Functional Annotation

Results

Team 1: Jiyeong Choi
Asmita Kishor Lagwankar
Chloe Pryor
Hannah Snyder
Likitha Venkatesh
Jiahong Zhang

Types of Functional Annotation

Ab-Initio

- *Ab initio* - “from the beginning”

- No external evidence is available to identify a gene
- Mathematical models
- Does not need experimental data

Homology Based

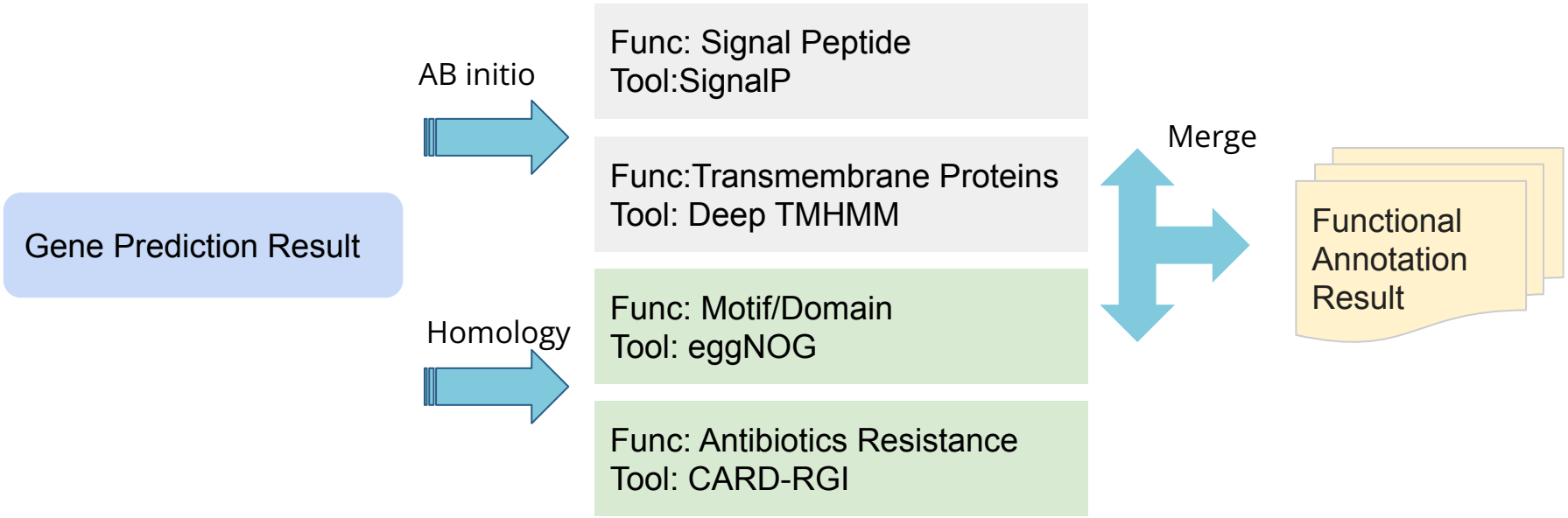
- Evidence Based Annotation

- Rely on comparison between sequences
- Uses information about known structure of related proteins to predict unknown

