TSSAR: TSS Annotation Regime for dRNA-seq data

Fabian Amman[1,2] and Michael T. Wolfinger[2,3,4] and Ronny Lorenz[2] and Ivo L. Hofacker[2,5,9] and Peter F. Stadler[1,2,5,6,7,8] and Sven Findeiß[2,9]

[1] Bioinformatics Group, Department of Computer Science and the Interdisciplinary Center for Bioinformatic, University of Leipzig, Härtelstraße 16-18, 04107 Leipzig, Germany.
[2] Institute for Theoretical Chemistry, University of Vienna, Währingerstraße 17, 1090 Vienna, Austria.
[3] Center for Integrative Bioinformatics Vienna (CIBIV), Max F. Perutz Laboratories , University of Vienna & Faculty of Computer Science, University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria.
[4] Department of Biochemistry and Molecular Cell Biology, Max F. Perutz Laboratories, University of Vienna, Dr. Bohr-Gasse 9, 1030 Vienna, Austria.
[5] Center for RNA in Technology and Health, University of Copenhagen, Grønnegårdsvej 3, Frederiksberg C, Denmark.
[6] Max Planck Institute for Mathematics in the Sciences, Inselstraße 22, D-04103 Leipzig, Germany.
[7] Fraunhofer Institute for Cell Therapy and Immunology, Perlickstraße 1, D-04103 Leipzig, Germany.
[8] Santa Fe Institute, 1399 Hyde Park Road, Santa Fe NM 87501.
[9] Research group Bioinformatics and Computational Biology, Faculty of Computer Science, University of Vienna, Währingerstraße 29, 1090 Vienna, Austria.

Email: Fabian Amman*- afabian@bioinf.uni-leipzig.de;

*Corresponding author

Abstract

Background: Differential RNA sequencing (dRNA-seq) is a high-throughput screening technique designed to examine the architecture of bacterial operons in general and the precise position of transcription start sites (TSS) in particular. Hitherto, dRNA-seq data were analyzed by visualizing the sequencing reads mapped to the reference genome and manually annotating reliable positions. This is very labor intensive and, due to the subjectivity, error prone.

The numbers of sequencing reads starting at a certain genomic position within a transcriptional active region follow a Poisson distribution with a parameter that depends on the local strength of expression. The differences of two dRNA-seq library counts thus follow a Skellam distribution. This provides a statistical basis to identify significantly enriched primary transcription starts.

Results: Here, we present TSSAR, a tool for automated *de novo* TSS annotation from dRNA-seq data that respects the statistics of dRNA-seq libraries. We assessed the performance by analyzing a publicly available dRNA-seq data set using TSSAR and two simple approaches that utilize user-defined score cutoffs. We evaluated the power of reproducing the manual TSS annotation. Furthermore, the same data set was used to reproduce 74 experimentally validated TSS in *H. pylori* from reliable techniques such as RACE or primer extension. Both analyses showed that TSSAR has the potential to outperform the static cutoff-dependent approaches.

Conclusions: Having an automated and efficient tool for analyzing dRNA-seq data facilitates the use of the dRNA-seq technique and promotes its application to more sophisticated analyses. For instance, monitoring the plasticity and dynamics of the transcriptomal architecture triggered by different stimuli and growth conditions becomes possible.

The main asset of a novel tool for dRNA-seq analysis that reaches out to a broad user community is usability. As such, we provide TSSAR both as intuitive RESTful Web service (http://rna.tbi.univie.ac.at/TSSAR) together with a set of post-processing and analysis tools, as well as a stand-alone version for use in high-throughput dRNA-seq data analysis pipelines.

Keywords: differential RNA sequencing, dRNA-seq, TSS, Transcription start site annotation, Transcriptome, RESTful Web service, next generation sequencing

Background

Deep sequencing approaches were successfully applied to examine the architecture of primary bacterial transcriptomes and revealed that they are much more complex than previously believed [1–5]. While plain transcriptome sequencing can in principle be sufficient for transcription start site (TSS) prediction, extensive sequencing depth is generally required [6, 7]. In addition, the annotation of alternative TSS within operons poses difficulties for this strategy since they might lack a distinct signal. The recently developed differential RNA sequencing method dRNA-seq [4] makes use of the 5'-monophosphate dependent terminator RNA exonuclease (TEX) that specifically degrades processed RNA with a monophosphate at its 5' end. RNA with a protecting 5'-triphosphate, which is a characteristic feature of unprocessed 5' RNA fragment ends as they are produced in the course of transcription initiation, are left untouched. Treating RNA isolates with TEX prior to reverse transcription to cDNA, leads to a sequencing library ([+]-library or treated library) that is enriched in primary transcription starts, compared to an untreated total RNA library ([–]-library or untreated library).

Hitherto, the analysis of the dRNA-seq data consists of mapping sequencing reads for each library onto the reference genome, visualizing the read coverage in a genome browser, often with displayed gene and transcription unit annotation, promoter predictions and other available prior knowledge. In this context the genome is manually inspected for positions with a more pronounced peak in the [+]- compared to the [–]-library. This process is not only very time consuming, tedious, and error-prone, but also highly subjective and weakly reproducible. Additional annotation information from third-party sources has the potential to introduce biases, often resulting in re-annotation of already "known" features, and neglecting signals that are less obviously associated with current annotation data.

