Contents

1 Samenvatting

1.1 1.Training van het AUGUSTUS-programma voor het ontdekken van nieuwe genmodellen en hun patronen.

De training van AUGUSTUS vond plaats in verschillende stappen. In het begin werd de predictor uitgevoerd met de standaardinstellingen voor caenorhabditis, wat leidde tot 11.000 voorlopige genmodellen voor één chromosoom, maar met een vrij lage nauwkeurigheid in de voorspellingen. Voor het opstellen van een eerste trainingsset van genen werden RNA-sequencingdata gebruikt. De transcriptomereads werden met TopHat op het genoom gemapt (zie documentatie protocol1, data_processing). Dit resulteerde in 9.953 voorspelde genmodellen per chromosoom op basis van het transcriptoom. Gemiddeld waren de genen ongeveer 5.146 baseparen lang, en elk gen had meestal rond de 3,2 exons (zie protocol1, data_processing,

genemarkES, genemark.average_gene_length.out). De exons waren gemiddeld 1.719 baseparen, terwijl de introns gemiddeld 4.760 baseparen lang waren.

Daarna werden de genensets gefilterd met het Augustus-programma filterGenemark.pl. Na de filtratie bleven er 1.975 genen over op één chromosoom. Etrain werd uitgevoerd met genen die uit het transcriptoom kwamen. De uiteindelijke parameters werden gebruikt om de gff-annotatie te genereren. Een de novo-model met hoge specificiteit en sensitiviteitsscores van 8-9 voor de Lumbricus Terresstris werd verkregen via de mRna-pijplijn.

1.2 2.. De training van het AUGUSTUS-programma met proteïne van langere evolutionaire afstand

Het AUGUSTUS-programma is getraind met proteïne die een langere evolutionaire afstand hebben. Hiervoor is een database uit Ortho DB, Arthropoda ("Bioinformatics Web Server - University of Greifswald" n.d.) gebruikt. Deze database is voorbereid met "ProtHints", wat een onderdeel is van de Braker-pipline(**GaiusAugustusBRAKER2024?**). Voor de BLAST-analyse werd de versie ncbi-blast-2.16.0+ toegepast (zie protocol 2 documentatie). De OrthoDB-database diende als referentie. Om redundantie te verminderen, zijn alle trainingsgen aminozuursequenties met elkaar vergeleken en zijn alleen die eiwitsequenties behouden die minder dan 80% redundant zijn met andere sequenties in de set (zie protocol 2, scripts en documenten). Hieruit is een model (species) afgeleid dat de annotatie heeft opgeleverd.

Na het verwijderen van redundante genstructuren in de proteïne-pijplijn, zijn de specificiteit en gevoeligheid gestegen van 0,01 naar 0,4-0,5 punten voor het de novo-model van de eiwitpijplijn.

2 Alignment. Ruwe gegevens inspecteren

Voor de Alignment zijn Bowtie en Tophat gebruikt. Het transcriptome ID49 is afkomstig van Project: PRJEB59399("ENA Browser" n.d.), dat een verzameling genomische en transcriptomische data bevat voor Lumbricus terrestris, ook wel de gewone regenworm genoemd. Dit project is opgezet om de assemblage en annotatie van het genoom te ondersteunen. Je kunt de ruwe gegevens hier bekijken: https://www.ebi.ac.uk/ena/browser/view/PRJEB59399.

Eerst wordt de index opgebouwd met bowtie2. Daarna vindt de Alignmenet plaats met Tophat. Cufflinks voegt alle reads samen tot transcripties met: cufflinks accepted_hits.bam.

```
bowtie2-build -f  Lumbricus_terrestris-GCA_949752735.1-softmasked.fa  lumter  --large-index
```

```
tophat lumter     sample_1.fastq  sample_2.fastq \
 --output-dir TopHAT \
```

```
 cufflinks  accepted_hits.bam
```

Cufflink zal transcripts.gtf genereren, terwijl TopHat accepted_hits.bam aanmaakt met de resultaten van de uitlijning en een lijst van uitlijningen in jucntions.bed. Elke junction bestaat uit twee verbonden BED-blokken, waarbij elk blok zo lang is als de maximale overhang van een lees die de junction overspant. De score is het aantal uitlijningen dat de junction overspant. Het uitvoerbestand introns.gff bevat informatie over de strengen die gebruikt kan worden voor ET-training.

OX457036.1 TopHat2 intron 253060 254504 12 + . .

Ten eerste moeten we de ruwe gegevens bekijken die we hebben van de genoom-Alignment . Elke junction bestaat uit twee verbonden BED-blokken, waarbij elk blok zo lang is als de maximale overhang van een lees die de junction overspant. De score is het aantal uitlijningen dat de junction overspant. Junctions.bed (TopHat, protocol1, script2):

```
head(junctions)
```

```
##           V1     V2     V3           V4 V5 V6     V7     V8      V9 V10    V11
## 1 OX457036.1 135689 136300 JUNC00000001 16  - 135689 136300 255,0,0   2 143,22
## 2 OX457036.1 136278 139661 JUNC00000002 13  - 136278 139661 255,0,0   2  22,37
## 3 OX457036.1 139624 150988 JUNC00000003  9  - 139624 150988 255,0,0   2  37,70
## 4 OX457036.1 150918 153142 JUNC00000004  1  - 150918 153142 255,0,0   2  70,16
## 5 OX457036.1 150929 156647 JUNC00000005  1  - 150929 156647 255,0,0   2  59,92
## 6 OX457036.1 155453 155919 JUNC00000006  2  - 155453 155919 255,0,0   2  92,59
##       V12
## 1   0,589
## 2  0,3346
## 3 0,11294
```

```
## 4  0,2208
## 5  0,5626
## 6   0,407
```

Cufflink verwerkt de uitgelijnde RNA-Seq-reads die van Tophat komen en bouwt ze op in de transcripten en exonen.

```
transcripts <- read.table("lumbricus/protocol1/data_processing/TOPHAT/transcripts.gtf", sep="\t")

colnames(transcripts) <-  c("chr", "versie", "feature", "start", "end", "score", "strain", "v8")

transcripts  %>%  select(1:5)  %>% head()
```

```
##          chr    versie    feature  start    end
## 1 OX457036.1 Cufflinks transcript 109191 109546
## 2 OX457036.1 Cufflinks       exon 109191 109546
## 3 OX457036.1 Cufflinks transcript 124949 125423
## 4 OX457036.1 Cufflinks       exon 124949 125423
## 5 OX457036.1 Cufflinks transcript 135006 155436
## 6 OX457036.1 Cufflinks       exon 135006 135832
```

1. We gaan de outputbestanden van Tophat+Cufflink, namelijk accepted_hits.bam en junctions.bed, in IGV zetten, samen met het transcriptbestand van Cufflinks. Eerst hebben we een bed-bestand nodig.

```
awk '{if($3=="exon" ) {print $1,$4,$5, $7, $3 }}'  transcripts.gtf  > exon_ids.bed

awk '{if($3=="transcript" ) {print $1,$4,$5, $7, $3 }}'  transcripts.gtf  > tranasctips_ids.b
```

1. Bekijk de bed-bestanden voor de genoombrowser:

```
exons_ids <- read.table("lumbricus/protocol1/data_processing/TOPHAT/igv/exon_ids.bed", sep="\t")

transcript_ids <- read.table("lumbricus/protocol1/data_processing/TOPHAT/igv/transctipts_ids.bed", sep="\t")
```

```
head(exons_ids)
```

```
##                                V1
## 1 OX457036.1 109191 109546 . exon
## 2 OX457036.1 124949 125423 . exon
## 3 OX457036.1 135006 135832 - exon
## 4 OX457036.1 136279 136300 - exon
## 5 OX457036.1 139625 139661 - exon
## 6 OX457036.1 150919 150988 - exon
```

```
head(transcript_ids)
```

```
##                                    V1
## 1 OX457036.1 109191 109546 . transcript
## 2 OX457036.1 124949 125423 . transcript
## 3 OX457036.1 135006 155436 - transcript
## 4 OX457036.1 135006 156649 - transcript
## 5 OX457036.1 135006 156649 - transcript
## 6 OX457036.1 135006 156649 - transcript
```

Bekijk de exonen (diepblauw) en transcripties (lichtblauw) in IGV:

Vervolgens plaatsen we junctions.bed (rood) en geaccepteerde hits of reads (grijs) op dezelfde track om de exon-intronstructuur te visualiseren.

Voordat we GeneMarkET uitvoeren, verzamelen we enkele statistieken uit de primaire analyse. Eerst bekijken we de gemiddelde introns, exonen en lengtes.

```
introns  <- read.table("lumbricus/protocol1/data_processing/TOPHAT/introns.gff", sep="\t")

colnames(introns) <- c("chr","aligner","structure", "start", "end", "score", "strand", "v8", "v9")

head(introns)
```