Functional Annotation Strategy - Recap

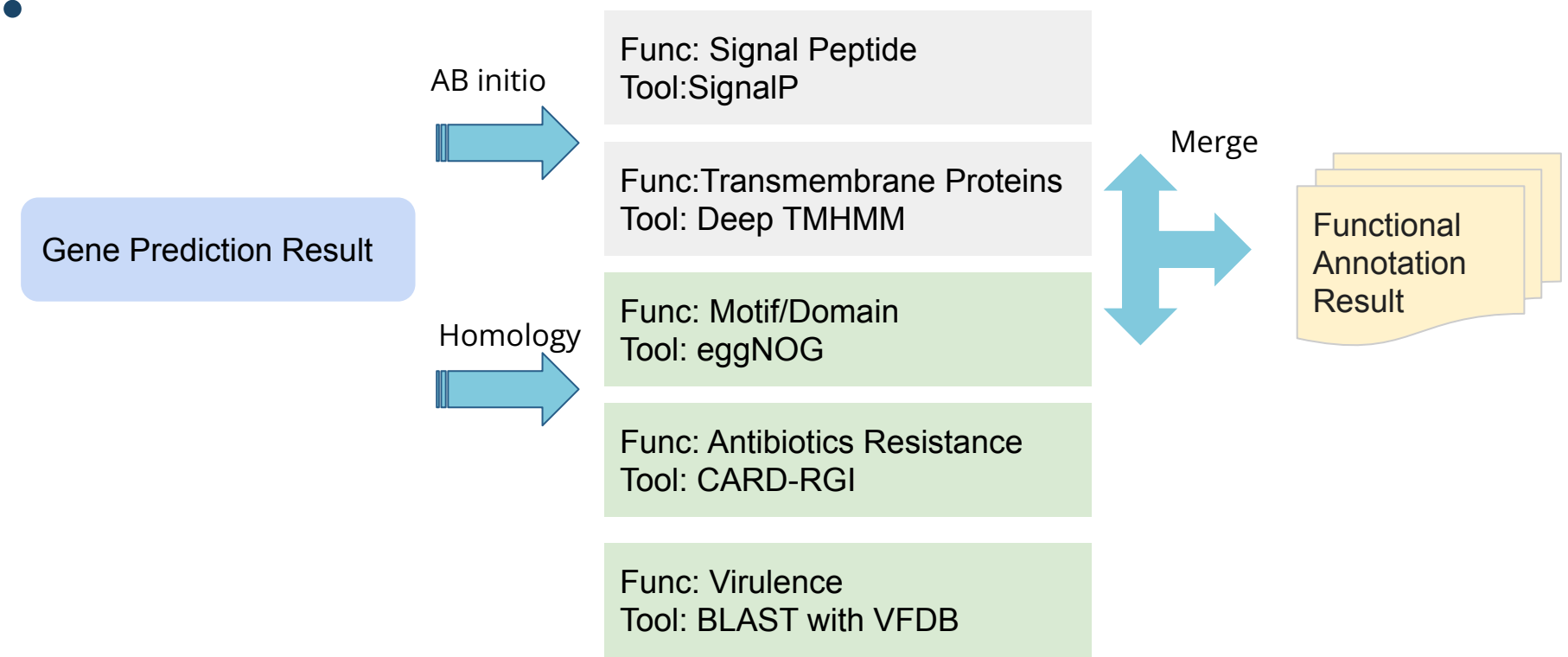
We were going to choose 4 different aspects of functions to predict. (Signal Peptide, Transmembrane Proteins, Motif/Domain and Antibiotics Resistance).

Use 2 homology based methods to predict 2 functions and use 2 ab initio methods to predict other 2 functions.



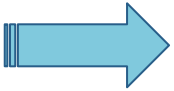
Functional Annotation Strategy - Updates

We updated our methods to cover 5 different aspects of functions to predict. (Signal Peptide, Transmembrane Proteins, Motif/Domain and Antibiotics Resistance, **Virulence**).



Final Work Delegation

AB initio



Homology

Func: Signal Peptide Tool:SignalP	Likitha Venkatesh
Func:Transmembrane Proteins Tool: Deep TMHMM	Asmita Lagwankar
Func: Motif/Domain Tool: eggNOG	Jiyeong Choi
Func: Antibiotics Resistance Tool: CARD-RGI	Jiahong Zhang
Func: Virulence Tool: BLAST with VFDB	Chloe Pryor

