Evolutionary Bioinformatics

TECHNICAL ADVANCE

EDGE-pro: Estimated Degree of Gene Expression in Prokaryotic Genomes

Tanja Magoc[1], Derrick Wood[2] and Steven L. Salzberg[1,3]

[1]Center for Computational Biology, McKusick-Nathans Institute of Genetic Medicine, Johns Hopkins University School of Medicine, Baltimore, MD, USA. [2]Center for Bioinformatics and Computational Biology, University of Maryland, College Park, MD, USA. [3]Department of Biostatistics, Bloomberg School of Public Health, Johns Hopkins University, Baltimore, MD, USA. Corresponding author email: edge.comments@gmail.com

Abstract

Background: The expression levels of bacterial genes can be measured directly using next-generation sequencing (NGS) methods, offering much greater sensitivity and accuracy than earlier, microarray-based methods. Most bioinformatics software for estimating levels of gene expression from NGS data has been designed for eukaryotic genomes, with algorithms focusing particularly on detection of splicing patterns. These methods do not perform well on bacterial genomes.

Results: Here we describe the first software system designed explicitly for quantifying the degree of gene expression in bacteria and other prokaryotes. EDGE-pro (Estimated Degree of Gene Expression in PROkaryotes) processes the raw data from an RNA-seq experiment on a bacterial or archaeal species and produces estimates of the expression levels for each gene in these gene-dense genomes.

Software: The EDGE-pro tool is implemented as a pipeline of C++ and Perl programs and is freely available as open-source code at http://www.genomics.jhu.edu/software/EDGE/index.shtml.

Keywords: RNA-seq, bacteria, prokaryotes, gene expression

Evolutionary Bioinformatics 2013:9 127–136

doi: 10.4137/EBO.S11250

This article is available from http://www.la-press.com.

Introduction

Measuring the expression of genes in bacterial genomes has a very broad range of applications, from developing treatments for infections to creating synthetic genomes. Gene expression studies in bacteria have been used to study metabolic pathways, identify properties of mutants, and otherwise better understand the molecular processes in their genomes.[1,2] The first step in this process is to quantify gene expression for all the genes expressed in a particular experiment.

For more than a decade, microarrays have been the main means for studying gene expression. However, microarray technology can only capture transcripts for which probes are designed, therefore limiting its applicability to known genes in well-studied strains of a species. Alternatively, the use of quantitative PCR (qPCR) allows one to quantify specific genes rather than all genes in the genome, although this technique is far more costly on the scale of the whole genome. Recent improvements in the quality, efficiency, and cost of second-generation sequencing have led to an explosion in experiments (known as RNA-seq) that directly capture and sequence RNA, which has been replacing microarray analysis in recent years. In a microarray experiment, differences between the reference and a novel strain might prevent hybridization to some probes on the microarray. In contrast, in an RNA-seq experiment, transcribed genes are sequenced and aligned to the genome. Alignment algorithms can tolerate many mismatches, thereby allowing sensitive measurement of gene expression even when the target genome has diverged from the reference. In addition, because the entire transcript is sequenced, RNA-seq data also reveals the operon structure of a genome.[3,4]

Since the introduction of RNA-seq technology,[5,6] software methods have been developed to quantify gene expression[7] in a sample and to find differences in gene expression between multiple samples.[7–9] However, the current family of tools for estimating gene expression has been developed with the goal of identifying the structure of eukaryotic genes. These tools focus much of their effort on finding intronic regions within a gene and on finding alternative splice variants, which are common in higher eukaryotes. On the contrary, bacterial genes do not have introns and are not alternatively spliced; thus, there is no need to look for splice variants when analyzing their transcripts.

Bacterial genomes are also very densely packed with genes, many of them overlapping one another. Previous RNA-seq software methods generally do not provide a means of dealing with genes that overlap because these are extremely rare in humans and other mammals (the main targets of most RNA-seq experiments). In contrast, approximately 90% of a typical bacterial genome codes for proteins.[10] A study of 220 prokaryotic genomes spanning a wide evolutionary range[10] revealed that 29% of all genes in these species overlap another gene on either the 5′ or the 3′ end. These overlaps range from just a few base pairs (bp) to well over 100 bp. Overlapping genes can occur on the same strand or on opposite strands; thus, strand-specific RNA-seq offers at best a partial solution. For RNA reads that map within the overlapping region, it may be impossible to determine which of the 2 genes yielded the read, thus producing a challenge in determining gene expression of each prokaryotic gene.

Further complicating the requirements for analysis, bacterial RNA-seq technology has at least one major difference from eukaryotic RNA sequencing protocols, due to the absence of polyadenylation. The long poly-A tail on eukaryotic transcripts can be used as a capture probe, but bacterial RNA-seq method must instead rely on random priming to capture transcripts.[11] Another challenge is that more than 80% of captured bacterial transcripts are ribosomal RNA (rRNA). Although methods have been developed for removing rRNA, a large amount of rRNA still appears in some RNA-seq experiments.