To overcome these shortcomings we developed `TSSAR` (TSS Annotation Regime), a tool for automated *de novo* TSS annotation from dRNA-seq data. Incorporation of information like gene annotation or promoter predictions is deferred to post-processing steps.

Implementation

Theory

Detailed knowledge of the underlying background distribution is required to quantify the significance of differential read start count signals. The background is variable along the genome, depending on the transcription activity of the considered region. In general we assume that the distribution of read starts within an expressed genomic region can be modeled by a Poisson distribution with parameter λ. Given λ the Poisson probability $P(Y = k) = \frac{\lambda^k e^{-\lambda}}{k!}$ describes the probability that k reads start at a genomic position. In dRNA-seq data genomic positions with significantly enriched differences between the Poisson distributions of [+]- and [–]-library are potential TSS. Therefore, we are concerned with finding positions where the observed difference cannot be explained easily by the local model of the background expression in the [–]-library. In the context of dRNA-seq [5], the differences can be modeled adequately by a Skellam distribution [8] with the cumulative distribution function

$$F(D, \lambda_{[+]}, \lambda_{[-]}) = \sum_{d=-\infty}^{D} e^{-(\lambda_{[+]}+\lambda_{[-]})} (\frac{\lambda_{[+]}}{\lambda_{[-]}})^{\frac{k}{2}} I_{|k|}(2\sqrt{\lambda_{[+]}\lambda_{[-]}}) \qquad (1)$$

Here $\lambda_{[+]}$ and $\lambda_{[-]}$ are the parameters describing the average read start rate in the [+]- and the [–]-library, respectively. $I_{|k|}$ is the modified Bessel function of the first kind and integer order $|k|$.

A major practical issue is the estimation of the parameters $\lambda_{[\pm]}$ for the two libraries. We assume that read start counts per position within transcriptional active regions follow a Poisson distribution, with the expected value λ depending on the transcription rate. Within untranscribed regions the background, neglecting sequencing and mapping errors, ideally follows a uniform distribution with the expected value zero. Consequently, randomly selected genomic regions are most likely a mixture of transcribed and untranscribed regions. To separate the two underlying distributions and estimate the parameter λ, describing only the transcriptionally active region, a zero-inflated Poisson model regression [9, 10] is applied. For each sample Y the probability ϕ that an observed zero is a structural zero (i.e., part of a transcriptional inactive region and thus from a uniform zero distribution) and not part of the transcriptional active region is estimated, such that

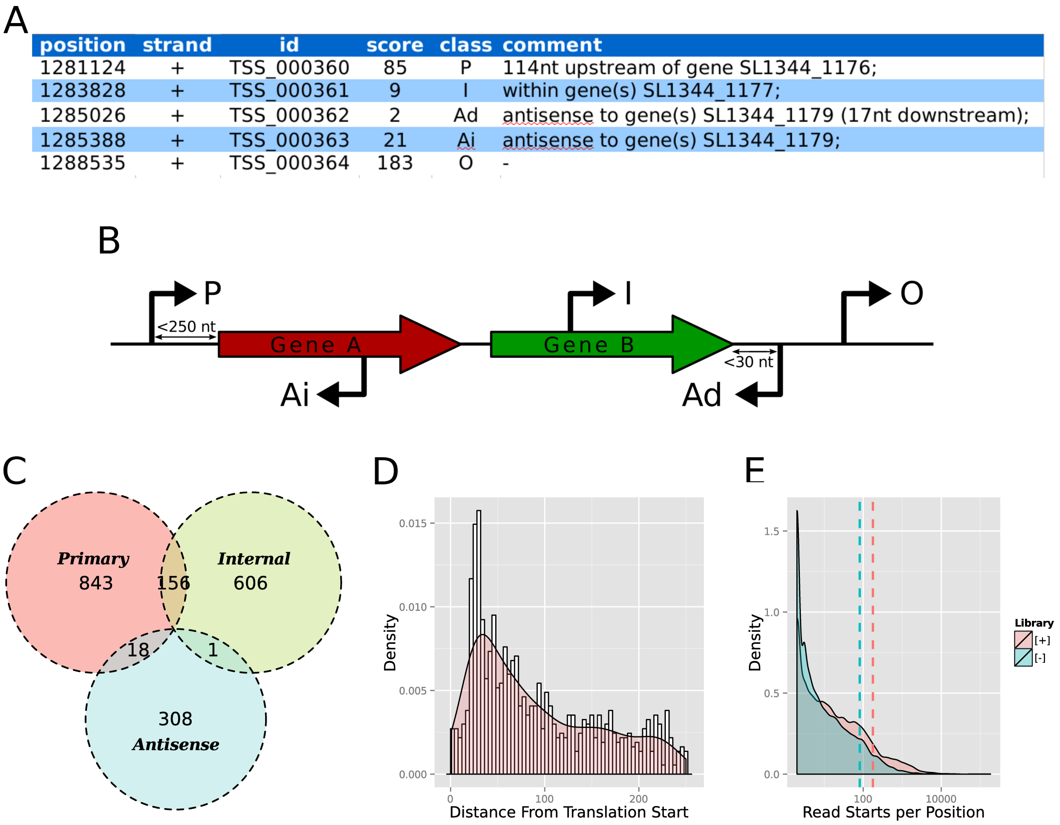
$$P(Y = 0) = \phi + (1 - \phi) \cdot e^{-\lambda} \tag{2}$$

where $e^{-\lambda}$ is the probability for a position within the Poisson distributed part to have zero reads starting there (sampling zero). These positions are part of transcriptional active regions. This framework can be used to determine how many positions without read starts are structural and sampling zeros, respectively. Only the latter and positions that have at least one read start are used to estimate λ of the [+]- and [–]-library, respectively. The estimation of λ thus effectively considers the transcriptionally active regions only.