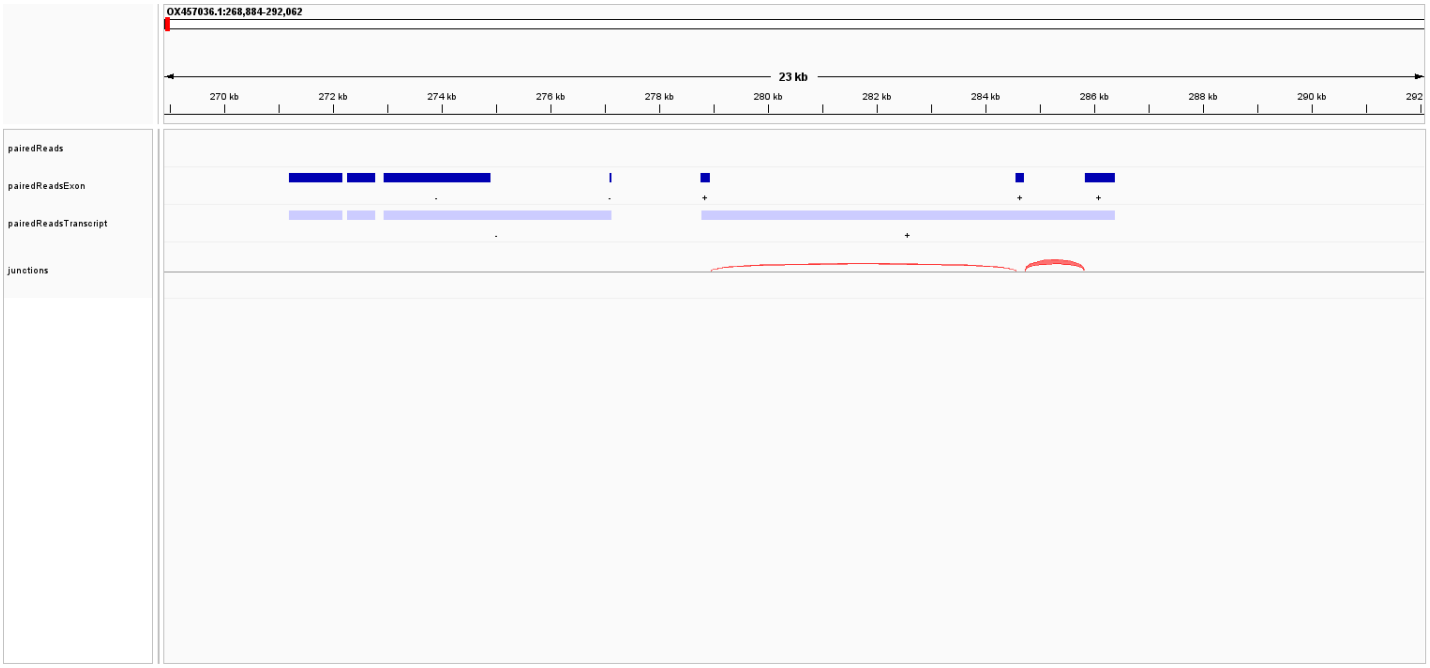


Figure 1: exon-transcripts structure chr1:23kb

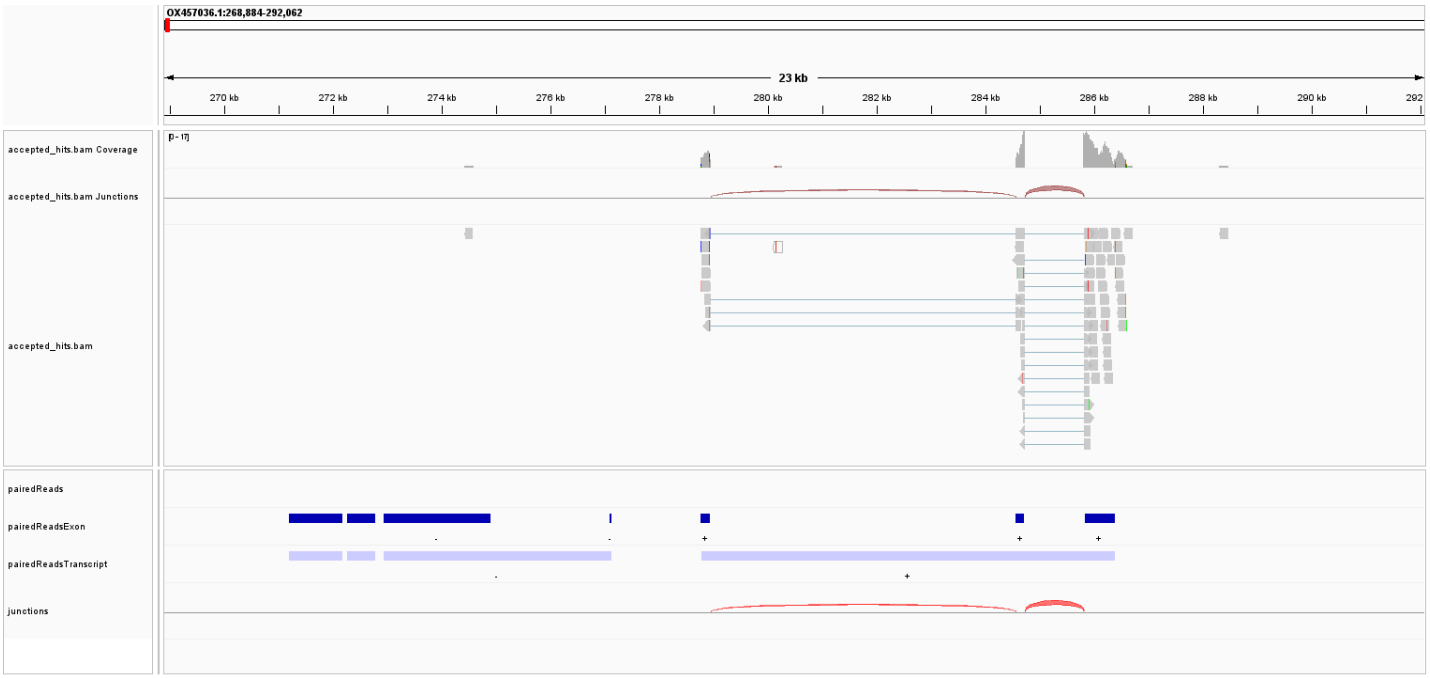


Figure 2: exon-intron structure chr1:23kb, reads in grey

```
##          chr aligner structure  start    end score strand v8 v9
## 1 OX457036.1 TopHat2    intron 135833 136278    16      -  .  .
## 2 OX457036.1 TopHat2    intron 136301 139624    13      -  .  .
## 3 OX457036.1 TopHat2    intron 139662 150918     9      -  .  .
## 4 OX457036.1 TopHat2    intron 150989 153126     1      -  .  .
## 5 OX457036.1 TopHat2    intron 150989 156555     1      -  .  .
## 6 OX457036.1 TopHat2    intron 155546 155860     2      -  .  .
```

```
introns_length <- introns %>%   mutate(ilength=end-start)


max_intron <-  max(introns_length$ilength)  %>%  round(digits = 1)


 avr_intron <-  mean(introns_length$ilength)  %>%  round(digits = 1)
```

```
exons  <- read.table("lumbricus/protocol1/data_processing/TOPHAT/transcripts.gtf", sep="\t")


exons  <- exons %>%  select(1:5)
colnames(exons) <- c("chr","aligner","structure", "start", "end")
exons_length <- exons %>%   mutate(elength=end-start)


max_exon <-  max(exons_length$elength)  %>%  round(digits = 1)
max_exon
```

```
## [1] 255549
```

```
avr_exon <-  mean(exons_length$elength) %>%  round(digits = 1)
```

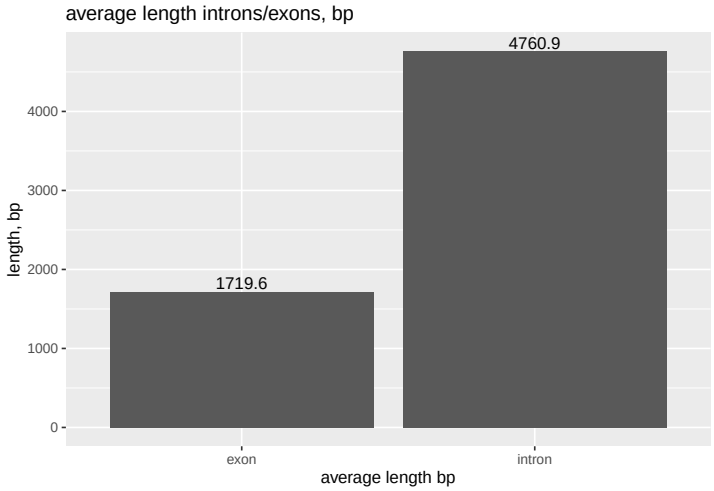
maximale lengte van intron : 2.91919×10^5

gemiddelde intronlengte 4760.9

maximale lengte van exon : 2.55549×10^5

gemiddelde lengte exon 1719.6

```
plot
```



3 GenemarkET. Model opbouwen (protocol 1). mRna pijplijnne

3.1 Deel 1. Model opbouwen

Het script bed_to_gff.pl van GeneMarkES maakt introns.gff aan vanuit de TopHat junctions.bed. Dit bestand bevat informatie over de strengen en kan direct gebruikt worden met GeneMarkET (protocol 1, script 2).

```
introns  <- read.table("lumbricus/protocol1/data_processing/TOPHAT/introns.gff", sep="\t")


colnames(introns) <- c("chr","aligner","structure", "start", "end", "score", "strand", "v8", "v9")


head(introns)
```

```
##          chr aligner structure  start    end score strand v8 v9
## 1 OX457036.1 TopHat2    intron 135833 136278    16      -  .  .
## 2 OX457036.1 TopHat2    intron 136301 139624    13      -  .  .
## 3 OX457036.1 TopHat2    intron 139662 150918     9      -  .  .
## 4 OX457036.1 TopHat2    intron 150989 153126     1      -  .  .
## 5 OX457036.1 TopHat2    intron 150989 156555     1      -  .  .
## 6 OX457036.1 TopHat2    intron 155546 155860     2      -  .  .
```

Om genemark met introns.gff uit te voeren:

```
../../gmes_petap.pl     --verbose --sequence   genome.fa   --ET  introns.gff
```

2. GeneMarkET gaat een ghmm-model en genemark.gtf produceren. Dit bestand(gtf) bevat informatie over de start- en eindcoördinaten van genen, die in de daaropvolgende stap gebruikt zal worden.

```
cut -f 2,3,4,5  lumbricus/protocol1\
/data_processing/GeneMarkES/genemark.gtf  | head
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## cut: lumbricus/protocol1: Is a directory
## cut: /data_processing/GeneMarkES/genemark.gtf: No such file or directory
```

1. Genemark maakt gebruik van filterGenemark.pl voor kwaliteitscontrole. Dit zorgt ervoor dat alleen de genmodellen die geregistreerd zijn in de exon-intronstructuur behouden blijven. (protocol1, script3) Na het filteren van de primaire resultaten wordt er een set van 1.975 genmodellen voor één chromosoom opgeslagen in genemark.f.good.gtf.

```
cut -f 2,3,4,5  lumbricus/protocol1/data_processing/GeneMarkES/genemark.f.good.gtf   | head
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## GeneMark.hmm3    stop_codon  51009   51011
## GeneMark.hmm3    CDS 51009   54860
## GeneMark.hmm3    exon    51009   54860
## GeneMark.hmm3    start_codon 54858   54860
## GeneMark.hmm3    stop_codon  82883   82885
## GeneMark.hmm3    CDS 82883   86734
## GeneMark.hmm3    exon    82883   86734
## GeneMark.hmm3    start_codon 86732   86734
## GeneMark.hmm3    stop_codon  116110  116112
## GeneMark.hmm3    CDS 116110  117048
## bash: [4151828: 2 (255)] tcsetattr: Inappropriate ioctl for device
```

1. Genemark.f.good.gtf is nu klaar om een trainingsset te maken van (protocol1, stap 4 en 5). Eerst wordt gtf omgezet naar gb. Zie protocol1, data_processing, Bonafide.

```
gff2gbSmallDNA.pl bonafide.gtf genome.fa 450 tmp.gb
filterGenesIn_mRNAname.pl bonafide.gtf tmp.gb > bonafide.gb
```

```
cat lumbricus/protocol1/data_processing/bonafide/bonafide.gb | head
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## LOCUS       OX457036.1 Lumbricus terrestris genome assembly, chromosome: 1_50559-55310   4752 bp  DNA
## FEATURES             Location/Qualifiers
##      source          1..4752
##      mRNA            complement(451..4302)
##                      /gene="1_t"
##      CDS             complement(451..4302)
##                      /gene="1_t"
## BASE COUNT     1219 a   994 c  853 g   1655 t   31 n
## ORIGIN
##         1 catccgtctt tttggaatcg atttttatcg tattctgaaa tgttcttatc aatcttacac
## bash: [4151844: 2 (255)] tcsetattr: Inappropriate ioctl for device
```

3.2 Etrain (protocol7)

Op basis van de genen die we hebben verkregen via mRNA-alignment, gaan we een trainingsset opstellen om een nieuw model te trainen. In de vorige sectie hebben we bonafide.gb aangemaakt, waarin 1.975 geverifieerde genen voor een specifiek chromosoom zijn opgenomen. We zijn nu klaar om de ontwikkeling van een nieuwe species te starten.

```
conda activate c
new_species.pl --species=lumter
```

```
etraining --species=lumter bonafide.gb &> bonafide.out
```

Check for Stop Codonds:

```
grep -c "Variable stopCodonExcludedFromCDS set right" bonafide.out
```

0

We hoeven geen bad lijst op te stellen, omdat er geen stopcodons in de CDS aanwezig zijn.

```
grep -c LOCUS bonafide.gb
```