Due to differences between eukaryotic and bacterial genomes and between RNA-seq protocols, the existing programs for expression analysis often perform poorly or break down entirely when applied to RNA-seq data from bacterial genomes. Therefore, new bioinformatics methods are required to estimate levels of gene expression in bacterial RNA-seq data. Currently, no stand-alone tool exists for this purpose. Multiple bacterial RNA-seq projects have been published,[12–20] but these have used ad hoc methods to quantify gene expression. All of these ad hoc methods first align input sequences ("reads") to a reference genome using a next-generation sequence aligner such as Bowtie, MAQ, SOAP, BWA, ELAND, Novoalign, or other custom-built aligners.[21–26] From these alignments, they count the number of reads mapped to each gene, usually normalizing counts by each gene's length.

Reviews of some of these ad hoc approaches can be found in Guell et al[27] and Van Vliet.[28]

One of the challenges faced by the standard alignment approach is that in every data set, some reads align to multiple places in the genome. It can be difficult and sometimes impossible to determine which of these multiple locations is the true source of the read, particularly if the source is repeated identically elsewhere in the genome. To avoid this problem, some previous methods simply discard multi-aligned reads. This strategy may significantly (and incorrectly) reduce the apparent expression levels of genes that contain repetitive sequences. A more serious problem arises for gene families, in which reads from any gene in the family may map equally well to all copies of the gene. Other methods of counting multi-aligned reads assign a fractional count to each location where a read maps or randomly assign a read to one of its multi-mapped locations. None of these provide a perfect solution, although, as we will show, the use of fractional read counts works well in practice.

While currently used ad hoc methods for estimating gene expression level in bacteria perform reasonably well, they are often not easy to use. Some require the user to run several software tools in succession, and the output of one program is sometimes not the correct input for the next tool, requiring ancillary programs to reformat the data. In this paper, we present a new program called EDGE-pro (Estimated Degree of Gene Expression in PROkaryots), the first stand-alone method specifically designed for estimating gene expression level in prokaryotic genomes. EDGE-pro is an efficient software system that provides solutions to the challenges mentioned above.

Methods

The EDGE-pro pipeline operates on four main inputs: the reference genome, a protein translation table (ptt) containing coordinates of protein coding genes in that genome, another table (rnt) containing coordinates of tRNA and rRNA genes, and the RNA-seq reads themselves. If the ptt and rnt tables are not available, they can be generated from the genome sequence separately, for example, by running Glimmer[29] to find protein-coding genes and running tRNAscan-SE[30] and RNAmmer[31] to find RNA genes.

The EDGE-pro pipeline consists of four mandatory steps: (1) the main mapping step, in which all reads are aligned to the reference genome; (2) a filtering step dedicated to multi-aligned reads; (3) computation of the depth of coverage of each base in the reference genome; and (4) conversion of raw coverage depth statistics into the RPKM value (reads per kilobase of gene per million reads mapped) for each gene.

Reads mapping

EDGE-pro maps reads to the reference genome using Bowtie2[32] with its default parameters, allowing up to 10 alignments for each read. Bowtie2 allows mismatches and small indels in the aligned reads, which gives it improved sensitivity over the mapping capability of its predecessors and especially over the ungapped aligners that have been used in some of the previous, ad hoc analysis methods for bacterial RNA-seq. The ability to allow indels is especially important when the reference genome is a different strain of bacterium from the source of the RNA, since even closely related strains often differ in many small insertions and deletions.

Filtering multi-mapped reads

In many cases, with up to 10 alignments per read available, some alignments are clearly better than others. For example, if 1 alignment contains 2 mismatches and a second alignment reports 10 mismatches, the first alignment is much more likely to be the correct one. On the other hand, if 1 alignment contains 1 mismatch and a second alignment contains 2 mismatches, the better match is not clear; for example, the second mismatch might be due to a sequencing error, in which case the 2 alignments are equally good.

The EDGE-pro pipeline filters reads that map to multiple locations based on the alignment score assigned by Bowtie2. Bowtie2 assigns a score of 0 to a perfect match and a negative score to each mismatch and indel. A mismatch gets a negative value between −6 and −1, with more negative values given to bases with higher quality values. The default penalty for an indel is −5 for opening a gap and −3 for each base in the gap. EDGE-pro considers multiple alignments using a threshold that is set based on the best-scoring read. If we let S be the highest score of all alignments for a read, then all alignments whose score is greater than minimum ($1.15 \cdot S$, S-3) are considered to be good alignments and all other alignments of the read

are discarded. By allowing mappings with the score S-3 to be considered as good alignments, EDGE-pro in practice allows for mismatches of 3 very low quality bases or 1 medium quality base (in addition to all penalties due to mismatches and indels of the highest-scoring alignment). Also, S-3 allows for an extra base in an already opened gap (ie, an indel that exists in the best scoring alignment). The penalty of −3 does not allow for an additional gap to be open in an alignment. Thus all alignments with scores higher than $1.15 \cdot S$ are also considered good alignments. This amount of deviation from the best-scoring alignment allows for additional gaps to be opened and extended if the best scoring alignment is not too high.

