# Gene Prediction - Results

Team 2 (Group 2):
Gautham Krishna Sankar Ramalaxmi, Jyothi Guruprasad,
Kaize H Ali, Lee Ellen Mullins, Varsha Srinivasan

# Gene Prediction Background

## Homology based prediction

- Finding similarity in gene sequences between the target genome and other sequences.
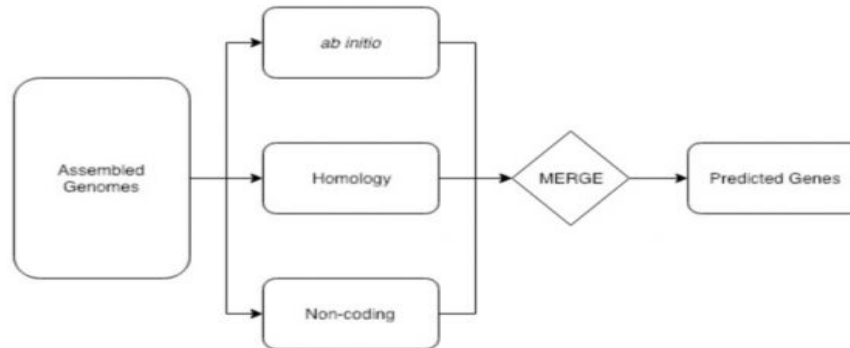
## Ab-initio prediction

- Finding protein sequences using sequence information alone
- Does not use homolog (template) structure

# Proposed Pipeline

Tools available:

- Ab-initio : GeneMarkS2, GLIMMER, Prodigal
- Homology-based : BLAST
- Merging results : Bedtools intersect, ORForise

# GeneMarkS2

- Ab-initio algorithm that finds species-specific genes to identify other genes
- Software is built using Hidden Markov Model (HMM)
- Robust method of detection
- Latest iteration was developed to increase the prediction accuracy of *where* the gene starts

Modeling leaderless transcription and atypical genes results in more accurate gene prediction in prokaryotes

Alexandre Lomsadze,[1,2,6] Karl Gemayel,[3,6] Shiyuyun Tang,[4] and Mark Borodovsky[1,2,3,4,5]

# GeneMarkS2

Command line argument:

```
perl ../../GeneMarkS2/gms2.pl --genome-type bacteria --gcode 11 --output gms2_$FILE_NAME --format gff3 --gid 'gene_' --seq ../../Contigs/$FILE_NAME_FASTA.gff
```

Input parameters:

- **--genome-type** : describes type of input genome (auto/bacteria/archaea)
- **--gcode** : 11 - indicates that it is a bacteria
- **--format** : tells the output file format
- **--seq** : input file sequence (either .fna or .fasta format)
- **--output** : tells the output file name

# GeneMarkS2

Output obtained:

# GLIMMER

- Find genes in prokaryotic DNA
- 1st system to use interpolated Markov model (IMM) to identify coding regions
- Has 2 built-in programs:
  - Build IMM - takes an input of many sequences, output is an IMM
  - Glimmer - using the above IMM, identifies the presumes gene
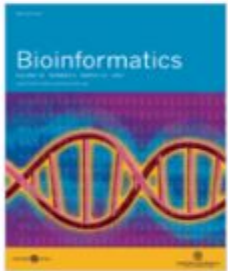
# GLIMMER

Command line argument:

```
$glimmerpath/glimmer3 $glimmeropts -b $tag.motif -P $startuse $genome $tag.icm $tag
```

Input parameters:

- **-‑options** : can be gc_percent/codon list/gene length etc
- **-‑sequence-file** : input file given for analysis
- **Other parameters/flags exist, but is outside the scope of our intended goal**

# GLIMMER

- For our situation, we used g3-iterated.sh, a script that uses g3-from-scratch.sh and g3-from-training.sh
- These pre-configured scripts exists in glimmer3.02/scripts
- Flags used were -o50: minimum nucleotide length set to 50. -g110: minimum distance between adjacent genes. -t30: minimum threshold score is set to 30
- Ran on Macbook Pro M2 with a core of 8 CPU

