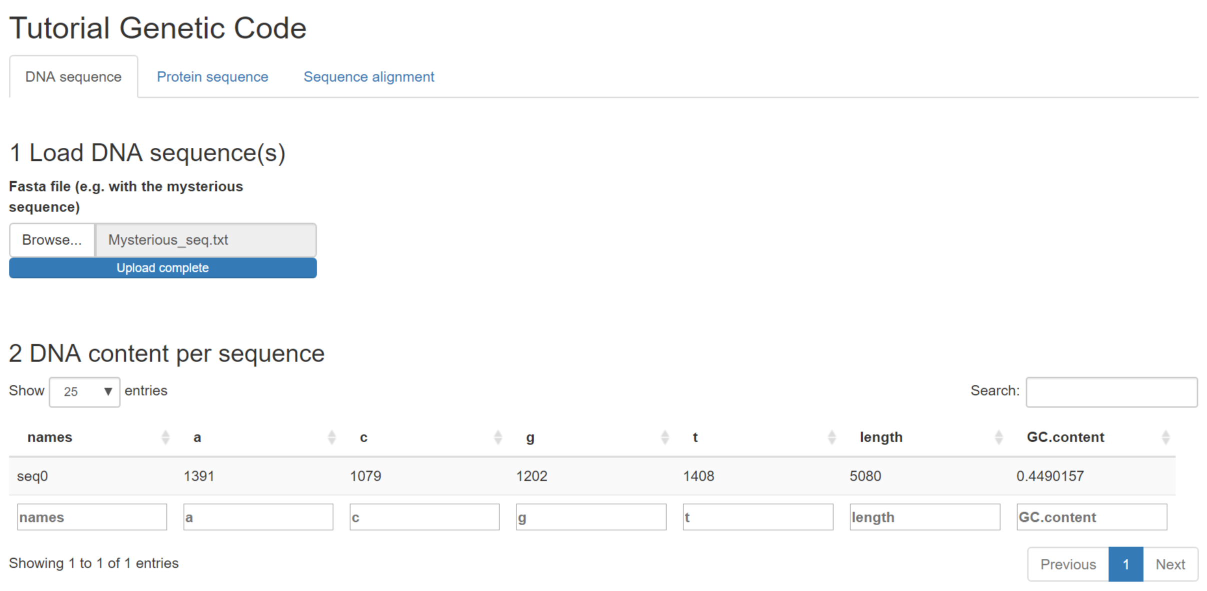
Then we can get the reverse of the complementary sequence in combining the command `rev` and `complementary`:

```
rev(complementary(sequences$seq0))
```

4. Introduction to Shiny

We add the function `compte` to the `util.R` file.

Firstly, we run the app with the `Mysterious_seq.txt` file:


Tutorial Genetic Code

1 Load DNA sequence(s)

Fasta file (e.g. with the mysterious sequence)