1975

Het randomSplit.pl-script splitst de data op in twee segmenten: een kleinere sectie genaamd test.gb voor trainingsdoeleinden, en een grotere sectie die train.gb wordt genoemd voor de evaluatie van het trainingsproces.

```
randomSplit.pl bonafide.gb 200
mv bonafide.gb.test test.gb
mv bonafide.gb.train train.gb
```

```
etraining --species=lumter train.gb &> etrain.out
```

Deze configuratie kan worden aangepast in het configuratiebestand (map config, species, lumter_parameters.cfg).

tag: 511 (0.259) taa: 700 (0.354) tga: 764 (0.387)

Evaluatie van de voorspelling:

```
augustus --species=lumter test.gb > test.out
```

******* Evaluation of gene prediction *******

```
-----------------------------------------\
                | sensitivity | specificity |
-----------------------------------------|
nucleotide level |       0.963 |       0.972 |
-----------------------------------------/


-----------------------------------------------------------------------------------------------------------\
           |  #pred |  #anno |      |    FP = false pos. |    FN = false neg. |             |             |
           | total/ | total/ |   TP |--------------------|--------------------| sensitivity | specificity |
           | unique | unique |      | part | ovlp | wrng | part | ovlp | wrng |             |             |
-----------------------------------------------------------------------------------------------------------|
           |        |        |      |                 78 |                 87 |             |             |
exon level |    389 |    398 |  311 | ------------------ | ------------------ |       0.781 |       0.799 |
           |    389 |    398 |      |   56 |    6 |   16 |   56 |    6 |   25 |             |             |
-----------------------------------------------------------------------------------------------------------/


-------------------------------------------------------------------------\
transcript | #pred | #anno |   TP |   FP |   FN | sensitivity | specificity |
-------------------------------------------------------------------------|
gene level |   389 |   398 |  311 |   78 |   87 |       0.781 |       0.799 |
-------------------------------------------------------------------------/
# total time: 31.2
# command line:
# augustus --species=wormETO test.gb
```

See also: lumbricus/protocol1/test/test.out

Hier eindigt onze mRNA-pijplijn, waarbij we een hoge specificiteitsscore hebben bereikt voor het model dat we hebben gemaakt voor Lumbricus Terrestris. Dit model zal dienen voor visualisatie.

4 ProtHints en de eiwitpijplijn

4.1 ProtHints

Er zijn veel genen in verschillende genoom die door hun evolutionaire oorsprong met elkaar verbonden zijn. De gelijkenis tussen eiwitsequenties is goed zichtbaar. OrthoDB is een belangrijke bron voor eiwitten en dient als een database die eiwitten met een uitgebreider evolutionair verleden omvat. Zie protocol 2.

```
../bin/prothint.py  ../OX457036.1.fasta  ../Arthropoda.fa
```

```
grep ">" seed_proteins.faa  | wc -l
```

14733

Prothint heeft een database met eiwitten voorbereid voor startAlign.pl. Het resultaat was 14.733 eiwitten in het bestand seed_proteins.faa. Dit seed-bestand kan worden gebruikt met startAlign.pl om een gth.concat.alg-object te verkrijgen, dat vervolgens wordt gebruikt om bonafide.gb te genereren.

```
head lumbricus/protocol2/data_processing/ProtHints/seed_proteins.faa
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## >6249_g
## MPSVSGLIEMMMMMTATITVMMTVTTVRIVERLGWGSYDTDGDDGDDDDDDDDDDDDDDDDDDDDSNNNSSNPPQVTAELCRRELRRCRHRFRSTSSEMTAPPAASA
##
## >10626_g
## MLGRGDCERKKQGNGILETAIHEHAWLQYLEGTDERNGKKSKAGNLKAKREKLQKMRKGDIEEIGLLRGFAERKEKQGETEGLTGQVEEMEIDGPTTEKARHCLVAK
##
## >2633_g
## MSSAHAHVNASRRQQRQTINVRQRKDGEGRRLKRGVLVGNSEDLTVNWVWKATRCRPVPLRYQGVSNETLRMNCNSTSGEGRFGTAIAIGVRRQKKGAKRQQDEKLP
##
## >1749_g
```

Naast het seed_proteins.faa genereert protHints een prothint_augustus.gff hintsbestand dat je direct kunt gebruiken met augustus.

```
head lumbricus/protocol2/data_processing/\

ProtHints/prothint_augustus.gff
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device

## bash: no job control in this shell

## OX457036.1   ProtHint    stop    51009   51011   2   -   0   src=P;mult=9;pri=4;al_score=0.163636;

## OX457036.1   ProtHint    start   52806   52808   2   -   0   src=P;mult=2;pri=4;al_score=0.2;

## OX457036.1   ProtHint    intron  53104   53221   2   -   .   src=P;mult=1;pri=4;al_score=0.361685;

## OX457036.1   ProtHint    intron  53515   53655   0   -   .   src=P;mult=1;pri=4;al_score=0.108387;

## OX457036.1   ProtHint    start   55225   55227   0   -   0   src=P;mult=1;pri=4;al_score=0.104132;

## OX457036.1   ProtHint    stop    82883   82885   2   -   0   src=P;mult=11;pri=4;al_score=0.163636;

## OX457036.1   ProtHint    intron  84978   85095   2   -   .   src=P;mult=1;pri=4;al_score=0.361685;

## OX457036.1   ProtHint    intron  85389   85529   0   -   .   src=P;mult=1;pri=4;al_score=0.108387;

## OX457036.1   ProtHint    start   87099   87101   0   -   0   src=P;mult=1;pri=4;al_score=0.104132;

## OX457036.1   ProtHint    intron  144544  144597  0   +   .   src=P;mult=1;pri=4;al_score=0.13595;
```

2. We kunnen augustus meteen draaien met de prothint_augustus.gff die door de eiwitten zijn gemaakt, voordat we de trainingsset aanpakken.

```
augustus --species=lumter\

--predictionStart=2000000 --predictionEnd=3000000\

OX457036.1.fasta\

--extrinsicCfgFile=extrinsic.cfg\

--hintsfile=prothint_augustus.gff \

> augustus.hints.prots.orthodb.arthropoda.2-3mb.gff
```

Hierdoor ontstaat een annotatie voor 2mb-3mb van het chromosoom, gebaseerd op de eiwitindicaties van eiwitten die een lange evolutionaire afstand hebben.

```
cat   lumbricus/protocol2/data_processing\
/ProtHints/augustus.hints.prots.orthodb.arthropoda.2-3mb.gff | \
tail -n 50
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## # 3'UTR exons and introns: 0/0
## # hint groups fully obeyed: 0
## # incompatible hint groups: 2
## #      P:   2 (407821_0:000ad4_584_g)
## # end gene g81
## ###
## # start gene g82
## OX457036.1   AUGUSTUS    gene    2981898 2982863 1   -   .   g82
## OX457036.1   AUGUSTUS    transcript  2981898 2982863 1   -   .   g82.t1
## OX457036.1   AUGUSTUS    stop_codon  2981898 2981900 .   -   0   transcript_id "g82.t1"; gene_id "g82";
## OX457036.1   AUGUSTUS    CDS 2981898 2982863 1   -   0   transcript_id "g82.t1"; gene_id "g82";
## OX457036.1   AUGUSTUS    start_codon 2982861 2982863 .   -   0   transcript_id "g82.t1"; gene_id "g82";
## # protein sequence = [MDDEETVPYSLPRTTSTPATKGAAEASAFGQSRAEAYRTFEDPEYQFLDLPKKDRKKVLISETTVSDSKRWEDASHLM
## # GPRKIQMKPGKFDGTSSLESFLTQFEVCARHNRWDDSDKVDFLRCALDKAATQLLWDFGARADVTYDQLVGRLRQRYGVEGQAETYRAQLYYRRQRAD
## # ESLSDLLHDIRRLVVLAYPVPSNETTEIVARDSFLEAIRDRELSLKVREREPKSIDEAYRVALRLSAYQQMTDVDDRRRPPNRVRQTQEADAGNQLQT
## # QLDGFLAAQRKWQRDFEDRISLQLNELRNQSQTHPDVAPATRNPASP]
## # Evidence for and against this transcript:
## # % of transcript supported by hints (any source): 100
## # CDS exons: 1/1
## #      P:   1
## # CDS introns: 0/0
## # 5'UTR exons and introns: 0/0
## # 3'UTR exons and introns: 0/0
## # hint groups fully obeyed: 0
```

```
## # incompatible hint groups: 1
## #      P:   1 (407821_0:000ad4_584_g)
## # end gene g82
## ###
## # start gene g83
## OX457036.1   AUGUSTUS    gene    2983320 2984174 0.91    -   .   g83
## OX457036.1   AUGUSTUS    transcript  2983320 2984174 0.91    -   .   g83.t1
## OX457036.1   AUGUSTUS    stop_codon  2983320 2983322 .   -   0   transcript_id "g83.t1"; gene_id "g83";
## OX457036.1   AUGUSTUS    CDS 2983320 2984174 0.91    -   0   transcript_id "g83.t1"; gene_id "g83";
## OX457036.1   AUGUSTUS    start_codon 2984172 2984174 .   -   0   transcript_id "g83.t1"; gene_id "g83";
## # protein sequence = [MEKAGLYFNLKKTKLMTTENWTSFEVDGEEMKVVTCFCFFGAMIENDGGCERYCGSLAGGINFFAVCVFFACSERTCL
## # SEPLVASSSCPLEPAPSSRLFARSNLPLRAARCLFELPDRTCPSEPPVRCSSRTCPSEPLAASSSCPLEPAPPSRLSACSGRLRPLRAASSLFELLAR
## # TSLFRTFAAESNLFVRAACLLLRTGLETYKRRKKKKPSFAVGIEVGESQSLRVNPSGVSQRNEKGSSSSVVRSPSPRKVISSIRQSEVSSSFKLRLKL
## # RLNSGQFVVE]
## # Evidence for and against this transcript:
## # % of transcript supported by hints (any source): 0
## # CDS exons: 0/1
## # CDS introns: 0/0
## # 5'UTR exons and introns: 0/0
## # 3'UTR exons and introns: 0/0
## # hint groups fully obeyed: 0
## # incompatible hint groups: 0
## # end gene g83
## ###
## # command line:
## # augustus --species=lumter --predictionStart=2000000 --predictionEnd=3000000 OX457036.1.fasta --extrinsicC
```

4.2 Protocol 2. Het creëren van genstructuren voor training op basis van eiwitten.

4.3 GenomeThreader

We hebben 14.733 eiwitten verzameld uit de eerdere secties. Nu gaan we een trainingsset opzetten met deze eiwitten. Uit de oorspronkelijke 14.733 eiwitten hebben we een klein deel gekozen om de trainingsset te vormen.

```
startAlign.pl  --genome OX457036.1.fasta  \
--prot seed_proteins.faa  \
--pos OX457036.1:1-10000000 \
--prg gth
```

2. Hierdoor ontstaat het object gth.concat.aln, dat vervolgens kan worden geconverteerd naar het gtf-formaat (protocol2 ,data_processing, protHints).

```
 gth.concat.aln bonafide.gtf
```

Controleer het gtf-bestand :

```
head lumbricus/protocol2/data_processing/Bonafid/bonafide.gtf
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## OX457036.1   gth CDS 51009   54860   .   -   0   gene_id "OX457036.1_g_gene1_mRNA1"; transcript_id "OX45703
## OX457036.1   gth exon    51009   54860   .   -   0   gene_id "OX457036.1_g_gene1_mRNA1"; transcript_id "OX4
## OX457036.1   gth CDS 82883   86734   .   -   0   gene_id "OX457036.1_g_gene2_mRNA2"; transcript_id "OX45703
## OX457036.1   gth exon    82883   86734   .   -   0   gene_id "OX457036.1_g_gene2_mRNA2"; transcript_id "OX4
## OX457036.1   gth CDS 104626  104645  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104626  104645  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
## OX457036.1   gth CDS 104696  104750  .   -   0   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104696  104750  .   -   0   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
## OX457036.1   gth CDS 104904  105745  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104904  105745  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
```

```
computeFlankingRegion.pl bonafide.gtf
```

Output van computeFlankingRegion.pl:

Total length gene length (including introns): 5412279. Number of genes: 1090. Average Length: 4965.39357798165 The flanking_DNA value is: 2482 (the Minimum of 10 000 and 2482)

```
gff2gbSmallDNA.pl bonafide.gtf genome.fa 2482 bonafide.gb
```

Bonafide.gb wordt in de volgende pipeline gebruikt om redundantie te verwijderen.