# Prodigal

- PROkaryotic DYnamic programming Genefinding ALgorithm
- Constructs its training set by examining the GC frame plot
- Uses dynamic programming to identify heavily overlapped ORFs
- Lowers false positives (<5%)

## Prodigal: prokaryotic gene recognition and translation initiation site identification

Doug Hyatt ✉, Gwo-Liang Chen, Philip F LoCascio, Miriam L Land, Frank W Larimer & Loren J Hauser

# Prodigal

Command line argument:

```
prodigal -i "../../../../Contigs/$FILE_NAME_FASTA" -c -m -f gff -o cds_$FILE_NAME.gff 2>&1 | tee log_$FILE_NAME
```

Input parameters:

- **–i** : input file parameter (.fasta format)
- **-o** : name of output file generated
- **-f** : format of output file (.gff format in this case)
- **-c** : closed ends - does not allow partial genes at end of sequence
- **-m** : gap-handling behavior - genes cannot run into gaps

# Prodigal

Output obtained:

# Prodigal Stats



```
    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
3954486 gtg875q   20   0  200072  54292   2276 R  93.8   0.1   0:05.06 prodigal
      1 root      20   0  257112   8464   5164 S   0.0   0.0   5:48.46 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:05.10 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     12 root      20   0       0      0      0 S   0.0   0.0   2:26.37 ksoftirqd/0
     13 root      20   0       0      0      0 I   0.0   0.0  14:19.56 rcu_sched
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.87 migration/0
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.82 watchdog/0
top - 06:53:15 up 25 days,  1:22,  8 users,  load average: 0.92, 0.63, 0.39
Tasks: 258 total,   2 running, 255 sleeping,   0 stopped,   1 zombie
%Cpu(s):  9.9 us,  0.6 sy,  0.0 ni, 89.5 id,  0.0 wa,  0.0 hi,  0.0 si,  0.0 st
MiB Mem :  64146.0 total,  54089.9 free,   2332.1 used,   7724.1 buff/cache
MiB Swap:  16383.0 total,  15928.9 free,    454.1 used.  60854.1 avail Mem

    PID USER      PR  NI    VIRT    RES    SHR S  %CPU  %MEM     TIME+ COMMAND
3954534 gtg875q   20   0  198468  53520   1940 R  93.8   0.1   0:01.01 prodigal
      1 root      20   0  257112   8464   5164 S   0.0   0.0   5:48.46 systemd
      2 root      20   0       0      0      0 S   0.0   0.0   0:05.10 kthreadd
      3 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_gp
      4 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 rcu_par_gp
      6 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 kworker/0:0H-events_highpri
      9 root       0 -20       0      0      0 I   0.0   0.0   0:00.00 mm_percpu_wq
     10 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_rude_
     11 root      20   0       0      0      0 S   0.0   0.0   0:00.00 rcu_tasks_trace
     12 root      20   0       0      0      0 S   0.0   0.0   2:26.37 ksoftirqd/0
     13 root      20   0       0      0      0 I   0.0   0.0  14:19.58 rcu_sched
     14 root      rt   0       0      0      0 S   0.0   0.0   0:00.87 migration/0
     15 root      rt   0       0      0      0 S   0.0   0.0   0:00.82 watchdog/0
```

```bash
#!/bin/bash

# Set the filename for the log file
LOGFILE=tops/prodigal_tops.log

# Clear the log file before we start
> $LOGFILE

# Run top and capture the top 50 lines every 5 seconds
while true; do
    top -b -n 1 | head -n 20 >> $LOGFILE
    sleep 60
done
```

# Performance Comparison

| Parameters/Tools | GeneMarkS2 | GLIMMER | Prodigal |
|---|---|---|---|
| Time | 47m 2.262s in real time 46m 14.530s in user-mode 0m 24.795s in system mode | 3m 39s and in user-mode 1.63 seconds in system (Kernel) mode | 4 min 53.985s in real time 4 min 51.031s inn user-mode 0 min 2.551s in system-mode |
| CPU Capacity | 97% | 98% | 93.8% |

