Genomics 102 (2013) 507–514

Contents lists available at ScienceDirect

Genomics

journal homepage: www.elsevier.com/locate/ygeno

SSP: An interval integer linear programming for de novo transcriptome assembly and isoform discovery of RNA-seq reads

Zhaleh Safikhani [a], Mehdi Sadeghi [b,*], Hamid Pezeshk [c], Changiz Eslahchi [d,e]

[a] Department of Bioinformatics, Institute of Biochemistry and Biophysics, University of Tehran, Tehran, Iran
[b] National Institute of Genetic Engineering and Biotechnology (NIGEB), Tehran, Iran
[c] School of Mathematics, Statistics and Computer Sciences, Center of Excellence in Biomathematics, College of Science, University of Tehran, Tehran, Iran
[d] Department of Computer Science, Shahid Beheshti University, GC., Tehran, Iran
[e] School of Computer Science, Institute for Research in Fundamental Sciences (IPM), Tehran, Iran

ARTICLE INFO

Article history:
Received 10 May 2013
Accepted 16 October 2013
Available online 23 October 2013

Keywords:
RNA sequencing
Transcriptome
Alternative splicing
Isoforms
Linear programming

ABSTRACT

Recent advances in the sequencing technologies have provided a handful of RNA-seq datasets for transcriptome analysis. However, reconstruction of full-length isoforms and estimation of the expression level of transcripts with a low cost are challenging tasks. We propose a novel de novo method named SSP that incorporates interval integer linear programming to resolve alternatively spliced isoforms and reconstruct the whole transcriptome from short reads. Experimental results show that SSP is fast and precise in determining different alternatively spliced isoforms along with the estimation of reconstructed transcript abundances. The SSP software package is available at http://www.bioinf.cs.ipm.ir/software/ssp.

1. Introduction

Comprehensive transcriptome annotation is an essential task in the study of a broad range of biological processes. The traditional methods for sequencing and identifying the expression level of the cell transcripts are based on the process of expression sequence tags (EST) or microarrays [1–5]. Despite the high cost of these techniques, they do not cover the transcriptome precisely [6]. Next generation sequencing (NGS) of expressed sequence mRNA (RNA-seq) has transformed this field while the cost has been sharply decreased [7–10]. These techniques can be used to identify the full set of transcripts including, novel transcripts from unannotated genes, several alternatively spliced isoforms and also trans splicing events [11,12].

Despite the advances in NGS techniques, sequence reads obtained from these methods are often very short. Hence, transcriptome reconstruction from this huge amount of short sequence reads is a hard computational task. It requires powerful tools to assemble the massive short reads with acceptable sensitivity and accuracy [13–16].

Transcriptome assembly is based on two computational methods. Ab initio or reference-based assembly is applied for the case where a reference genome for the target transcription is available. In this strategy, short reads are aligned to a reference genome and then the cluster of overlapping reads is used to build a graph representing all the possible isoforms [17–22]. Although this strategy has many advantages such as the feasibility of applying parallel computing, high sensitivity and assemble of low abundance transcripts, however, it is extremely dependent on the existence of a high quality reference genome. So this strategy cannot be applied for all the organisms.