Browse... Mysterious_seq.txt

Upload complete

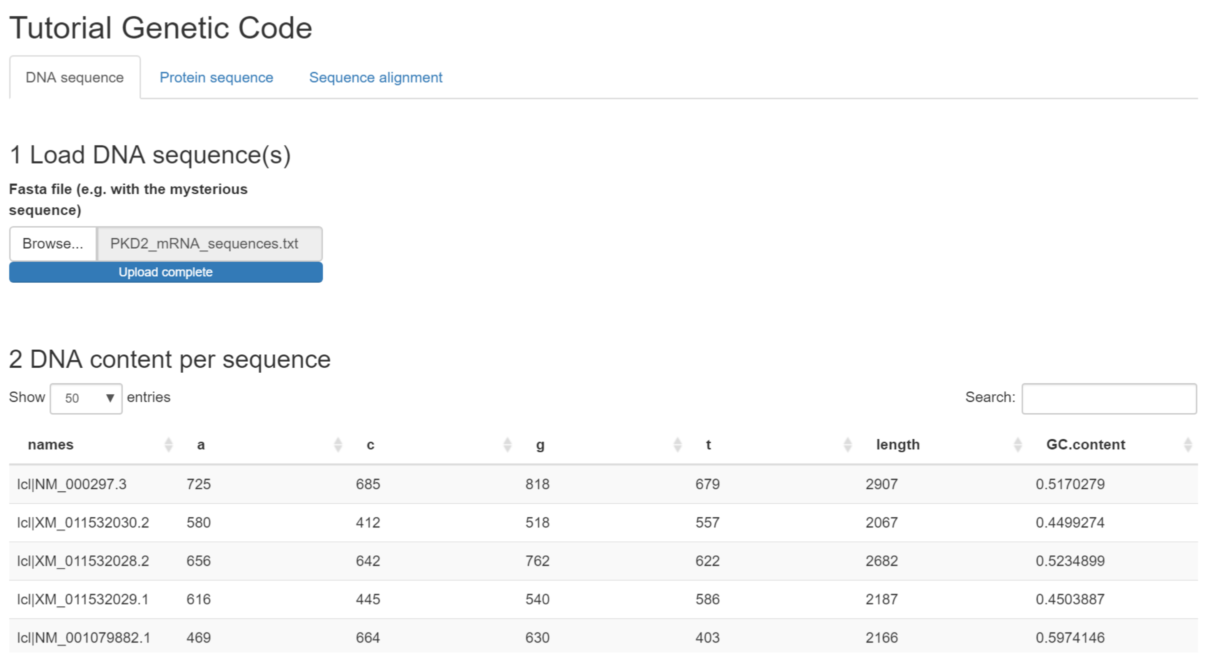
2 DNA content per sequence

Show 25 entries Search:

names	a	c	g	t	length	GC.content
seq0	1391	1079	1202	1408	5080	0.4490157

Showing 1 to 1 of 1 entries

Secondly, we run it with 31 sequences of the `PKD2_mRNA_sequences.txt` file :


Tutorial Genetic Code

1 Load DNA sequence(s)

Fasta file (e.g. with the mysterious sequence)

Browse... PKD2_mRNA_sequences.txt

Upload complete

2 DNA content per sequence

Show 50 entries Search:

names	a	c	g	t	length	GC.content
lcl|NM_000297.3	725	685	818	679	2907	0.5170279
lcl|XM_011532030.2	580	412	518	557	2067	0.4499274
lcl|XM_011532028.2	656	642	762	622	2682	0.5234899
lcl|XM_011532029.1	616	445	540	586	2187	0.4503887
lcl|NM_001079882.1	469	664	630	403	2166	0.5974146

5. Transcription

Transcription mechanism

Transcription is the process by which the information in a strand of DNA is copied into a new molecule of messenger RNA (mRNA) by the enzyme RNA polymerase.

It contains 3 main steps:

1. Initiation. The DNA molecule unwinds and separates to form a small open complex. RNA polymerase binds to the promoter of the template strand.
2. Elongation. RNA polymerase moves along the template strand, synthesising an mRNA molecule.
3. Termination. RNA polymerase crosses a stop (termination) sequence in the gene. The mRNA strand is complete, and it detaches from DNA.

4. Maturation. Introns are removed and the exons are spliced together to form a mature mRNA molecule consisting of a single protein-coding sequence.

cDNA and mRNA

The tested above mysterious sequence is a complementary DNA (cDNA) obtained from the messenger (mRNA) sequence.

The difference(s) between mRNA and cDNA:

- Composition: The the base uracil(U) in a RNA is replaced with base thymine(T) for the DNA
- Usage: different in cells(for we do the reverse transcription by man, we skip that difference)

The mRNA to cDNA reaction is catalyzed by an enzyme reverse transcriptase.

The differences between cDNA and the original DNA sequence

- cDNA has no introns. It does not contain any other original DNA that does not directly code for a protein (referred to as non coding DNA).
- Further more, not all genes in the original DNA are being transcribed into mRNA at any given time. As a result, cDNA will only contain genes that are actively being used by a specific cell or tissue at a point in time.

We can write a function `transcribe` that converts the mysterious cDNA sequence into RNA.

```r
transcribe <- function(cDNA){
  RNA <- chartr("A", "u", chartr("t", "a",
                                 chartr("a", "A", chartr("C", "g",
                                                         chartr("g", "c", chartr("c", "C", cDNA))))))
  return(RNA)
}
```

6. Translation

Exercise 6.1

- **Translation** is a process by which the genetic code contained within a messenger RNA (mRNA) molecule is decoded to produce a specific sequence of amino acids in a polypeptide chain.

