Bioinformatics, YYYY, 0–0
doi: 10.1093/bioinformatics/xxxxx
Advance Access Publication Date: DD Month YYYY
Manuscript Category

Applications Note

baerhunter: An *R* package for the discovery and analysis of expressed non-coding regions in bacterial RNA-seq data

A. Ozuna[1], D. Liberto[1], R. M. Joyce[1], K.B. Arnvig[2] and I. Nobeli[1,*],

[1] Institute of Structural and Molecular Biology, Department of Biological Sciences, Birkbeck, Malet Street, London WC1E 7HX, UK, [2]Institute of Structural and Molecular Biology, University College London, London WC1E 6BT, UK.

*To whom correspondence should be addressed.

Associate Editor: XXXXXXX

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Abstract

Summary: Standard bioinformatics pipelines for the analysis of bacterial transcriptomic data commonly ignore non-coding but functional elements e.g. small RNAs, long antisense RNAs or untranslated regions (UTRs) of mRNA transcripts. The root of this problem is the use of incomplete genome annotation files. Here, we present *baerhunter*, a coverage-based method implemented in *R*, that automates the discovery of expressed non-coding RNAs and UTRs from RNA-seq reads mapped to a reference genome. The core algorithm is part of a pipeline that facilitates downstream analysis of both coding and non-coding features. The method is simple, easy to extend and customize and, in limited tests with simulated and real data, compares favourably against the currently most popular alternative.

Availability: The *baerhunter R* package is available from: https://github.com/irilenia/baerhunter

Contact: i.nobeli@bbk.ac.uk

Supplementary information: Supplementary data are available at *Bioinformatics* online.

1 Introduction

Next-generation sequencing has facilitated global surveys of the transcriptome, largely focused on studying differential expression of genes across different conditions. Studies of eukaryotic transcriptomes are increasingly embracing the analysis of non-coding transcript expression but in bacteria, where intergenic regions tend to be a lot shorter (Thorpe *et al.,* 2017) and less well annotated, automated differential gene expression is still largely synonymous with differential expression of the coding regions (CDS). As the functional importance of bacterial non-coding RNAs (ncRNAs - the term used here to cover long antisense RNA, small regulatory RNA (sRNA), and untranslated parts of mRNAs) is becoming evident (Michaux *et al.,* 2014), so is the need for including them in differential expression studies.

A major obstacle in studying non-coding RNA expression in bacteria is that relatively few ncRNAs are reliably annotated and, with the exception of well-known cases (such as tRNAs, ribosomal RNAs and, more recently, some members of the RFAM (Kalvari *et al.,* 2018) families), the majority are not included in the standard annotation files required by computational pipelines. Requiring the non-coding RNAs to be included in the annotation is prohibiting their analysis by methods such as TrBorderEx (Wang *et al.,* 2015), which identifies the transcript boundaries but does not find new non-coding RNAs. An alternative to waiting for annotations to improve is to identify ncRNAs using the expression data signal. Early efforts in this direction relied on a combination of manual inspection and in-house written scripts to identify clusters of reads falling outside known CDS regions (Arnvig *et al.,* 2011)(Wilms *et al.,* 2012)(Pfeifer-Sancar *et al.,* 2013). These studies offered great insights into the non-coding transcriptome but applying their approach in a different context is time-consuming and prone to errors due to the need for recreating the pipelines from scratch. Selected studies have led to publicly available software for the study of ncRNAs in bacteria. However, some methods are limited to specific species (Pellin *et al.,* 2012) or rely on specialized sequencing protocols (Peña-Castillo *et al.,* 2015; Amman *et al.,* 2014). Two notable exceptions have appeared in recent years. DETR'PROK (Toffano-Nioche *et al.,* 2013) employs the Galaxy platform (Afgan *et al.,* 2018) to classify clusters of RNA-seq reads not

