

Prokka: rapid prokaryotic genome annotation

Torsten Seemann^{1,2}¹Victorian Bioinformatics Consortium, Monash University, Clayton 3800 and ²Life Sciences Computation Centre, Victorian Life Sciences Computation Initiative, Carlton 3053, Australia

Associate Editor: Alfonso Valencia

ABSTRACT

Summary: The multiplex capability and high yield of current day DNA-sequencing instruments has made bacterial whole genome sequencing a routine affair. The subsequent *de novo* assembly of reads into contigs has been well addressed. The final step of annotating all relevant genomic features on those contigs can be achieved slowly using existing web- and email-based systems, but these are not applicable for sensitive data or integrating into computational pipelines. Here we introduce Prokka, a command line software tool to fully annotate a draft bacterial genome in about 10 min on a typical desktop computer. It produces standards-compliant output files for further analysis or viewing in genome browsers.

Availability and implementation: Prokka is implemented in Perl and is freely available under an open source GPLv2 license from <http://vicbioinformatics.com/>.

Contact: torsten.seemann@monash.edu

Received on November 10, 2013; revised on February 6, 2014; accepted on March 13, 2014

1 INTRODUCTION

Genome annotation is the process of identifying and labeling all the relevant features on a genome sequence (Richardson and Watson, 2012). At minimum, this should include coordinates of predicted coding regions and their putative products, but it is desirable to go beyond this to non-coding RNAs, signal peptides and so on.

There are various online annotation servers (Stewart *et al.*, 2009). The NCBI provides a Prokaryotic Genomes Automatic Annotation Pipeline service via email, with a turn-around time measured in days. RAST is a web server for annotating bacterial and archaeal genomes that provides annotation results in under a day (Aziz *et al.*, 2008), and xBASE2 does similar in a few hours (Chaudhuri *et al.*, 2008). These classes of tools are valuable, but they are not useful where throughput or privacy is critical.

Here we present Prokka, a command line software tool that can be installed on any Unix system. Prokka coordinates a suite of existing software tools to achieve a rich and reliable annotation of genomic bacterial sequences. Where possible, it will exploit multiple processing cores, and a typical bacterial genome can be annotated in ~10 min on a quad core desktop computer. It is well suited to iterative models of sequence analysis and integration into genomic software pipelines.

2 DESCRIPTION

2.1 Input

Prokka expects preassembled genomic DNA sequences in FASTA format. Finished sequences without gaps are the ideal input, but it is expected that the typical input will be a set of scaffold sequences produced by *de novo* assembly software. This sequence file is the only mandatory parameter to the software.

2.2 Annotation

Prokka relies on external feature prediction tools to identify the coordinates of genomic features within contigs. These tools are listed in Table 1, and all of them, except for Prodigal, provide coordinates and appropriate labels to describe the feature.

Proteins coding genes are annotated in two stages. Prodigal identifies the coordinates of candidate genes, but does not describe the putative gene product. The traditional way to predict what a gene codes for is to compare it with a large database of known sequences, usually at a protein sequence level, and transfer the annotation of the best significant match.

Prokka uses this method, but in a hierarchical manner, starting with a smaller trustworthy database, moving to medium-sized but domain-specific databases, and finally to curated models of protein families. By default, an *e*-value threshold of 10^{-6} is used with the following series of included databases:

- (1) An optional user-provided set of annotated proteins. These are expected to be trustworthy curated datasets and will be used as the primary source of annotation. They are searched using BLAST+ `blastp` (Camacho *et al.*, 2009).
- (2) All bacterial proteins in UniProt (Apweiler *et al.*, 2004) that have real protein or transcript evidence and are not a fragment. This is ~16 000 proteins, and typically covers >50% of the core genes in most genomes. BLAST+ is used for the search.
- (3) All proteins from finished bacterial genomes in RefSeq for a specified genus. This captures domain-specific naming, and the databases vary in size and quality, depending on the popularity of the genus. BLAST+ is used for this and is optional.
- (4) A series of hidden Markov model profile databases, including Pfam (Punta *et al.*, 2012) and TIGRFAMs (Haft *et al.*, 2013). This is performed using `hmmsearch` from the HMMER 3.1 package (Eddy, 2011).
- (5) If no matches can be found, label as 'hypothetical protein'.

Table 1. Feature prediction tools used by Prokka

Tool (reference)	Features predicted
Prodigal (Hyatt 2010)	Coding sequence (CDS)
RNAmmer (Lagesen <i>et al.</i> , 2007)	Ribosomal RNA genes (rRNA)
Aragorn (Laslett and Canback, 2004)	Transfer RNA genes
SignalP (Petersen <i>et al.</i> , 2011)	Signal leader peptides
Infernal (Kolbe and Eddy, 2011)	Non-coding RNA

Table 2. Description of Prokka output files

Suffix	Description of file contents
.fna	FASTA file of original input contigs (nucleotide)
.faa	FASTA file of translated coding genes (protein)
.ffn	FASTA file of all genomic features (nucleotide)
.fsa	Contig sequences for submission (nucleotide)
.tbl	Feature table for submission
.sqn	Sequin editable file for submission
.gbk	Genbank file containing sequences and annotations
.gff	GFF v3 file containing sequences and annotations
.log	Log file of Prokka processing output
.txt	Annotation summary statistics

2.3 Output

Prokka produces 10 files in the specified output directory, all with a common prefix. These are described in Table 2.