  It has three stages:

  1. Initiation. The ribosome assembles around the target mRNA and the start codon 5' AUG is recognized.
  2. Elongation. The tRNA transfers an amino acid to the tRNA bound to the next codon, forming a peptide bond between the two amino acids. The ribosome then translocates to the next codon to continue the process, creating an amino acid chain.
  3. Termination. One of the three stop codons is recognised. The ribosome then folds the polypeptide into its final structure and release it into the cytoplasm.

- **The genetic code** is the set of rules used by living cells to translate information encoded within genetic material (DNA or mRNA sequences of nucleotide triplets, or codons) into proteins.

  It has some main characteristics as listed below:

  - Number of nucleotides used for decoding: 3 nucleotides for 1 amino acid
  - The ribosome moves forward from the start to the stop codon in exact steps of three bases at a time
  - Universal. All known living organisms use the same genetic code.
  - Unambiguous. Each codon codes for just one amino acid (or start or stop)
  - Redundant. Most amino acids are encoded by more than one codon.

- There are 64 amino acids in the standard genetic code. The genetic code defined as "redundant" because most amino acids are encoded by more than one codon.

A protein is delimited by a **Start and a Stop codon** located in the same "**reading frame**".

- A reading frame is a way of dividing the sequence of nucleotides in a nucleic acid (DNA or RNA) molecule into a set of consecutive, non-overlapping triplets--codons. There are six possible reading frames for one sequence--three in the forward direction and three in the reverse.

- The amino acid methionine (with the code AUG) acts as the START codon.
- There are 3 STOP codons in the genetic code - UAG, UAA, and UGA. The one-letter code "X" is used here to describe stop codons.

Exercise 6.2

Here is the function `dna2peptide` which return the peptidic sequence giving the reading frame and the sens :

```
dna2peptide <- function(sequence, frame = 0, sens = "F"){
  indices = seq(frame + 1, length(sequence), 3) # indexes for first nucleotides in codons
  if (sens == "R"){
    seq = rev(complementary(sequence))
  }
  codons = paste0(seq[indices], seq[indices+1], seq[indices+2]) # creat list of codons
  return(codon.table[codons, "letter"]) # translate codons to amino acids
}
```

Exercise 6.3

Here is the first version of `find.mysterious.protein` function (without the `p.length` argument) :

```
find.mysterious.proteins = function(sequence){
  out = list()
  for (f in 0:2){ # vary the frame
    for (s in c("F", "R")){ # vary the direction
      # gregexpr: find all the matches with a list
      temp = gregexpr('M[ACDEFGHIKLMNPQRSTVWY]{80,}(X|$)',
                      paste0(dna2peptide(sequence, f, s), collapse = ""),
                      extract = TRUE)
      if (temp != ""){ # if find, add to list
        for (elt in temp){
          out = append(out, elt)
        }
      }
    }
  }
  return(out)
}
```

Here is what returns the function `find.mysterious.proteins` when applied to the mysterious sequence :