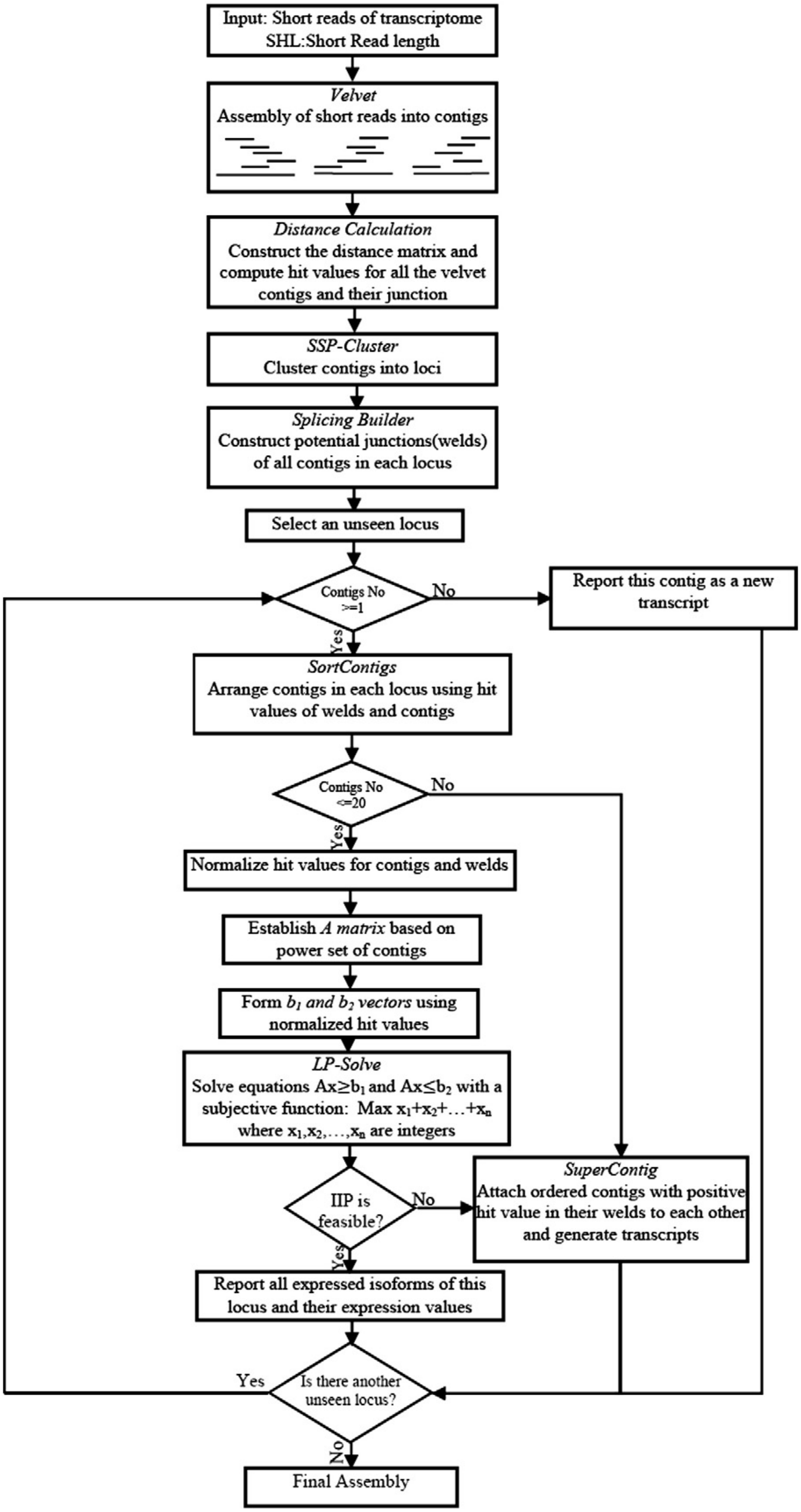
De novo or reference-independent strategy is used to directly assemble transcripts by finding overlaps between the reads [23–27]. This strategy is applied when a reference genome is not available or is poorly annotated. Moreover, it can be exploited as a preliminary step to provide longer assembled contigs (contiguous sequence of a transcript assembled from shorter sequence reads) before alignment to a reference genome. Several de novo assembly programs use De Bruijn graph [28] to process short reads into larger contiguous sequences [29–33]. Huge amount of short reads related to higher eukaryotic transcriptome increases the size of these graphs. Therefore, it enhances the difficulty to determine related reads to join them into contigs. The main problem involved is identifying alternatively spliced isoforms. The variant coverage depth of transcript isoforms in the graphs, that changes the density of the short reads, can be employed to overcome this difficulty. In addition to the assembly of various isoforms, the expression level of them should be computed using the coverage of exons in each cluster.