# Tools Comparison

Prodigal vs GeneMarkS2 vs GLIMMER
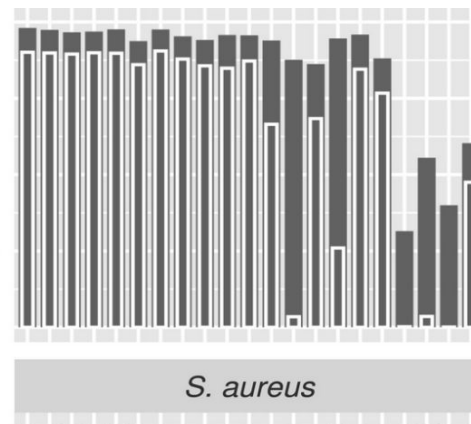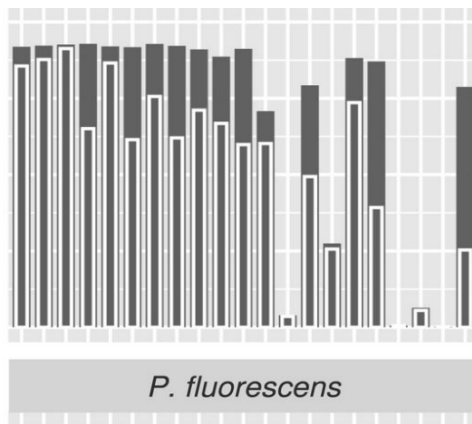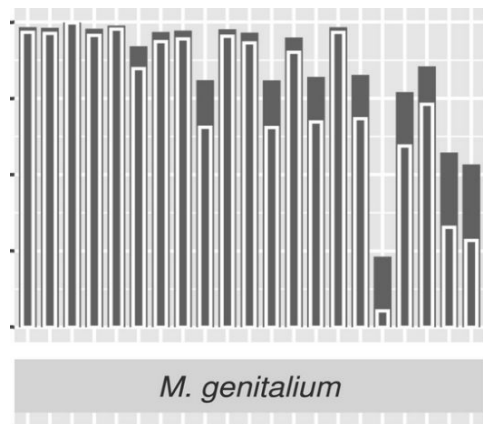


A : Prodigal
B : GeneMarkS −2
C : MetaGeneAnnotator
D : MetaGeneMark
E : GeneMarkS
F : MetaGene
G : GeneMark.hmm / *E.coli* Model
H : GeneMark −HA
I : Augustus / *E.coli* Model
J : FragGeneScan
K : GLIMMER −3