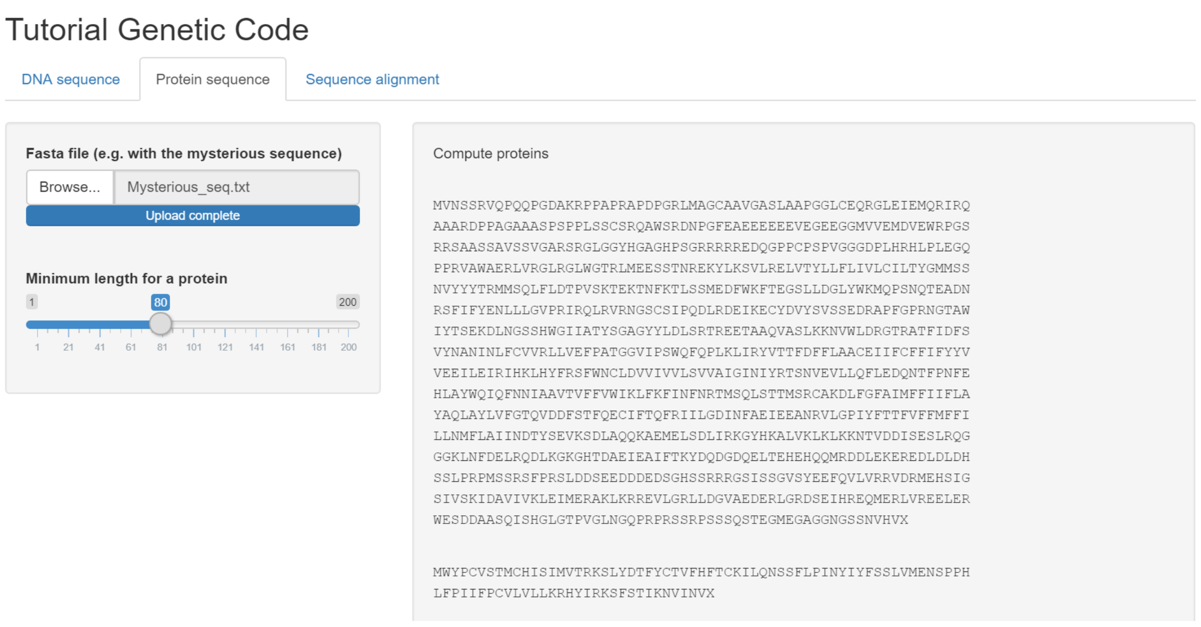
```
> find.mysterious.proteins(sequences$seq0)
[[1]]
[1] "MVNSSRVQPQQPGDAKRPPAPRAPDPGRLMAGCAAVGASLAAPGGLCEQRGLEIEMQRIRQAAARDPPAGAAASPSPPLSSCSRQAWSRDNPGFEAEEEEEEVEGEEGGMVVEMDVEWRPGS

[[2]]
[1] "MWYPCVSTMCHISIMVTRKSLYDTFYCTVFHFTCKILQNSSFLPINYIYFSSLVMENSPPHLFPIIFPCVLVLLKRHYIRKSFSTIKNVINVX"

[[3]]
[1] "MQTLVCSNKELCVAYSRESKDRDEAMRKNLQKPSMFHFQPGKVYLLMFSLEIGLTFYQLNEQLVKLKYSIKRQDSLGSPSEGYSTALHALLLISAQNVTLLNISVKEDFFFX"

[[4]]
[1] "MQRVPAADWAGARRALVLSPAPPPARVARAVVAPKPPAPRAHGAHGRGGGRPPAARAPLYVHLHHHSAFFPFHLLLLLLGLEAGVIAAPRLPGARRERRRRGGRGSGRGVPRGRLPDALHLD

[[5]]
[1] "MHKTMRKRRYVTSSRKTLLRYFSRLVLLSSMSLVPQRPRSPRTSLSAQATRGGCPSRGRWRCSGSPPPTGLGHGGPWSSRRRRLPLGWPAPWX"
```

We save the resulting sequences in `mysterious_proteins.txt` file.

```
write.table(mysterious.proteins,
            file = "./TP-Student/Students/mysterious_proteins.txt",
            quote = FALSE)
```

Exercise 6.4

We fill the declaration of output$aa.sequences in `server.R` so that if we load a sequence file via shiny, the protein sequences appear in the main panel as below.

