732A51 Bioinformatics

Lab 1

Johannes Hedström

STIMA
Institutionen för datavetenskap
Linköpings universitet

2024-11-15

Contents

1 Hardy–Weinberg equilibrium

We consider a gene locus with two possible alleles (say A and a) and a diploid population with N individuals. Hence, there are 2N alleles in the population. Let p be the proportion of As in the allele population and q the population of as (of course p + q = 1). A population is said to be in Hardy–Weinberg equilibrium if the proportion of AA homozygotes is p 2, aa homozygotes is q2 and the proportion of heterozygotes (Aa) is 2pq.

1.1 Question 1.1

Show that with random mating (i.e. both alleles of the offspring are just randomly, with proportions p and q, drawn from the parental allele population) Hardy–Weinberg equilibrium is attained in the first generation. What is the proportion of A and a alleles in the offspring population? Hence, with random mating, can a population in Hardy–Weinberg equilibrium ever deviate from it?

$$P(AA) = p^2$$

$$P(Aa) = 2pq$$

$$P(aa) = q^2$$

When its random mating the proportions will always stay the same for the next generation.

1.2 Question 1.2

We look at the MN blood group (https://en.wikipedia.org/wiki/MNS_antigen_system), it has two possible co–dominating (both contribute to heterozygotes) alleles L M (denoted M) and L N (denoted N). In a population of 1000 Americans of Caucasian descent the following genotype counts were observed, 357 individuals were MM, 485 were MN and 158 were NN. Use a chi– square goodness of fit test to test if the population is in Hardy–Weinberg equilibrium.

chi– square goodness of fit test:

$$\chi^2 = \sum \frac{(O_i - E_i)^2}{E_i}$$

$$df = 3 - 1$$

$$p - value = P(\chi^2 > Observed\chi^2)$$

```r
# data
MM <- 357
NM <- 485
NN <- 158

# all M's

p <- (2*MM+NM)/(2*1000)
q <- 1-p

# expected values
exp_mm <-p*p*1000
exp_nm <-2*p*q*1000
exp_nn <- q*q*1000

# built in function
# chisq.test(c(MM,NM,NN), p=c(p*p,2*p*q,q*q))
act <- c(MM,NM,NN)
expected <- c(exp_mm,exp_nm,exp_nn)

# Calculate Chi-Square statistic manually
chi_sq_statistic <- sum((act-expected)^2 / expected)

df <- 3-1
p_value <- 1 - pchisq(chi_sq_statistic, df)

p_value
```

```
## [1] 0.951259
```

The p-value is very high, which indicates that there is no significant statistaical difference between the groups, we cant reject that the population is in Hardy–Weinberg equilibrium.

2 Question 2: Exploring a genomic sequence

In this exercise, you will use GenBank (https://www.ncbi.nlm.nih.gov/genbank/) with the default "Nucleotide" database. We will be interested in the sequence with accession number in GenBank: MK465080. You will find the relevant information in the FEATURES section of the returned record and access the nucleotides of the sequence under CDS (protein coding sequence, from CoDing Sequence). Remember that the coding strand (https://en.wikipedia.org/wiki/Coding_strand) is the strand of the gene that is identical to the transcript (see the lecture slides for the genetic code—translation of DNA triples to amino acids). The complimentary to it strand is called the template strand.

2.1 Question 2.1

From what species does the sequence come from? Name the protein product of the CDS.

The protein product of the CDS is "cytochrome c oxidase subunit I"(COI) and the species is branchipus schaefferi, which seem to be a kind of crustacean.