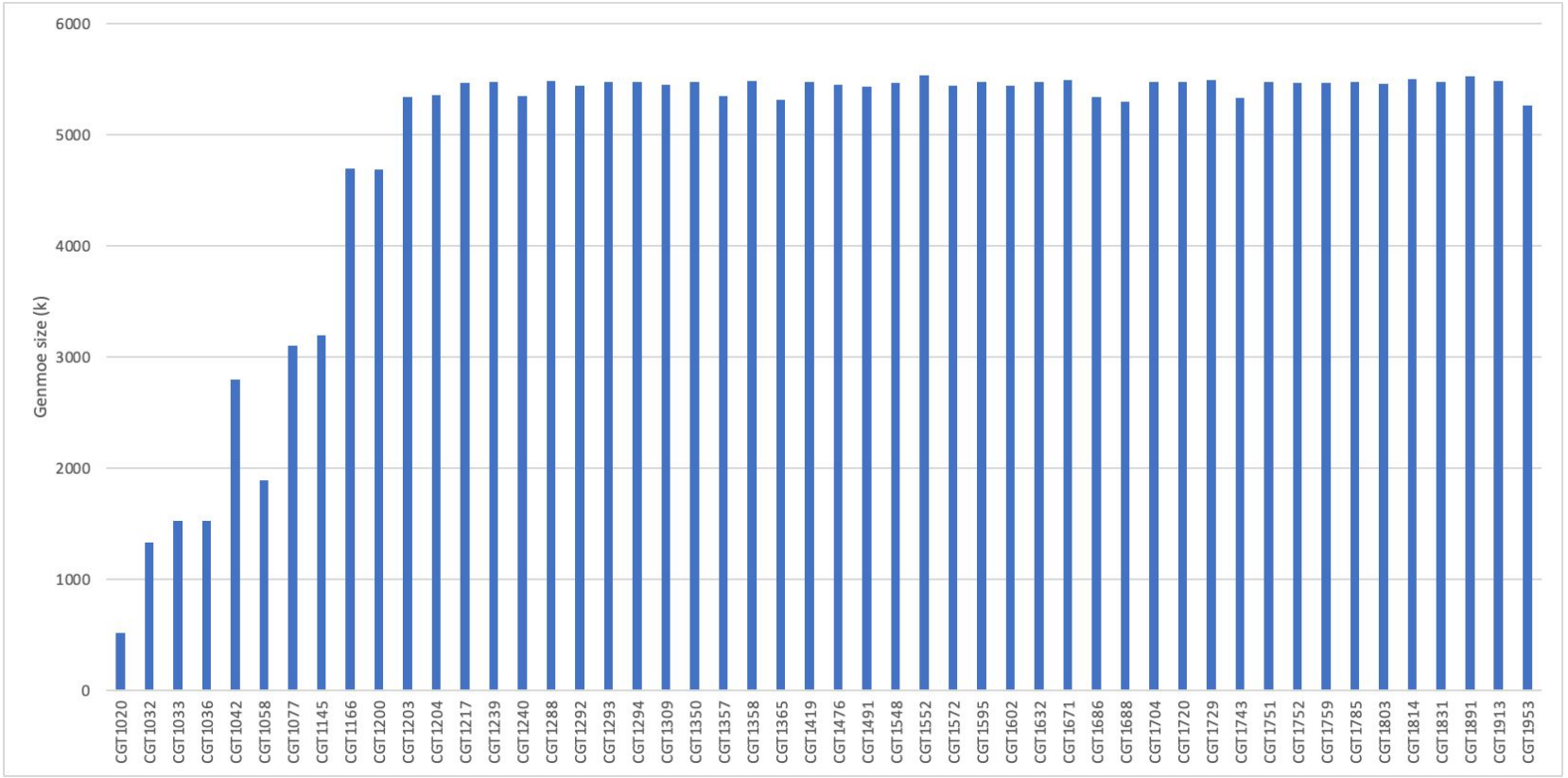
Hannah Snyder

Functional Annotation Result

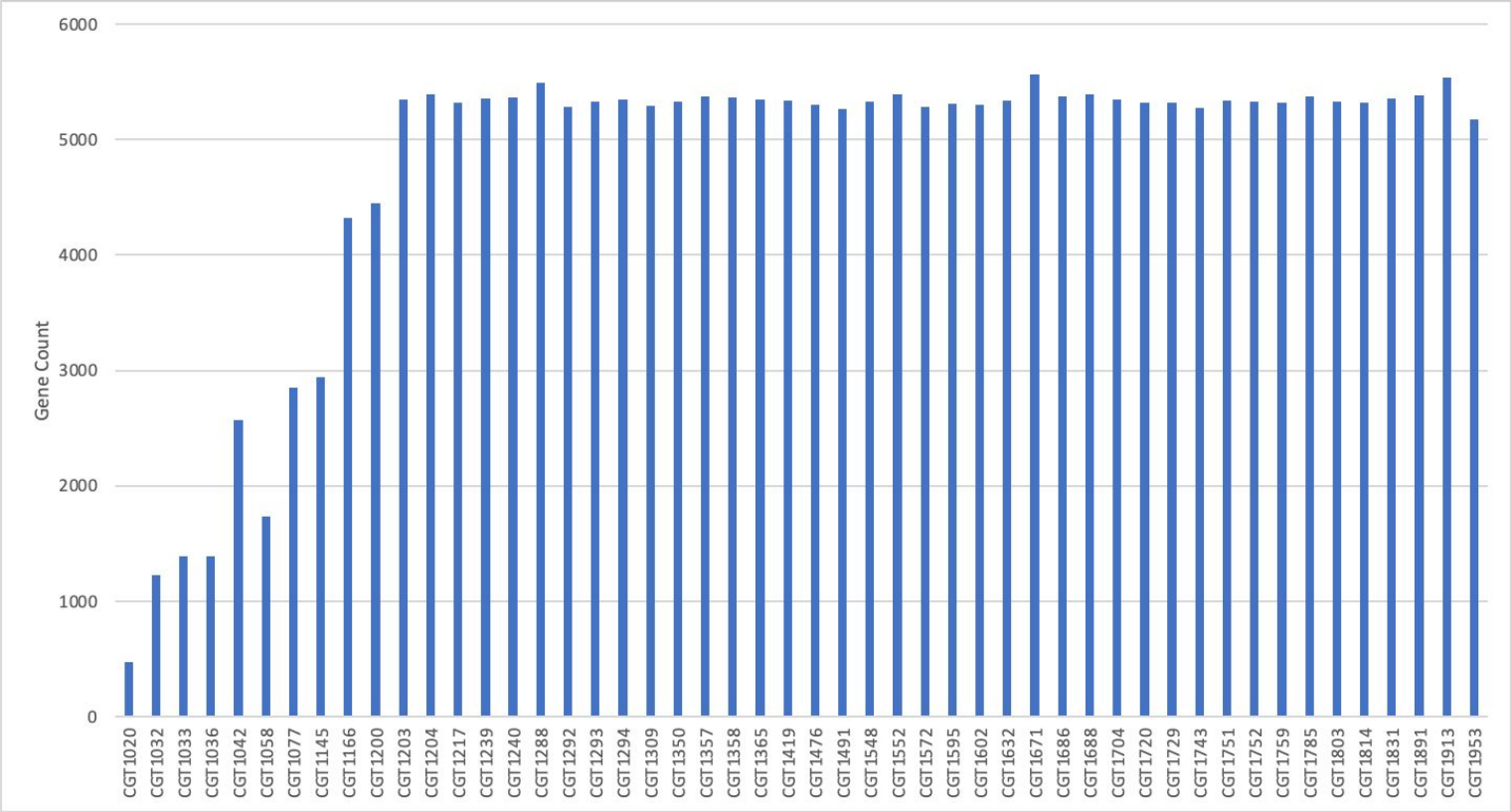
Jiyeong Choi

GitHub Readme

Genome Size per contig



Gene count per contig