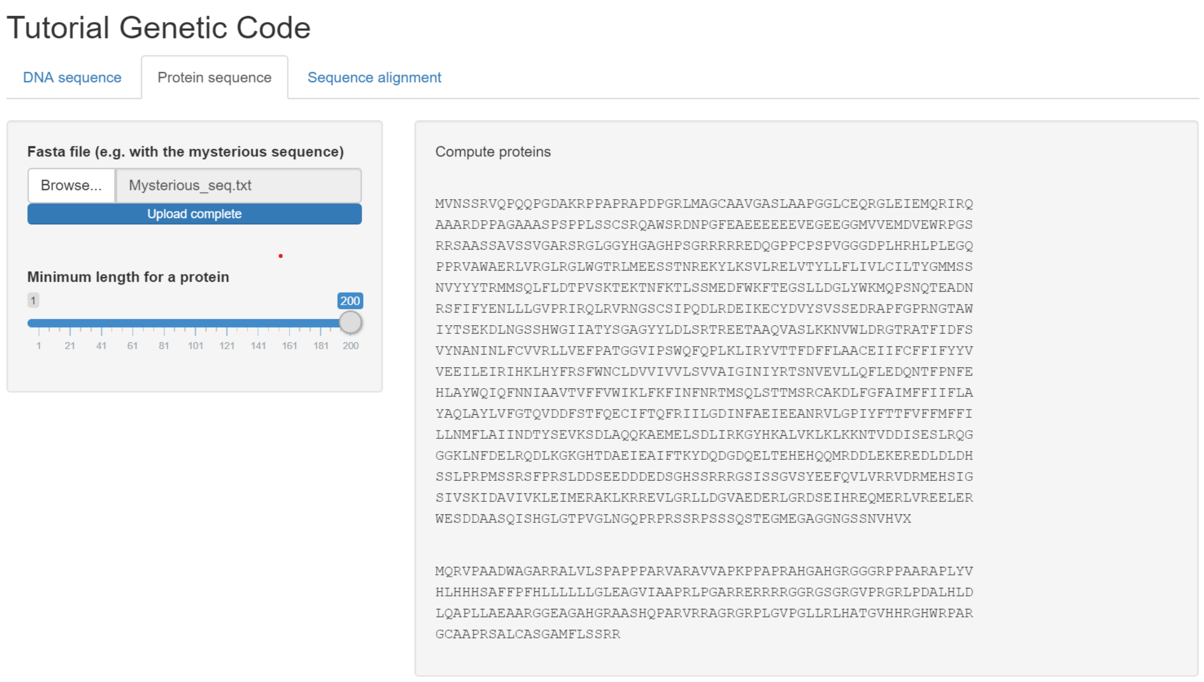


Fasta file (e.g. with the mysterious sequence): Mysterious_seq.txt

Minimum length for a protein: 80

Compute proteins

```
MVNSSRVQPQQPGDAKRPPAPRAPDPGRLMAGCAAVGASLAAPGGLCEQRGLEIEMQRIRQ
AAARDPPAGAAASPSPPLSSCSRQAWSRDNPGFEAEEEEEEVEGEEGGMVVEMDVEWRPGS
RRSAASSAVSSVGARSRGLGGYHGAGHPSGRRRRREDQGPPCPSPVGGGDPLHRHLPLEGQ
PPRVAWAERLVRGLRGLWGTRLMEESSTNREKYLKSVLRELVTYLLFLIVLCILTYGMMSS
NVYYYTRMMSQLFLDTPVSKTEKTNFKTLSSMEDFWKFTEGSLLDGLYWKMQPSNQTEADN
RSFIFYENLLLGVPRIRQLRVRNGSCSIPQDLRDEIKECYDVYSVSSEDRAPFGPRNGTAW
IYTSEKDLNGSSHWGIIATYSGAGYYLDLSRTREETAAQVASLKKNVWLDRGTRATFIDFS
VYNANINLFCVVRLLVEFPATGGVIPSWQFQPLKLIRYVTTFDFFLAACEIIFCFFIFYYV
VEEILEIRIHKLHYFRSFWNCLDVVIVVLSVVAIGINIYRTSNVEVLLQFLEDQNTFPNFE
HLAYWQIQFNNIAAVTVFFVWIKLFKFINFNRTMSQLSTTMSRCAKDLFGFAIMFFIIFLA
YAQLAYLVFGTQVDDFSTFQECIFTQFRIILGDINFAEIEEANRVLGPIYFTTFVFFMFFI
LLNMFLAIINDTYSEVKSDLAQQKAEMELSDLIRKGYHKALVKLKLKKNTVDDISESLRQG
GGKLNFDELRQDLKGKGHTDAEIEAIFTKYDQDGDQELTEHEHQQMRDDLEKEREDLDLDH
SSLPRPMSSRSFPRSLDDSEEDDDEDSGHSSRRRGSISSGVSYEEFQVLVRRVDRMEHSIG
SIVSKIDAVIVKLEIMERAKLKRREVLGRLLDGVAEDERLGRDSEIHREQMERLVREELER
WESDDAASQISHGLGTPVGLNGQPRPRSSRPSSSQSTEGMEGAGGNGSSNVHVX


MWYPCVSTMCHISIMVTRKSLYDTFYCTVFHFTCKILQNSSFLPINYIYFSSLVMENSPPH
LFPIIFPCVLVLLKRHYIRKSFSTIKNVINVX
```