4.4 Protocol 6.Verwijderen van Redundant Genstructuren (protocol 6)

Voor NCBI Blast, controleer de link en stel het Path in naar de Blast uitvoerbare bestanden.

```
wget ftp://ftp.ncbi.nlm.nih.gov/blast/executables/blast+/LATEST/
```

export PATH=$PATH$:HOME/ncbi-blast-2.16.0+

Maak gebruik van de opgegeven commandoregel om het GTF-bestand van de trainingsgenstructuur te transformeren naar een FASTA-bestand dat de eiwitsequentie omvat.

```
gtf2aa.pl genome.fa bonafide.f.gtf prot.aa
```

Inspecteer prot.aa :

```
head lumbricus/protocol2/data_processing/Redundancy/prot.aa
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## >OX457036.1_t_gene395_mRNA512
## ESLLPRCCPAGRGGGSQDSIAHARCFDRRITFSMMTLVGLGKEGLKRRKGGMDGERDLNWLEGGMGGEVQNWRVIGIERRY*
## >OX457036.1_t_gene508_mRNA640
## MEESRPVTPAQPSRPPSSMEVLLEAIQTNAKSTHDAMTSIQSSLQLNARDTQEAIATVELNVLAVQSNVREEISSVKSYVRDTQDAISSVQSNVSDAISSVQLNVRE
## >OX457036.1_t_gene532_mRNA668
## MTCLRRIEGVTRRERIRNTEIHNRLKIQRDIVDRIQIRRMRYFGHVVRMQSGRYPKVALQGYVHGKRRRGRPRKRWMDVAEEDCLRMGLTVGEATRRAQDRDDWRLS
```

```
## >OX457036.1_t_gene891_mRNA1213
## GKGRVNGCCFWRIRSGKLVREISTFCDIEFCEFCKFGRDSFEVSCRGGKMASLEEELIPEFGDVRDIPSDTLRLVSETYGEEVEDVSRSQVRRMAMKPLSPKLGSAA
## >OX457036.1_t_gene296_mRNA397
## WSEEPEEGDGVWLVMIVEELQKIGIHEADHSMVDHIRNEEVLKLAGSRYLEYIIMGRRGRLAGHILRLPKERIARTAIKWVPEGGKRRRGRPRNTWRRTFKGDLERM
```

Voer een Blast uit van alle eiwitsequenties uit de vorige stap met elkaar en toon alleen de eiwitsequenties die minder dan 80% identiek zijn aan een andere sequentie in de groep.

```
aa2nonred.pl prot.aa prot.nr.aa
```

```
head lumbricus/protocol2/data_processing/Redundancy/prot.nr.aa
```

```
grep ">"  lumbricus/protocol2/data_processing/Redundancy/prot.nr.aa | wc -l
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## >OX457036.1_t_gene753_mRNA930
## SIVGAATEVYNRMSSDFLPTPTKSHYIFNLRDLSKCIQESKQVFRLFCHEALRVFHDRLTTSEDKMSFYAILAEIAPKFFNENADAQSFLKHPIIFGDFIKVAAPRE
## >OX457036.1_t_gene803_mRNA1030
## SQKSRASATIVVCDLDHMMIRLPHFTAKRSVEPFQSTEEQVLGRIRSFPQGSSGGPDGLRPQHLSDLVNCVEIGSELIFAITGLVNLLLKGECPEDIRPVLFGGTLM
## >OX457036.1_t_gene103_mRNA135
## MSINFAQRIQMPGIERVHGVTKVRNEFNILGYSVSFRYVISVFEDRIPYRLRKEIRLTGIRNAVDIGSSENANCLYVSDYEEKCVRKITRERDGGHKIIKWLITAYR
## >OX457036.1_t_gene500_mRNA632
## IMRAEIQGRLNRGRQKKSWMDMIQQDMEFLGLRKEEVRDRTTWRQRIRINGLKYVYVYGHVSVNMKDIIIEHRLTVAELHFLKRAEILDRREKPLDVERKRQTETET
## >OX457036.1_t_gene503_mRNA635
## MCEVAEYFENGELVIFDDSDPAPSYADEMESDEMDDSKSDFPEAECAMALLELAQSFGLVSSLNSFGHINDETGLRNAATEPSNVPLNNTAENLASTADARQHFSAF
## 602
```

Daarna hebben we 602 niet-redudante eiwitten om mee verder te gaan:

```
grep ">"  lumbricus/protocol2/data_processing/Redundancy/prot.nr.aa | wc -l
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## 602
```

```
cat bonafide.gb | perl -ne 'if(m/\/gene=\"(\S+)\"/){ \
print "\"".$1."\"\n";}' | sort -u > traingenes.lst
```

regel 1: syntaxisfout bij onverwacht token '('

Dit leverde een syntaxisfout op, waarna alle perl -ne regex werden vervangen door Python regex, die werden uitgevoerd in de IDE.

```
import re
import subprocess

# Read from the file 'bonafide.gb'
with open('bonafide.gb', 'r') as file:
    content = file.read()

# Find all unique gene names
gene_names = set(re.findall(r'/gene="(\S+)"', content))

# Writing unique gene names to a file
with open('traingenes.lst', 'w') as f:
    for gene in sorted(gene_names):
        f.write(f'"{gene}"\n')
```

De uitvoer bevat de strings die als transcriptnamen worden gebruikt in het bonafide.gtf-bestand, waaruit bonafide.gb oorspronkelijk is gemaakt, met aanhalingstekens.

```
head lumbricus/protocol2/data_processing/Redundancy/traingenes.lst
```

```
## "OX457036.1_t_gene1000_mRNA1441"
## "OX457036.1_t_gene1001_mRNA1448"
## "OX457036.1_t_gene1002_mRNA1452"
## "OX457036.1_t_gene1003_mRNA1454"
## "OX457036.1_t_gene1004_mRNA1455"
## "OX457036.1_t_gene1006_mRNA1462"
## "OX457036.1_t_gene1007_mRNA1463"
## "OX457036.1_t_gene1008_mRNA1464"
## "OX457036.1_t_gene1009_mRNA1468"
## "OX457036.1_t_gene100_mRNA132"
```

Hierna volgt een reeks scripts/opdrachten die alleen bedoeld zijn om een lijst te verkrijgen van niet-redudante genen en hun bijbehorende loci in GeneBank.Dit is voornamelijk een bewerking voor tekstbestanden

```
grep -oE '(OX457036[A-Za-z1-9._]{1,})\w+'  prot.nr.aa > nonred.lst
```

```
head lumbricus/protocol2/data_processing/Redundancy/nonred.lst
```

```
## OX457036.1_t_gene753_mRNA930
## OX457036.1_t_gene803_mRNA1030
## OX457036.1_t_gene103_mRNA135
## OX457036.1_t_gene500_mRNA632
## OX457036.1_t_gene503_mRNA635
## OX457036.1_t_gene504_mRNA636
## OX457036.1_t_gene573_mRNA720
## OX457036.1_t_gene618_mRNA766
## OX457036.1_t_gene384_mRNA500
## OX457036.1_t_gene496_mRNA628
```

Isoleer de genen in traingenes.lst van bonafide.gtf:

```
grep -f traingenes.lst -F bonafide.gtf > bonafide.f.gtf
```

```
head lumbricus/protocol2/data_processing/Redundancy/bonafide.f.gtf
```

```
## OX457036.1   gth CDS 51009   54860   .   -   0   gene_id "OX457036.1_g_gene1_mRNA1"; transcript_id "OX45703
## OX457036.1   gth exon    51009   54860   .   -   0   gene_id "OX457036.1_g_gene1_mRNA1"; transcript_id "OX4
## OX457036.1   gth CDS 82883   86734   .   -   0   gene_id "OX457036.1_g_gene2_mRNA2"; transcript_id "OX45703
## OX457036.1   gth exon    82883   86734   .   -   0   gene_id "OX457036.1_g_gene2_mRNA2"; transcript_id "OX4
## OX457036.1   gth CDS 104626  104645  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104626  104645  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
## OX457036.1   gth CDS 104696  104750  .   -   0   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104696  104750  .   -   0   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
## OX457036.1   gth CDS 104904  105745  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX45703
## OX457036.1   gth exon    104904  105745  .   -   2   gene_id "OX457036.1_g_gene3_mRNA3"; transcript_id "OX4
```

```
grep -oE '(OX457036[A-Za-z1-9._]{1,})\w+'  prot.nr.aa > nonred.lst
```

```
head lumbricus/protocol2/data_processing/Redundancy/nonred.lst
```

```
## OX457036.1_t_gene753_mRNA930
## OX457036.1_t_gene803_mRNA1030
## OX457036.1_t_gene103_mRNA135
## OX457036.1_t_gene500_mRNA632
## OX457036.1_t_gene503_mRNA635
## OX457036.1_t_gene504_mRNA636
## OX457036.1_t_gene573_mRNA720
## OX457036.1_t_gene618_mRNA766
## OX457036.1_t_gene384_mRNA500
## OX457036.1_t_gene496_mRNA628
```

In nonred.lst gaan we nu een niet-redundante subset van genen vinden.