overlapping with annotated genes as sRNAs, antisense RNAs, and UTRs. The updated annotation file can be used to carry out differential gene expression. However, the DETR'PROK workflow is composed of a large number of steps, requires an active user input at several stages and depends on access to a Galaxy instance. Rockhopper (McClure *et al.,* 2013), a standalone, Java-based program that allows both identification of features in a bacterial transcriptome and differential expression between conditions, is primarily aimed at non-bioinformaticians. A user-friendly graphical interface masks a fairly sophisticated set of algorithms that are presented as a black box with only a handful of parameters accessible to the user. Although straightforward to use, the set up is inflexible with little scope for extending or altering the pipeline without expert interfering with the code. Finally, very recently, two more tools have been added to aid the discovery of non-coding RNAs in bacterial RNA-seq data. ANNOgesic (Yu *et al.,* 2018) is an ambitious and comprehensive pipeline that aims to fully characterize non-coding expressed regions in bacterial genomes. It has a large number of dependencies and its full potential is likely to be only realized when parameters for all modules included in the pipeline are suitably optimized. APERO (Leonard *et al.*), on the other hand, is a method that does not rely directly on read coverage, is implemented as an R package, so installation is straightforward, and performs well in comparison with many other methods. However, it requires paired-end reads and so cannot be used on many of the legacy RNA-seq datasets currently present in databases.

Here, we present *baerhunter* ("baer" stands for **ba**cterial **e**xpressed **r**egions), a new coverage-based method implemented in *R* (R Core Team, 2018), for automating the detection and quantification of expressed putative non-coding RNAs (including UTRs) in bacterial strand-specific RNA-seq data. At the core of the baerhunter pipeline is a simple but effective method of capturing expressed intergenic regions across sets of RNA-seq data samples. The method is designed to provide predictions of approximate locations of non-coding elements, reflecting our belief that accurate definitions of transcript ends are best achieved by targeted experimental methods rather than computational predictions from noisy data. We refer to these predictions as either "UTRs", if they are thought to be the untranslated part of a coding mRNA, or putative sRNAs, which in this context encompasses all other types of non-coding RNA in bacteria, including long antisense RNAs. The pipeline built around this method facilitates the analysis of differential expression of these regions in parallel with the more traditional protein-coding-focused analysis. Below, we describe our method and present the results of testing its performance both on simulated and real data from *Mycobacterium tuberculosis* (*Mtb*). In addition, we compare baerhunter to Rockhopper, chosen as the most widely used alternative method, as well as to ANNOgesic and APERO, chosen as the most recently developed methods.

2 Methods

The core algorithm of baerhunter carries out a search for intergenic features on each strand, displaying a minimum length and coverage depth in the RNA-seq signal (see Supplementary Methods for details). The algorithm is wrapped within a "driver" *R* script that can be easily edited to include, exclude or modify steps, depending on the user's requirements. Individual functions of baerhunter can also be used in isolation or incorporated within different pipelines. In the default mode, baerhunter reads in a set of Binary Alignment Map (BAM) files with RNA-seq reads mapped to a reference genome and an annotation file in the Generic Feature Format (GFF3) for the same genome. It will then identify expressed intergenic regions on each strand ("features") and combine overlapping features across multiple BAM files to create a full set of non-overlapping genomic features. In addition, baerhunter allows for new transcripts to be filtered by their expression level (normalised to transcripts per million (TPM) values), as many very low-expression features are likely to be the result of transcriptional noise or ambiguous read mapping. Finally, differential expression analysis, including all newly annotated putative features, is facilitated by a wrapper script that utilizes the DESeq2 method (Love *et al.*, 2014).

To test baerhunter, a simulated RNA-seq dataset was created using the package *polyester* (Frazee *et al.*, 2015). In addition, RNA-seq data from the study of (Cortes *et al.*, 2013), six samples from exponentially growing and starved cultures of *Mtb*, were downloaded from Array Express (E-MTAB-1616) and processed as detailed in Supplementary Methods. Following analysis with baerhunter, the sRNA/UTR predictions were compared to a set of experimentally confirmed and predicted mycobacterial sRNAs from the comprehensive review of (Haning *et al.*, 2014). Transcription start sites reported by (Cortes *et al.*, 2013) for the same samples were also used to assess the accuracy of our predictions. The genome browser Artemis (version 17.0.1) (Carver *et al.*, 2012) was used for visualization.