SignalPv6

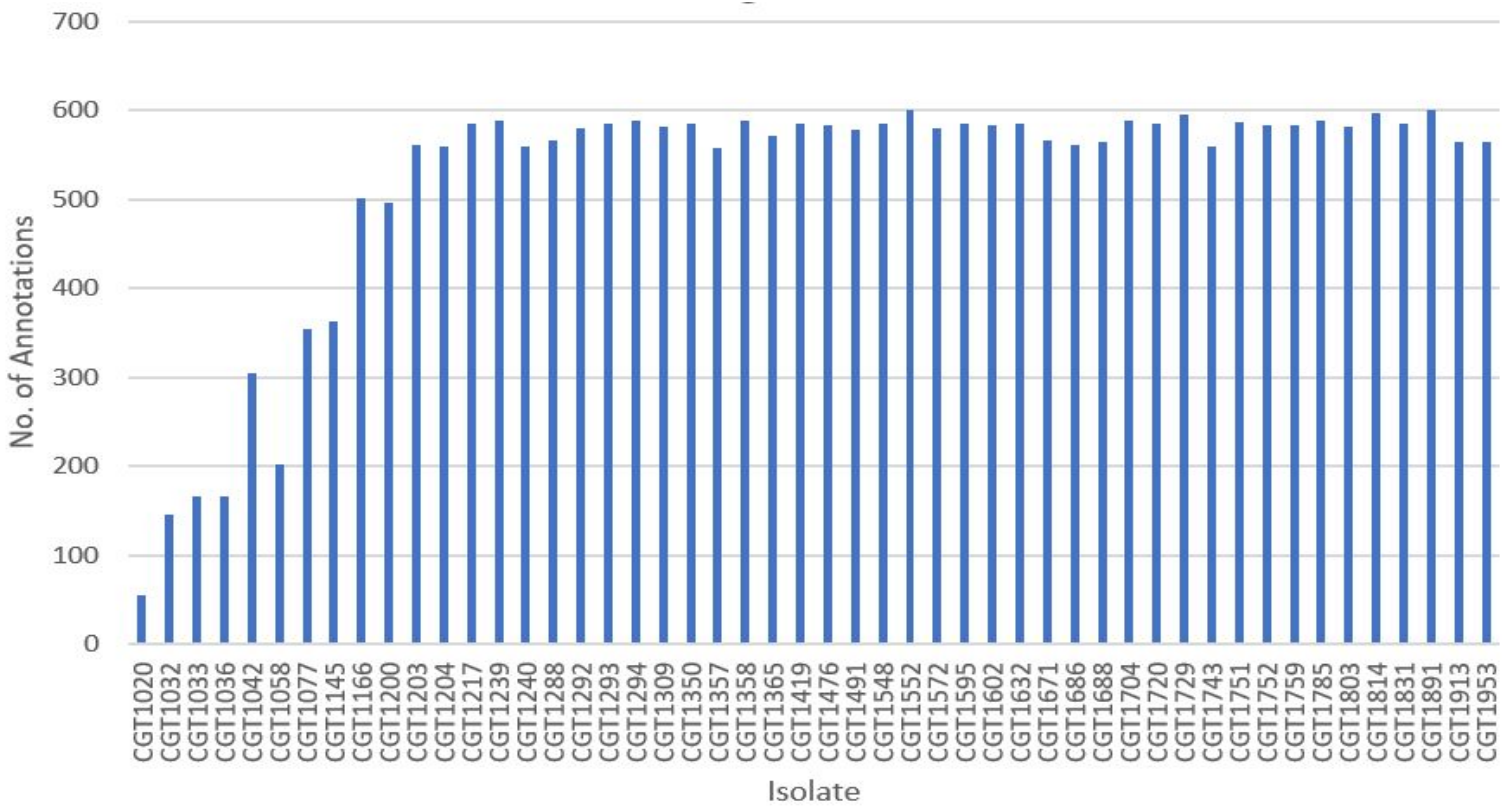
- Command: signalp6 --fastafile <input_file> --organism <organism type> --output_dir <output directory> --format <output_format> --mode <mode_type> --write_procs 8