For example, if the best scoring alignment has only 1 mismatch, then no alignments with any gap will be considered. However, if the highest scoring alignment has 5 mismatches with low quality values, its score will be −10 (assuming that each base gets penalized by score of −2). EDGE-pro will keep all alignments with a score of at least $1.15 \cdot (-12) = -13.8$, which allows for a 2-base indels (−5 for opening the gap and −3 for each of 2 bases in the gap). It is not clear which of these 2 alignments is better (5 mismatches at low quality bases or 1 2-bases indel); thus, EDGE-pro considers both alignments as good alignments. The value of 1.15 was determined empirically by looking at hundreds of situations manually to determine which alignments are "reasonable competition" to the highest scoring alignment.

Per-base coverage

After filtering multi-mapped reads, the EDGE-pro pipeline determines the read coverage for each position in the genome. Uniquely mapped reads are simple: for these, EDGE-pro increments its coverage counts by 1 for each position in the genome covered by the read. Multi-mapped reads in bacterial genomes often map to duplicate genes, which contain identical or near-identical sequences, and it is not possible to determine which of 2 duplicate genes is expressed. EDGE-pro offers 2 options to handle these reads. The default option is to divide these multi-mapped reads, giving partial credit to each location where they might map, as follows. For each multi-mapped read remaining after filtering, the coverage of each position that the read spans is incremented by 1/N, where N is the number of "good" alignments for that read.

This heuristic will give the correct answer if all copies of a duplicated gene are expressed equally. Suppose instead that we have 2 identical copies of the gene, 1 of which is expressed at an RPKM level of 10 and the other silent. EDGE-pro will undercount the expression level and will assign half the reads to the unexpressed copy, reporting both copies with an RPKM of 5. If the 2 genes are identical, then this represents the same quantity of transcripts as if only 1 copy were expressed, and, in this respect, EDGE-pro's behavior is still correct; however, it might have different implications depending on the physical location and nearby regulatory sequences for each of the 2 copies.

The second option that EDGE-pro provides to deal with multi-mapped reads is to randomly choose 1 of all the positions where the read mapped and assign a full count to this position and ignore all the other positions.

The pipeline up to this point computes the per-base coverage for all genes across the genome. Calculating coverage per base overcomes the problem of trying to determine the source of a read that maps in the overlapping portion of 2 genes. Moreover, per-base coverage leaves open possibilities for further analysis such as visualization of uneven coverage across a gene and improvements to gene annotation, particularly at the start and stop coordinates of annotated genes.

Average gene coverage and RPKM calculation

In the last step, EDGE-pro converts coverage per base to the RPKM value for each gene. Note that the current version of EDGE-pro provides RPKM rather than FPKM values, even if paired-end reads are used. This provides a count for each read rather than each fragment that maps to a gene, therefore giving 2 counts for a pair of reads that map to the same gene rather than giving just 1 as FPKM does. This may provide a small disadvantage if an experiment uses paired-end sequencing and if a significant fraction of the fragments only yield 1 rather than 2 reads. However, because bacterial RNA-seq analysis does not need to link together exons across splice junctions, paired-end sequencing does not provide as much of an advantage. For single-end sequencing, RPKM and FPKM are equivalent.

Because many bacterial genes overlap, even to the point that a gene may completely contain another

gene on the opposite strand, the coverage in the overlapping regions should not be counted toward both genes. The easiest method to distribute the coverage in the overlapping region is to distribute it proportionally to the coverage in the nonoverlapping segments. This scheme should work on average, but it might be distorted due to nonuniform coverage of genes. To illustrate, Figure 1 shows a gene for which coverage is much higher at its 5′ end, skewing its average coverage. The relative coverage levels of genes 1 and 2 in the overlapping region, as shown in Figure 1, are better approximated by the coverage in the smaller windows (red boxes) adjacent to the overlapping region. The window size used here is a parameter that can be changed by a user. By default, EDGE-pro takes the window of length 100 bp, which is approximately the length of current Illumina reads.

Figure 2 shows the regions used to determine coverage of the overlapping portion of a gene. After identifying the region of overlap, EDGE-pro adjusts the coordinates of the nonoverlapping parts by a predefined length in order to exclude the untranslated region (UTR) on each side of the overlap. Because the RefSeq annotation provides only the coordinates of the protein coding regions and because the precise coordinates of UTRs are generally unknown, this heuristic adjustment reduces the bias in determining the coverage of each gene in its UTR regions. The UTR length is a parameter that could be adjusted by a user and is set to 40 bp by default since most bacterial UTRs are approximately 30 to 40 bp long. The coverage of a gene is calculated as follows:

1. The average coverage of the nonoverlapping portion of a gene is computed as the number of reads per base covering that region.
2. The average coverages of window1 (W1), window2 (W2), and the overlapping segment (O1) are calculated.
3. The ratio of these coverage values (W1/W2) is used to distribute the total coverage of the overlap regions, with each gene getting a proportional share of the coverage in the overlapping segment.
4. The ratio W1/W2 is also used to determine which portion of coverage in UTR1 and UTR2 comes from gene 1 and gene 2, respectively.