Voor het filteren van het bestand bonafide.gb hebben we een lijst met loci-namen nodig in plaats van genenamen.

```
cat bonafide.gb | perl -ne '

if ( $_ =~ m/LOCUS\s+(\S+)\s/ ) {

$txLocus = $1;

} elsif ( $_ =~ m/\/gene=\"(\S+)\"/ ) {

$txInGb3{$1} = $txLocus

}

if( eof() ) {

foreach ( keys %txInGb3 ) {

print "$_\t$txInGb3{$_}\n";

}

}' > loci.lst
```

```
Unrecognized character \xE2; marked by <-- HERE after <-- HERE near column 1 at -e line 1.

cat: write error: Broken pipe

./test.sh: line 2: syntax error near unexpected token `('

./test.sh: line 2: `if ( $_ =~ m/LOCUS\s+(\S+)\s/ ) {'
```

Deze commando van het protocol veroorzaakte een fout en is vervangen. Het is nu locilist.py (scripts, protocol2).

```
import re


txInGb3 = {}

txLocus = ""


with open("bonafideOrtho.gb.db") as file:

    for line in file:

        if re.search(r'LOCUS\s+(\S+)\s', line):

            txLocus = re.search(r'LOCUS\s+(\S+)\s', line).group(1)

        elif re.search(r'/gene="(\S+)"', line):

            gene = re.search(r'/gene="(\S+)"', line).group(1)
```

```
            txInGb3[gene] = txLocus


with open("loci.lst", "w") as output_file:
    for key in txInGb3.keys():
        output_file.write(f"{key}\t{txInGb3[key]}\n")
```

en nonred.loci.py (scripts, protocol2)):

```
import subprocess


with open('nonred.lst', 'r') as f:
    patterns = f.read().splitlines()


with open('loci.lst', 'r') as f:
    loci = f.read().splitlines()


matched_loci = [locus.split('\t')[1] for locus in loci if any(pattern in locus for pattern in patterns)]


with open('nonred.loci.lst', 'w') as f:
    f.write('\n'.join(matched_loci))
```

wat nonred.loci.lst en loci.lst (met 2 kolommen) produceert:

```
head lumbricus/protocol2/data_processing/Redundancy/nonred.loci.lst
```

```
## OX457036.1_102144-115856
## OX457036.1_161655-167728
## OX457036.1_180282-185623
## OX457036.1_225887-235418
## OX457036.1_345964-351295
## OX457036.1_411769-417637
```

```
## OX457036.1_418604-428585
## OX457036.1_428586-437296
## OX457036.1_468333-473965
## OX457036.1_488481-495418
```

```
head lumbricus/protocol2/data_processing/Redundancy/loci.lst
```

```
## OX457036.1_t_gene1_mRNA1 OX457036.1_48527-57342
## OX457036.1_t_gene2_mRNA2 OX457036.1_80401-89216
## OX457036.1_t_gene3_mRNA3 OX457036.1_102144-115856
## OX457036.1_t_gene4_mRNA4 OX457036.1_138781-147529
## OX457036.1_t_gene5_mRNA5 OX457036.1_161655-167728
## OX457036.1_t_gene6_mRNA6 OX457036.1_180282-185623
## OX457036.1_t_gene7_mRNA7 OX457036.1_225887-235418
## OX457036.1_t_gene8_mRNA8 OX457036.1_321850-327440
## OX457036.1_t_gene9_mRNA9 OX457036.1_345964-351295
## OX457036.1_t_gene10_mRNA10   OX457036.1_389861-394620
```

```
filterGenesIn.pl nonred.loci.lst bonafide.gb > bonafide.f.gb
```

Deze commando haalt enkel de laatste locus uit de bonafide.gb. Het doel is om alle unieke loci uit de bonafide.gb te verzamelen, niet alleen de laatste.

Om alle unieke loci te krijgen, moeten we dit in een loop zetten (protocol2, scripts, bonafide.nonred.f.py).

```
import re
origfilename ="bonafideRED.gb"
goodfilename ="nonred.loci.lst"

goodlist = {}
with open(goodfilename, 'r') as goodfile:
    for line in goodfile:
```

```
        goodlist[line.strip()] = 1



with open(origfilename, 'r') as origfile:
    content = origfile.read().split('\n//\n')
    for gendaten in content:
        match = re.match(r'^LOCUS +(\S+) .*', gendaten)
        if match:
            genname = match.group(1)
            if genname in goodlist:
                with open('bonafide.filtered.nonred.gb', 'a') as f2:
                    f2.write( gendaten+ '\n'+'//'+'\n')
                    f2.close()
```

```
grep -c LOCUS lumbricus/protocol2/data_processing/Redundancy/bonafide.f.nonred.gb
```

```
## 602
```

Na deze fase zijn er 602 verschillende loci in Bonafide.

4.5 Trainingsset van Proteins.Etrain

We hebben in de vorige sectie 602 niet-redundante genstructuren ontdekt die kunnen dienen om een nieuwe soort te ontwikkelen.

Creëer een nieuwe species

```
new_species.pl --species=wormNonredEP
```

```
etraining --species=wormNonredEP bonafide.gb &> bonafide.out
```

Check for stop-codons:

```
grep -c "Variable stopCodonExcludedFromCDS set right" bonafide.out
```

49

We moeten 49 stopcodons uitfilteren.Bad List:

```
etraining --species=wormNonredEP    bonafide.gb 2>&1\
| grep "in sequence" \
|  sed -E 's/.*n sequence (\\S+):.*/\\1/' \
| sort -u   > bad.pre.list


grep -oE  "in sequence.*(OX457036.[1-9A-Za-z_0-]{1,})\w+" \
bad.pre.list\
| grep -oE  "(OX457036.[1-9A-Za-z_0-]{1,})\w+"> bad.list
```

```
head  ~/lumbricus/protocol2/data_processing/bad-list/bad.list
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## OX457036.1_80264327-80269533
## OX457036.1_3169142-3174603
## OX457036.1_3169142-3174603
## OX457036.1_82306032-82311964
## OX457036.1_83519819-83526493
## OX457036.1_85356189-85367403
## OX457036.1_3254876-3258078
## OX457036.1_87513969-87519492
## OX457036.1_3258079-3261568
```

```
## OX457036.1_92067632-92073579
```

Vervolgens fitler bad.list uit bonafide.gb:

```
perl filterGenes.pl  bad.list  bonafide.filtered.nonred.gb  \
> bonafide.filtered.gb
```

```
grep -c LOCUS bonafide.gb  bonafide.filtered.gb
```

bonafide.gb:602 bonafide.filtered.gb:373

```
ln -s bonafide.filtered.gb  bonafide.gb
```

test.gb is een klein bestand dat dient voor training. Train.gb is een groot bestand dat gebruikt wordt om de training te evalueren.

```
randomSplit.pl bonafide.gb 200
mv bonafide.gb.test test.gb
mv bonafide.gb.train train.gb
```

```
etraining --species=wormNonredEP  train.gb &> etrain.out
```

```
cat  lumbricus/protocol2/data_processing/Bonafid/etrain.out
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## # Read in 373 genbank sequences.
## Quantiles of the GC contents in the training set:
## 0%   0.351
```

```
## 5%   0.385   10% 0.388
## 15%  0.393   20% 0.397
## 25%  0.4 30% 0.403
## 35%  0.405   40% 0.407
## 45%  0.412   50% 0.415
## 55%  0.417   60% 0.419
## 65%  0.425   70% 0.429
## 75%  0.432   80% 0.438
## 85%  0.446   90% 0.456
## 95%  0.478   100%    0.596
## HMM-training the parameters...
## i= 0 bc= (0.237, 0.263, 0.263, 0.237)
##  ** building model for exons *EXON*
## gene OX457036.1_t_gene1093_mRNA1640 transcr. 1 in sequence OX457036.1_98424851-98432363: Initial exon does
## start codon frequencies: ATG(372)
## # admissible start codons and their probabilities: ATG(1), CTG(0), TTG(0)
##  number of bases in the reading frames: 160917 161284 161285
## --- frame = 0 ---    minPatSum = 233
## --- frame = 1 ---    minPatSum = 233
## --- frame = 2 ---    minPatSum = 233
## --- initial frame = 0 ---    minPatSum = 233
## --- initial frame = 1 ---    minPatSum = 233
## --- initial frame = 2 ---    minPatSum = 233
## --- internal exon terminal frame = 0 ---    minPatSum = 233
## --- internal exon terminal frame = 1 ---    minPatSum = 233
## --- internal exon terminal frame = 2 ---    minPatSum = 233
## single, initial, internal, terminal mean exon lengths :
## 934  275 199 246
## single exon : 66
## initial exon 0 : 134
```

```
## initial exon 1 : 79
## initial exon 2 : 93
## internal exon 0 : 511
## internal exon 1 : 196
## internal exon 2 : 193
## terminal exon : 307
## Frequency of stop codons:
## tag:   97 (0.26)
## taa:  102 (0.273)
## tga:  174 (0.466)
## end *EXON*
## Storing parameters to file...
## Writing exon model parameters [1] to file /home/alena/anaconda3/envs/c/config/species/wormNonredEP/wormNonr
```

```
tail -6 etrain.out | head -3
```

tag: 97 (0.26) taa: 102 (0.273) tga: 174 (0.466)

Je moet deze waarden corrigeren in je wormNonredEP_parameters.cfg in config map

```
augustus --species=wormNonredEP test.gb > test.out
```

Eerst werd er een test gedaan op het model voordat het geoptimaliseerd werd, waarbij redudante structuren werden verwijderd. Deze test gaf een gevoeligheid en specificiteit van 0.01.

Na het toepassen van het protocol voor het verwijderen van redundante genstructuren, nam de specificiteit toe met 0,3 tot 0,5 punten.

```
*******      Evaluation of gene prediction     *******

-----------------------------------------------\
```

```
                | sensitivity | specificity |
-------------------------------------------|
nucleotide level |       0.942 |       0.762 |
-------------------------------------------/


-----------------------------------------------------------------------------------------------------------\
           |  #pred |  #anno |      |    FP = false pos. |    FN = false neg. |             |             |
           | total/ | total/ |   TP |--------------------|--------------------| sensitivity | specificity |
           | unique | unique |      | part | ovlp | wrng | part | ovlp | wrng |             |             |
-----------------------------------------------------------------------------------------------------------|
           |        |        |      |              1071 |                767 |             |             |
exon level |   1884 |   1580 |  813 | ------------------ | ------------------ |       0.515 |       0.432 |
           |   1884 |   1580 |      |  436 |  104 |  531 |  456 |  145 |  166 |             |             |
-----------------------------------------------------------------------------------------------------------/


-------------------------------------------------------------------------\
transcript | #pred | #anno |   TP |   FP |   FN | sensitivity | specificity |
-------------------------------------------------------------------------|
gene level |   454 |   373 |   88 |  366 |  285 |       0.236 |       0.194 |
-------------------------------------------------------------------------/
```

Zie lumbricus/protocol2/test/test.out voor meer informatie.

5 Identificatie en visualisatie

5.1 Gen-identificatie

Alle voorspellingen zijn gebaseerd op een DNA-fragment van 1 mb, wat overeenkomt met 1% van chromosoom. De exacte locatie is aangeduid als 2000000-3000000. (2-3 mb) van chr1. De predictor is toegepast op het nieuwe lumtermodel (zie, protocol 1, model) dat in deel 1 is ontwikkeld. Alle stappen voor identificatie zijn vastgelegd in prediction.xlsx (map identification).

```
augustus --species=lumter lumter.fasta --predictionStart=2000000 --predictionEnd=3000000 --gff3=on
```

Voor het identificeren van genen hebben we de qblast() functie gebruikt uit de Bio.Blast.NCBIWWW module van Biopython. De qblast functie heeft verschillende opties die vergelijkbaar zijn met de parameters die je kunt instellen op de BLAST webpagina. Wij hebben nucleotide blast ("blastn", "nt") gebruikt. Deze functie is bedoeld om nucleotidesequenties te vinden die vergelijkbaar zijn met die van andere organismen, en deze gegevens zijn beschikbaar in de NCBI-database. Hulp voor de qblast functie:

```
from Bio.Blast import NCBIWWW

help(NCBIWWW.qblast)
```

Some useful parameters:

```
 - program        blastn, blastp, blastx, tblastn, or tblastx (lower case)
 - database       Which database to search against (e.g. "nr").
 - sequence       The sequence to search.
 - ncbi_gi        TRUE/FALSE whether to give 'gi' identifier.
 - descriptions   Number of descriptions to show.  Def 500.
 - alignments     Number of alignments to show.  Def 500.
 - expect         An expect value cutoff.  Def 10.0.
 - matrix_name    Specify an alt. matrix (PAM30, PAM70, BLOSUM80, BLOSUM45).
 - filter         "none" turns off filtering.  Default no filtering
 - format_type    "HTML", "Text", "ASN.1", or "XML".  Def. "XML".
 - entrez_query   Entrez query to limit Blast search
 - hitlist_size   Number of hits to return. Default 50
 - megablast      TRUE/FALSE whether to use MEga BLAST algorithm (blastn only)
 - short_query    TRUE/FALSE whether to adjust the search parameters for a
                  short query sequence. Note that this will override
                  manually set parameters like word size and e value. Turns
                  off when sequence length is > 30 residues. Default: None.
 - service        plain, psi, phi, rpsblast, megablast (lower case)
```

```
This function does no checking of the validity of the parameters
and passes the values to the server as is.  More help is available at:
https://ncbi.github.io/blast-cloud/dev/api.html
</p>
```