- Runtime was ~3-4 minutes for each isolate

- Output Files:
  - Predicted_results.txt
  - Output.gff3
  - Single sequence files

- Mode could be specified as slow or fast

SignalPv6


700
600
500
400
300
200
100
0
No. of Annotations
CGT1020
CGT1032
CGT1033
CGT1036
CGT1042
CGT1058
CGT1077
CGT1145
CGT1166
CGT1200
CGT1203
CGT1204
CGT1217
CGT1239
CGT1240
CGT1288
CGT1292
CGT1293
CGT1294
CGT1309
CGT1350
CGT1357
CGT1358
CGT1365
CGT1419
CGT1476
CGT1491
CGT1548
CGT1552
CGT1572
CGT1595
CGT1602
CGT1632
CGT1671
CGT1686
CGT1688
CGT1704
CGT1720
CGT1729
CGT1743
CGT1751
CGT1752
CGT1759
CGT1785
CGT1803
CGT1814
CGT1831
CGT1891
CGT1913
CGT1953
Isolate

Deep TMHMM

Command :

biolib run DTU/DeepTMHMM --fasta input.fasta

Output:

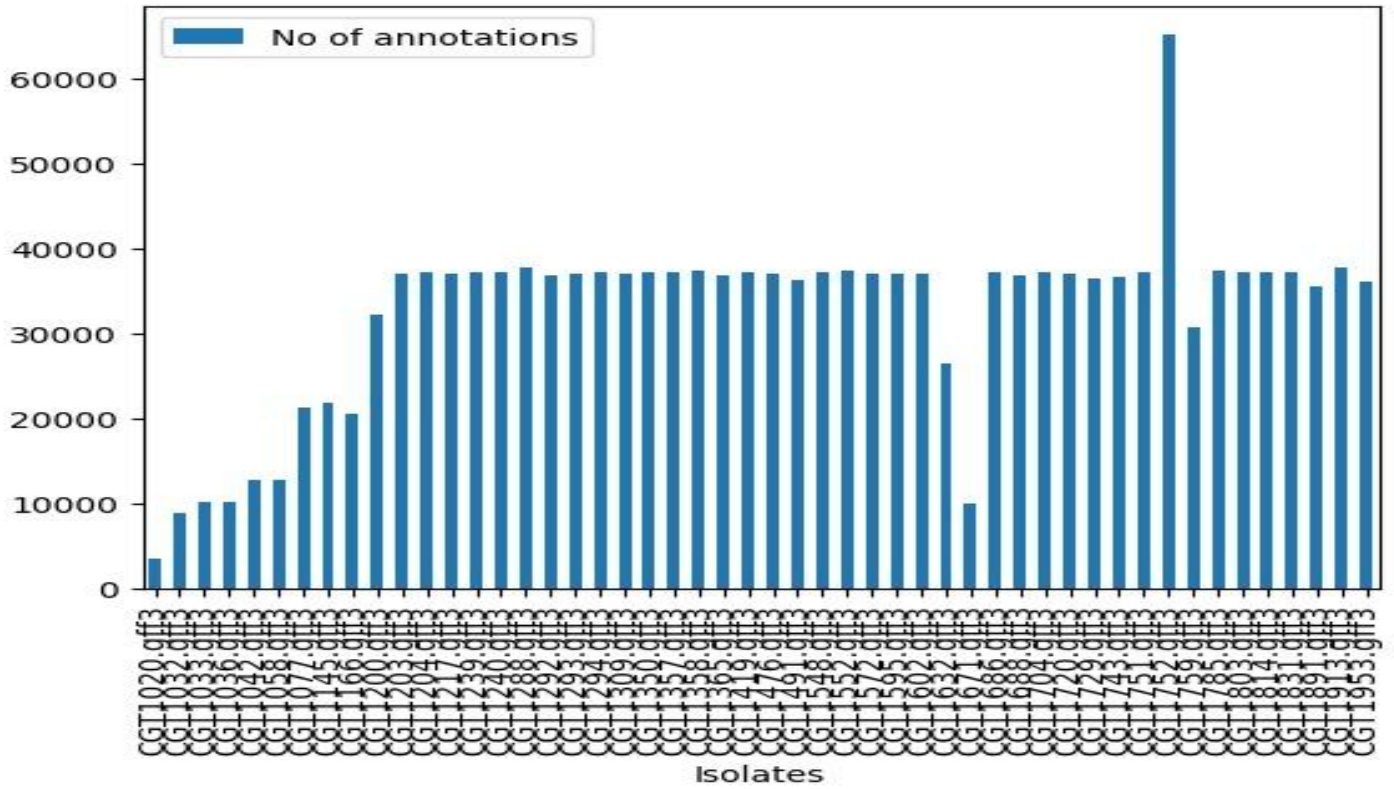
- A gff file with all TMRs
- A predicted topologies file

DeepTMHMM by default is run on the BioLib Cloud.