In Figure 2, the nonoverlapping part of each gene is longer than the window used to estimate coverage

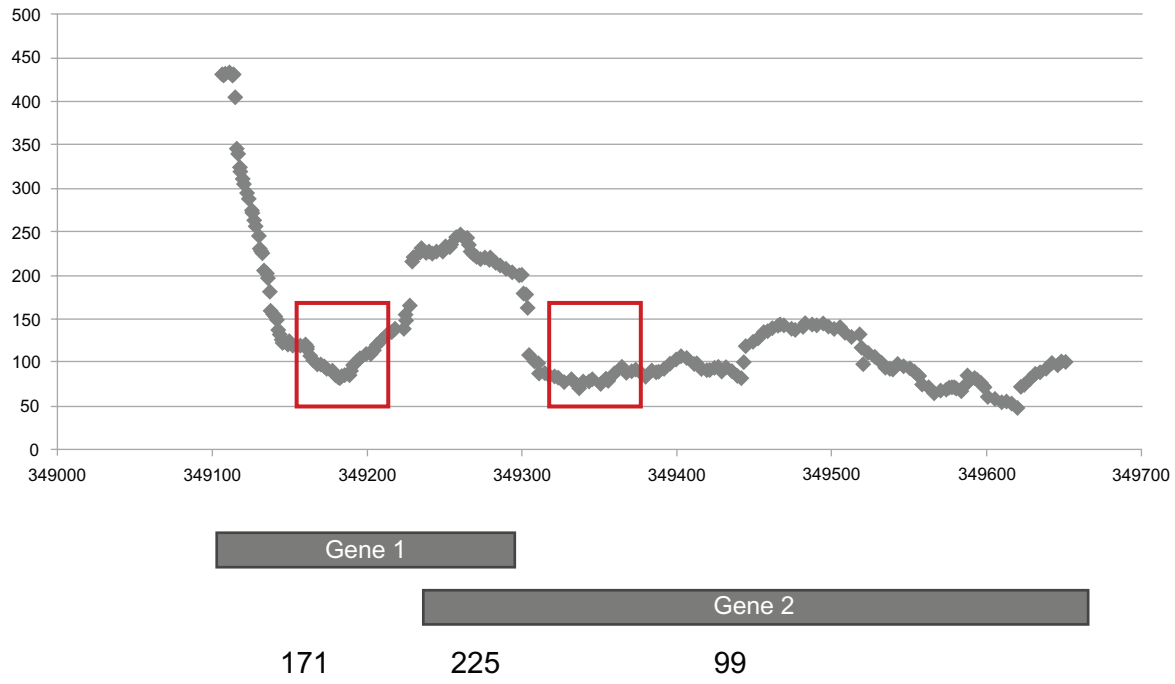

Figure 1. Nonuniform coverage of genes using data from an experiment on *E. coli*.
Notes: The horizontal axis represents the position in the genome, and the y-axis shows the number of reads mapped to the corresponding position. The positions of overlapping genes are represented by rectangles under the graph, and the average coverage in the nonoverlapping parts (171,99) and the overlapping region (225) are denoted under the genes. Under perfectly uniform coverage, the coverage in the overlapping region would be equal to the sum of the nonoverlapping parts. However, $171 + 99 > 225$. Gene 1 has much higher coverage at its 5′ end, distal from the overlapping region. Samples taken near the overlap (red boxes) provide a better approximation of each gene's coverage in the overlapping region.

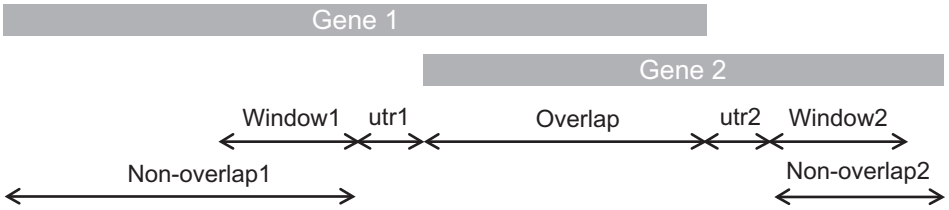

Figure 2. Overlapping genes.
Notes: After identifying the region of overlap, EDGE adjusts the coordinates of the nonoverlapping parts by a predefined length in order to exclude the untranslated region (UTR) on each side of the overlap. Because the RefSeq annotation provides only the coordinates of the protein coding regions and because the precise coordinates of UTRs are generally unknown, this heuristic adjustment reduces the bias in determining the coverage of each gene in its UTR regions. The ratio of coverage in windows of predefined sizes on both sides of the overlapping region is used to distribute the total coverage of the overlap region, with each gene getting a proportional share of the coverage in the overlapping segment.

(100 bp by default). If the nonoverlapping part of only 1 gene is long enough, only this gene is used to determine distribution of overlap coverage between the 2 genes.