Program architecture

TSSAR has been implemented in Perl and R and is available in two variants: A stand-alone version incorporates the core statistic routines and is best suited to be used in custom high-throughput dRNA-seq analysis. The Web service (available at http://rna.tbi.univie.ac.at/TSSAR/) comprises additional components for pre- and post-processing, thus providing a Web-based, cross-platform compatible pipeline for dRNA-seq analysis.

The TSSAR **Web service** is built on top of the Perl Dancer [11] framework and adheres to the Representational State Transfer (REST) [12] principles of Web architecture. The first step in using the TSSAR online pipeline is pre-processing of mapped reads, i.e. extracting the essential information of read start counts per genomic position. To avoid the necessity of uploading huge mapping files (typically for bacterial genomes up to several gigabytes), we implemented the TSSAR **client** for local pre-processing of mapped reads in SAM/BAM or BED format on the user's computer. To grant platform independence, the TSSAR client is implemented in Java. Once the relevant data is extracted from the mapping files assisted by the Picard tools [13], files are compressed using XZ utils [14] and automatically transferred, using the Apache HttpComponents [15] package, to the TSSAR Web server. On the Web server the statistical calculations are conducted and potential TSS are predicted. The TSSAR Web service provides an assortment of post-processing steps. The list of predicted TSS can be reduced by merging consecutive TSS to the most prominent position. For samples where the reference genome annotation was specified, all annotated TSS are classified into primary, internal, anti-sense or orphan, according to their position relative to nearby genes, see Figure 1B. Based on the classification the 5' UTR length distribution is determined. All results are visualized and provided for download. Figure 1 depicts part of the output for a showcase data set of *Salmonella*



position	strand	id	score	class	comment
1281124	+	TSS_000360	85	P	114nt upstream of gene SL1344_1176;
1283828	+	TSS_000361	9	I	within gene(s) SL1344_1177;
1285026	+	TSS_000362	2	Ad	antisense to gene(s) SL1344_1179 (17nt downstream);
1285388	+	TSS_000363	21	Ai	antisense to gene(s) SL1344_1179;
1288535	+	TSS_000364	183	O	-

Figure 1: **Post-processing and Visualization.** (A) Each annotated Transcription start site is classified according to its genomic context. Therefore the following scheme (B) is used: If it is positioned within 250 nt upstream of an annotated gene, it is classified as **P**rimary. TSS within an annotated gene is labeled **I**nternal. TSS which is on the opposite strand of an annotated gene, is labeled **Ai** and **Ad**, for internal antisense and downstream antisense, respectively. The later is reserved for a TSS which points in the opposite reading direction and is less than 30 nt downstream of an annotated gene. A TSS that falls in none of these classes is reported to be **O**rphan. (C) As a matter of fact, one TSS can have several labels as it might fall into more than one of the aforementioned classes. The `TSSAR` Web service summarizes the counts for each class and their overlaps graphically. (D) For TSS which are annotated as 'Primary' the 5'UTR lengths are deduced and the corresponding distribution is plotted. (E) To assess the efficiency of the TEX treatment, the distribution of read starts per position is provided as a helpful indicator. If the enrichment in the [+]-library worked efficiently, its distribution and its mean (dashed line) is expected to be shifted to the right, i.e. more positions with more reads compared to the minus library.

typhimurium [16]. Beside the shown results, the output additionally contains all annotated TSS and the clustered TSS list in BED and GFF format. All tables are available in comma and tab-separated lists, as excel and html files. With the assistance of the pre-computed plots, it is easy to gain a quick overview of the quality of the analysis.

While the TSSAR Web service provides convenient usability for routine dRNA-seq analysis tasks, there is also a demand for integrating third-party bioinformatics tools into custom analysis pipelines. To address this issue, we provide a **TSSAR stand-alone version**. In this version, the implementation is restricted to processing of SAM files, analysis based on the statistical calculations, and output of annotated TSS in BED format [17]. The stand-alone version is available for download from the TSSAR Web site.

Statistical calculation

We chose a sliding window approach with a dynamic assessment of each position in the context of its local surrounding in order to account for different transcription rates across the genome. The default window size is 1,000 nt, matching the average length of prokaryotic genes. It can be easily adjusted by the user. For each window the parameters describing the Poisson distribution are estimated in the following manner:

First, the sample values are winsorized [18], i.e., the highest read start count is substituted with the second highest count. The same procedure is done for the lowest value. This increases the robustness of the method against outliers, which may be caused by mis-mapping and/or abundant RNA fragments e.g. arising from rRNA loci.