Eerst hebben we het ruwe GFF-bestand voorbereid voor de Blast API door alle spaties en het '#' symbool te verwijderen. Om de gencoördinaten te krijgen, maakten we gebruik van een regex-patroon.

```
pattern_a = r'gene.*\s+(OX457036.*AUGUSTUS\sgene.*g\d+)'
```

Voor het ophalen van de coderingssequentie uit het GFF-bestand maakten we gebruik van een andere regex.

```
pattern_b = r"coding sequence =.*[actg\s\]]{1,}".
```

Nadat je het GFF-bestand hebt geparsed, is het klaar voor gebruik met de Blast API. Elke coderingssequentie heeft een unieke identificatie die de start- en eindcoördinaten bevat: genomisch OX457036.1:2000789-2003917

Voor meer details kun je de scripts bekijken, vooral parsegtf.py, deel identification.

```
head lumbricus/identification/prediciton/genome.fa.gff
```

```
## bash: cannot set terminal process group (4151773): Inappropriate ioctl for device
## bash: no job control in this shell
## >genomic OX457036.1:2000789-2003917
## atggaggagtctaggccagtcactcccgctcagccttctaggcccccttcttctatggagatattgctcgaggcaatac
## aaactaatgctaggtccactcatgaagcaatacagactaacgctaagtcttcacaagaggctatgcaagcgcatgctaagtcaactcatgatgctatg
## acttctatacagtcgtctttgcaactgaatgccagagagacgcaagaggcgattgccacggtggagtttaatgtcctggcagtgcaatcaaatgttag
## cgaagctatttcctcagtgcaatcaaatgtaagagaggagataagagaagagatctcggctgtaagagataatgtcagggaagcgctgacggaaatgg
## tatcacgattggaaaggctagaggcgtcgccggtacccaagcctgctgtggattcgaaccctggttacctcaccgctattacccctgcggacgcgcca
## taccactcgaccatcggcctgggggaaactttgggtgctaggcctaaagatttcacgcaacctggtatattgcggagaagtgatagattggctggtag
## gccgccaatttcatatagggagtacggtagtcgaaaagactggccgcctttcctgggttgggattcgaacccagaagtcacctcctcctgtcctccct
## ctatctctagagctcgtccacagcagcacgcggtcccatcaggcgaggatccggaagtggcgactccggggatgccgataggggcgggcgttacaatt
## ggtcccagccagtggggtcaaattagttctagagattttggtgatgataggttagaagaggaaactgactatgctagaacaggcgaaatggcaatttc
```

De blast-query's via Bio.Blast.NCBIWWW.qblast zijn uitgevoerd en de resultaten zijn teruggegeven in XML-formaat (voor meer informatie, zie: blast.py).

```
from Bio.Blast import NCBIWWW
from Bio.Blast import NCBIXML



genomic="genome.fa"


sequence_data = open(genomic).read()


sequence_data


result_handle = NCBIWWW.qblast("blastn", "nt", sequence_data, hitlist_size=5, alignments=50)



with open('reults.xml', 'w') as save_file:
   blast_results = result_handle.read()
   save_file.write(blast_results)
```

Voor de blast-analyse is het bestand genome.fa opgedeeld in drie verschillende fracties, wat resulteerde in 3 xml-bestanden (identificatie->xml). Elke DNA-sequentie die je invoert in nucleotide BLAST krijgt een bepaald aantal hits, en het geeft ook wat statistieken over die hits.

Een voorbeeld van een hit: .

```
<Iteration_hits>
<Hit>
  <Hit_num>1</Hit_num>
  <Hit_id>gi|11071239|emb|AJ299434.1|</Hit_id>
  <Hit_def>Lumbricus rubellus mt2A gene for metallothionein 2A, exons 1-4</Hit_def>
  <Hit_accession>AJ299434</Hit_accession>
```

```
  <Hit_len>7302</Hit_len>
  <Hit_hsps>
    <Hsp>
      <Hsp_num>1</Hsp_num>
      <Hsp_bit-score>85.143</Hsp_bit-score>
      <Hsp_score>93</Hsp_score>
      <Hsp_evalue>7.19655e-12</Hsp_evalue>
      <Hsp_query-from>70</Hsp_query-from>
      <Hsp_query-to>246</Hsp_query-to>
      <Hsp_hit-from>306</Hsp_hit-from>
      <Hsp_hit-to>490</Hsp_hit-to>
      <Hsp_query-frame>1</Hsp_query-frame>
      <Hsp_hit-frame>1</Hsp_hit-frame>
      <Hsp_identity>131</Hsp_identity>
      <Hsp_positive>131</Hsp_positive>
      <Hsp_gaps>8</Hsp_gaps>
      <Hsp_align-len>185</Hsp_align-len>
      <Hsp_qseq>AGATTGAACATCAAACAGGATATAGTTGACAAAGTGCGGAATAGAAGAATGCGATACTTTGGACATGTGA------CAAGAATGGGGAACGAAA
      <Hsp_hseq>AGACTGAATATTCAACATGATATAATACACAAGATCCAAAGTAAACGACTACGCTACTTTGGCCACGTATATATATCCAGAATGAGGGATGAGA
      <Hsp_midline>||| |||| ||  |||| |||||| |  ||||  | |  | || | || | || |||||||| || ||        | |||||| || | |
    </Hsp>
  </Hit_hsps>
</Hit>
```

De XML-resultaten van de blast-uitvoer laten zien hoe goed de Alignment overeenkomt, samen met de eval-waarde. De gevonden Hits worden bewaard met het NCBI-referentienummer, zoals “ref XM_003731435.1”, of het Ensemble-referentienummer, zoals “emb OE003277.1”. Zodra je de XML-resultaten hebt, is de eerste stap om ze te parseren. De XML-resultaten zijn geparsed en gesorteerd op coördinaten en e-waarde (sort-blast-by-coords.py, sort-blast-by-pval.py).

```
import os

cwd = os.getcwd()

print(cwd)



import sys

from Bio.Blast import NCBIXML

OUT = open("sorted_by_coordinates.fraction3.txt", 'w')

OUT.write("Query Name\tQuery Length\tAlignment ID NCBI\teValue\n")

result_handle = open("blast.results.fraction3.xml")

blast_records = NCBIXML.parse(result_handle)

for rec in blast_records:

        for alignment in rec.alignments:

            for hsp in alignment.hsps:

                fields = [rec.query_id, rec.query[:100], str(rec.query_length), alignment.hit_id,

                          alignment.accession, str(hsp.expect)]

                OUT.write("\t".join(fields) + "\n")

OUT.close()

print('Done')
```

```
sorted_by_coordinate <- read_excel("lumbricus/identification/prediction.xlsx", sheet = 6 )



sorted_by_p <- read_excel("lumbricus/identification/prediction.xlsx", sheet = 5 )



# sorted by coordinates

head(sorted_by_coordinate )
```

```
## # A tibble: 6 x 6

##   `Query Name`  `Query Length`        `Alignment ID NCBI` eValue Column1    `_1`

##   <chr>         <chr>                                 <dbl> <chr>  <chr>     <dbl>

## 1 Query_1234140 genomic OX457036.1:2~                   459 gi|26~ XM_063~ 6.18e-5

## 2 Query_1234140 genomic OX457036.1:2~                   459 gi|26~ XM_063~ 3.20e-2
```

```
## 3 Query_1234140 genomic OX457036.1:2~                459 gi|26~ XM_062~ 4.75e-1
## 4 Query_1234140 genomic OX457036.1:2~                459 gi|26~ XM_062~ 4.75e-1
## 5 Query_1234140 genomic OX457036.1:2~                459 gi|26~ XM_062~ 4.75e-1
## 6 Query_1234141 genomic OX457036.1:2~                408 gi|28~ OZ0783~ 4.46e-6
```

```
# sorted by p-val
head(sorted_by_p)
```

```
## # A tibble: 6 x 2
##   Column1 Column2
##   <chr>   <chr>
## 1 <NA>    <NA>
## 2 query:  genomic OX457036.1:2108840-2109808
## 3 match:  gi|2739567124|gb|CP157508.1| Candidozyma auris strain BA03 chromosome~
## 4 query:  genomic OX457036.1:2108840-2109808
## 5 match:  gi|2739567124|gb|CP157508.1| Candidozyma auris strain BA03 chromosome~
## 6 query:  genomic OX457036.1:2108840-2109808
```

Eerst moeten we naar alle voorspellingen kijken, ook naar de voorspellingen met ongunstige eval-waarden (vergelijkbaar met p-waarden). Alle voorspellingen: .

```
all_predictions <- read_excel("lumbricus/identification/prediction.xlsx", sheet = 1 )


all_predictions
```

```
## # A tibble: 89 x 5
##    `OX457036.1:2000789-2003917` AUGUSTUS gene  predicted:not satisfac~1 `185403`
##    <chr>                        <chr>    <chr> <chr>                   <chr>
##  1 OX457036.1:2007959-2008723   AUGUSTUS gene  predicted:not satisfact~ 1852
##  2 OX457036.1:2039309-2039692   AUGUSTUS gene  predicted:not satisfact~ 881419
##  3 OX457036.1:2062296-2062562   AUGUSTUS gene  predicted:not satisfact~ 0
##  4 OX457036.1:2087020-2087265   AUGUSTUS gene  predicted: Lumbricus ru~ 7.38114~
```

```
##  5 OX457036.1:2089471-2089899   AUGUSTUS gene  predicted:Lampetra plan~ 8.69409~
##  6 OX457036.1:2090721-2091137   AUGUSTUS gene  predicted:not satisfact~ 965729
##  7 OX457036.1:2106048-210639    AUGUSTUS gene  predicted:Mus musculus ~ 6.12276~
##  8 OX457036.1:2106538-2106948   AUGUSTUS gene  predicted:not satisfact~ 640374
##  9 OX457036.1:2107471-2108487   AUGUSTUS gene  predicted:not satisfact~ 3.24628~
## 10 OX457036.1:2108840-2109808   AUGUSTUS gene  predicted: Candidozyma ~ 1.27857~
## # i 79 more rows
## # i abbreviated name: 1: `predicted:not satisfactory p-value`
```