2.2 Question 2.2

```
# importing fasta file
library(seqinr)
branchipus <- read.fasta('branchipus_schaefferi.fasta')

sequance <- branchipus$MK465080.1
print(branchipus)
```

```
## $MK465080.1
##   [1] "t" "c" "t" "c" "c" "t" "a" "g" "g" "a" "g" "a" "t" "g" "a" "c" "c" "a"
##  [19] "a" "c" "t" "t" "t" "a" "t" "a" "a" "c" "g" "t" "c" "a" "t" "t" "g" "t"
##  [37] "t" "a" "c" "t" "g" "c" "t" "c" "a" "c" "g" "c" "t" "t" "t" "t" "g" "t"
##  [55] "a" "a" "t" "g" "a" "t" "t" "t" "t" "c" "t" "t" "c" "a" "t" "a" "g" "t"
##  [73] "t" "a" "t" "a" "c" "c" "a" "a" "t" "c" "c" "t" "t" "a" "t" "t" "g" "g"
##  [91] "a" "g" "g" "a" "t" "t" "t" "g" "g" "a" "a" "a" "t" "t" "g" "a" "t" "t"
## [109] "a" "g" "t" "c" "c" "c" "t" "t" "t" "a" "a" "t" "a" "c" "t" "a" "g" "g"
## [127] "g" "g" "c" "t" "c" "c" "t" "g" "a" "t" "a" "t" "g" "g" "c" "t" "t" "t"
## [145] "c" "c" "c" "c" "c" "g" "a" "c" "t" "a" "a" "a" "t" "a" "a" "c" "t" "t"
## [163] "a" "a" "g" "c" "t" "t" "t" "t" "g" "a" "a" "t" "a" "c" "t" "t" "c" "c"
## [181] "t" "c" "c" "c" "t" "c" "a" "t" "t" "a" "a" "c" "t" "c" "t" "t" "c" "t"
## [199] "a" "g" "t" "g" "g" "c" "c" "a" "g" "c" "t" "c" "a" "a" "t" "g" "g" "t"
## [217] "a" "g" "a" "a" "a" "g" "a" "g" "g" "g" "g" "t" "a" "g" "g" "a" "a" "c"
## [235] "a" "g" "g" "a" "t" "g" "a" "a" "c" "a" "g" "t" "t" "t" "a" "t" "c" "c"
## [253] "a" "c" "c" "c" "c" "t" "a" "t" "c" "t" "g" "c" "t" "g" "c" "t" "a" "t"
## [271] "t" "g" "c" "c" "c" "a" "t" "g" "c" "t" "g" "g" "t" "c" "c" "t" "t" "c"
## [289] "t" "g" "t" "t" "g" "a" "t" "t" "t" "a" "g" "c" "a" "a" "t" "c" "t" "t"
## [307] "t" "t" "c" "a" "c" "t" "t" "c" "a" "c" "c" "t" "c" "g" "c" "a" "g" "g"
## [325] "g" "a" "t" "c" "t" "c" "t" "t" "c" "a" "a" "t" "t" "t" "t" "a" "g" "g"
```

```
## [343] "a" "g" "c" "t" "g" "t" "a" "a" "a" "t" "t" "t" "c" "a" "t" "t" "a" "c"
## [361] "a" "a" "c" "t" "g" "t" "a" "a" "t" "t" "a" "a" "t" "a" "t" "a" "c" "g"
## [379] "g" "c" "c" "t" "c" "a" "t" "t" "c" "c" "a" "t" "a" "a" "g" "a" "t" "t"
## [397] "a" "g" "a" "c" "c" "g" "a" "a" "t" "a" "c" "c" "t" "t" "t" "a" "t" "t"
## [415] "t" "g" "c" "a" "t" "g" "a" "g" "c" "g" "g" "t" "a" "g" "t" "t" "a" "t"
## [433] "c" "a" "c" "a" "g" "c" "a" "g" "t" "t" "c" "t" "t" "c" "t" "t" "c" "t"
## [451] "c" "c" "t" "t" "t" "c" "t" "c" "t" "c" "c" "c" "a" "g" "t" "a" "t" "t"
## [469] "a" "g" "c" "a" "g"
## attr(,"name")
## [1] "MK465080.1"
## attr(,"Annot")
## [1] ">MK465080.1 Branchipus schaefferi isolate ST2-2 cytochrome c oxidase subunit I (COI) gene, parti
## attr(,"class")
## [1] "SeqFastadna"
```

Save (and submit) the nucleotide sequence of the coding strand that corresponds to these amino acids as a FASTA format file.. Use transeq (https://www.ebi.ac.uk/Tools/st/emboss_transeq/) to translate the nucleotides to a protein. Do you obtain the same protein sequence? Check what is the ORF and codon table (these are provided by GenBank in the FEATURES section). Use backtranseq (https://www.ebi.ac.uk/Tools/st/emboss_backtranseq/) to obtain the sequence from the protein sequence.