Second, the zero-inflated Poisson regression is applied to estimate ϕ, the probability that an observed zero is a structural zero from an untranscribed region instead of a sampling zero from a transcribed region. The R package VGAM is used for the regression [10, 19]. Here, the parameters describing the Poisson distribution are fitted by full maximum likelihood estimation (MLE). In case the MLE algorithm fails to converge, which might happen because the underlying assumption of a well behaved Poisson distribution is violated, the respective window is excluded from further analysis. While this might seem to be a drawback, it can be used as a minimal plausibility check, ensuring the data fulfills the underlying assumption of following a Poisson distribution. Sequencing libraries with low complexity but many PCR duplicates might otherwise feign confidence in the results, which can actually not be deduced from the data. A BED file listing the omitted genome segments which typically correspond to less than 1% of the genome is provided (see Figure 2). In the evaluation data set (see section Evaluation) modeled with a window size of 1,000, 24 regions with a total length of 12,000 bases could not be modeled (~0.5% of the genome). The majority (10 and 5) correspond to

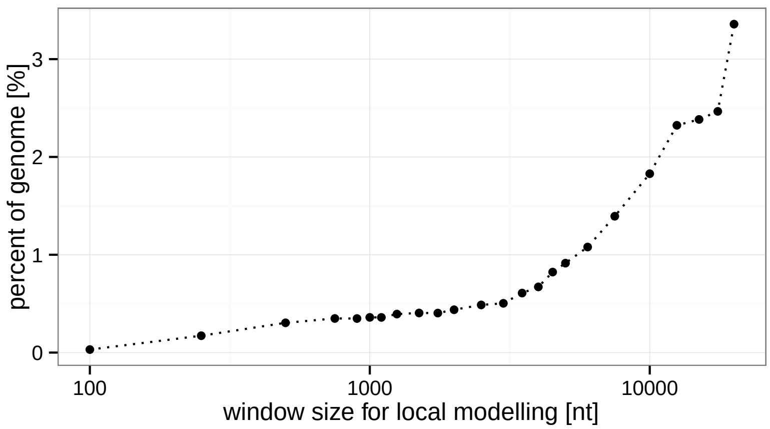


Figure 2: **Regions of non-convergence.** Regions where the applied zero-inflated Poisson regression does not converge are omitted from the analysis and need manual inspection. Since the basic unit which cannot converge is the step size (equals a tenth part of the windows size) there is a correlation between the parameter window size and the percentage of the genome which can not be modeled. The *H. pylori* dRNA-seq data (see section Evaluation) shows that for all practical useful window sizes below 5,000 nt, less then 1% of the genome eludes analysis.

tRNA and rRNA coding loci. Additionally, 4 regions overlapped with annotated protein coding genes and the remaining 5 did not overlap with any annotated gene.

Third, a regression procedure is applied to each window and the [+]- and the [–]-library separately. For each library the probability ϕ is transformed into an expected number of excess structural zeros. Since the same genomic region is under consideration in both libraries, a similar proportion of untranscribed and transcribed regions can be expected. To increase robustness, the average between the number of structural zeros in both libraries is calculated and the estimated number of zeros are removed from each library. To determine λ for each library, describing the Poisson distribution of the sample, the arithmetic mean of the remaining counts is calculated.

In the next step the probability that the read start differences between [+]- and [–]-library can be explained by the aforementioned background model is calculated. For this purpose, the original read start counts are normalized by

$$\widehat{p_i} = \begin{cases} p_i \cdot \frac{\sum M}{\sum P} & , \sum M > \sum P \\ p_i \cdot 1 & , \sum M \leq \sum P \end{cases} \tag{3}$$

$$\widehat{m_i} = \begin{cases} m_i \cdot \frac{\sum P}{\sum M} & , \sum P > \sum M \\ m_i \cdot 1 & , \sum P \leq \sum M \end{cases} \tag{4}$$

Thereby, p_i and $\widehat{p_i}$ are the raw and normalized values in a window of the [+]-library at position i, respectively.

$\sum P$ and $\sum M$ are the native sums of all read start counts in the total [+]- and [–]-library, respectively. The same applies to the [–]-library, i.e. m_i and $\widehat{m}_i$. The effect of this step is to scale the read counts of the larger library relative to the smaller one, hence avoiding artificial distending of the sample variance. The estimated parameters $\lambda_{[+]}$ and $\lambda_{[-]}$ are therefore normalized accordingly.

For each sequence position i in the current window, the difference $\widehat{d_i} = \widehat{p}_i - \widehat{m}_i$ of the normalized counts between [+]- and [–]-library is calculated. Unexpectedly large positive values of $\widehat{d_i}$ indicate TSS, while exceptional negative values may indicate processing sites. The probability of observing $\widehat{d_i}$ is evaluated w.r.t. the Skellam distribution with the estimated normalized Poisson parameters.

The window slides along the genome with a step size equal to $1/10^{th}$ of the window size, hence each position is evaluated in 10 slightly different contexts. The geometric mean of all ten *p*-values is calculated in order to obtain the final position-wise *p*-value. Finally, each position that falls below a user-specified average *p*-value cutoff and whose total read start count in the [+]-library exceeds a user specified noise cutoff is reported as a significant TSS.