Rockhopper was used with default parameters, except for the minimum transcript length that was set to 40 nucleotides to match the baerhunter settings. Two minimum expression thresholds were tested (0.5, the default, and 0.2, to increase sensitivity).

Comparison with APERO and ANNOgesic was limited to the use of two datasets available in the corresponding publications. Baerhunter was run with default settings on a single file of mapped reads derived from the *Salmonella enterica* SRA dataset SRX1036363 (sample SRR2149882, paired-end data) and kindly provided to us by the APERO developers. We followed APERO's approach and assessed the accuracy of our ncRNA predictions using the Jaccard index (the fraction of the overlap over the union of intervals) for each predicted ncRNA with known coordinates. We then compared our predictions to predictions listed for both APERO and ANNOgesic in the Supplementary File S2 of the APERO publication. For a fairer comparison to ANNOgesic, we processed the *Campylobacter jejuni* GEO dataset GSE38883 used in the ANNOgesic publication (4 samples of single-end reads) and ran baerhunter on the mapped reads files with default settings. We then compared our overall predictions to the list of known sRNAs available from the ANNOgesic authors and to ANNOgesic results, as presented in their publication. In addition, we assessed the accuracy of the baerhunter 5' UTR predictions using the manually annotated subset of TSS available from the ANNOgesic authors.

All code and data required to reproduce this analysis are available from Zenodo:

(scripts) http://doi.org/10.5281/zenodo.3353102

(data) http://doi.org/10.5281/zenodo.3352787

The baerhunter code version used here is archived at Zenodo:

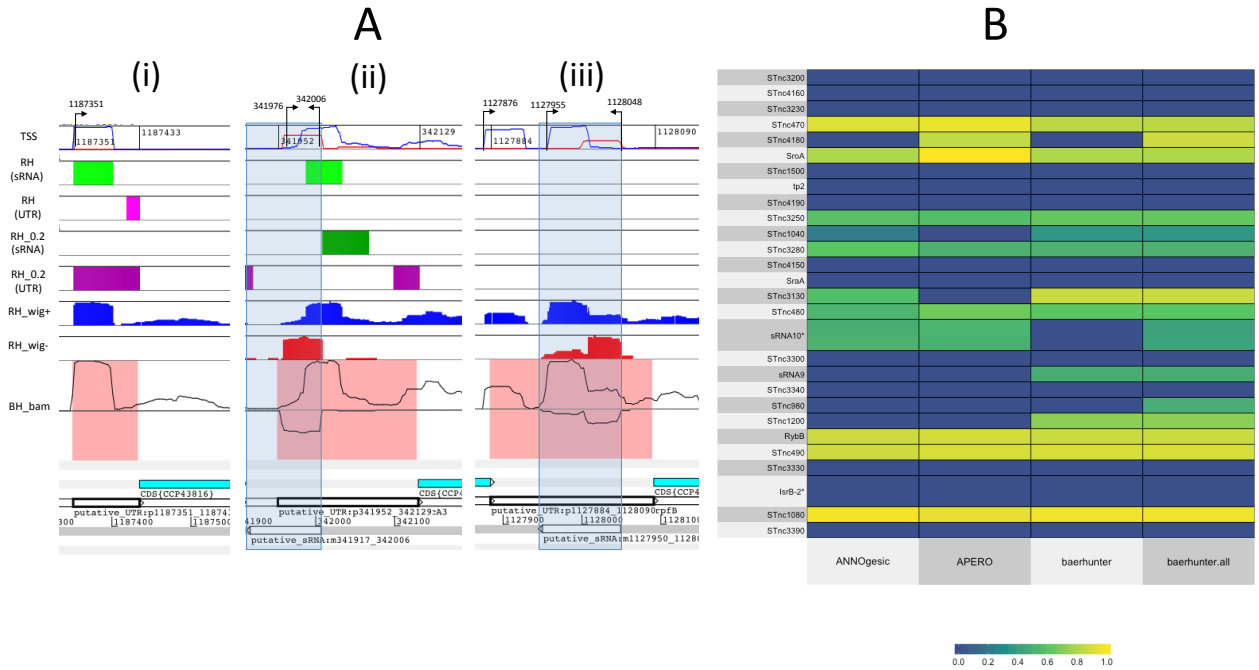
http://doi.org/10.5281/zenodo.3253339

The latest version of baerhunter is available from:

https://github.com/irilenia/baerhunter

3 Results

3.1 Simulated dataset



We tested the ability of baerhunter to recover expressed intergenic regions and UTRs using simulated data. 1000 genomic features were randomly selected from the *Mtb* genome, including twenty-four short RNAs included in the original annotation (see Supplemental Methods). As the genome annotation file does not include UTR information for *Mtb*, artificial UTRs were added to a random subset of 200 genes. These 1000 features were simulated in 10 samples belonging to two groups (with fold changes between 1 and 5 applied to 20% of the features). Our pipeline applied to paired-end read simulations recovered all short RNAs and all UTRs, with exact predictions for the start and end coordinates of all 24 sRNAs and over half of the UTRs (the remaining being in their vast majority within 5 nucleotides of the true range). Results were relatively insensitive to small changes in the program parameters (Supp Table 1). Rockhopper performed similarly on sRNAs, recovering 23 of 24 when run at default sensitivity (22 of the 23 sRNAs had their coordinates exactly predicted) but was less successful in the prediction of UTRs, missing 4 of the 200 and estimating the lengths of approximately a quarter of the ones it predicted to be at least 20 nucleotides shorter than expected (Supp Figure 4).