In deze fase hadden we voorspellingen(Hits) voor 92 genen op een 1mb chromosoom (tussen 2mb en 3mb), zelfs met enkele genen die niet zo'n goede eval-waarden hadden.

```
colnames(all_predictions ) <- c("id","source","feature", "predicted", "eval")
```

```
all_predictions $eval <- parse_number(all_predictions $eval)


df.f.pavlue <-  all_predictions %>% filter(eval<= 1e-4) %>%  filter(eval!=0)
head(df.f.pavlue)
```

```
## # A tibble: 6 x 5
##   id                         source   feature predicted                     eval
##   <chr>                      <chr>    <chr>   <chr>                        <dbl>
## 1 OX457036.1:2087020-2087265 AUGUSTUS gene    predicted: Lumbricus rub~ 7.38e- 7
## 2 OX457036.1:2089471-2089899 AUGUSTUS gene    predicted:Lampetra plane~ 8.69e-99
## 3 OX457036.1:2106048-210639  AUGUSTUS gene    predicted:Mus musculus c~ 6.12e-10
## 4 OX457036.1:2108840-2109808 AUGUSTUS gene    predicted: Candidozyma a~ 1.28e-22
## 5 OX457036.1:2108840-2109808 AUGUSTUS gene    predictied: Phaeodactylu~ 2.82e-13
## 6 OX457036.1:2112894-2113442 AUGUSTUS gene    predicted:  Ixodes scapu~ 1.27e-14
```

```
write.table( df.f.pavlue, +
               "lumbricus/identification/prediction/df.filtered.txt",sep="\t")
```

```
predictions <-read.table("lumbricus/identification/prediciton/df.filtered.txt")
```

In de daaropvolgende fase hebben we een eval, evaluatiedrempel van 1e-4 ingesteld, wat redelijk mild is.Na het filteren van de voorspellingen met ongunstige eval-waarden, hebben we 32 voorspellingen gevonden die betrekking hebben op 32 genen voor een 1 Mb segment van het eerste chromosoom, wat 1% van het totale chromosoom is. De uiteindelijke voorspelling voor het fragment dat we onderzoeken, is als volgt.

predicition:

```
table7 <-  predictions   %>%  select(V7)

table7 %>%

  kable("html") %>%

  kable_styling(font_size = 7)
```

V7

predicted: Lumbricus rubellus mt2A gene for metallothionein 2A, exons 1-4;AJ299434.1;

predicted:Lampetra planeri genome assembly, chromosome: 62; emb OZ078387.2

predicted:Mus musculus chromosome 8, clone RP23-339I14, complete sequence;AC121136.11

predicted: Candidozyma auris strain BA03 chromosome; 1 eval; CP157508.1

predictied: Phaeodactylum tricornutum CCAP 1055/1 predicted protein partial mRN;XM_002176960.1

predicted: Ixodes scapularis G-protein coupled receptor dmsr; XM_029969893.4

predicted :Melanogrammus aeglefinus genome assembly, chromosome: 10; emb OZ180142.1

predicted : Earthworm (L.terrestris) extracellular globin chain c gene, complete cds; gb J05161.1 LUMHBC

predicted:Zymobacter palmae IAM14233 DNA, complete genome;dbj|AP018933.1

predicted: Hylaeus volcanicus uncharacterized LOC128877144 (LOC128877144), transcript variant X5, mRNA;XM_054124195.1

predicted:Mus musculus BAC clone RP23-95F15 from chromosome 1, complete sequence;AC165443.5

predicted:4_Tte_b3v08;emb|OE003277.1

predicted:Earthworm (L.terrestris) extracellular globin chain c gene, complete cds;J05161.1 LUMHBC

predicted: XM_009033761.1| Helobdella robusta hypothetical protein mRNA

predicted:XM_069820523.1| PREDICTED: Periplaneta americana carbonic anhydrase beta (CAHbeta), transcript variant X3, mRNA

preddicted:Loxodonta africana zinc finger protein 252-like (LOC100666328), transcript variant X4, mRNA

predicted:gb|KX592814.1| Bos taurus isolate Dominette_000065F genomic sequence

predicted: gb|J05161.1|LUMHBC Earthworm (L.terrestris) extracellular globin chain c gene, complete cds

predicted:ref|XM_637462.1| Dictyostelium discoideum AX4 hypothetical protein (DDB_G0277655) mRNA, complete cds

predicted:Rattus norvegicus uncharacterized LOC134482949 (LOC134482949), ncRNA

Melanogrammus aeglefinus genome assembly, chromosome: 13

predicted:PREDICTED: Portunus trituberculatus putative uncharacterized protein DDB_G0271982 (LOC123514901), partial mRNA

predicted:emb|LN021320.1| Spirometra erinaceieuropaei genome assembly S_erinaceieuropaei ,scaffold SPER_scaffold0020968

predicted: gb|L12688.1|LUMBT Earthworm DNA sequence

prediction:emb|OZ078459.1| Lampetra fluviatilis genome assembly, chromosome: 56

emb|OZ180149.1| Melanogrammus aeglefinus genome assembly, chromosome: 17

predicted: ef|NC_043824.1|;Passiflora obovata chloroplast, complete genome;gb|MK694931.1|

predicted:ref|XM_005559078.4;Macaca fascicularis piggyBac transposable element derived 4 (PGBD4), mRNA

predicted:emb|OE179951.1| 2_Tcm_b3v08

predicted:emb;BX544872.8;Zebrafish DNA sequence from clone DKEY-58L12 in linkage group 3, complete sequence

predicted:XM_023356947.1;Centruroides sculpturatus uncharacterized LOC111615539 (LOC111615539), mRNA

predicted:XM_066083420.1| PREDICTED: Magallana gigas retrovirus-related Pol polyprotein from transposon 412 (LOC105343682), mRNA

For more details,see Voor meer informatie, kijk in de map identification, prediction.xlsx, sheet “df_fitlered”.

5.2 Visualisatie

5.3 GenViz

Voor het voorbereiden van de data kun je de volgende bestanden bekijken: genviz-features.py, map visualisatie en GenomeViz. De genen die zijn gevonden, worden weergegeven in grafieken, met speciale aandacht voor de eerste 2-3 megabases van chromosoom 1 (coördinaten 2000000-3000000).

Om te scrollen door de features, kun je de webversie gebruiken:

https://alenagrrr3.github.io/2-3mb-terrsetris/

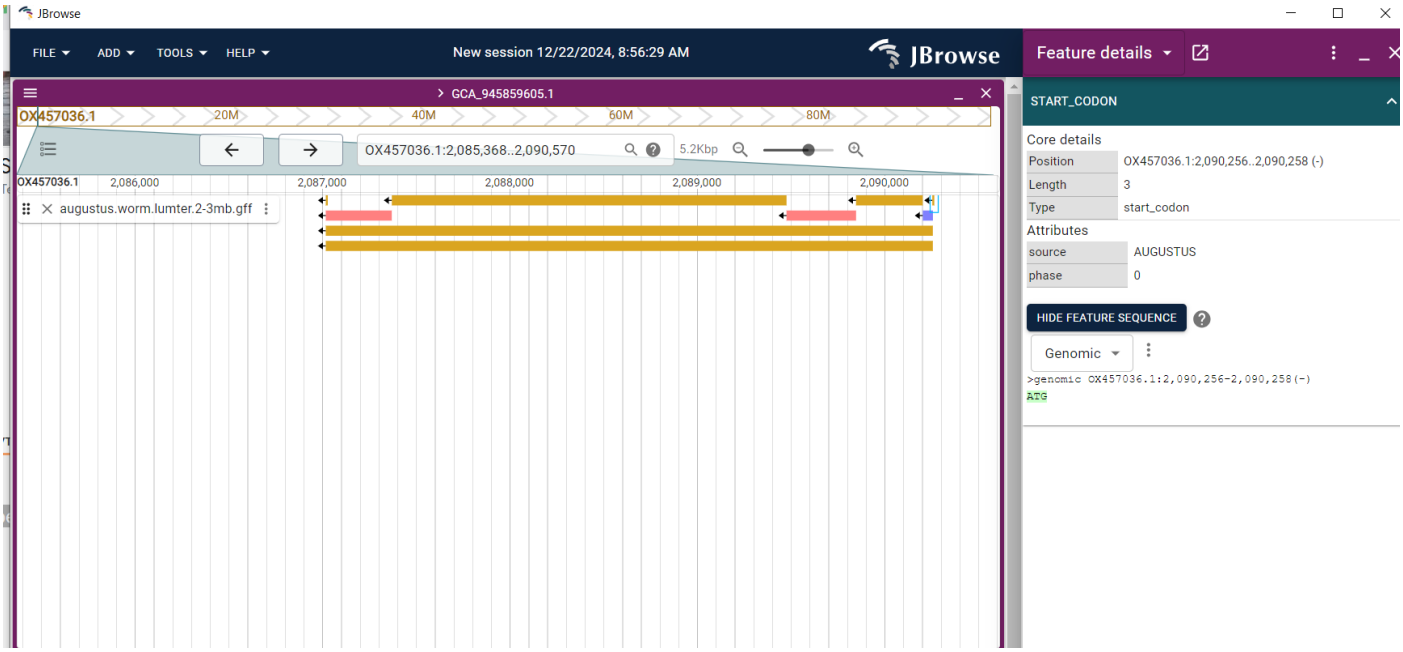
De totale representatie van het chromosoom /OX457036.1.

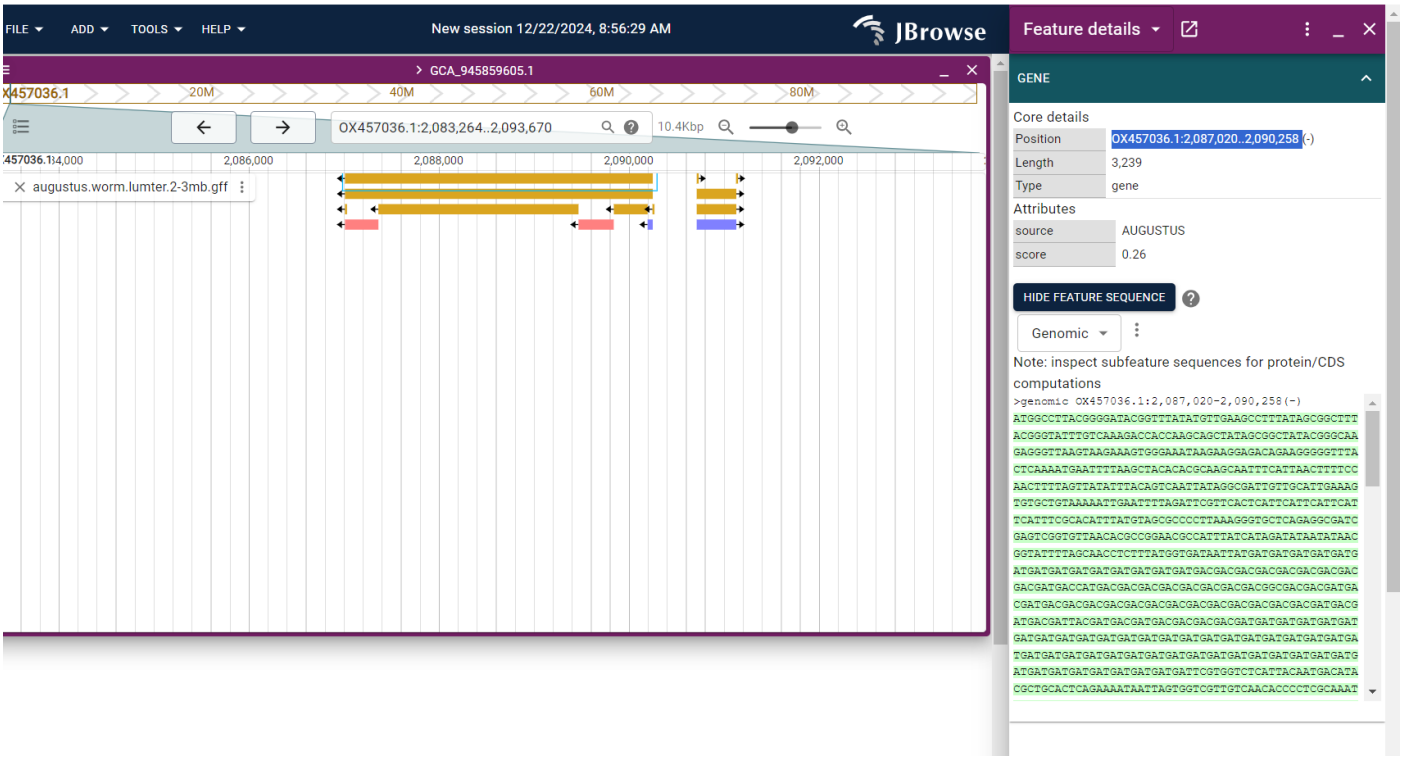
https://alenagrrr3.github.io/OX457036.1.html/

5.4 JBrowse

Het gen met de coördinaten OX457036.1:2,087,020 - 2,090,258 is geïdentificeerd als het mt2A-gen voor metallothioneïne 2A van Lumbricus rubellus, inclusief exons 1-4; AJ299434.1. is onderzocht in de in Jbrowser (“JBrowse | JBrowse” n.d.)