Predicts the following:

- protein topology prediction tool
- protein secondary structure prediction tool
- protein transmembrane helices prediction
- membrane protein structures
- Signal Peptides

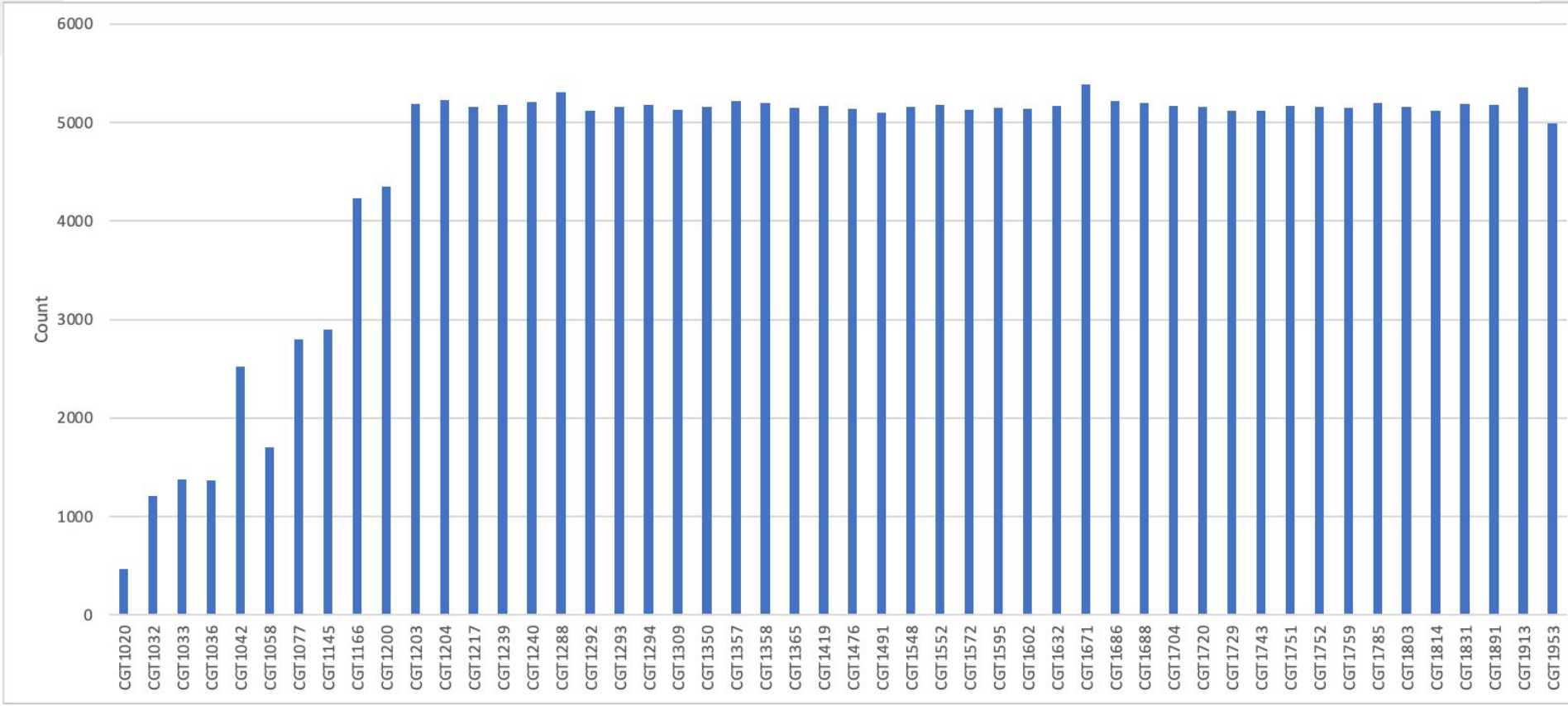
Deep TMHMM


No of annotations
60000
50000
40000
30000
20000
10000
0
Isolates

eggNog-mapper

- Diamond mode was used as it has the best balance of speed and accuracy
- Specified number of CPUs
  - `--cpu 10`
  - Took about 3 hours to generate all output files
- Domain/motif information obtained
  - `--pfam_realign realign`
  - Realigning queries to the PFAM domains found on the orthologous groups
- 4 output files were generated for each input
  - *.emapper.annotation
  - *.emapper.hits
  - *.emapper.pfam
  - *.emapper.seed_orthologous

eggNog-mapper



Antibiotics Resistance - CARD RGI

Command: `rgi main --input_sequence /path/to/*.faa --output_file /path/to/result --local --clean -t protein`

```
  --clean               removes temporary files
  -t {contig,protein}, --input_type {contig,protein}
```