In this work, we present a new de novo transcriptome assembly algorithm, named SSP. The essence of this algorithm is to use interval integer linear programming to efficiently distinguish the possible alternative splicing isoforms. We use the primary contigs produced by *velvet* [30] and the cluster contigs are sorted based on their similarity

[*] Corresponding author at: National Institute of Genetic Engineering and Biotechnology, Pajohesh Blvd, 17 Km Tehran Karaj Highway, P.O. Box: 161/14965, Tehran, Iran.
E-mail address: sadeghi@nigeb.ac.ir (M. Sadeghi).

0888-7543/$ – see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.ygeno.2013.10.003

obtained by common short reads between them. Eventually, an interval integer linear programming system reports all the possible transcripts in each cluster. The performance of SSP using a simulated dataset derived from Rat alternative transcripts is evaluated. The effect of the coverage depth, the number and the size of the alternatively spliced isoforms are also examined. The assembly results are compared to those from other widely used de novo assemblers, including Trinity [23] and Oases [24]. The SSP results in more accurate transcriptome including alternative-spliced isoforms. The SSP pipeline also significantly reduces false identification of transcripts.


Input: Short reads of transcriptome
SHL:Short Read length
Velvet
Assembly of short reads into contigs
Distance Calculation
Construct the distance matrix and compute hit values for all the velvet contigs and their junction
SSP-Cluster
Cluster contigs into loci
Splicing Builder
Construct potential junctions(welds) of all contigs in each locus
Select an unseen locus
Contigs No >=1
No
Report this contig as a new transcript
Yes
SortContigs
Arrange contigs in each locus using hit values of welds and contigs
Contigs No <=20
No
Yes
Normalize hit values for contigs and welds
Establish *A matrix* based on power set of contigs
Form b_1 *and* b_2 *vectors* using normalized hit values
LP-Solve
Solve equations $Ax \geq b_1$ and $Ax \leq b_2$ with a subjective function: Max $x_1+x_2+\ldots+x_n$ where $x_1, x_2, \ldots, x_n$ are integers
SuperContig
Attach ordered contigs with positive hit value in their welds to each other and generate transcripts
IIP is feasible?
No
Yes
Report all expressed isoforms of this locus and their expression values
Yes
Is there another unseen locus?
No
Final Assembly

2. Methods

2.1. Overview

De novo transcriptome assembly can be significantly improved by the application of SSP. It is a pipeline consisting of different software modules and it aims to produce transcriptome assembly of an RNA-seq experiment. The ultimate goal is to reconstruct the maximum number of true transcripts wherein the number of incorrectly reported transcripts becomes as small as possible. A schematic demonstration of the steps applied to achieve this goal is presented in Fig. 1.

Similar to the other de novo transcriptome assemblers the primary RNA-seq reads are assembled using a De Bruijn graph in our approach. This transcriptome sequencing graph is a collection of small connected components that are representatives for loci (specific location of genes or positions on the chromosomes). The SSP assembles the contigs that belong to each locus, using interval integer linear programming (IILP) and reconstructs all the possible alternative spliced isoforms. It exploits the fact that different isoforms are variant combinations of the contigs that are representatives for exons, retained introns or combinations of these parts. The required parameters to solve this interval integer linear programming system are produced using the number of aligned short reads with contigs and welds (potential junction site between contigs). The required software modules for this pipeline are illustrated in Fig. 1. The SSP assembly process is explained in detail in the following sections.

2.2. Data pre-processing

Data pre-processing is for eliminating sequencing errors in NGS reads. This step is optional which means that the pipeline can be executable ignoring pre-processing of RNA-seq datasets prior to the assembly. But it improves the accuracy and computational efficiency of the final assembly [13]. This step can be executed using several tools including *fastx* [34], *Seqtrim* [35], *Quake* [36] and *Tagdust* [37]. We took advantage of *fastx*, which is accurate and also easy to use. This preprocessing step was mainly applied to remove sequencing errors from the raw RNA-seq data and has a significant impact on the transcriptome assembly. The most common way to remove or correct sequencing errors in reads, is by analyzing the quality scores. Most of the NGS technologies report quality scores to measure the probability that a base is called incorrectly. Low quality scores indicate possible sequencing errors. It should be noted that the *fastx* also refines sequences using these quality scores.