Figure 3 illustrates the algorithm for computing coverage when a gene is completely contained within the coordinates of a gene on the other strand. The average coverage in sampled regions on either side of the contained gene (window1 and window2 in Fig. 3) is used as the estimated coverage of the longer gene. If the total coverage in the overlapping part is similar to or lower than the overall coverage of the larger gene, then EDGE-pro assumes that gene 2 is not expressed. By default, the coverages are considered to be similar if the coverage of overlapping region is within 15% of the overall coverage of the larger gene. This parameter is adjustable by a user. If the overlap coverage is not lower than or similar to the overall coverage of the larger gene, the difference between the overlapping segment's coverage and the coverage of the larger gene is assigned to the smaller, contained gene.

Once the average coverage of each gene is determined, we use the average coverage to estimate the number of reads that mapped to the gene: $R = C \cdot L/r$, where R is the number of reads mapped to the gene, C is the average coverage of the gene, L is the length of the gene, and r is the read length.

Next, we compute the RPKM for each gene, which normalizes the expression level with respect to gene length and the total number of reads mapped. To avoid counting low-level background noise as transcription, we set the RPKM value to 0 if the average coverage of a gene is less than 3 (a parameter that can be adjusted by the user). Otherwise, we calculate RPKM as $RPKM = R/(T/10^6)\cdot(L/10^3)$, where L is gene length, R is the calculated read count defined above, and T is the total number of reads mapped to the genome minus the number of reads mapped to rRNAs. The total number of reads T is divided by 10^6 and gene length is divided by 10^3 following the standard definition of RPKM, which normalize the read count per million bases mapped and per thousand bases in the gene. RNA sequencing protocols contain steps to remove ribosomal RNA molecules from the sample; however, these steps are not completely effective, and many rRNA reads remain in the sample that is sequenced. Because the number of remaining

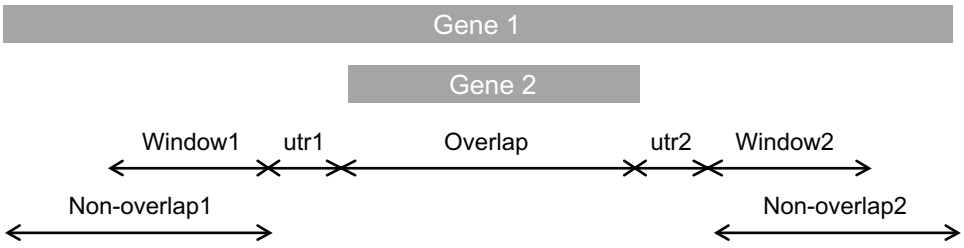

Figure 3. Contained genes.
Notes: When a gene is completely contained within the coordinates of a gene on the other strand, the average coverage in sampled regions on either side of the contained gene (window1 and window2 in Fig. 3) is used as the estimated coverage of the longer gene. If the total coverage in the overlapping part is similar to or lower than the overall coverage of the larger gene, then EDGE-pro assumes that gene 2 is not expressed. Otherwise, the difference between the overlapping segment's coverage and the coverage of the larger gene is assigned to the smaller, contained gene.

rRNA reads depends critically on the rRNA subtraction step rather than the inherent expression level of rRNA, subtracting these reads from T normalizes RPKM levels in a manner that should permit comparisons between multiple samples.

The EDGE-pro pipeline outputs for each chromosome and plasmid a report that contains the identity and coordinates of each gene, its average coverage *C*, the number of reads *R* mapped to the gene, and the gene's RPKM value. Additional outputs include the coverage per base by uniquely mapped reads, coverage per base by multi-aligned reads, and total coverage per base. EDGE-pro does not provide differential expression analysis between multiple samples, but the package provides a script to convert EDGE-pro output to format used by DESeq tool,[6] a stand-alone tool for differential expression analysis. Because EDGE-pro is open source software, it should be a simple task to convert its output for use by other differential expression packages.

Results

We tested the performance of EDGE-pro on RNA-seq data generated from the bacterial pathogen *Campylobacter jejuni*, strain NCTC11168, and a mutant strain with a defect in the gene *rpoN* (identified as *rpoN::cat*). The *rpoN* gene is involved in the transcription of flagellar genes,[33,34] but its precise function is not known. Inactivating *rpoN* was expected to downregulate the expression of some flagellar genes and possibly have other consequences in the mutant.

We downloaded RNA-seq data collected from 2 samples of *C. jejuni* NCNC11168 and 2 samples of the *rpoN* mutants (NCBI accession numbers ERR036497-ERR036500), all of which were sequenced in an earlier study by Chaudhuri et al.[15] Each sample contained over 7 million 37-bp single-end reads (Table 1), and the normal and mutant samples were technical replicates of each other. We ran EDGE-pro on each of the 4 samples and used the output values of EDGE-pro to find differentially expressed genes between wild type and mutant. We compared our results with those presented in Chaudhuri et al.[15]

Table 1. Reads used in the differential expression study.