Fig. 1. Comparison of baerhunter's predictions with Rockhopper, APERO and ANNOgesic. A. In all three figures (i, ii and iii), the panels arranged in rows are: TSS: reads from the 5' sensitive sequencing data used to derive the TSS (shown as arrows pointing to the direction of transcription) are shown as blue (+ strand) and red (- minus) coverage lines based on normalised counts per base from the *Cortes et al.* (Cortes *et al.*, 2013) dataset; RH (sRNA) and RH (UTR): sRNA and UTR predictions from Rockhopper run with default sensitivity, 0.5; RH_0.2 (sRNA) and RH_0.2 (UTR): sRNA and UTR predictions from Rockhopper run with increased sensitivity, 0.2; RH_wig+ and RH_wig-: RNA-seq trace from Rockhopper mapping of reads for sample ERR262980 (blue: positive strand, red: negative strand); and BH_bam: the RNA-seq trace corresponding to read coverage in sample ERR262980 mapped by our own pipeline (see supplemental Methods). Filled pink rectangles in row "BH_bam" highlight the putative UTR region as predicted by baerhunter. Transparent blue rectangles (panels ii and iii) highlight the two short RNAs predicted by baerhunter on the negative strand. **(i).** The experimentally detected TSS (arrow in row "TSS") supports a 90 nt 5' UTR for the uncharacterized protein Rv1065. The baerhunter's prediction is 84 nt long (region highlighted with pink in row "BH_ham"), closely following the RNA-seq trace. Rockhopper's prediction is similar when run with the more sensitive detection threshold of 0.2 (dark pink rectangle, row "RH_0.2 sRNA") but run at the default 0.5, it splits the prediction to a "non-coding RNA" (green rectangle, row "RH sRNA") and a much shorter 5' UTR (bright pink, row "RH UTR") that is not supported by TSS data. **(ii).** The baerhunter program predicts a long (178 nt) 5' UTR ahead of the Rv0282 gene and an antisense RNA (90nt) partially overlapping this UTR. Both predictions are supported by experimentally detected TSS (black arrows pointing in opposite directions; row "TSS"). Rockhopper, run with default precision parameters, predicts a non-coding RNA (light green; panel "RH sRNA") and no 5' UTR (row "RH UTR"), whereas when run at higher sensitivity, it predicts a non-coding RNA further downstream (dark green, row "RH_0.2 sRNA") as well as a short 5' UTR (dark pink, row "RH_0.2 UTR"), corresponding to a weaker TSS just ahead of the coding region of the gene (small blue peak, row "TSS"). In both cases, Rockhopper misses the antisense RNA that is clearly seen in the RNA-seq trace of the exponentially growing bacteria (black trace on the negative strand, row "BH_bam" and red-fill trace, row "RH_wig-")**.(iii).**The baerhunter program discovers both the 5'UTR ahead of the Rv1009 (*rpfB*) gene and the antisense RNA overlapping it on the negative strand (both of which have experimental TSS support). Rockhopper does not predict any ncRNA features in the same region, although its own mapping of reads results in clear expression on both strands (rows "RH_wig+" and "RH_wig-"). The similarities between read coverage as reported by Rockhopper (rows "RH_wig+" and "RH_wig-") and our own pipeline (row "BH_bam") indicate that differences between the Rockhopper and baerhunter predictions are not due to differences in the way the reads were mapped to the reference genome but instead are due to the different ways the programs identify expressed regions. **B.** The heatmap represents Jaccard index values for coordinate predictions of a list of known *S. enterica* sRNAs. A Jaccard index ranges from 0 (no overlap between predicted and known sRNA transcripts) to 1 (predicted and known transcripts overlap fully and are on the same strand). Only the first 30 sRNAs from the list of 208 available in the APERO paper (Leonard *et al.*) are shown in this figure (full heatmap can be seen in Supp. Fig. 9 and a table of binned Jaccard indices is given in Supp. Table 3). The column labeled "baerhunter.all" includes predictions of transcripts that have been classified as putative UTRs by baerhunter but overlap known sRNA transcripts. It is clear that some of the UTR predictions contain known sRNAs but as they are found adjacent to gene features, they are considered UTRs by baerhunter. The full heatmap (Supp Figure 9 and Supp. Table 3) indicates that baerhunter's and APERO's predictions are more accurate than ANNOgesic's on this dataset (Wilcoxon signed-rank (paired) test, testing the hypothesis that the baerhunter and APERO Jaccard indices are greater than ANNOgesic's; *p*-value =

0.00608 and 0.00046 respectively). APERO is marginally better than baerhunter (Wilcoxon signed-rank (paired) test, *p*-value = 0.039) but if UTR predictions are included in baerhunter's predictions of these known sRNAs, then the APERO and baerhunter distributions of Jaccard indices show significant overlap (Wilcoxon signed-rank (paired) test, alternative hypothesis = "two sided", *p*-value = 0.078).

3.2 Real datasets

Comparison with Rockhopper

In addition to using simulated data, we applied baerhunter to the RNA-seq data from starved and exponentially grown cultures of *Mtb* (Cortes *et al.*, 2013). In this case, the true number of non-coding RNAs is unknown, so baerhunter was benchmarked against transcription start-site data from the same samples, as well as lists of known and predicted non-coding RNAs (Haning *et al.*, 2014) in order to assess the likely accuracy of our predictions.