Fasta file (e.g. with the mysterious sequence): Mysterious_seq.txt

Minimum length for a protein: 200

Compute proteins

```
MVNSSRVQPQQPGDAKRPPAPRAPDPGRLMAGCAAVGASLAAPGGLCEQRGLEIEMQRIRQ
AAARDPPAGAAASPSPPLSSCSRQAWSRDNPGFEAEEEEEEVEGEEGGMVVEMDVEWRPGS
RRSAASSAVSSVGARSRGLGGYHGAGHPSGRRRRREDQGPPCPSPVGGGDPLHRHLPLEGQ
PPRVAWAERLVRGLRGLWGTRLMEESSTNREKYLKSVLRELVTYLLFLIVLCILTYGMMSS
NVYYYTRMMSQLFLDTPVSKTEKTNFKTLSSMEDFWKFTEGSLLDGLYWKMQPSNQTEADN
RSFIFYENLLLGVPRIRQLRVRNGSCSIPQDLRDEIKECYDVYSVSSEDRAPFGPRNGTAW
IYTSEKDLNGSSHWGIIATYSGAGYYLDLSRTREETAAQVASLKKNVWLDRGTRATFIDFS
VYNANINLFCVVRLLVEFPATGGVIPSWQFQPLKLIRYVTTFDFFLAACEIIFCFFIFYYV
VEEILEIRIHKLHYFRSFWNCLDVVIVVLSVVAIGINIYRTSNVEVLLQFLEDQNTFPNFE
HLAYWQIQFNNIAAVTVFFVWIKLFKFINFNRTMSQLSTTMSRCAKDLFGFAIMFFIIFLA
YAQLAYLVFGTQVDDFSTFQECIFTQFRIILGDINFAEIEEANRVLGPIYFTTFVFFMFFI
LLNMFLAIINDTYSEVKSDLAQQKAEMELSDLIRKGYHKALVKLKLKKNTVDDISESLRQG
GGKLNFDELRQDLKGKGHTDAEIEAIFTKYDQDGDQELTEHEHQQMRDDLEKEREDLDLDH
SSLPRPMSSRSFPRSLDDSEEDDDEDSGHSSRRRGSISSGVSYEEFQVLVRRVDRMEHSIG
SIVSKIDAVIVKLEIMERAKLKRREVLGRLLDGVAEDERLGRDSEIHREQMERLVREELER
WESDDAASQISHGLGTPVGLNGQPRPRSSRPSSSQSTEGMEGAGGNGSSNVHVX


MQRVPAADWAGARRALVLSPAPPPARVARAVVAPKPPAPRAHGAHGRGGGRPPAARAPLYV
HLHHHSAFFPFHLLLLLLGLEAGVIAAPRLPGARRERRRRGGRGSGRGVPRGRLPDALHLD
LQAPLLAEAARGGEAGAHGRAASHQPARVRRAGRGRPLGVPGLLRLHATGVHHRGHWRPAR
GCAAPRSALCASGAMFLSSRR
```

7. Alignment

Our goals in this part are:

1. to characterize the mysterious sequence (PKD1 vs PKD2)
2. to identify the gene corresponding to your sequence (PKD1 vs PKD2) and locate your patient mutation
3. to compare your sequence (for our is `seq8`) with your classmates' sequence.

We use the Needleman Wunsch algorithm provided by the `pairwiseAlignment` function from the `Biostrings` package.

Exercise 7.1

Pros and cons of aligning amino acids instead of nucleotides

Alignment amino acids is almost always preferable to alignment nucleotides:

1. Redundant codons mean about1/3 of DNA mutations often don't matter.

2. A smaller sized alphabet requires more matches. DNA sequences are made with an alphabet of four nucleotides while most proteins have twenty. It is easier to achieve statistically significant alignment by comparing a larger alphabet of characters because you are much less likely to get a match by chance.
3. The DNA database is cluttered with non-coding sequences.