Using frame 2 and codon table 5 'Invertebrate Mitochondrial'.

```
library(stringr)
branchipus_2 <- read.fasta('emboss_transeq-I20241115-093926-0606-64392434-p1m.fasta')

print(branchipus_2)
```

```
## $MK465080.1_2
##   [1] "l" "l" "g" "d" "d" "q" "l" "y" "n" "v" "i" "v" "t" "a" "h" "a" "f" "v"
##  [19] "m" "i" "f" "f" "m" "v" "m" "p" "i" "l" "i" "g" "g" "f" "g" "n" "w" "l"
##  [37] "v" "p" "l" "m" "l" "g" "a" "p" "d" "m" "a" "f" "p" "r" "l" "n" "n" "l"
##  [55] "s" "f" "w" "m" "l" "p" "p" "s" "l" "t" "l" "l" "v" "a" "s" "s" "m" "v"
##  [73] "e" "s" "g" "v" "g" "t" "g" "w" "t" "v" "y" "p" "p" "l" "s" "a" "a" "i"
##  [91] "a" "h" "a" "g" "p" "s" "v" "d" "l" "a" "i" "f" "s" "l" "h" "l" "a" "g"
## [109] "i" "s" "s" "i" "l" "g" "a" "v" "n" "f" "i" "t" "t" "v" "i" "n" "m" "r"
## [127] "p" "h" "s" "m" "s" "l" "d" "r" "m" "p" "l" "f" "a" "w" "a" "v" "v" "i"
## [145] "t" "a" "v" "l" "l" "l" "l" "s" "l" "p" "v" "l" "a" "x"
## attr(,"name")
## [1] "MK465080.1_2"
## attr(,"Annot")
## [1] ">MK465080.1_2 Branchipus schaefferi isolate ST2-2 cytochrome c oxidase subunit I (COI) gene, par
## attr(,"class")
## [1] "SeqFastadna"
```

```
seqq <- branchipus_2$MK465080.1_2

true_seq <- str_split(tolower("LLGDDQLYNVIVTAHAFVMIFFMVMPILIGGFGNWLVPLMLGAPDMAFPRLNNLSFWMLPPSLTLLVASSMVE

print(true_seq==seqq)
```

```
##   [1]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [13]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [25]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [37]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [49]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [61]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [73]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [85]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
##  [97]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [109]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [121]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [133]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [145]  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE  TRUE
## [157]  TRUE FALSE
```

We get the same protein sequence as this is the sequence from https://www.ncbi.nlm.nih.gov/nuccore/MK465080 .

“LLGDDQLYNVIVTAHAFVMIFFMVMPILIGGFGNWLVPLMLGAPDMAFPRLNNLSFWMLPPSLTLLVASSMVESGVGT

Then in the end of the sequence we got there is an X which i guess is a stop codon.