Results

The goal of the TSSAR method is to provide user-friendly tools for rapid annotation of significant TSS based on dRNA-seq data. We therefore implemented a stand-alone version and a Web service. The first is intended to be used in high-throughput analysis pipelines whereas the latter represents an easy to use and platform independent user interface. For a Web service it is important to avoid the transfer and storage of gigabyte-sized mapping files. We therefore provide a Java client that extracts the necessary information and asks the user for only two parameters, namely genome size and window size. The data is pre-processed locally on the user's computer. The essential information, i.e. the number of sequencing reads starting at each position, is automatically uploaded and analyzed on the TSSAR Web server. All relevant cutoffs like *p*-value and noise threshold are subsequently selectable for precomputed values.

Evaluation

To evaluate the performance of TSSAR in analyzing dRNA-seq data, we resort to the published data set for *Helicobacter pylori* [4]. We used the publicly available raw sequencing data from the Sequence Read Archive [20] (study accession number SRP001481), restricting ourselves to the dRNA-seq data from mid-logarithmic growth phase and acid stress growth condition. The reads were pooled and mapped to the reference genome (NCBI accession ID NC_000915) using segemehl version 0.1.4 [21] with default parameters.

Based on this data we predicted putative TSS with three different approaches. The first two represent a naïve benchmark. First, we calculated the difference $(p_i - m_i)$ for each position i of the [+]- and [–]-library read start counts. We applied a different cutoff threshold between 1 and 300, thereby denoting every position with a difference higher than the cutoff to be a putative TSS. The resulting list of potential TSS was compared to the manual annotation from [4] using `bedtools intersect` [22], allowing ±2 nt inaccuracy to call a manual and an automated annotated TSS the same. The second approach is quotient based. Analogous to the difference based approach, the quotient $\frac{p_i+1}{m_i+1}$ is calculated for each position i (+1 is used as pseudo-count to avoid division by zero). Again we use different cutoff values between 1.1 and 20. These two approaches have their static nature in common. For the whole genome the same threshold is applied. A similar approach was already applied by [23]. Albeit, there it was used to identify differentially induced TSS between different strains and growth conditions and additional information about promoter sequences was used to gain specificity.

Finally, we applied the dynamic `TSSAR` model, which analyzes the transcriptome locally and thus is able to model the different dynamics within the transcriptome. Here, we used a window size of 1,000 nt (approximately the mean gene length in *H. pylori*) and a noise cutoff of 3 reads per position. We filtered with different p-value threshold from $1 \cdot 10^{-15}$ to $9 \cdot 10^{-1}$.

From these results, each threshold based prediction is evaluated using standard measurements: recall rate $(\frac{TP}{TP+FN})$, precision $(\frac{TP}{TP+FP})$, accuracy $(\frac{TP+TN}{TP+FP+FN+TN})$ and the F-measure $(2 \times \frac{precision \times recall}{precision+recall})$ [24], where TP, TN, FP and FN are true positive, true negative, false positive and false negative predictions, respectively. Figure 3 depicts the results of this analysis. `TSSAR` shows a much higher precision and simultaneously a less sharp decrease of the recall rate. In terms of the F-measure, it outperforms the fixed-threshold approaches by about 2-fold. A further advantage is the smoother course of the F-measure along different p-value cutoffs. This makes the resulting annotation less dependent on the cutoff choice. The optimal cutoff value for the basic annotation strategies based on difference or ratio might be very variable for different experiments and difficult to deduce without a reference annotation.

Additionally, besides comparing our automated annotation to the manual annotation by the authors, we sought to reproduce *H. pylori* TSS which were annotated by independent methods, such as primer extension or 5' RACE. From the 74 examples described in the literature (summarized in supplementary material of [4]), we calculated the distance to the closest position which we annotated as TSS. If the discrepancy was more then 10 nt, we considered the TSS as not recovered. Figure 4 shows the result of this analysis for two `TSSAR` annotations with different parameters. The first one with lenient threshold values (aiming for sensitivity),