	Number of reads	Number of reads mapped	Number and % of rRNA reads
Wild type 1	7169195	5523244	1427655 (26%)
Wild type 2	7064429	5377250	1382597 (26%)
Mutant 1	7255306	6180817	2594288 (42%)
Mutant 2	7007005	6046930	3091831 (51%)

For each pair of replicate samples, we simply averaged the RPKM values of the 2 samples and used the averages for comparison between the wild type and mutant strains, similarly to the method of Chaudhuri et al.[15] We defined upregulated and downregulated genes as those whose expression level increased or decreased at least 4-fold. Note that no conclusions about significance can be drawn from this comparison; our goal was simply to compare the computational analysis produced by EDGE-pro to previously published results on the same data. The differential expression in the Chaudhuri study was computed with DeSeq,[6] which computed a *P* value despite the very small sample size. We compared these results with those produced by our simple heuristic method.

We identified 20 genes that were downregulated in the mutant strain, *rpoN::cat*. These genes are listed in Table 2 along with the fold decrease. As expected,

Table 2. Downregulated genes in the mutant.

Gene	Product	Fold decrease
Cj0040	Hypothetical protein Cj0040	384.6
flgE2	Flagellar hook subunit protein	142.9
flgD	Putative flagellar hook assembly protein	85.5
fliK	Putative flagellar hook-length control protein	46.5
flgE	Flagellar hook protein	43.5
flgI	Flagellar P-ring protein	41.3
flgJ	Hypothetical protein Cj1463	36.8
Cj1242	Hypothetical protein Cj1242	36.2
flgH	Putative flagellar L-ring protein precursor	28.2
flgG2	Flagellar basal-body rod protein	18.3
flgK	Putative flagellar hook-associated protein	15.3
Cj0243c	Hypothetical protein Cj0243c	12.4
Cj1465	Hypothetical protein Cj1465	9.6
flgD2	Putative flagelline	7.2
flgG	Flagellar basal-body rod protein	6.7
flgB	Flagellar basal-body rod protein	6.4
Cj1650	Hypothetical protein Cj1650	6.1
Cj0716	Putative phosphor-2-dehydro-3-deoxyheptonate aldolase	4.7
Cj0044c	Hypothetical protein Cj0044c	4.4
Cj1004	Putative periplasmic protein	4.3

11 of these were flagellar genes (*flgE2, flgD, flgE, flgI, flgJ, flgH, flgG2, flgK, flgD2, flgG, flgB*). We compared our conclusions with the results presented in Chaudhuri et al,[15] which found 17 downregulated genes. Our 20 genes include these 17 genes as well as 3 additional genes (*Cj0716*, *Cj0044c*, *Cj1004*). The 3 additional genes had the smallest change in expression; if we used a threshold of 5-fold rather than 4-fold, we would identify precisely the same 17 genes as in the previous study (Table 2).

Supporting this hypothesis, the fold-change for 2 of these genes, *Cj1004* and *Cj0044c*, were just below the threshold for reporting a change in the Chaudhuri et al study.[15] Only *Cj0716*, which had a 2-fold change in Chaudhuri et al, was well below their threshold. Sixteen of the 17 downregulated genes were also reported as significantly downregulated in a separate microarray experiment described in Chaudhuri et al.

Thus, the differences between results presented in Chaudhuri et al and our results are mostly due to different thresholds used: the *P* value threshold used by Chaudhuri et al and 4-fold threshold used in our computation.

Another possible explanation for the small difference between these results is that the mutant samples contained a much higher proportion of rRNA reads (see Table 1), which EDGE-pro discards before computing RPKM values. Chaudhuri et al's analysis included these reads, which would have altered both the absolute and relative RPKM values for some genes. We also note that Chaudhuri et al used a different (commercial) alignment program, Novoalign, which may have changed some of the statistics.

We found that 11 genes were upregulated in the *rpoN* mutants, as shown in Table 3. Chaudhuri et al reported 10 upregulated genes, including all those in Table 3 except *hupB*. *HupB* was upregulated by a factor of 4 in Chaudhuri et al's data, just under their threshold for reporting. Thus the 2 methods also agree very closely on upregulated genes. Note that these comparisons do not speak to the significance of any of the upregulated or downregulated genes, which would require a larger number of samples. They do, however, show that EDGE-pro provides a reliable, streamlined computational solution to measuring expression levels in bacterial RNA-seq data.

The EDGE-pro pipeline only provides the expression level of each gene and does not compute differential expression statistics. The simple heuristic that we used here to compare our results with those presented in Chaudhuri et al[15] shows that even a simple heuristic applied to the EDGE-pro output matches the results produced by previously published, less automated ad hoc methods for bacterial RNA-seq analysis. Researchers who want to compute differential expression can easily feed the output of EDGE-pro into a separate program (eg, DeSeq)[6] designed for that task.

Table 3. Upregulated genes in the mutant.