2.3. Assemble reads into contigs

2.3.1. Reconstruction of contigs using velvet

The first step of de novo transcriptome assembly is assembling reads into unique contiguous sequences. In this step *velvet* [30] is applied. It is a fast, reliable and easy to use de novo genome assembler. The strategy employed by velvet, like the other de novo genome assemblers, is to use the redundancy of short-read sequencing to find overlaps between the reads. In this way a De Bruijn graph, a directed graph with sequences of length *k*, is used. So that if there are k-1 overlaps between two nodes, they will be connected to each other. The main problem of using these kinds of assemblers for transcriptome assembly is their procedures for error removing and repeat resolving. So that, alternatively spliced transcripts may be recognized as repeat and low-coverage transcripts mistaken by error. The most important parameter of these tools is the k-mer size.

2.3.2. Construction of reverse complement

It is possible that the reported contigs by *velvet* are the reverse complement of the true contigs. Here we report both forward and reverse strands of each contig and the right one will be determined after clustering step. *Oases* [24], which is another transcriptome assembler that uses *velvet* as its contig assembler, determines the correct order of *velvet* contigs using the comparison of the number of stop codons between reverse and forward strands. The algorithm applied in this work is more accurate compared to the one obtained by *Oases*.

2.4. Clustering of contigs

2.4.1. Building of distance matrix

To distinguish the overlapping contigs that likely belong to alternative spliced isoforms of one locus, it is needed to reconstruct the sequencing graph between contigs. *Velvet* has a read-tracking option that can be used to report the read tracking information. It reports the reads that constructed each contig along with the location of reads in contigs. We used this information to produce a distance matrix D in which element D_{ij} is the number of paired reads in which one end is in the *i*th contig and the other end is in the *j*th contig. Matrix D is used to cluster contigs in the next step. Moreover, we computed the number of reads in each contig and the number of reads spanning the junction sites of contigs in this step. These values will be used by *SortContigs* to determine the order of contigs in each cluster and finding the solution of the IILP system will be discussed in Section 2.5.

2.4.2. Clustering of contigs using SSP-Cluster

Since alternative splicing occurs just between contigs that belong to a specific locus, we need a tool to cluster related contigs that correspond to a specific portion of alternatively spliced transcripts. We developed *SSP-Cluster*, which categorizes contigs into distinct set of clusters named loci. These contigs are likely to be derived from alternatively spliced isoforms. The clustering approach works in a top-down manner. In this way, a graph of all contigs is constructed using a distance matrix of the previous step. The vertices of this graph are contigs and the edges are the distances between the contigs. The low weighted edges are eliminated based on a threshold with a default value of 10. Then connected components of the remained sub-graph will be specified. To obtain the connected components of the graph, the following steps are taken to the point that there is no more unvisited vertex in the graph.

1. Choose an unvisited vertex in the graph, then marked it as visited and consider it as a new connected component.
2. Obtain all the accessible vertices through this vertex using a depth first traverse approach and insert them to the created connected components.

The primary graph and the specified connected components that are representative for the loci of a sample RNA-seq experiment are shown in Fig. 2a. Moreover, the schematic representation of a locus of this experiment is displayed in Fig. 2b. These graphs are drawn using Gephi [38] which is an open source software for graph and network analysis.