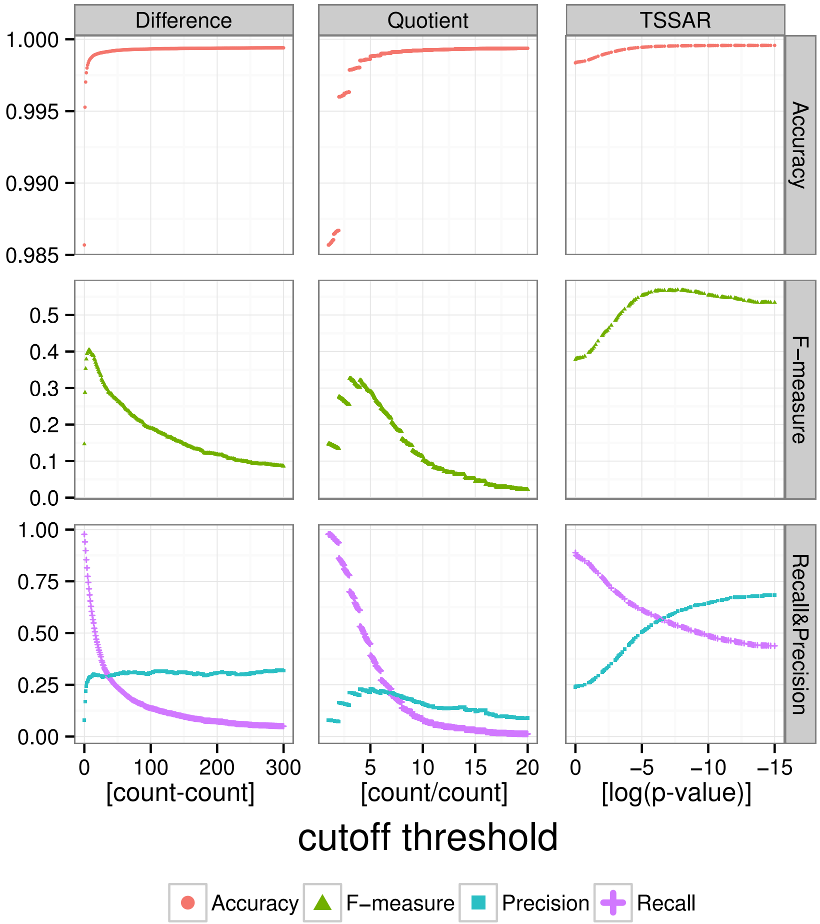


Figure 3: **Evaluation of `TSSAR` performance.** Comparison of the prediction power of `TSSAR` against two fixed-cutoff approaches *Difference* and *Quotient*. For each method different cutoff thresholds were applied. The difference, quotient and logarithm of the p-value are plotted along the x-axis. Please note, for comparability the log(p-value) is plotted in descending order from left to right. The resulting predictions were evaluated by calculating the recall rate, precision, F-measure and accuracy. The dynamic approach of `TSSAR` clearly outperforms the remaining in all aspects.

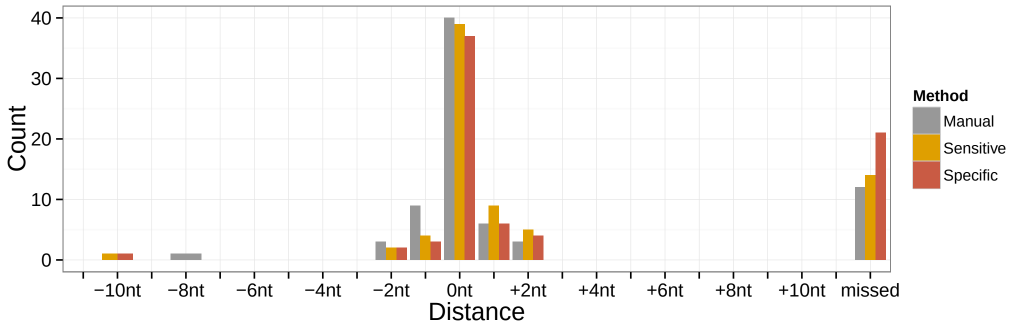


Figure 4: **Recall experimental validated TSS.** Comparison of 74 experimentally validated TSS described in literature [4] with TSSAR results. The *Manual* TSS annotation recovered 40, 15 and 6 TSS with a 0, ±1 and ±2 nt offset, respectively. Here 12 TSS were annotated more than 10 nt away from the experimentally determined position (summarized as *missed* in the plot). TSSAR was run with a *Sensitive* and a *Specific* parameter set (p-value cutoff 0.05 and 0.0001; noise cutoff 1 and 3, respectively). With sensitive parameters 39 TSS (53%) were annotated on the exact same position. Of the remaining TSS 13 and 7 were annotated with ±1 and ±2 nt variance, respectively, whereas 14 TSS (19%) were annotated more than 10 nt away. The specific TSSAR prediction annotated 37, 9 and 6 TSS with 0, ±1 and ±2 nt offset, respectively, relative to the experimentally validated position. In this case 21 TSS (28%) were annotated more than 10 nt away, and therefore annotated as missed.

and the later with more stringent values (aiming for specificity). In both cases the majority of experimentally confirmed TSS could be detected at the exact same position (39 and 37 TSS, respectively). TSSAR missed 14 and 21 TSS, respectively, compared to the 12 TSS that were also not detectable in the manual annotation by the authors of [4]. We have to emphasis that, in contrast to a manual annotation, our method is not aware of any annotation information, which might induce a human curator to prefer certain positions.

Discussion

The automated TSS prediction is only as good as the underlying dRNA-seq libraries. We therefore emphasize that a thoughtful investigation of the input sequencing reads, especially for PCR duplicates, is advised.

A major advantage of an automated TSS annotation, based on a sound statistical analysis, neglecting *a priori* knowledge of the whereabouts of promoters and other already established annotation, lies in the avoidance of any bias towards certain genomic positions. This ensures an unbiased analysis as well as a comparable and reproducible TSS annotation procedure.

Although our approach checks whether the basic assumption that the read starts of a sequencing library are Poisson distributed holds, a manual inspection of the produced data is still recommended. Manual inspection is of course necessary for those genomic regions that are not annotated by TSSAR due to non-convergence in the estimation of the expression parameters. For TSSAR's output, we recommend at least a basic sanity check, since very complex regions, such as tRNA and rRNA loci, might be misconstrued. In spite of these precautions, the work load to check hundreds or a few thousands of predicted TSS positions is significantly reduced compared to screening millions of genomic positions in the first place.