At the more stringent parameter values (5-20), 74-83% of the predicted sRNA features in samples from either condition are supported by the presence of a TSS, even at one-nucleotide resolution (Supp Figure 5A & Supp Table 2). Relaxation of the cut-off (to 5-10) increases false positives but, importantly it also increases true predictions, thus allowing more transcripts to be discovered at the cost of a more noisy output (Supp Fig. 5B & Supp Table 2). Although more than half of the baerhunter-predicted sRNAs do not correspond to sRNAs in the published list of (Haning *et al.*, 2014), visual examination of the RNA-seq signal confirms expression at these loci (Supp Figure 6), usually from a very weak TSS that has not passed the inclusion cut-off in the original study by (Cortes *et al.*, 2013). These transcripts are often expressed at very low levels and can be easily filtered out using expression strength. Rockhopper, run at default expression cut-offs, not only predicts fewer sRNAs but also a smaller percentage of these predictions (~50-75%) are supported by TSS evidence (Supp Figure 5C&D). The prediction of UTRs is harder to assess. In the absence of ground truth for 3' UTRs, we compared the start of the 5' UTR predictions to TSS data. Differences were generally small when default parameters were used (Supp Figure 7). Rockhopper run at increased sensitivity predicted more of the 5' UTRs backed by TSS evidence but this increase was accompanied by an increase in the number of predictions without a match to a known TSS.

Comparison with APERO and ANNOgesic

To compare baerhunter to the more recently developed APERO and ANNOgesic methods, we used two datasets included in the publications describing these methods. Baerhunter's predictions of known sRNAs in *S. enterica* are comparable in accuracy to results from APERO and better than results from ANNOgesic on the same data (see Figure 1B and Supp Figure 9). As some of the sRNAs border CDS regions, baerhunter by design will label these as putative UTRs. Hence, if UTR predictions are also taken into account then baerhunter's accuracy is even higher (the number of sRNA transcripts predicted with some overlap to known transcripts goes up from 125 to 147 (out of a total of 208) and the median Jaccard index goes up from 0.49 to 0.62; the corresponding numbers for APERO are 125 and 0.56; for ANNOgesic they are 128 and 0.44).

Results for the *C. jejuni* dataset are less accurate but remain comparable to ANNOgesic for sRNA (see Supp Figure 10). Baerhunter predicts transcripts overlapping 22 of the 31 *C. jejuni* known sRNAs but this number goes up to 29, if UTR predictions are included. Hence, baerhunter recovers more of the transcripts in this list than the number reported by ANNOgesic (26 of 31). However, baerhunter's predictions of the transcript limits are a lot less accurate for this dataset than all others examined. The reason is that 9 of the 31 transcripts are incompatible with baerhunter's algorithm (one falls within the limits of a genomic feature in the original annotation file and is missed by design; the remaining eight would be too short to pass baerhunter filtering on minimum length and would have been missed entirely, had it not been for the fact that they form a cluster in the genome with overlapping expression peaks; these are seen by baerhunter's naïve algorithm as one long transcript). Nevertheless, the median Jaccard index for the set of both predicted sRNAs and UTRs is reasonable (0.53) and indicates that even in difficult cases, baerhunter's predictions can be a useful starting point for further exploration.

Baerhunter is much less sensitive in predicting 5' UTRs in the *C. jejuni* dataset, when benchmarked against the manually annotated subset of TSS on which ANNOgesic TSS prediction was trained. In this region where 162 5' UTRs (of length greater than 50 bases) are expected given the TSS list, baerhunter predicts only 33 5' UTRs (23 of which are within 10 bases of a TSS and the rest are associated with a clear signal in RNA-seq). Visual examination of missed cases suggests that the low sensitivity is due to two facts: a) baerhunter cannot detect 5' UTRs that are preceded by signal above the noise level (as is the case in intergenic regions of operons) and b) many TSS in the curated list do not correspond to significant read coverage in the RNA-seq signal. Although (a) can be addressed with future improvements to the software, we believe that baerhunter behaves correctly in cases covered by (b), as it was not designed to detect TSS but to report expressed UTR regions.

4 Conclusions

Our new method, baerhunter, allows the extraction of bacterial putative non-coding expressed regions directly from RNA-seq data and facilitates the integration of differential expression studies of coding and non-coding elements in bacterial transcriptomes. Importantly, baerhunter's results are of similar accuracy to two recent, more sophisticated methods and compare favourably with the most popular alternative method in tests with both simulated and real data.