# Bedtools



FASTA → GFF → FASTA

```
>k141_19:106802-108947
TTAACGCCGTCGGCGGTGTGCAGGTTTTGCTTTTGGTTTTTACTATAACGTCTGGCACGGCAGGCTGCTCTATCGGAAAAACAGGTAAGGCATTCAGTAAACGCTGTCCGTAGTTTTTGGTCAACAAACGCTTATCATAAATGACCACTT
CTCCCCGACAGGCGTGGCTACGGATCAATCGCCCTACCTGCTGAATTAAATTAAAAGACGCGCTTGGTAAGCTTTGTACCTCAAACGGATAACGATTCAGGCTTTTTAGCCATTCGCCTTCGGTGATCACGACCGGGCTGTCGATCGGC
GGGAAGGCAATTTTTATGGATATGGACCTGCGTCAGCAGCTCGCCTTTAAGATCCAGCCCTTCGGCAAAAGACTGAAGCCCAACCAGCACGCTGCGTTCGCCGCTCTCCACCCGTTTACGGTGTAGCTCCACCAGCCGGTAACGCGGCTG
GTCGCCCTGGACCAGGAGGAGTAAGCGCACATCAGCGACATGCTCCAGAAAACGCTGCATCGCCCGCCCGCTGGCGAACAGTACCAGCATACCATGATGTTTTTTGCTCTCCAGTTGCTCACGAAAATAGGCCGCCATTTCGGCAATAT
GCTGCTCTTCGTTATCAATGGTGGGTTCGTAGCGCATTTGAGGAATCACCAGTTTCCCCTGTTCCACATGATTAAACGGCGAATCCAGCGCCACAAAGCGGTCGCCGGCTTTTTCTTTTAAGCCGCTCATTTCCTGCAGACGCGAAAAG
CTATTGAGCGAGCGCAGTGTCGCCGAGGTGACAATAATGTGCGGTACGCTGCGCCATAACAATCTTTCCAGCTGATCGCTGCGCGGATACCGACACAATGGAACCAGACGTGGAGTTGCCCTTCGCGTATTTCACGCGTCGCCCATTT
GGACACCGGCGCGCCTGAAGACTGCGCCATCGAAGCCAGCCGCCATAATTTACTTTGCGCCTCGAACATTCCCAGCGCCCGGTTCATCTGTAAAATCACCCGATGCAAACGCACAATATCGTGCGACCCGGTTTTTTCGCTGAGATCGT
TGAGAAAGGATTCCGCCAGCCCACGCAGCGTCTCGGTCAGCTTCGCCAGTCGCTGACAGATTTCCATGACCTCATCAGGCAACTCGCCCATTGCAAAACGGTGCTCCGCCTCCTGCGCTGCGGGCATATACAGATTAAGAATGGCATTG
AGCGAGGCGATCAGTTCGTATACCTCCTCGCAATGGGCGTTAAGTCGCTCAGGGTTCGCTAACGGCGGCGTGGTTTTAGGGCGAAACTGCTCCATGCAGGTCGCCACCAGTTTGCTGAACAGATCCAGTTGCAAGCGATACCAGGAGGC
GGTAATTTCCGCGCTCATCTCCAGCGCATCGCGAGCGACATCGGGCAGATGATGACCTTCATCCAGTACCAACAGCAGATGCTTTGGTTCCGGCAGTACGGCTTCACTTTCCATCGCCGCCATTACCAGCGCGTGATTCGCCACCACGA
CTTCCGCTTCCTGTATTTCACGTCGGGCGACAAAGAATGGACATTCGCGGTAATAGTGACAGTTGCGGTTCAGACAACTGGCTTTATCGGTACTTAATCGCCGCCACAGATCGTCGTCAATAGCAATGTCGGTATGATCCCGCAGACCA
TCCCATTTGTAACCGTCAAGATCCCCTTTCAGCCTGGCGCAACGCTTTTGCTCTTCCTGGTTGTTAGGCGTCAGTTCGTCATCAAGAAAGGCGAGCAGATCCTGCTGCGTGGCGTCGCTACTGGCGAGCGCCGCCAGGTTTCGCGGACA
CACGTAACGCCCGCGGCCCAAAGGCGGCGGTAAAGCGCAGATCGGGGATGATTTTTTCGTAGCAGCGGCAAATCTTTGCTAAAGATCTGATCCTGTAACGCCACGTTAGCGGTACTGACCACCAGTGTTTTTTGCTCTTCCCGGGCAATGG
CGATACCGGGAATAAGATAGGAGAGGGTTTTCCCGACGCCGGTCGGCGCTTCAATAGCCAGATGTCTCCCTTCTTCCCCGGCCAGCGTTCTGGCGACGTCAGCAATCATCTGCCGCTGCGGCGCACGGGGAATAAAGTCGGGGATCTGG
TCCTGAAGCGCCTTATACCAGGCGGCGATTTGCGCTTTCAGCGCAGCGGTCAATGCCAT