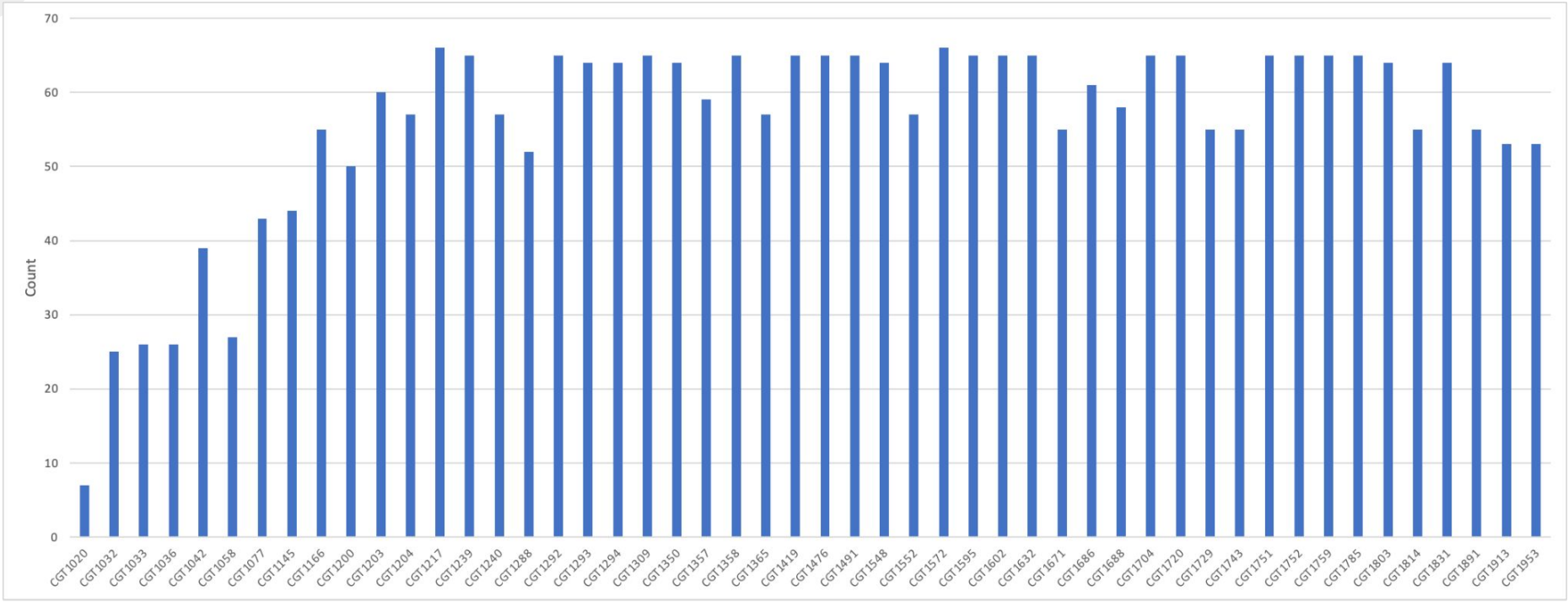
Key results:

Best_Hit_Bitscore	Bitscore value of match to top hit in CARD
Best_Hit_ARO	ARO term of top hit in CARD
Best_Identities	Percent identity of match to top hit in CARD

Drug Class	ARO Categorization
Resistance Mechanism	ARO Categorization
AMR Gene Family	ARO Categorization

Output: json & txt

Antibiotics Resistance - CARD RGI



Virulence - BLAST with VFDB

Ran a BLAST search with Virulence Factor Database and converted output with MGKit

44,502 annotations for all genomes relatively quickly

making the blast database

makeblastdb -in /home/team1/annotation/vfdb_database/VFDB_setA_pro.fas -parse_seqids -dbtype prot -out /home/team1/annotation/vfdb_output/vfdb_prot