Acknowledgements

We would like to thank Drs Sylvie Reverchon-Pescheux, Stephan Lacour and Simon Leonard for their help with accessing data relevant to the APERO publication.

Conflict of Interest: none declared.

References

Afgan,E. *et al.* (2018) The Galaxy platform for accessible, reproducible and collaborative biomedical analyses: 2018 update. *Nucleic Acids Res*, **46**, W537–W544.

Amman,F. *et al.* (2014) TSSAR: TSS annotation regime for dRNA-seq data. *BMC Bioinformatics*, **15**, 89.

Arnvig,K.B. *et al.* (2011) Sequence-Based Analysis Uncovers an Abundance of Non-Coding RNA in the Total Transcriptome of Mycobacterium tuberculosis. *PLOS Pathogens*, **7**, e1002342.

Carver,T. *et al.* (2012) Artemis: an integrated platform for visualization and analysis of high-throughput sequence-based experimental data. *Bioinformatics*, **28**, 464–469.

Cortes,T. *et al.* (2013) Genome-wide Mapping of Transcriptional Start Sites Defines an Extensive Leaderless Transcriptome in Mycobacterium tuberculosis. *Cell Reports*, **5**, 1121–1131.

Frazee,A.C. *et al.* (2015) Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics*, **31**, 2778–2784.

Haning,K. *et al.* (2014) Small RNAs in mycobacteria: an unfolding story. *Front. Cell. Infect. Microbiol.*, **4**.

Kalvari,I. *et al.* (2018) Rfam 13.0: shifting to a genome-centric resource for non-coding RNA families. *Nucleic Acids Res*, **46**, D335–D342.

Leonard,S. *et al.* APERO: a genome-wide approach for identifying bacterial small RNAs from RNA-Seq data. *Nucleic Acids Res*.

Love,M.I. *et al.* (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biology*, **15**, 550.

McClure,R. *et al.* (2013) Computational analysis of bacterial RNA-Seq data. *Nucleic Acids Res*, **41**, e140–e140.

Michaux,C. *et al.* (2014) Physiological roles of small RNA molecules. *Microbiology*, **160**, 1007–1019.

Pellin,D. *et al.* (2012) A Genome-Wide Identification Analysis of Small Regulatory RNAs in Mycobacterium tuberculosis by RNA-Seq and Conservation Analysis. *PLOS ONE*, **7**, e32723.

Peña-Castillo,L. *et al.* (2015) Detection of bacterial small transcripts from rna-seq data: a comparative assessment. In, *Biocomputing 2016*. WORLD SCIENTIFIC, pp. 456–467.

Pfeifer-Sancar,K. *et al.* (2013) Comprehensive analysis of the Corynebacterium glutamicum transcriptome using an improved RNAseq technique. *BMC Genomics*, **14**, 888.

R Core Team (2018) R: A Language and Environment for Statistical Computing R Foundation for Statistical Computing, Vienna, Austria.

Thorpe,H.A. *et al.* (2017) Comparative Analyses of Selection Operating on Nontranslated Intergenic Regions of Diverse Bacterial Species. *Genetics*, **206**, 363–376.

Toffano-Nioche,C. *et al.* (2013) Detection of non-coding RNA in bacteria and archaea using the DETR'PROK Galaxy pipeline. *Methods*, **63**, 60–65.

Wang,Y. *et al.* (2015) An empirical strategy to detect bacterial transcript structure from directional RNA-seq transcriptome data. *BMC Genomics*, **16**, 359.

Wilms,I. *et al.* (2012) Deep sequencing uncovers numerous small RNAs on all four replicons of the plant pathogen Agrobacterium tumefaciens. *RNA Biology*, **9**, 446–457.

Yu,S.-H. *et al.* (2018) ANNOgesic: a Swiss army knife for the RNA-seq based annotation of bacterial/archaeal genomes. *Gigascience*, **7**.