Prokaryotic genome analysis

Progenepredict: A python-based tool for *de novo* gene prediction in prokaryotic genomes

L. Hohmann

Department of Biology, Box 118, 221 00, Lund University, Sweden

Abstract

Motivation: *De novo* gene prediction plays a crucial role in annotating prokaryotic genomes and is therefore an indispensable tool for gaining insight into a species' underlying biological processes. Its prediction process is based on using selected genomic features to identify potential coding sequences. The development of a tool that presents users with the option to specify several parameters related to these genomic features enables them to tailor the tool's application to their exact needs.

Results: Progenepredict performs *de novo* gene prediction in prokaryotic genomes based on the identification of open reading frames and Shine-Dalgarno sequences by employing a local sequence alignment algorithm. Besides being able to specify the Shine-Dalgarno sequence, users may determine how much the sequence preceding a gene may differ from the Shine-Dalgarno sequence template while still being considered a valid match.

Implementation: Progenepredict was built using the Python programming language (v3.9.2) in combination with the package for array computing NumPy (v1.20.1) and is freely available on GitHub for noncommercial users.

Contact: lennart.hohmann.7524@student.lu.se

Availability: The software is available at: https://github.com/lphohmann/Progenepredict.

1 Introduction

An important step in understanding the genomes of new prokaryotic strains is called gene prediction. It is preceded by sequencing and genome assembly processes which yield a reconstruction of the strain's genome. During gene prediction, genomic DNA sequences are investigated to identify regions that encode proteins or functional RNA-products (Stothard & Wishart, 2006). Since a DNA strand may be divided into three sets of consecutive and non-overlapping triplets, called reading-frames, and genes may overlap in different reading-frames, gene prediction tools have to take all three into account (see **Figure 1**). Subsequently, sequences of predicted genes are utilized for their functional annotation by using them as queries when carrying out sequence similarity searches against selected nucleotide or protein databases (Ranganathan et al., 2019). Opposed to the reference-based type of gene prediction which relies on using genes from related organisms as training sets, *de novo* gene prediction is based on selected genomic features that can be used to identify coding sequences. The most prominent genomic features used in prokaryotic *de novo* gene prediction are open reading-frames (ORFs). An open reading-frame consists of a start codon, a stop codon and the sequence in between the two codons. While not every ORF necessarily encodes a gene, every coding sequence in prokaryotes correspond to one ORF. Furthermore, may the length of an ORF serve as a valuable indicator of whether the ORF at hand does indeed represent a coding sequence as the majority of short ORFs are not part of any genes (Pevsner, 2016).

Another genomic feature which guides prokaryotic gene prediction is the Shine-Dalgarno sequence. After transcription, the sequence is complementary to the 16S rRNA and therefore serves as a consensus sequence for ribosome binding. Its value in gene prediction stems from the fact that it occurs only a few nucleotides upstream of the start codon, thereby serving as a useful indicator of likely ORF start sites (Alberts et al., 2015; Pevsner, 2016).

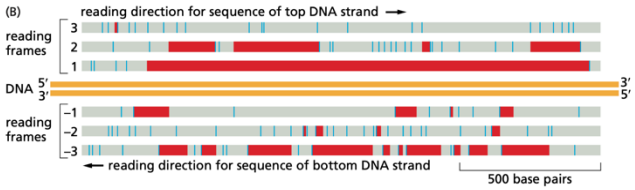

Figure 1: *Visualization of reading frames leading to different open reading frames overlapping on the same nucleotide sequence (Alberts et al., 2015).*

The study at hand focused on developing a novel tool for gene prediction in prokaryotes based on the investigation of all three genome-reading-frames for ORFs preceded by a Shine-Dalgarno sequence. While these genomic features are commonly employed for gene prediction in other programs as well, the distinguishing feature of Progenepredict is that it provides the user with a greater range of flexibility when scanning a genome for these features. Besides being able to specify which start and stop codons should be considered, Progenepredict allows the user to specify the exact Shine-Dalgarno sequence that predicted genes should be preceded by, as well as how much the Shine-Dalgarno sequence preceding a gene may differ from the user-specified template. The latter is assessed by employing a local alignment algorithm for the alignment of the Shine-Dalgarno sequence to the nucleotides preceding the start codon and subsequent alignment-scoring based on the number of shared identical positions.

2 Software

Progenepredict was built using the Python programming language (v3.9.2) in combination with the package for array computing NumPy (v1.20.1) (Harris et al., 2020; Van Rossum & Drake, 2009).

3 Methods and results

3.1 Workflow

Progenepredict is a tool for gene prediction in prokaryotic genomes based on the investigation of all three reading-frames for the presence of ORFs with Shine-Dalgarno (SD) sequences in the vicinity of their start codon. It takes a genome in FASTA format as input, with the sequence in the file corresponding to the forward strand (5' to 3').

First, the program creates the reverse complement strand based on the leading strand. It then iterates over the leading strand and employs the following mechanism to scan for ORFs.