To use backtranseq we could’nt find the correct codon table so we use ‘Homo’ Sapiens’ and got the following results.

```
branchipus_3 <- read.fasta('emboss_backtranseq-I20241115-101532-0798-77094578-p1m.fasta')


print(branchipus_3)
```

```
## $MK465080.1_2
##   [1] "c" "t" "g" "c" "t" "g" "g" "g" "c" "g" "a" "c" "g" "a" "c" "c" "a" "g"
##  [19] "c" "t" "g" "t" "a" "c" "a" "a" "c" "g" "t" "g" "a" "t" "c" "g" "t" "g"
##  [37] "a" "c" "c" "g" "c" "c" "c" "a" "c" "g" "c" "c" "t" "t" "c" "g" "t" "g"
##  [55] "a" "t" "g" "a" "t" "c" "t" "t" "c" "t" "t" "c" "a" "t" "g" "g" "t" "g"
##  [73] "a" "t" "g" "c" "c" "c" "a" "t" "c" "c" "t" "g" "a" "t" "c" "g" "g" "c"
##  [91] "g" "g" "c" "t" "t" "c" "g" "g" "c" "a" "a" "c" "t" "g" "g" "c" "t" "g"
## [109] "g" "t" "g" "c" "c" "c" "c" "t" "g" "a" "t" "g" "c" "t" "g" "g" "g" "c"
## [127] "g" "c" "c" "c" "c" "c" "g" "a" "c" "a" "t" "g" "g" "c" "c" "t" "t" "c"
## [145] "c" "c" "c" "a" "g" "g" "c" "t" "g" "a" "a" "c" "a" "a" "c" "c" "t" "g"
## [163] "a" "g" "c" "t" "t" "c" "t" "g" "g" "a" "t" "g" "c" "t" "g" "c" "c" "c"
## [181] "c" "c" "c" "a" "g" "c" "c" "t" "g" "a" "c" "c" "c" "t" "g" "c" "t" "g"
## [199] "g" "t" "g" "g" "c" "c" "a" "g" "c" "a" "g" "c" "a" "t" "g" "g" "t" "g"
## [217] "g" "a" "g" "a" "g" "c" "g" "g" "c" "g" "t" "g" "g" "g" "c" "a" "c" "c"
## [235] "g" "g" "c" "t" "g" "g" "a" "c" "c" "g" "t" "g" "t" "a" "c" "c" "c" "c"
## [253] "c" "c" "c" "c" "t" "g" "a" "g" "c" "g" "c" "c" "g" "c" "c" "a" "t" "c"
## [271] "g" "c" "c" "c" "a" "c" "g" "c" "c" "g" "g" "c" "c" "c" "c" "a" "g" "c"
## [289] "g" "t" "g" "g" "a" "c" "c" "t" "g" "g" "c" "c" "a" "t" "c" "t" "t" "c"
## [307] "a" "g" "c" "c" "t" "g" "c" "a" "c" "c" "t" "g" "g" "c" "c" "g" "g" "c"
## [325] "a" "t" "c" "a" "g" "c" "a" "g" "c" "a" "t" "c" "c" "t" "g" "g" "g" "c"
```

```
## [343] "g" "c" "c" "g" "t" "g" "a" "a" "c" "t" "t" "c" "a" "t" "c" "a" "c" "c"
## [361] "a" "c" "c" "g" "t" "g" "a" "t" "c" "a" "a" "c" "a" "t" "g" "a" "g" "g"
## [379] "c" "c" "c" "c" "a" "c" "a" "g" "c" "a" "t" "g" "a" "g" "c" "c" "t" "g"
## [397] "g" "a" "c" "a" "g" "g" "a" "t" "g" "c" "c" "c" "c" "t" "g" "t" "t" "c"
## [415] "g" "c" "c" "t" "g" "g" "g" "c" "c" "g" "t" "g" "g" "t" "g" "a" "t" "c"
## [433] "a" "c" "c" "g" "c" "c" "g" "t" "g" "c" "t" "g" "c" "t" "g" "c" "t" "g"
## [451] "c" "t" "g" "a" "g" "c" "c" "t" "g" "c" "c" "c" "g" "t" "g" "c" "t" "g"
## [469] "g" "c" "c" "n" "n" "n"
## attr(,"name")
## [1] "MK465080.1_2"
## attr(,"Annot")
## [1] ">MK465080.1_2 Branchipus schaefferi isolate ST2-2 cytochrome c oxidase subunit I (COI) gene, par
## attr(,"class")
## [1] "SeqFastadna"
```

We get a different sequence compared to the one we saved from https://www.ncbi.nlm.nih.gov/nuccore/MK465080.1?report=fasta.

2.3 Question 2.3

Compare your obtained coding strand sequence with the nucleotide sequence provided (when following the CDS link). Are they the same or do they differ? Try reversing and taking the complement (e.g., http://arep.med.harvard.edu/labgc/adnan/projects/Utilities/revcomp.html or http://www.bioinformatics.nl/cgi-bin/emboss/revseq or write your own code) of the your coding strand DNA. Consider also backtranseqambiq (https://www.ebi.ac.uk/Tools/st/emboss_backtranseqambig/ and check if the resulting nucleotide sequence is compatible with the true one. Do not forget to check the codon table. Explain what happened and why. Save (and submit) the nucleotide sequence of the template strand that corresponds to these amino acids as a FASTA format file.