But all depends on what you want to find. For example, if you have a sequence read from a patient and you want to align that sequence read with the reference genome to see if there are any SNPs or mutations - then a protein alignment is overkill and it will be better to use a nucleic acid

Alignement against the two reference proteins

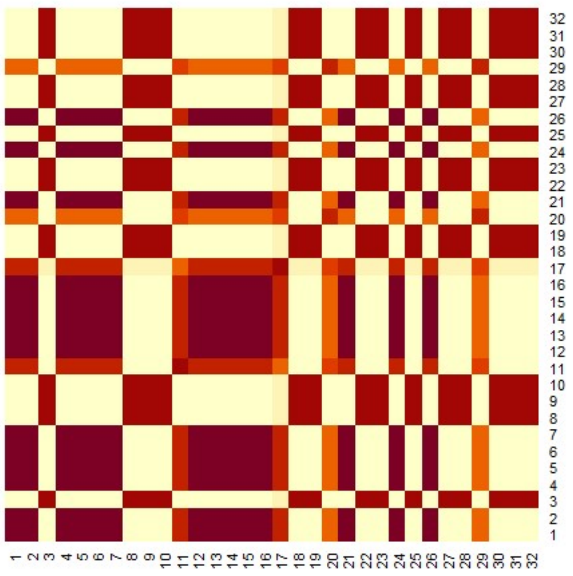
Back to the working R script, we align the longest mysterious translated protein from the previous part against the two reference proteins by `pairwiseAlignment`. The mysterious sequence is PKD2 protein with a identity up to 99%.

Then we align `sequence 8` against the two reference proteins and identified that PKD2 proteins was assigned to us with a identity up to 50.5% compared to 3.5% with PKD1. The mutation is F394I.

Exercise 7.2

In Computing the pairwise alignment for each of the 32 AA sequences against all the others in `Protein_sequences.txt`, we can build a 32 by 32 scores matrix `scores`.

Then we use `heatmap` to plot the clustered scores matrix as below.

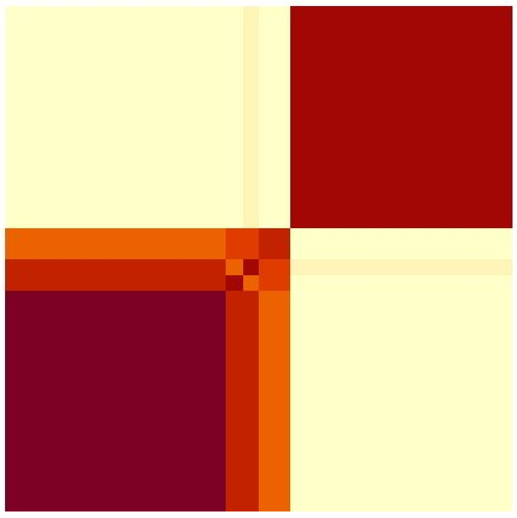


Alignment scores heatmap for 32 protein sequences

Comments:

It represents the zones of alignment with the identity more or less important according to the colours from yellow to dark red.

It has a symetry compared to the second diagonal.



By interchanging columns and lines we can see the two clusters distinctly (numbers are not correct on the second heat map so we delect it).

One composed of the sequences 1, 2, 4, 5, 6, 7, 11, 12, 13, 14, 15, 16, 17, 20, 21, 24, 26 and 29 and the other one of the rest. On the second heat map we can also see two other smaller clusters, which matches to sequences 11, 17 and 20, 29.

Application with Shiny: refine sequence alignment

Exercise 7.3

We use substitution matrices to deal with amino acids changes instead of constant penalties, because this allows us to maximize the score according to our target and its properties of the amino acids.

Exercise 7.4

We fill the `output$align.out` part of the `server.R` code.

The effects of every parameter:

- Select reference sequence: change PKD1 or PKD2
- Alignment type: change the global alignement or local alignement
- Substitution matrix: change the type of substution matrix for alignement
- Gap opening cost and Gap extension cost: to change the scores cost for GO and GE