Gene	Product	Fold increase
Cj0425	Putative periplasmic protein	16.23
Cj0423	Putative integral membrane protein	12.14
Cj0424	Putative acidic periplasmic protein	11.87
cysM	Cysteine synthase	6.10
cstA	Putative integral membrane protein	5.72
Cj0454c	Putative membrane protein	5.67
Cj0667	Putative S4 domain protein	5.45
Cj0898	Putative histidine triad (HIT) family protein	5.12
metB	Putative O-acetylhomoserine (thiol)-lyase	4.82
metA	Homoserine O-succinyltransferase	4.57
hupB	DNA-binding protein HU homolog	4.05

Computational Requirements

We measured time and memory requirements for EDGE-pro on a four-core 2.1 GHz AMD Opteron server with 512GB of RAM. To provide requirements relevant to current RNA-seq technologies, we used 101-bp reads for the timing experiments. We ran EDGE-pro on 2 samples containing 30 million RNA-seq reads from *Escherichia coli* strain E44, which were mapped to a very closely related strain, UTI89 (unpublished data). The first sample contained 12,587,318 multi-mapped reads (primarily rRNA reads), while the second sample contained only 410,024 multi-mapped reads. The runtime performance and memory requirements are shown in Table 4. EDGE-pro runs in both single- and multi-threaded mode, and the Table shows how the performance changes as additional threads (processors) are used. These experiments required a maximum of 4.2 GB of memory. The running time of EDGE-pro is dominated by the running time of Bowtie2, which in turn is linearly proportional to the read length and the number of reads.

Table 4. Time and memory requirements.

	Sample 1	Sample 2
Total number of reads	30000000	30000000
Number of multi-mapped reads	12587318	410024
Time in minutes: 1 thread	180	51
Time in minutes: 4 threads	42	17
Time in minutes: 8 threads	24	10
Time in minutes: 16 threads	14	7
Max memory requirement	4.2 GB	4.2 GB

Conclusion

Quantifying gene expression levels is the critical first step in analyzing RNA-seq experiments. Although several software systems have been developed for eukaryotic RNA-seq experiments, these algorithms do not perform well on bacterial organisms due to multiple differences between the experimental protocols and due to differences in gene structure and gene density. EDGE-pro is the first system to integrate, in a single software pipeline, a series of alignment and quantification methods specifically designed for prokaryotic genomes. The software includes algorithms specifically tailored for the overlapping gene regions commonly found in prokaryotic genomes.

Abbreviations

EDGE-pro, Estimated Degree of Gene Expression of PROkaryotic organisms; RNA, ribonucleic acid; rRNA, ribosomal RNA; tRNA, transfer RNA; RPKM, reads per kilobase of gene per million reads mapped; ptt, protein translation table.

Acknowledgement

Thanks to Joel Schildbach for providing the 101-bp *E. coli* reads used in the computational benchmarking tests.

Author Contributions

TM designed and implemented most of the software. DW designed and implemented some parts of the software. TM and SLS wrote the manuscript. All authors reviewed and approved of the final manuscript.

Funding

This work was supported in part by the National Institutes of Health under grants R01-LM006845S, R01-HG006677, and R01-HG006102 to S.L.S.

Competing Interests

Authors disclose no potential conflicts of interest.

Disclosures and Ethics

As a requirement of publication author(s) have provided to the publisher signed confirmation of compliance with legal and ethical obligations including but not limited to the following: authorship and contributorship, conflicts of interest, privacy and confidentiality and (where applicable) protection of human and animal research subjects. The authors have read and confirmed their agreement with the ICMJE authorship and conflict of interest criteria. The authors have also confirmed that this article is unique and not under consideration or published in any other publication, and that they have permission from rights holders to reproduce any copyrighted material. Any disclosures are made in this section. The external blind peer reviewers report no conflicts of interest.

References

1. Blencke HM, Homuth G, Ludwig H, Mader U, Hecker M, Stulke J. Transcriptional profiling of gene expression in response to glucose in Bacillus subtilis: regulation of the central metabolic pathways. *Metabol Eng*. 2003;5(2):133–49.
2. Shi J, Romero PR, Schoolnik GK, Spormann AM, Karp PD. Evidence supporting predicted metabolic pathways for Vibrio cholera: gene expression data and clinical tests. *Nucleic Acids Res*. 2006;34(8):2438–44. Accessed November 15, 2012.
3. Wurtzel O, Sapra R, Chen F, Zhu Y, Simmons BA, Sorek R. A single-base resolution map of an archaeal transcriptome. *Genome Res*. 2010;20:133–41.
4. Martin J, Zhu W, Passalacqua KD, Bergman N, Borodovsku M. Bacilus anthracis genome organization in light of whole transcriptome sequencing. *BMC Bioinformatics*. 2010;11(Suppl 3):S10.
5. Mortazavi A, Williams BA, McCue K, Schaeffer L, Wold B. Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat Method*. 2008;5(7):621–8.
6. Lister R, O'Malley RC, Tonti-Filippini J, et al. Highly integrated single-base resolution maps of the epigenome in Arabidopsis. *Cell*. 2008;133(3):523–36.
7. Trapnell C, Williams BA, Pertea G, et al. Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nat Biotechnol*. 2010;28:511–5.
8. Anders S, Huber W. Differential expression analysis for sequence count data. *Genome Biol*. 2010;11:R106.
9. Langmead B, Hansen K, Leek J. Cloud-scale RNA-sequencing differential expression analysis with Myrna. *Genome Biol*. 2010;11:R83.
10. Kingsford C, Delcher AL, Salzberg SL. A unified model explaining the offsets of overlapping and near-overlapping prokaryotic genes. *Mol Biol Evol*. 2007;24(9):2091–8.
11. Deutscher MP. Degradation of stable RNA in bacteria. *Journal of Biological Chemistry*. 2003;278:45041–4.
12. Albrecht M, Sharma CM, Dittrich MT, et al.The transcriptomal landscape of Chlamydia pneumonia. *Genome Biol*. 2011;12:R98.
13. Arnvig KB, Comas I, Thomson NR, et al. Sequence-based analysis uncovers as abundance of non-coding RNA in the total transcriptome of Mycobacterium tuberculosis. *PLoS Pathog*. 2011;7(11):e1002342.

