4. Intro - Functional annotation.
Download AUGUSTUS results and extract protein sequences (fasta) from the prediction output
```
./getAnnoFasta.pl augustus.whole.gff
```
Count the number of obtained proteins
```
cat augustus.whole.aa | grep ">" | wc -l
```
Output: 16435

Or
```
grep -c '>' augustus.whole.aa
```
Output: 16435

5. Physical localization
Build a local database
```
makeblastdb -in augustus.whole.aa -parse_seqids -title "Tardigrada"
-dbtype prot
```

or
```
makeblastdb -in ./augustus.whole.aa -dbtype prot -input_type fasta -out
TardProt
```

Blast proteins
```
blastp -query peptides.fa -db augustus.whole.aa > blasted_peptides.fa
```
Extract ID of blasted proteins (without >)
```
cat blasted_peptides.fa | grep  ">" | cut -d " " -f1 | cut -d ">" -f2 |
sort | uniq > blasted_peptides_names.fa
```
Find seq for names of matched peptides in initial file augustus.whole.aa
```
xargs samtools faidx augustus.whole.aa < blasted_peptides_names.fa >
blasted_peptides_seq.fa
```
How many proteins were found?
```
cat blasted_peptides_seq.fa | grep ">" | wc -l
```
Output: 34

Or
Perform blastp of peptides associated with DNA using local database
```
 blastp -query ./peptides.fa -db TardProt -out
./peptides_TardProt.blastp.outfmt6 -outfmt 6
```

Extract ids of subject sequences from local database:
```
 cut -f2 ./peptides_TardProt.blastp.outfmt6 | sort | uniq  >
subject_seqid.txt
```

```
xargs samtools faidx ./augustus.whole.aa < subject_seqid.txt > subject_seq.fa
```

6. Localization prediction

6a. WoLF PSORT
https://wolfpsort.hgc.jp/
In browser, copied results to wolf_result.txt
Select only proteins with nucl localization
```
cat wolf_results.txt | grep "nucl" > wolf_result_nucl.txt
```
Get only information about localization for all proteins:
```
cat wolf_results.txt | cut -d " " -f3,4 | cut -d "," -f1
```
This column will be added to Supplementary table 1 (WoLF PSORT localization)

6b. TargetP 1.1 Server
http://www.cbs.dtu.dk/services/TargetP/
In browser, copied results to TargetP_result.txt
Get only information about localization (signal peptide prediction) for all proteins:
```
cat TargetP_output_protein_type.txtoutput_protein_type.txt | cut -f2
```
This column will be added to Supplementary table 1 (TargetP
localization)

7. BLAST search
BLAST
In browser, saved results to 3FHR30Z701R-Alignment-Descriptions.csv and 3FHR30Z701R-Alignment.txt and 3FHR30Z701R-Alignment_hit.txt.

The information about Accession Number, E-value, % Ident was obtained directly from 3FHR30Z701R-Alignment-Descriptions.csv using a simple R script:

```
setwd("/media/daria/DaryaNika/IB_fall2020/project4")

df <- read.csv("3FHR30Z701R-Alignment-HitTable.csv", header = FALSE)
head(df)

# Colnames were obtained from 3FHR30Z701R-Alignment_hit.txt
colnames(df) <- c("query_acc.ver", "subject_acc.ver", "identity",
"alignment length", "mismatches", "gap_opens", "q.start", "q.end",
"s.start", "s.end", "evalue", "bit_score", "%positives")

# Select only Accession Number, E-value, % Ident. % Query coverage and
annotation are absent so must be added manually later
df1 <- df %>%
  dplyr::select(1,2, evalue, 3) %>%
  group_by(query_acc.ver) %>%
  summarise(subject_acc.ver = subject_acc.ver[1],
            evalue = evalue[1],
            identity = identity[1])

write.table(df1, 'BLAST_output.tsv', sep='\t', row.names = F)
```

After that % Query coverage and annotation were added manually from 3FHR30Z701R-Alignment.txt

8. Pfam prediction

https://www.ebi.ac.uk/Tools/hmmer/

In browser, results copied to Supplementary table 1