Gene 5, with intron, Cds, and transctipt:



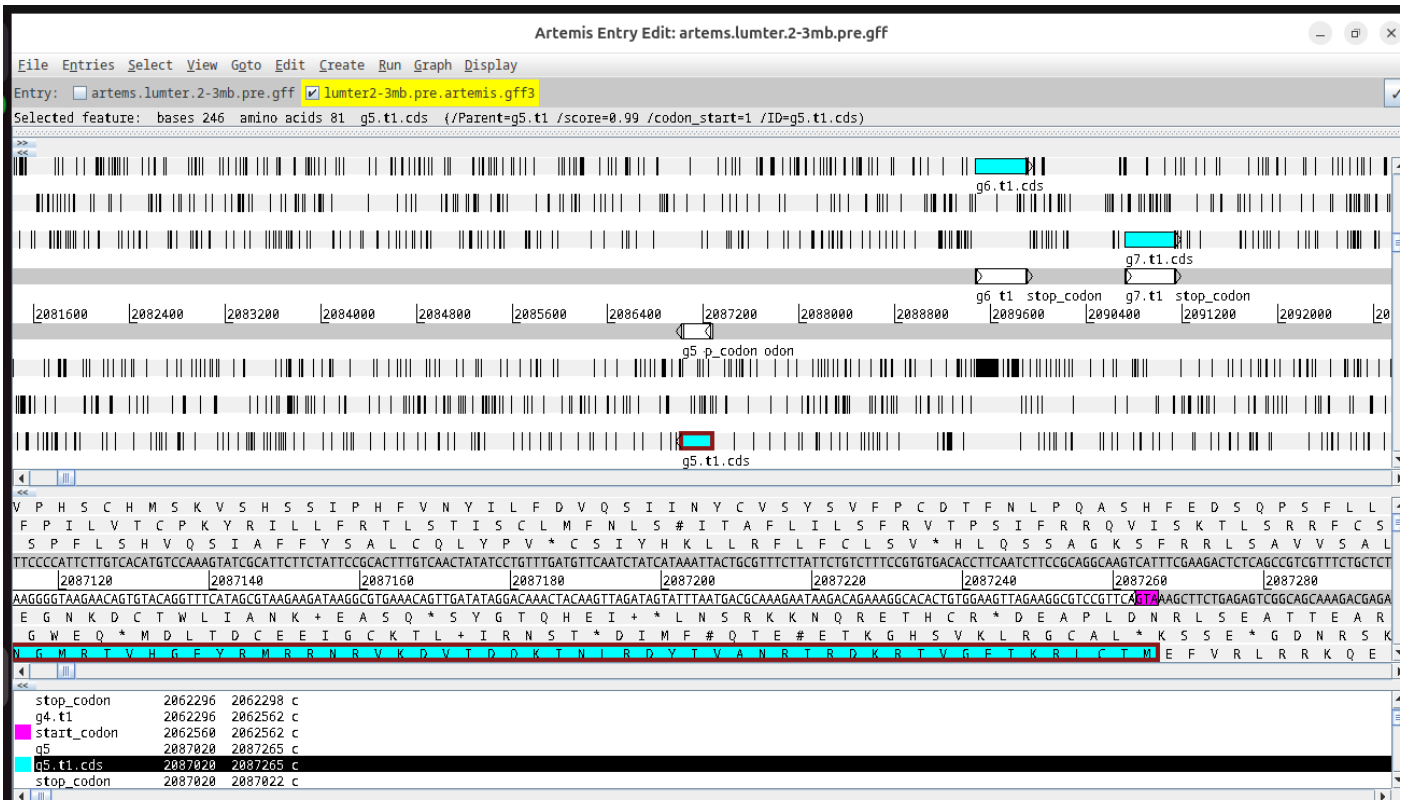


To zoom in, you can ues the link:

https://raw.githubusercontent.com/alenagrrr3/OX457036.1.html/refs/heads/main/lumterAM182481.1-gene5.svg

5.5 Artemis

```
gen "g5" (OX457036.1:2,087,020 - 2,090,258 )   in Artemis Browser  met startcodon en CDS (minus streng):
```



To zoom in, you can ues the link:

https://raw.githubusercontent.com/alenagrrr3/OX457036.1.html/refs/heads/main/artemis-g5-startcodon.webp

References

"Augustus." n.d. Bioinformatics Notebook. Accessed November 25, 2024. https://rnnh.github.io/bioinfo-notebook/docs/augustus.html.

"Augustus/Docs/RUNNING-AUGUSTUS.md at Master · Gaius-Augustus/Augustus." n.d. GitHub. Accessed November 25, 2024. https://github.com/Gaius-Augustus/Augustus/blob/master/docs/RUNNING-AUGUSTUS.md.

Baum, Dr Julia. n.d. "Ever Thought about Earthworms?" African Wildlife Economy Institute. Accessed November 25, 2024. https://www0.sun.ac.za/awei/articles/ever-thought-about-earthworms.

"Bioinformatics and Other Bits - Creating a Local RefSeq Protein Blast Database." n.d. Accessed November 28, 2024. https://dmnfarrell.github.io/bioinformatics/local-refseq-db.

"Bioinformatics Web Server - University of Greifswald." n.d. Accessed December 18, 2024. https://bioinf.uni-greifswald.de/bioinf/partitioned_odb11/.

Blaxter, Mark L., David Spurgeon, and Peter Kille. 2023. "The Genome Sequence of the Common Earthworm, Lumbricus Terrestris (Linnaeus, 1758)." *Wellcome Open Research* 8 (October): 500. https://doi.org/10.12688/wellcomeopenres.20178.

1.

Brůna, Tomáš, Katharina J Hoff, Alexandre Lomsadze, Mario Stanke, and Mark Borodovsky. 2021. "BRAKER2: Automatic Eukaryotic Genome Annotation with GeneMark-EP+ and AUGUSTUS Supported by a Protein Database." *NAR Genomics and Bioinformatics* 3 (1): lqaa108. https://doi.org/10.1093/nargab/lqaa108.

Buchfink, Benjamin, Chao Xie, and Daniel H Huson. 2015. "Fast and Sensitive Protein Alignment Using DIAMOND." *Nature Methods* 12 (1): 59–60. https://doi.org/10.1038/nmeth.3176.

colauttilab.github.io/. n.d. "De Novo Assembly Tutorial." Accessed November 30, 2024. https://colauttilab.github.io/NGS/deNovoTutorial.html.

ebi.ac.uk. n.d. "ENA Browser." Accessed November 25, 2024. https://www.ebi.ac.uk/ena/browser/view/PRJEB59400.

"ENA Browser." n.d. Accessed December 18, 2024. https://www.ebi.ac.uk/ena/browser/view/ERR10851549.

Erxleben, Anika, and Björn Grüning. 12:19:56 +0000. "Genome Annotation." Text. Galaxy Training Network; Galaxy Training Network. 12:19:56 +0000. https://translated.turbopages.org/proxy_u/en-ru.ru.dd5ab9ec-67446c58-5ab2c0cb-74722d776562/https/training.galaxyproject.org/training-material/topics/genome-annotation/tutorials/genome-annotation/tutorial.html.

"Gaius-Augustus/BRAKER." (2018) 2024. Gaius-Augustus. https://github.com/Gaius-Augustus/BRAKER.

"Gene Cluster Visualizations in R." n.d. Accessed November 27, 2024. https://nvelden.github.io/geneviewer/.

"Genome Annotation / Tutorial List." 13:32:22 +0000. Text. Galaxy Training Network; Galaxy Training Network. 13:32:22 +0000. https://training.galaxyproject.org/training-material/topics/genome-annotation/.

"Home · TransDecoder/TransDecoder Wiki." n.d. Accessed November 28, 2024. https://github.com/TransDecoder/TransDecoder/wiki.

"Index of /Genomes." n.d. Accessed November 28, 2024. https://ftp.ncbi.nlm.nih.gov/genomes/.

"Index of /Genomes/MapView." n.d. Accessed November 28, 2024. https://ftp.ncbi.nlm.nih.gov/genomes/MapView/.

"Index of /Releases/Dfam_3.8/Families/FamDB/." n.d. Accessed December 18, 2024. https://www.dfam.org/releases/Dfam_3.8/families/FamDB/.

"JBrowse | JBrowse." n.d. Accessed November 26, 2024. https://jbrowse.org/jb2/.

Leung, Maxwell C. K., Phillip L. Williams, Alexandre Benedetto, Catherine Au, Kirsten J. Helmcke, Michael Aschner, and Joel N. Meyer. 2008. "Caenorhabditis Elegans: An Emerging Model in Biomedical and Environmental Toxicology." *Toxicological Sciences* 106 (1): 5–28. https://doi.org/10.1093/toxsci/kfn121.

"LumbriBASE." n.d. Accessed November 30, 2024. http://xyala2.bio.ed.ac.uk/Lumbribase/lumbribase_php/lumbribase.shtml.

ncbi.nlm.nih.gov. n.d. "The NCBI Eukaryotic Genome Annotation Pipeline." Accessed November 25, 2024. https:

//www.ncbi.nlm.nih.gov/refseq/annotation_euk/process/.

Pilato, Giovanni. n.d. "The significance of musculature in the origin of the Annelida." Accessed November 30, 2024. http://ouci.dntb.gov.ua/en/works/ldperODl/.

"Sanger-Pathogens/Artemis." (2009) 2024. Pathogen Informatics, Wellcome Sanger Institute. https://github.com/sanger-pathogens/Artemis.

Short, Stephen, Amaia Green Etxabe, Alex Robinson, David Spurgeon, and Peter Kille. 2023. "The Genome Sequence of the Red Compost Earthworm, Lumbricus Rubellus (Hoffmeister, 1843)." *Wellcome Open Research* 8 (August): 354. https://doi.org/10.12688/wellcomeopenres.19834.1.

Stanke, Mario. 2005. "Augustus Online." Service. Institute for Mathematics and Computer Science, University of Greifswald. February 4, 2005. https://bioinf.uni-greifswald.de/augustus/submission.php.

"The Genome Sequence of the Common ... | Wellcome Open Research." n.d. Accessed December 19, 2024. https://wellcomeopenresearch.org/articles/8-500.

University of Greifswald. n.d. "Bioinformatics Web Server - University of Greifswald." Accessed December 28, 2024. https://bioinf.uni-greifswald.de/bioinf/partitioned_odb11/.