14. Camarena L, Bruno V, Euskirchen G, Poggio S, Snyder M. Molecular mechanisms of ethanol-induced pathogenesis revealed by RNA-sequencing. *PLoS Pathog*. 2010;6(4):e10000834.
15. Chaudhuri RR, Yu L, Kanji A, et al. Quantitative RNA-seq analysis of the transcriptome of Campylobacter jejuni. *Microbiology*. 2011;157(10): 2922–32.
16. Liu JM, Livny J, Lawrence MS, Kimball MD, Waldor MK, Camilli A. Experimental discovery of sRNAs in Vibrio cholera by direct cloning, 5S/ tRNA deplation and parallel sequencing. *Nucleic Acids Res*. 2009;37:e46.
17. Mandlik A, Livny J, Robins WP, Ritchie JM, Mekalanos JJ, Waldor MK. RNA-seq-based monitoring of infection-linked changes in Vibrio cholera gene expression. *Cell Host Microbe*. 2011;10(2):165–74.
18. Perkins TT, Kingaley RA, Fookes MC, et al. A strand-specific RNA-seq analysis of the transcriptome of the typhoid bacillus Salmonella typhi. *PLoS Genet*. 2009;5:e10000569.
19. Sharma CM, Hoffmann S, Darfeuille F, et al. The primary transcriptome of the major human pathogen Helicobacter pylori. *Nature*. 2010;464:250–5.
20. Yoder-Himes DR, Chain PS, Zhu Y, et al. Mapping the Burkholderia cenocepacia niche response via high-throughput sequencing. *Proc Natl Acad Sci U S A*. 2009;106:3976–81.
21. Hercus C. Novoalign. Novocraft. http://www.novocraft.com. Accessed November 15, 2012.
22. Langmead B, Trapnell C, Pop M, Salzberg SL. Ultrafast and memory efficient alignment of short DNA sequences to the human genome. *Genome Biol*. 2009;10:R25.
23. Li H, Durban R. Fast and accurate short read alignment with Burrows-Wheeler Transform. *Bioinformatics*. 2009;25:1754–60.
24. Li R, Li Y, Kristiansen K, Wang J. SOAP: short oligonucleotide alignment program. *Bioinformatics*. 2008;24(5):713–4.
25. Li R, Yu C, Li Y, et al. SOAP2: an improved ultrafast tool for short read alignment. *Bioinformatics*. 2009;25(15):1966–7.
26. Maq: mapping and assembly with qualities. Sourceforge.net. http://maq.sourceforge.net. Accessed November 15, 2012.
27. Guell M, Yus E, Lluch-Senar M, Serrano L. Bacterial transcriptomics: what is beyond the RNA horiz-ome? *Nat Rev Microbiol*. 2011;9:658–69.
28. Van Vliet AHM. Next generation sequencing of microbial transcriptomes: challenges and opportunities. *FEMS Microbiol Lett*. 2010;302:1–7.
29. Delcher AL, Bratke KA, Powers EC, Salzberg SL. Identifying bacterial genes and endosymbiont DNA with Glimmer. *Bioinformatics*. 2007;23(6): 673–9.
30. Lowe TM, Eddy SR. tRNAscan-SE: a program for improved detections of transfer RNA genes in genomic sequence. *Nucleic Acids Res*. 1997;25(5): 955–64.
31. Lagesen K, Hallin P, Rodland EA, Staerfeldt HH, Rognes T, Ussery DW. RNAmmer: consistent and rapid annotation of ribosomal RNA genes. *Nucleic Acids Research*. 2007;35(9):3100–8.
32. Langmead B, Salzberg SL. Fast gapped-read alignment with Bowtie 2. *Nat Methods*. 2012;9:357–9.
33. Hendrixson DR, DiRita VJ. Transcription of sigma54-dependent but not sigma28-dependent flageller genes in Campylobacter jejuni is associated with formation of the flagellar secretory apparatus. *Molecular Microbiology*. 2003;50:687–702.
34. Wosten MM, Wagenaar JA, van Putten JP. The FlgS/FlgR two-component signal transduction system regulates the fla regulon in Camppylobacter jejuni. *J Biol Chem*. 2004;279:16214–22.