Gene Prediction
Background & Strategy

Presenters: Sachin Kumar, Mansi Gupta, Vineeth Aljapur, Manu Tej Sharma Arrojwala, Mingming Cao

Date: 02-14-2019
Gene Prediction is the process of identifying regions that encode for genes. These identified regions can be then functionally annotated to understand specific protein coding genes, RNA genes and regulatory genes.

**Goal for the project:**
Identify new coding regions to understand variant based pathogenicity.
Prokaryotic Gene Structure

- Long Open Read Frames (ORFs) followed by short non-coding sequences.
- Easy to predict genes due to consistency in coding regions.

[Diagram of Prokaryotic Gene Structure]

Open Reading Frames (ORFs)

- Continuous stretch of codons that begin with a start codon and end with a stop codon having ability to be translated.
- Gene prediction methods identify ORFs to indicate the genes coding for protein and functional RNAs.

https://en.wikipedia.org/wiki/Open_reading_frame - Sample sequence with 3 possible reading frames
Methods used for Gene Prediction

Homology based tools:

- Homology based method uses local alignment to find similarities in protein coding regions and mRNA regions in the sequences in comparison to the extensive databases, e.g., BLAST, HMMER

Ab Initio tools:

- Ab-initio method utilises component detection like promoter sequences, start and stop codons and GC content to predict ORFs, e.g., GLIMMER, PRODIGAL, GENEMARK, EASYGENE
A comparison of Ab-initio and Homology based methods

<table>
<thead>
<tr>
<th>Pros</th>
<th>Homology</th>
<th>Ab Initio</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accurate</td>
<td></td>
<td>Fast</td>
</tr>
<tr>
<td>Increasing number of sequences</td>
<td></td>
<td>Not Limited by existing knowledge</td>
</tr>
<tr>
<td>Works well for eukaryotes</td>
<td></td>
<td>Inexpensive</td>
</tr>
<tr>
<td>Reliable for known genes</td>
<td></td>
<td>Easy and lightweight tools</td>
</tr>
<tr>
<td>Cons</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Limited by existing knowledge</td>
<td></td>
<td>Could have False positives</td>
</tr>
<tr>
<td>Experimental errors could propagate</td>
<td></td>
<td>No experimental verification</td>
</tr>
<tr>
<td>Requires larger databases</td>
<td></td>
<td>May not work well with eukaryotes</td>
</tr>
<tr>
<td>Not as many tools as Ab Initio</td>
<td></td>
<td>May not be as robust as homology</td>
</tr>
</tbody>
</table>
Homology-based methods

**Algorithms**
- BLAST
- Hidden Markov Model (HMM)

**Tools**
- BLAST+
- HMMER
Homology-based gene prediction - BLAST

**Local Alignment**

Target Sequence
5’ ACTACTAGATTTACGGATCAGGTACTTTAGAGGCTTGCAACCA 3’

Query Sequence
5’ TACTCAGGATGAGGTACTTTAGAGGC 3’

- **BLAST**: Basic Local Alignment Search Tool
- BLAST approximates the Smith–Waterman local alignment algorithm; it permits a chance of missing weak sequence similarities in exchange for greatly increased speed.
- **BLASTN** is used to compare a DNA query sequence to a database of DNA sequences.
- **MEGABLAST** is a faster variant on BLASTN best used for very long query sequences and for finding alignments of very closely related sequences.
Homology-based gene prediction - BLAST+ 

BLAST+ 2.8.1
January 4, 2019

NCBI Insights
Providing insights into NCBI resources and the science behind them

Posted on January 4, 2019

BLAST+ 2.8.1 with New Databases and Better Performance
Homology-based gene prediction - HMMER

- **HMMER** is a software suite includes database search programs for both protein and DNA sequence analysis: *phmmer, hmmscan, hmmsearch, jackhmmers, nhmmer*(DNA) and etc.

- **HMMER** is used for searching sequence databases for sequence homologs, and for making sequence alignments, using Hidden Markov Models (HMMs)

- **HMMER** makes a profile (profile HMM) of the query that assigns a position-specific scoring system for substitution, insertions and deletions
Homology-based gene prediction - HMMER

A profile HMM is a linear state machine consisting of a series of nodes, each of which corresponds roughly to a position (column) in the alignment from which it was built.