Reliable and automated TSS annotation is a prerequisite for many applications. So far, most genome-wide TSS annotations focused on a static picture of the transcriptomal architecture [2,25] (of course there are also notable exceptions, e.g. [23,26]). One reason is that data analysis was more laborious than data generation. Relieving the experimenter from this time-consuming burden might liberate the resources to investigate more of the dynamics and alteration of the transcriptome, due to external stimuli or evolutionary differences. During manuscript preparation the latter was demonstrated by conducting a comparative transcriptomics approach [27]. There, TSS annotation was also conducted in an automated manner. The problem of selecting an educated cutoff, which is immanent to all methods but especially troublesome for classifiers which directly depend on variable conditions such as sequencing depth, was neatly circumvented by using a comparative approach. Transcriptomes of different *Campylobacter jejuni* isolates were used to dynamically adjust thresholds if signals in different strains could be observed. In the more typical application scenarios, where such comparative information is not available, a robust p-value estimate that takes the dynamic range of transcription activity along the whole genome into account for the classification seems to be preferable.

Currently, `TSSAR` is based on the assumption that a [+]- and [–]-library is analyzed and only positions with a significant enrichment in the [+]-library are reported as potential TSS. At least two other application scenarios of the statistical framework are possible. One is to detect RNA processing sites and the other to analyze deferentially induced transcription starts. In principle the latter could be achieved by comparing two TEX treated libraries resulting from dRNA-seq runs of different growth conditions. In that case, a large positive and negative $\widehat{d_i}$ is of importance as it indicates (growth phase dependent) induction of a TSS in the one or the other library. RNA processing sites are in principle detectable using the "standard" dRNA-seq approach. Positions where a significant enrichment in the [–]-minus over the [+]-library is observable are of interest. Extreme negative values of $\widehat{d_i}$ point to these positions. Tackling both issues, processing sites and induced transcription initiation, is however currently hampered by the lack of experimentally verified training sets. Furthermore, although tailored for analyzing dRNA-seq data, in principle, the `TSSAR` method should be applicable to other RNA-seq protocols, e.g., [28], which aim to enrich read starts at certain positions in the sequencing library.

The modularity of the `TSSAR` framework makes it possible to extend the current approach e.g., by improving the statistical model. Alternative approaches based on a different (non-Poisson) distribution or the Pittman sampling method [6] can be implemented in the TSSAR core module, without the necessity to change the Java client or the Web service front end. The RESTful architecture of the TSSAR Web service provides additional extensibility, rendering implementation of new functionality such as promoter or operon characterization straightforward.

Conclusion

Here, we presented an automated analysis of dRNA-seq data which aims to detect significantly enriched TSS positions. The background distributions of sequencing read starts are modeled locally by a zero inflated Poisson distribution. Positions with a larger difference between the TEX treated and the untreated library than expected, considering the background, are annotated as significant transcription start sites. We could show that our method reproduces manually analyzed dRNA-seq data better than two simple approaches that use a global cutoff to discriminate between true and false signals. Furthermore, the choice of a *p*-value cutoff is more intuitive and less arbitrary.

TSSAR is available both as a stand alone tool and as a Web service at http://rna.tbi.univie.ac.at/TSSAR/. The latter provides additional post-processing functionality like TSS classification or merging of consecutive TSS. The TSSAR Web service offers user-friendly and intuitive online access to the TSSAR framework whereas the stand-alone version is intended for integration into third-party annotation pipelines.

Availability and requirements

- **Project name:** TSSAR
- **Project home page:** http://rna.tbi.univie.ac.at/TSSAR
- **Operating system:** Platform independent
- **Programming language:** Java, Perl and R
- **Other requirements:** Client needs Java 1.6 or higher, Perl 5, R 2.15
- **License:** Java client under Apache License, Statistics module under GPL2.
- **Any restrictions to use by non-academics:** For non-profit use only.

Competing interests

None.

Author's contributions

FA implemented the statistical analysis and evaluated the performance, SF programmed the Java client, MTW, FA and RL implemented the Web service. All authors contributed to the implementation details and testing, and collaborated in writing and approved the final manuscript.

Acknowledgments

We thank David Hot for initiating our interest in dRNA-seq, Thomas Yee for valuable advice and Peter Kerpedjiev for proof reading.

Funding This work was partly funded by the Austrian Science Fund (FWF) projects "In silico annotation of noncoding RNAs and their targets" (AF 0430511), "RNA regulation of the transcriptome" (F43), the German ministry of science (0316165C as part of the e:Bio initiative), and the Deutsche Forschungsgemeinschaft (STA 850/10-2 within the SPP 1258 "Sensory and regulatory RNAs in Prokaryotes").