3 RESULTS

Prokka was designed to be both accurate and fast. To assess accuracy, we compared the annotations of Prokka, RAST and xBase2 for the highly curated *Escherichia coli* K-12 genome. All methods were told it was an *E.coli* genome. Table 3 shows that Prokka produced an overall better annotation than both RAST and xBase2. This result could vary for less well-studied or draft genomes.

Prokka uses parallel processing to decrease running time on multicore computers. The most time-consuming steps are BLAST+ and hmmscan, which both support multiple CPUs natively. However, Prokka is more efficient if it runs multiple single CPU threads on subsets of the data, which it achieves using GNU parallel (Tange, 2011). Experiments on our 64-core AMD Opteron server on single genomes show linear speedup with up to eight cores and sublinear gain thereafter. However, for much larger bacterial meta-genome datasets, linear speedup is observed for many more CPUs. To annotate the *E.coli* K-12 genome on a typical quad-core desktop computer takes about 6 min.

ACKNOWLEDGEMENTS

The author thanks Dieter Bulach, Simon Gladman, Tim Stinear, Connor Skennerton, Scott Chandry, David Powell, Adam

Table 3. Comparison of annotation of *E.coli* K-12 accession U00096.2

Feature	Reference	Prokka	RAST	xBase2
Total CDS	4321	4305	4512	4444
<i>Matching start</i>	–	3828	3571	3025
<i>Different start</i>	–	318	533	1052
<i>Missing CDS</i>	–	172	214	241
<i>Extra CDS</i>	–	159	405	367
<i>Hypothetical protein</i>	18	276	638	156
<i>With EC number</i>	1114	1050	1118	0
Total tRNA	89	88	86	88
Total rRNA	22	22	22	22

The bold denotes the best performing tool (column) for that attribute (row). The italics are “subsets” of the “Total CDS” section.

Caldwell, Roderick Felsheim, John Nash, Nick Loman, Heikki Lehvaslaiho, Bastien Chevreux, Nicola Soranzo, Jon Graf, Harald Gruber-Vodicka, Haruo Suzuki, Geoff Winsor, Lionel Guy, Andrew Page and Ole Tange for suggestions and bug reports.

Funding: This research was supported in part by the Victorian Life Sciences Computation Initiative, an initiative of the Victorian Government hosted by the University of Melbourne, Australia.

Conflicts of Interest: none declared.

REFERENCES

- Aziz, R.K. *et al.* (2008) The RAST Server: rapid annotations using subsystems technology. *BMC Genomics*, **9**, 75.
- Apweiler, R. *et al.* (2004) UniProt: the universal protein knowledgebase. *Nucleic Acids Res.*, **32**, D115–D119.
- Camacho, C. *et al.* (2009) BLAST+: architecture and applications. *BMC Bioinformatics*, **10**, 421.
- Chaudhuri, R.R. *et al.* (2008) xBASE2: a comprehensive resource for comparative bacterial genomics. *Nucleic Acids Res.*, **36**, D543–D546.
- Eddy, S.R. (2011) Accelerated profile HMM searches. *PLoS Comput. Biol.*, **7**, e1002195.
- Haft, D.H. *et al.* (2013) TIGRFAMs and genome properties in 2013. *Nucleic Acids Res.*, **41**, D387–D395.
- Hyatt, D. *et al.* (2010) Prodigal: prokaryotic gene recognition and translation initiation site identification. *BMC Bioinformatics*, **11**, 119.
- Kolbe, D.L. and Eddy, S.R. (2011) Fast filtering for RNA homology search. *Bioinformatics*, **27**, 3102–3109.
- Lagesen, K. *et al.* (2007) RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Res.*, **35**, 3100–3108.
- Laslett, D. and Canback, B. (2004) ARAGORN, a program to detect tRNA genes and tmRNA genes in nucleotide sequences. *Nucleic Acids Res.*, **32**, 11–16.
- Petersen, T.N. *et al.* (2011) SignalP 4.0: discriminating signal peptides from transmembrane regions. *Nat. Methods*, **8**, 785–786.
- Punta, M. *et al.* (2012) The Pfam protein families database. *Nucleic Acids Res.*, **40**, D290–D301.
- Richardson, E.J. and Watson, M. (2013) The automatic annotation of bacterial genomes. *Brief. Bioinform.*, **14**, 1–12.
- Stewart, A.C. *et al.* (2009) DIYA: a bacterial annotation pipeline for any genomics lab. *Bioinformatics*, **25**, 962–963.
- Tange, O. (2011) *GNU Parallel—The Command-Line Power Tool*. ;login: The USENIX Magazine, Feb 2011, pp. 42–47.