It provides a formal probabilistic framework for sequence comparison and improve detection of remote homology by

(i) enabling position-specific residue and gap scoring based on a query profile

(ii) calculating the signal of homology based on the more powerful ‘Forward/Backward’ HMM algorithm that computes not just one best-scoring alignment, but a sum of support over all possible alignments
Homology-based gene prediction - HMMER

HMMER 3.1

Advantages

• More accurate
• Can detect remote homologs as sensitively as possible
• Can run as fast as BLAST (~100 fold than previous)

Homology-based gene prediction - *nhmmer*

The program nhmmer is used to search one or more nucleotide queries against a nucleotide sequence database.

Uses a series of acceleration filters that depend on simpler approximations of the final Forward score of a hit.

Searching the target database with a single query sequence, either (i) producing a consensus sequence to represent the sequence family, then using the consensus as query to search the database, or (ii) using the family pairwise (fpw) search method, in which each individual sequence from the family alignment is used as a query, the hit lists are merged, and overlapping hits are adjudicated by recording the hit with the best E-value.

As it said, more sensitive than blastn.
GeneMarkS-2

- S - self training
- Typical and atypical coding regions (based on the GC content of the ORF)
- The model which yields the highest log-odds score is selected
- Typical model - parameters estimated from iterative self training on the whole genome.
- Atypical model - precomputed parameters from 82 models
- Can account for leaderless transcription start sites too
- Generalized Hidden markov model (5th order for coding regions, 2nd order for non-coding regions)
- Viterbi algorithm is used to compute the maximum likely sequence of hidden state
GeneMarkS-2

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

Alexandre Lomsadze,1,5 Karl Gemayel,1,6 Shiuyun Tang,4 and Mark Borodovsky1,2,3,4,5
GeneMarkS-2

- Algorithm stops after 10 iterations in the final prediction step if it doesn’t converge
- Reduces the number of false positive predictions by using atypical models
- Inclusion of leaderless TSS reduced error rate of predictions

Cons:
- Small genes, programmed frameshifts and pseudogenes
Gene Locator and Interpolated Markov ModelER (GLIMMER)

- GLIMMER is a collection of programs for identifying genes in microbial DNA sequences.
- Utilizes training set of genes to generate Interpolated Markov Models.
- Executed in 2 parts – builds the interpolated context model (training part) and then uses it to identify regions certain to be genes.
GLIMMER Algorithm

- Generates training set data using possible overlapping long-orfs and with amino acid distribution.
- Trains all six Markov models of coding DNA from 0 to 8th order.
- Calculates probabilities from data and decision to use the MM or IMM.
- Obtains score for long-orfs.
- If greater than threshold, then predicts to be a gene.
## GLIMMER

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to install and use, lightweight</td>
<td>Output format differs from the standard gene prediction softwares</td>
</tr>
<tr>
<td>Efficient execution time ~ 10 seconds</td>
<td>High false positives</td>
</tr>
<tr>
<td>Reverse scoring from of ORFs from stop to start codon</td>
<td>Glimmer actually predicts more genes per nucl in the third order random seq than they do in genome</td>
</tr>
<tr>
<td>Built in scripts for executing in 3 different modes</td>
<td></td>
</tr>
<tr>
<td>Ribosome binding site finder with ELPH algorithm</td>
<td></td>
</tr>
</tbody>
</table>
# Prodigal

(PROkaryotic DYnamic programming Genefinding ALgorithm)

<table>
<thead>
<tr>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
<tbody>
<tr>
<td>Easy to Install and takes less space</td>
<td>Cannot Predict RNA Genes</td>
</tr>
<tr>
<td>Predicts Genes in 3 Formats (GFF/GenBank/Sequin)</td>
<td>Cannot Handle Introns (Works only on Prokaryotes)</td>
</tr>
<tr>
<td>Handles draft genomes and metagenomes too</td>
<td>Cannot do Functional Annotation</td>
</tr>
<tr>
<td>Fast (Usually runs in seconds)</td>
<td>Cannot deal with Frame Shifts (only TTG for start)</td>
</tr>
<tr>
<td>Don’t need to train (unsupervised learning)</td>
<td>May not work with Viral Genes</td>
</tr>
<tr>
<td>Gives the stats like contig len, GC Content to verify</td>
<td>No GUI</td>
</tr>
</tbody>
</table>