References

1. Croucher NJ, Thomson NR: **Studying bacterial transcriptomes using RNA-seq**. *Current opinion in microbiology* 2010, **13**(5):619–624.
2. Cho BK, Zengler K, Qiu Y, Park YS, Knight EM, Barrett CL, Gao Y, Palsson BØ: **The transcription unit architecture of the Escherichia coli genome**. *Nature biotechnology* 2009, **27**(11):1043–1049.
3. Wurtzel O, Sapra R, Chen F, Zhu Y, Simmons B, Sorek R: **A single-base resolution map of an archaeal transcriptome**. *Genome research* 2010, **20**:133–141.
4. Sharma C, Hoffmann S, Darfeuille F, Reignier J, Findeiß S, Sittka A, Chabas S, Reiche K, Hackermüller J, Reinhardt R, et al.: **The primary transcriptome of the major human pathogen Helicobacter pylori**. *Nature* 2010, **464**(7286):250–255.
5. Schmidtke C, Findeiß S, Sharma C, Kuhfuß J, Hoffmann S, Vogel J, Stadler P, Bonas U: **Genome-wide transcriptome analysis of the plant pathogen Xanthomonas identifies sRNAs with putative virulence functions**. *Nucleic acids research* 2012, **40**(5):2020–2031.
6. Tauber S, von Haeseler A: **Exploring the sampling universe of RNA-seq**. *Statistical applications in genetics and molecular biology* 2013, **12**(2):175–188.
7. Passalacqua KD, Varadarajan A, Ondov BD, Okou DT, Zwick ME, Bergman NH: **Structure and complexity of a bacterial transcriptome**. *Journal of bacteriology* 2009, **191**(10):3203–3211.
8. Skellam J: **The frequency distribution of the difference between two Poisson variates belonging to different populations.** *Journal of the Royal Statistical Society. Series A (General)* 1946, **109**(Pt 3):296.
9. Lambert D: **Zero-inflated Poisson regression, with an application to defects in manufacturing**. *Technometrics* 1992, **34**:1–14.
10. Yee TW: **The VGAM package for categorical data analysis**. *Journal of Statistical Software* 2010, **32**(10):1–34.
11. **The Perl Dancer Project**[http://www.perldancer.org].
12. Fielding RT: **REST: Architectural Styles and the Design of Network-based Software Architectures**. *Doctoral dissertation*, University of California, Irvine 2000.
13. **Picard tools version 1.85**[http://picard.sourceforge.net].
14. **XZ for Java version 1.2**[http://tukaani.org/xz/java.html].
15. **Appache HttpComponents version 4.2.3**[http://hc.apache.org/index.html].
16. Ramachandran V, Shearer N, Jacob J, Sharma C, Thompson A: **The architecture and ppGpp-dependent expression of the primary transcriptome of Salmonella Typhimurium during invasion gene expression**. *BMC genomics* 2012, **13**:25.
17. Quinlan AR, Hall IM: **BEDTools-User-Manual**[bedtools.googlecode.com/files/BEDTools-User-Manual.pdf].
18. Searls DT: **An estimator for a population mean which reduces the effect of large true observations**. *Journal of the American Statistical Association* 1966, **61**(316):1200–1204.
19. Yee TW, Yee MT, Suggests M: **Package 'VGAM'** 2012.
20. Sayers EW, Barrett T, Benson DA, Bolton E, Bryant SH, Canese K, Chetvernin V, Church DM, DiCuccio M, Federhen S, et al.: **Database resources of the national center for biotechnology information**. *Nucleic acids research* 2012, **40**(D1):D13–D25.
21. Hoffmann S, Otto C, Kurtz S, Sharma CM, Khaitovich P, Vogel J, Stadler PF, Hackermüller J: **Fast mapping of short sequences with mismatches, insertions and deletions using index structures**. *PLoS computational biology* 2009, **5**(9):e1000502.
22. Quinlan AR, Hall IM: **BEDTools: a flexible suite of utilities for comparing genomic features**. *Bioinformatics* 2010, **26**(6):841–842.
23. Mitschke J, Vioque A, Haas F, Hess WR, Muro-Pastor AM: **Dynamics of transcriptional start site selection during nitrogen stress-induced cell differentiation in Anabaena sp. PCC7120**. *Proceedings of the National Academy of Sciences* 2011, **108**(50):20130–20135.

24. Sokolova M, Japkowicz N, Szpakowicz S: **Beyond accuracy, F-score and ROC: a family of discriminant measures for performance evaluation**. In *AI 2006: Advances in Artificial Intelligence*, Springer 2006:1015–1021.

25. Mitschke J, Georg J, Scholz I, Sharma CM, Dienst D, Bantscheff J, Voß B, Steglich C, Wilde A, Vogel J, et al.: **An experimentally anchored map of transcriptional start sites in the model cyanobacterium Synechocystis sp. PCC6803**. *Proceedings of the National Academy of Sciences* 2011, **108**(5):2124–2129.

26. Nicolas P, Mäder U, Dervyn E, Rochat T, Leduc A, Pigeonneau N, Bidnenko E, Marchadier E, Hoebeke M, Aymerich S, et al.: **Condition-dependent transcriptome reveals high-level regulatory architecture in Bacillus subtilis**. *Science* 2012, **335**(6072):1103–1106.

27. Dugar G, Herbig A, Förstner KU, Heidrich N, Reinhardt R, Nieselt K, Sharma CM: **High-Resolution Transcriptome Maps Reveal Strain-Specific Regulatory Features of Multiple Campylobacter jejuni Isolates**. *PLoS genetics* 2013, **9**(5):e1003495.

28. Wurtzel O, Sesto N, Mellin JR, Karunker I, Edelheit S, Bécavin C, Archambaud C, Cossart P, Sorek R: **Comparative transcriptomics of pathogenic and non-pathogenic Listeria species**. *Molecular systems biology* 2012, **8**.