Fig. 1. Schematic representation of the SSP method. This pipeline is appropriate for determining alternative splice isoforms in RNA-seq experiments. The original short reads are assembled using *velvet* into unique contigs. Then, the distance matrix needed to cluster contigs is computed. Moreover, the read coverage of each contig and the number of short reads which span the junction (weld) of contigs are computed in this stage. These values called hit values are needed to order the contigs in each cluster and reconstruct the alternative transcripts. In the next stage, the *SSP-Cluster* clusters the related contigs to non-overlapping loci. If a contig is the only one in a locus it will be reported as a new transcript. Otherwise, if the number of contigs is less than a predefined threshold (considered 20 in our work), the interval integer linear programming is applied to find different isoforms in that locus. This IILP system is built from various linear inequalities relating to all the possible mixture of contigs of the locus. To constitute the IILP system it is needed to have the correct order of contigs in the locus. *SortContigs* uses an innovative algorithm to predict the order of contigs using the abovementioned hit values. Moreover, the normalized hit values and welds are used to solve the IILP system. Then, the IILP solver, *LP-Solve*, tries to find a feasible solution for this problem and reports the expression values for all the transcripts. If there is not a feasible solution for the IILP system or in the case of existence more than the threshold number of contigs in a locus, a super contig will be created by locating ordered contigs besides each other via *SuperContigBuilder*.

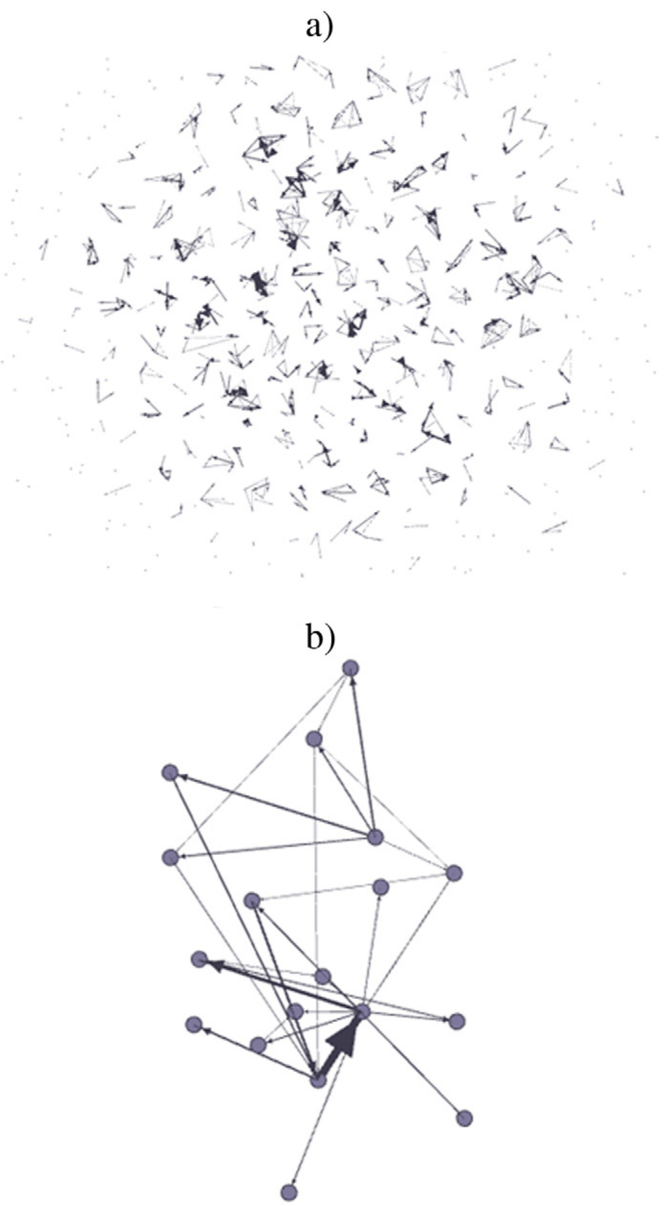


Fig. 2. (a) The clustering graph of all the contigs of a sample transcriptome reconstruction experiment. The sub-graphs are considered as loci. (b) A closer view of a sample sub graph in which different paths may be assumed as various alternative isoforms.