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

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

*BMC Bioinformatics* 2010 11:119

[https://doi.org/10.1186/1471-2105-11-119](https://doi.org/10.1186/1471-2105-11-119)  © Hyatt et al; licensee BioMed Central Ltd. 2010

Received: 20 July 2009 | Accepted: 8 March 2010 | Published: 8 March 2010
Prodigal Algorithm

- Uses **Dynamic Programming (DP)** to construct training set by GC frame plot in ORFs
- Traverses the sequence and looks at GC bias for each of three codon positions and chooses the one with **highest GC content** and normalizes it
- Calculates preliminary score by multiplying bias with length for ORF length of 90 bp
- Penalizes or gives bonus to intergenic spaces according to gene distance

**Figure [1] Gene Connections:** (a) 5’-3’ Gene (b) 3’-5’ Gene (c) 3’-3’ Overlapping gene (d) 3’-5’ FR Gene (e) 5’-3’ Intergenic space (f) 3’-5’ Rev Gene (g) 3’-3’ Overlapping gene (h) 5’-5’ Intergenic space (i) 3’-3’ Intergenic space **Red** - Gene Connections **Black** - Intergenic Connections
An automated gene-finding method, which estimates the statistical significance of a predicted gene. Based on hidden Markov Model. Statistical significance is calculated based on its HMM score and length of the ORF. Solves the problem of distinguishing between real and random ORFs by giving statistically significant values.
**EasyGene**

<table>
<thead>
<tr>
<th>Advantages</th>
<th>Disadvantages</th>
</tr>
</thead>
<tbody>
<tr>
<td>Self-training gene finder.</td>
<td>Large number of false positives</td>
</tr>
<tr>
<td>Sensitivity of EasyGene tends to be comparable to GeneMarkS</td>
<td>An old software which has GFF version 2</td>
</tr>
<tr>
<td>Specificity is generally high</td>
<td>Has models for 138 organisms, which might pose as a limitation.</td>
</tr>
<tr>
<td>Gives exact start sites</td>
<td></td>
</tr>
</tbody>
</table>
RNA gene prediction

- Introduction
  - Non-coding RNA

- Tools
  - Infernal (ncRNA)
  - Aragorn (tRNA)
  - tRNAScan-SE (tRNA)
  - Barrnap (rRNA)
  - RNAmmer (rRNA)
Non-coding RNAs are a group of RNA transcripts that do not necessarily code for protein products, as they make instead potentially functional RNAs.

Abundant and functionally important types of non-coding RNAs include transfer RNAs (tRNAs) and ribosomal RNAs (rRNAs), as well as small RNAs such as microRNAs, siRNAs, piRNAs, snoRNAs, snRNAs, exRNAs, scaRNAs and the long ncRNAs.
RNA gene prediction - Tools

ncRNA
- **Infernal 1.1, 2013**
  - Use Covariance models (CMs), probabilistic profiles of the sequence and secondary structure of an RNA family
  - RNA homology search based on HMM-banded CM alignment method
  - ~10,000-fold acceleration over exhaustive non-filtered CM searches

rRNA
- **Barrnap, 2013**
  - Bacterial ribosomal RNA predictor
  - Uses HMMER 3.1 tool for HMM searching in RNA:DNA style
  - The HMM models used are derived from Rfam, Silva and RefSeq
  - Fast but may will probably miss weird rRNAs

- **RNAmmers, 2007**
  - Nuvel, unannotated rRNAs are also predicted in many genomes
  - More accurate but slower than barrnap

**tRNA**
- **Aragorn 1.2, 2004**
  - tRNAs and tmRNAs
  - achieves a detection sensitivity of 99% from a set of 1290 tRNA genes and detects all complete tmRNA sequences in the tmRNA database

- **tRNAscan-SE 2.0, 2016**
  - improvements in tRNA classification with greatly enhanced biological context
  - Accurate
Prokka - rapid prokaryotic genome annotation

Prokka coordinates a suite of existing software tools to achieve a rich and reliable annotation of genomic bacterial sequences.