3 Här ska det in mer kod typ reversing och complement

Results from backtranseqambiq with codon table 5 'Invertebrate Mitochondrial'.

```
branchipus_3 <- read.fasta('emboss_backtranambig-I20241115-101314-0071-49255505-p1m.fasta')

print(branchipus_3)
```

```
## $MK465080.1_2
##   [1] "y" "t" "n" "y" "t" "n" "g" "g" "n" "g" "a" "y" "g" "a" "y" "c" "a" "r"
##  [19] "y" "t" "n" "t" "a" "y" "a" "a" "y" "g" "t" "n" "a" "t" "y" "g" "t" "n"
##  [37] "a" "c" "n" "g" "c" "n" "c" "a" "y" "g" "c" "n" "t" "t" "y" "g" "t" "n"
##  [55] "a" "t" "r" "a" "t" "y" "t" "t" "y" "t" "t" "y" "a" "t" "r" "g" "t" "n"
##  [73] "a" "t" "r" "c" "c" "n" "a" "t" "y" "y" "t" "n" "a" "t" "y" "g" "g" "n"
##  [91] "g" "g" "n" "t" "t" "y" "g" "g" "n" "a" "a" "y" "t" "g" "r" "y" "t" "n"
## [109] "g" "t" "n" "c" "c" "n" "y" "t" "n" "a" "t" "r" "y" "t" "n" "g" "g" "n"
## [127] "g" "c" "n" "c" "c" "n" "g" "a" "y" "a" "t" "r" "g" "c" "n" "t" "t" "y"
```

```
## [145] "c" "c" "n" "c" "g" "n" "y" "t" "n" "a" "a" "y" "a" "a" "y" "y" "t" "n"
## [163] "w" "s" "n" "t" "t" "y" "t" "g" "r" "a" "t" "r" "y" "t" "n" "c" "c" "n"
## [181] "c" "c" "n" "w" "s" "n" "y" "t" "n" "a" "c" "n" "y" "t" "n" "y" "t" "n"
## [199] "g" "t" "n" "g" "c" "n" "w" "s" "n" "w" "s" "n" "a" "t" "r" "g" "t" "n"
## [217] "g" "a" "r" "w" "s" "n" "g" "g" "n" "g" "t" "n" "g" "g" "n" "a" "c" "n"
## [235] "g" "g" "n" "t" "g" "r" "a" "c" "n" "g" "t" "n" "t" "a" "y" "c" "c" "n"
## [253] "c" "c" "n" "y" "t" "n" "w" "s" "n" "g" "c" "n" "g" "c" "n" "a" "t" "y"
## [271] "g" "c" "n" "c" "a" "y" "g" "c" "n" "g" "g" "n" "c" "c" "n" "w" "s" "n"
## [289] "g" "t" "n" "g" "a" "y" "y" "t" "n" "g" "c" "n" "a" "t" "y" "t" "t" "y"
## [307] "w" "s" "n" "y" "t" "n" "c" "a" "y" "y" "t" "n" "g" "c" "n" "g" "g" "n"
## [325] "a" "t" "y" "w" "s" "n" "w" "s" "n" "a" "t" "y" "y" "t" "n" "g" "g" "n"
## [343] "g" "c" "n" "g" "t" "n" "a" "a" "y" "t" "t" "y" "a" "t" "y" "a" "c" "n"
## [361] "a" "c" "n" "g" "t" "n" "a" "t" "y" "a" "a" "y" "a" "t" "r" "c" "g" "n"
## [379] "c" "c" "n" "c" "a" "y" "w" "s" "n" "a" "t" "r" "w" "s" "n" "y" "t" "n"
## [397] "g" "a" "y" "c" "g" "n" "a" "t" "r" "c" "c" "n" "y" "t" "n" "t" "t" "y"
## [415] "g" "c" "n" "t" "g" "r" "g" "c" "n" "g" "t" "n" "g" "t" "n" "a" "t" "y"
## [433] "a" "c" "n" "g" "c" "n" "g" "t" "n" "y" "t" "n" "y" "t" "n" "y" "t" "n"
## [451] "y" "t" "n" "w" "s" "n" "y" "t" "n" "c" "c" "n" "g" "t" "n" "y" "t" "n"
## [469] "g" "c" "n" "n" "n" "n"
## attr(,"name")
## [1] "MK465080.1_2"
## attr(,"Annot")
## [1] ">MK465080.1_2 Branchipus schaefferi isolate ST2-2 cytochrome c oxidase subunit I (COI) gene, par
## attr(,"class")
## [1] "SeqFastadna"
```