For each reading-frame:

(1) The current codon is created

(2) The codon is assessed to see if it is:

   a. A start codon preceded by a SD sequence (whose presence is assessed by employing the Smith-Waterman alignment algorithm (Tiefenauer, 2018))

   b. A stop codon

   c. None of the above

Step two is continuously executed until a start codon is found that identifies the start of an ORF associated with a coding sequence and then:

(3) The iteration over the strand and codon assessment continues until a stop codon is found (any additional start codons do not reinitialize the ORF)

(4) When a stop codon is found the length of the complete ORF is assessed

If the ORF fulfills the criterion of minimum length, the predicted gene and its associated information is saved. If it doesn't fulfill the criterion of minimum length the ORF is discarded. Then the process continues from (1) until the end of the input sequence is reached. Subsequently, the same process takes place for the reverse complement strand which is created based on the leading strand in the FASTA input file.

Progenepredict produces a tab-separated output file that contains the start and end position of each predicted gene, the start codon that initialized the ORF, as well as information on the reading frame and strand it was found on. A second output file containing the nucleotide sequences of the predicted genes in FASTA format is optional.

3.2 Usage and features

Users are able to specify several parameters to adapt the gene prediction process to their specific needs. The arguments that can be specified are summarized in **Table 1**. The default values for arguments were set based on the features of the E. coli genome and therefore are recommended to be adapted to the species the respective input genome originates from.

Table 1. Usage information for Progenepredict

Flag	Argument description
-h	*show help message with usage instructions*
-g	*the file containing the genome sequence in FASTA format*
-o	*the desired name of the output file containing the predicted genes*
-ml	*the minimum ORF length of predicted genes (default=300)*
-sc	*the possible start codons (default=ATG)*
-ec	*the possible stop codons (default=TAA TAG TGA)*
-sd	*the Shine-Dalgarno sequence (default=AGGAGG)*
-me	*the max. number of errors that may be present in the pairwise alignment when checking for a Shine-Dalgarno sequence (default=1)*
-f	*output a file with the nucleotide sequences of predicted genes in FASTA format*

Note: The -g flag that specifies the file that contains the genome sequence in FASTA format is the only mandatory argument.

4 Discussion

One of the most distinct features of Progenepredict is that the user is not only able to choose the Shine-Dalgarno sequence that identifies ORFs with coding functions but also how specific the pre-start codon sequences have to match to the previously defined SD sequence to be considered a valid match. This option may be especially of interest against the backdrop of previous studies findings that suggest a positive correlation between the conservation of Shine-Dalgarno sequences with translation efficiency. Therefore, it is a general characteristic of many prokaryotic genomes that evolutionary pressures led to frequently expressed genes being preceded by well conserved SD sequences (Barrick et al., 1994; Sakai et al., 2001). Furthermore, in cases where Progenepredict is employed to predict genes in a prokaryotic species with a poorly understood genome, the exact SD sequence occurring in the species might not be identified yet. In these cases, the option to decrease the required identity with the input SD sequence template may be preferable to prevent accidental exclusion of true genes.

4.1 Limitations and future directions

While Progenepredict is making use of the two most prominent features used in gene prediction, a third commonly used feature is not integrated in its gene prediction process. By implementing a Hidden Markov model, the bias in codon usage patterns of prokaryotic species may be used to assess coding potentials of suspected genes and thereby improve Progenepredict's gene prediction power. Besides the implementation of such a model, future development directions include the construction of a user interface to make the tool accessible for users without a background in bioinformatics. Furthermore, Progenepredict's gene prediction accuracy still has to be evaluated to extend its comparison with existing gene prediction tools beyond comparing available features.

5 Acknowledgements

The author is grateful to the editor and two anonymous reviewers for their valuable comments on the manuscript and for Lund University for their financial support.

6 Funding

This work has been supported by Lund University.

Conflict of Interest: none declared.

7 References

Alberts, B., Johnson, A., Lewis, J., Raff, M., Morgan, D., Roberts, K., & Walter, P. (2015). *Molecular biology of the cell*. Garland Science.

Barrick, D., Villanueba, K., Childs, J., Kalil, R., Schneider, T. D., Lawrence, C. E., Gold, L., & Stormo, G. D. (1994). Quantitative analysis of ribosome binding sites in E.coli. *Nucleic Acids Research*, *22*(7), 1287-1295. https://doi.org/10.1093/nar/22.7.1287

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, *585*(7825), 357-362. https://doi.org/10.1038/s41586-020-2649-2

Pevsner, J. (2016). Completed Genomes: Bacteria and Archaea. In *Bioinformatics and Functional Genomics* (pp. 797-837). Wiley-Blackwell. https://doi.org/https://doi.org/10.1002/9780470451496.ch15

Ranganathan, S., Gribskov, M. R., Nakai, K., & Schönbach, C. (2019). Encyclopedia of bioinformatics and computational biology. https://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=1724622

Sakai, H., Imamura, C., Osada, Y., Saito, R., Washio, T., & Tomita, M. (2001). Correlation Between Shine–Dalgarno Sequence Conservation and Codon Usage of Bacterial Genes. *Journal of Molecular Evolution*, *52*(2), 164-170. https://doi.org/10.1007/s002390010145

Stothard, P., & Wishart, D. S. (2006). Automated bacterial genome analysis and annotation. *Curr Opin Microbiol*, *9*(5), 505-510. https://doi.org/10.1016/j.mib.2006.08.002

Tiefenauer, D. (2018). *Smith-Waterman algorithm in Python*. https://tiefenauer.github.io/blog/smith-waterman/

Van Rossum, G., & Drake, F. L. (2009). *Python 3 Reference Manual*. CreateSpace.