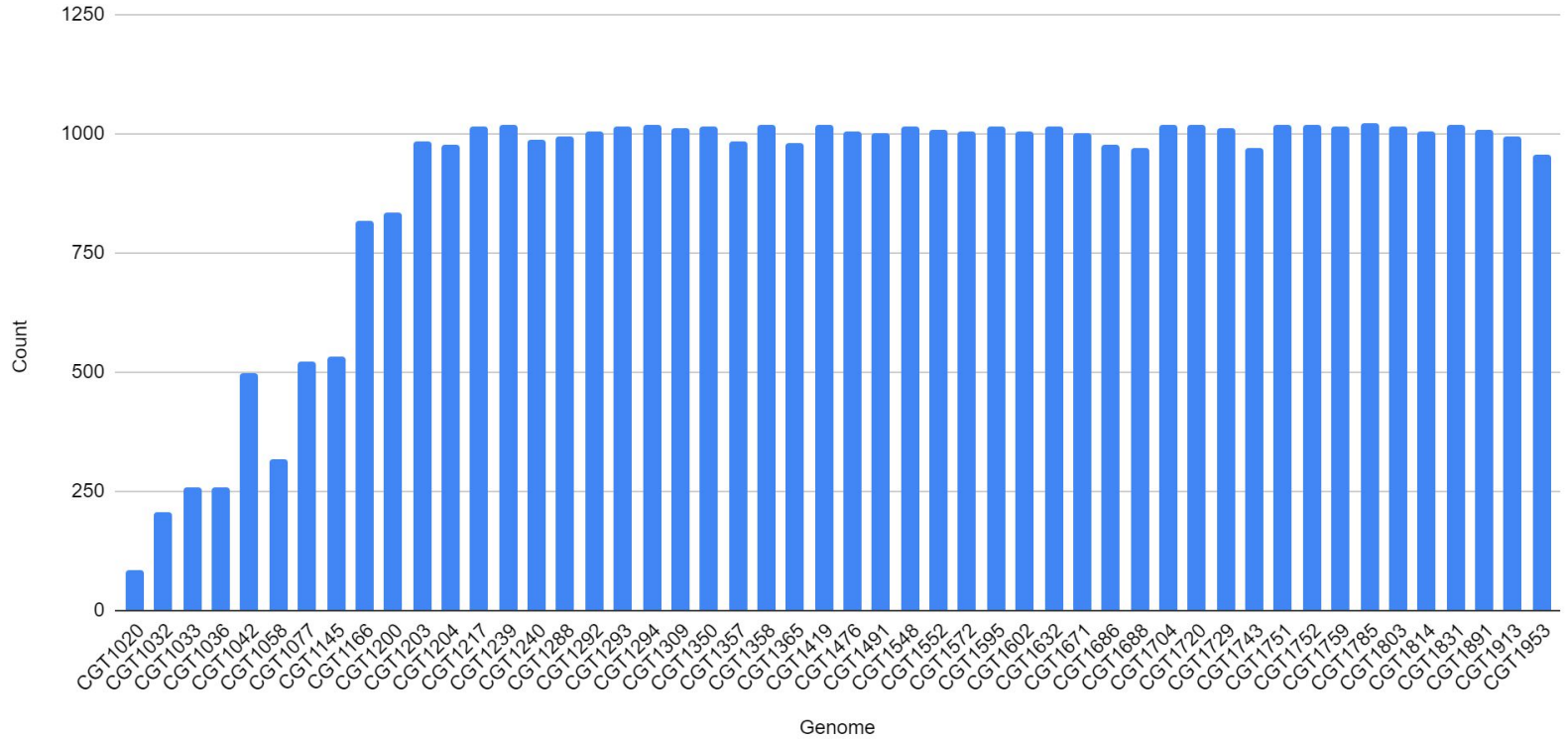
generating the blast output

for i in CGT*.faa; do blastp -db vfdb_prot -query $i -out “${i}_out” -outfmt 6 -num_threads 4 -evalue 1e-10 -max_hsps 1 -max_target_seqs 1; done

converting blast output to gff files

for i in CGT*.faa_out; do blast2gff blastdb $i “${i%_*}.gff”; done

Virulence - BLAST with VFDB



Tool Comparison

- Just for 5 test files that were the smallest

	CPU	Time
SignalP	10.0%	3-4 mins
TMHMM	13%	17-18 mins
eggNOG	9.97%	32.55 mins
CARD-RGI	9.96%	54.6s
Blast	9.95%	42.25s

(Also in .tsv file)

Merged GFF File

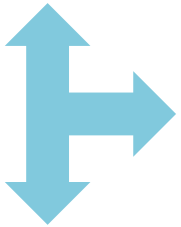
Func: Signal Peptide
Tool: SignalP

Func:Transmembrane Proteins
Tool: Deep TMHMM

Func: Motif/Domain
Tool: eggNOG

Func: Antibiotics Resistance
Tool: CARD-RGI

Func: Virulence
Tool: BLAST with VFDB



Merged gff Files

Information to pass on:

Pipeline: /home/team1/annotation/annotation_pipeline.py

Readme: GitHub! (https://github.gatech.edu/comgenomics2023/Team1-FunctionalAnnotation/blob/main/README.md)

Pathway for isolate CGT1020 Example:

- Merged gff file: /home/team1/annotation/merge/output/CGT1020
- Output of all tools: /home/team1/annotation/merge/CGT1020

All files are in the server

Questions?