4 Question 3: Exploring a genomic sequence

Eukaryotic genes are commonly divided into exons and introns and these needed to be spliced in order to produce an mRNA that can be translated into a protein. The gene starts with the promoter ("region of DNA that initiates transcription, i.e. DNA→RNA, of a particular gene", see https://en.wikipedia.org/wiki/Promoter_(genetics)), the the first exon, first intron, second exon, e.t.c. Multiple introns an alternative splicing (a single gene codes for different proteins, through different exons used, see https://en.wikipedia.org/wiki/Alternative_splicing) of mRNAs can make it difficult to identify genes. Finding genes and how they are organized is often due to searching for similar nucleotide sequences within already know protein amino acids, or rather their nucleotide sequences and the corresponding full–length cDNAs (complementary DNA—DNA synthesized from a single stranded RNA, see https://en.wikipedia. org/wiki/Complementary_DNA). cDNAs come from back transcription (reverse–transcription) of mRNAs and hence are without introns—and can be considered as equivalent to mRNA sequences. Comparing a part of the genome (that contains introns) with its cDNA will show the introns' start and end points. GenBank (https://www.ncbi.nlm.nih.gov/genbank/) contains both cDNA sequences and corresponding genomic sequences (if available). If order to discover the structure of the gene we need to compare the cDNA with the genomic sequence. In the file 732A51 BioinformaticsHT2024 Lab01Ex03.fasta you can find a genomic sequence from the species C. elegans, that contains a particular gene. You will use the Basic Local Alignment Sequence Tool (BLAST), to find the gene's organization. BLAST is used to compare a query sequence all sequences (i.e., cDNA sequences) in GenBank. Usually the top scoring hit is the one you want. The next ones will be less and less similar. It can happen that all hits have 100% identity—then consider the percent coverage.

4.1 Question 3.1

Read up on C. elegans and in a few sentences describe it and why it is such an important organism for the scientific community.

4.2 Question 3.2

Use the nucleotide BLAST tool to construct a schematic diagram that shows the arrangement of introns and exons in the genomic sequence. In the BLAST tool, https://blast.ncbi.nlm. nih.gov/Blast.cgi?PAGE_TYPE=BlastSearch, choose database RefSeq Genome Database and remember that the species source of the genomic sequence is Caenorhabditis elegans. Use the Genome Data Viewer button. Alternatively you may use https://wormbase.org/tools/blast_ blat.

4.3 Question 3.3

How are the sequences numbered in the alignment (i.e., pairing of query and database sequences)? Are the directions the same? What would happen if you reverse complement your query sequence (e.g., http://arep.med.harvard.edu/labgc/adnan/projects/Utilities/revcomp.html or http://www.bioinformatics.nl/cgi-bin/emboss/revseq or write your own code) and do the search with such a reverse complemented sequence?

4.4 Question 3.4

On what chromosome and what position is the query sequence found? At which position does the gene begin and end in your query sequence?

4.5 Question 3.5

Extract the DNA code of each exon and using transeq (https://www.ebi.ac.uk/Tools/st/ emboss_transeq/) find the protein code of the gene. You can also use blastx (https://blast. ncbi.nlm.nih.gov/Blast.cgi?PROGRAM=blastx&PAG or https://wormbase.org/tools/blast_blat) to obtain protein sequences. How do they compare to your translation?

4.6 Question 3.6

What gene is in the query sequence? Hovering over an exon you should see links to View GeneID and View WormBase. These point to pages with more information on the gene. Follow them and write a few sentences about the gene.