>k141_19:109154-110516
TTACTGGGCAGGCGTGATTCTGCGCGGGCGACGCGGGCGGACGTGCTTTGTCGCCAGCGGGTTTCCCCTCGCCTGTGCGCGGTTTTGGTTTCGCCTTCGCCGCCCCGCTATCCTGACGTCGCGGCTGCTGACTGCGCCCCGCTCCCCCCT
GATTTTGTCCGCGTCCACGACCGCCGCCGCCGCGCTGCTGACGACCGTTCTGAATCGGCTCCGCTTTAATCGACGGGTCCGGCTCATAGCCCGGCGTCGTAATGCGTGGGATCTCTTTTTTCAGCAACTTTTCGATATCACGCAACAAT
TTATGTTCGTCAACACAGACCAGCGACAGCGCCTCGCCAGTTGCCGCCGCGCGACCGGTACGGCCAATACGGTGTACATAATCTTCCGGCACGTTCGGCAGCTCATAGTTGACCACGTGCGGCAGCTCTTCAATATCCAGACCACGCGC
CGCAATATCCGTCGCCACCAGCACGCGAATATCGCCGGATTTAAAATCGGCCAGCGGTACGTGCGCCCTGCGATTTATTACCGTGGATCGCCGGCGCTGCGAATACCGTCTTTATTAAGCTGTTCCGCCAGATGGTTCGCGCCAT
GCTTGGTGCGGGTGAAGACCAACACCTGCTGCCAGTTGCCCTGGCCAATCATTTGTGACAGCAGTTCGCGCTTACGTTTTTTATCGACAAAGTGCACGTGCTGCGTCACCTGCTCTGAGGCCGTGTTGCGGCGCGCCGCCACTTCGATTTCC
AGCGGGTTGCGCAACAGCTTTTCGGCGAGCGCTTTGATGTCGTCGGAGAACGTGGCGGAGAACAGCAGATTCTGACGCTTCGCCAGCACGCGACGGGATGTCGTGAATAAAGCCCATATCCAGCATACGGTCGGCTTC
ATCCAGTACCAGAATCTCAATCTGATCCAGCTTCACGGCATTTTGATGTTCCAGGTCGAGCAGACGTCCCGGCGTCGCCACCAGCACATCCACGCCGCCGCGCAGTTTCATCATCTGCGGATTGATGCTTACGCCGCCAAACACCACCA
GCGAGCGAATGTTGAGGTATTTACTGTAATCTCTGACGTTTTCGCCAATTTGCCCGCCAGCTCGCGGGTTGGCGTGAGGATCAACGCGCGCACCGGGCGACGACCTTTAGCGTGCGGCTGATGGGTAATCAGATGCTGCAACAATGGC
AGGGTAAAACCTGCCGTTTTACCGGTTCCGGTCTGGGCGCTGGCCATCAGATCGCGGCCTTCCAGCACGGCGGGAATCGCCTGTTGCTGAATAGGGGTAGGTTCACGGTAACCCTGCTCGGCAATGGCGCGCAGGATATCAGGGTTTAA
TCCCAGGGAATCAAAAGACAT
>k141_17:32-209
TTACGCCTTTTCAACCTGAACGAGGTTCGTATGTGACGGGTTCCCCTTCGCCAGCGGAGAAGGACGTTGGGTCGTCAGTACGTTAATACATCCGCCCTGATCCACGCGCTTTGCATCCGGGTCATACCACGCGCGCCTTCTCCCAGGGCGA
CCACGCCCGGCATCATGCGCGGCGTCAC
(group2) [gtg875q@biogenome2023 lem]$ bedtools getfasta -fi CGT2006.fasta -bed ./prodigal_results/cds_CGT2006.gff -fo 2006.fasta
(group2) [gtg875q@biogenome2023 lem]$
```

# Task Delegation

- Jyothi Guruprasad - Background & PPT
- Gautham Krishna Sankar Ramalaxmi - GeneMarkS2
- Kaize H Ali - GLIMMER
- Varsha Srinivasan - Prodigal
- Lee Ellen Mullins - Bedtools & Comparison

# References

- https://github.com/nigyta/dfast_core/blob/master/docs/FAQ.md
- http://ccb.jhu.edu/software.shtml
- https://wiki.gacrc.uga.edu/wiki/GLIMMER-Teaching
- https://bmcbioinformatics.biomedcentral.com/articles/10.1186/1471-2105-11-119
- https://github.com/hyattpd/Prodigal