<table>
<thead>
<tr>
<th>Tools &amp; Version</th>
<th>Features predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>BLAST+ 2.7</td>
<td>Homology-based gene prediction</td>
</tr>
<tr>
<td>HMMER 3.1</td>
<td>Homology-based gene prediction</td>
</tr>
<tr>
<td>Prodigal 2.6</td>
<td>Coding sequence (CDS)</td>
</tr>
<tr>
<td>Aragorn 1.2</td>
<td>Transfer RNA genes</td>
</tr>
<tr>
<td>Barrnap 0.9</td>
<td>Ribosomal RNA genes</td>
</tr>
<tr>
<td>Infernal 1.1</td>
<td>Non-coding RNA</td>
</tr>
</tbody>
</table>

doi:10.1093/bioinformatics/btu153
GeneValidator - identifies problematic gene predictions

- Predicted Gene
- BLAST against an updated database (Diamond for ~100x speed)
- Up to 7 comparisons between gene prediction and significant hit sequences
- Each query either similar or different according to a P-value
- Result: Quality score (0 to 100)

- Length
- Coverage
- Conserved Regions
- Different genes
- ab initio Open Reading Frame (ORF)
- Similarity-based ORFs
- MAKER RNASeq Quality Index

GeneValidator: identify problems with protein-coding gene predictions
Monica-Andreea Drăgan, Ismail Moghul, Anurag Priyam, Claudio Bustos, Yannick Wurm
Published: 18 January 2016
GeneValidator - identifies problematic gene predictions

Percentage of good predictions:

<table>
<thead>
<tr>
<th></th>
<th>Acetobacter</th>
<th>Bacillus</th>
<th>Treponema</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneMarkS-2</td>
<td>30.87%</td>
<td>27.07%</td>
<td>23.07%</td>
</tr>
<tr>
<td>Prodigal</td>
<td>29.9%</td>
<td>27.08%</td>
<td>23.15%</td>
</tr>
</tbody>
</table>
### Evaluation

Based on Genes predicted by each method:

<table>
<thead>
<tr>
<th>Organism</th>
<th>Base (Score)</th>
<th>GMS2 (Score)</th>
<th>Glimmer (Score)</th>
<th>EasyGene (Score)</th>
<th>Prodigal (Score)</th>
<th>Prokka (Score)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Acetobacter Ghanensis</em></td>
<td>2698 (1.00)</td>
<td>2687 (0.99)</td>
<td>2867 (1.06)</td>
<td>-</td>
<td>2712 (1.01)</td>
<td>2804 (1.04)</td>
</tr>
<tr>
<td><em>Bacillus Anthracis</em></td>
<td>5706 (1.00)</td>
<td>5761 (1.01)</td>
<td>5900 (1.03)</td>
<td>5583 (0.97)</td>
<td>5768 (1.01)</td>
<td>5999 (1.05)</td>
</tr>
<tr>
<td><em>Treponema Denticola</em></td>
<td>2770 (1.00)</td>
<td>2592 (0.94)</td>
<td>2668 (0.96)</td>
<td>2511 (0.91)</td>
<td>2578 (0.93)</td>
<td>2637 (0.95)</td>
</tr>
</tbody>
</table>
Evaluation

Based on Sensitivity:

Sensitivity = TP / (TP + FN)
Specificity = TN / (TN + FP)

Since we cannot get TN (non-coding regions are not given) we evaluated them based on sensitivity (in rounded percentages for start + stop / start regions).

<table>
<thead>
<tr>
<th>Organism</th>
<th>GeneMarkS2</th>
<th>Glimmer</th>
<th>EasyGene</th>
<th>Prodigal</th>
<th>Prokka</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Acetobacter Ghanensis</em></td>
<td>74/85</td>
<td>31/33</td>
<td>-</td>
<td>78/86</td>
<td>78/86</td>
</tr>
<tr>
<td><em>Bacillus Anthracis</em></td>
<td>92/95</td>
<td>31/32</td>
<td>90/93</td>
<td>96/97</td>
<td>96/97</td>
</tr>
<tr>
<td><em>Treponema Denticola</em></td>
<td>81/86</td>
<td>33/34</td>
<td>78/84</td>
<td>81/86</td>
<td>82/86</td>
</tr>
</tbody>
</table>
Final Pipeline

GeneMark-S2 → Merged Results → Functional Annotation

The intersection of Prodigal and GeneMark-S2 results are used.
References


References

13. Lagesen K, Hallin PF, Rødland E, Stærfeldt HH, Rognes T Ussery DW RNammer: consistent annotation of rRNA genes in genomic sequences
Thank You