2.5. Resolving alternatively spliced transcripts using interval integer linear programming

In order to improve not only the accuracy but also the sensitivity of alternatively spliced isoforms, we designed a method which performs a comprehensive process in each locus to report all the expressed isoforms. We used interval integer linear programming (IILP), a mathematical method for determining a way to achieve the best outcome in a list of requirements represented by linear inequalities [39]. In the context of transcriptome assembly we try to define alternatively spliced isoforms of locus $\mathcal{L}$ as linear inequalities. Then, all the isoforms represented as linear inequalities constitute a complete interval integer linear programming system. The principal idea of this work is based on the fact that alternatively spliced isoforms are different combinations of exons and retained introns. Let $A=[\alpha_{ij}]$ be a binary matrix where the columns are all the possible isoforms of $\mathcal{L}$ and the rows are contigs and welds of $\mathcal{L}$. In this matrix, $\alpha_{ij}=1$ if and only if *j*th isoform includes contig or weld *i*. Otherwise $\alpha_{ij}=0$. If the number of contigs of $\mathcal{L}$ is *c*, there are 2^c–1 possible different isoforms and $\binom{c}{2}$ welds. Therefore, *A* has 2^c–1 columns and $c+\binom{c}{2}$ rows. An example of matrix *A* for a locus of three contigs is shown in Fig. 3. It is remarkable that the order of the contigs is needed to construct matrix *A*. As mentioned in previous section, *SortContigs* arrange the contigs in each locus.

Let $b=[b_{i1}]$ be a matrix with $c+\binom{c}{2}$ rows and one column, such that b_{i1}denoted the number of read counts falling into contig or weld *i* divided by the length of contig or weld *i*. Because of the existence of errors and repeats in short reads, the values of coordinates of *b* are not accurate enough. In this way, we build b^+ and b^- in which $b_{i1}^+=(1+\alpha)\times b_{i1}$ and $b_{i1}^-=(1-\alpha)\times b_{i1}$. The value for α parameter is determined by the user as an input value which can be any number between 0 and 1. The default value of α is 0.5.

In this work we solve the following IILP system

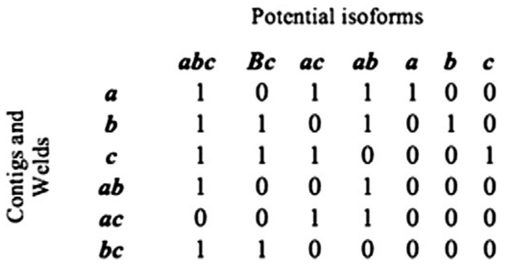
$$\begin{aligned} &\textit{Maximize} && \textstyle\sum_{i=1}^{2^c-1} x_{i1} \\ &\textit{Subject to} && b^- \le A\begin{bmatrix} x_{11} \\ \vdots \\ x_{2^c-11} \end{bmatrix} \le b^+ \end{aligned} \qquad (1)$$

and

$$x_{i1}\in N\cup\{0\},\ 1\le i\le 2^c-1.$$

By solving this system, we can obtain the best supporting expression values for all the isoforms of $\mathcal{L}$. In fact, x_{i1} indicates the expression level of the *i*th isoform of $\mathcal{L}$.

To solve Eq. (1), we use *LP-Solve* (version 5.5.2.0) which is an open-source mixed integer linear programming solver [40]. If there is no feasible solution for linear inequalities, *SuperContig* constructs some super contigs by assembling the ordered contigs with the number of welds more than 0.04 times the average k-1-mer coverage of each contig. This limitation corresponds to twice the sequence error rate in a read, the upper bound which we set 2% based on the value reported in the literature [23]. Moreover, in the case of the existence of more than a limited number of contigs in a locus, the required time for solving Eq. (1) grows exponentially. Hence, we use an upper bound, which is estimated to be 20 contigs in each locus. When the number of contigs is more than the upper bound, the super contigs are created using *SuperContig*.



Contigs and Welds		Potential isoforms						
		abc	**Bc**	**ac**	**ab**	**a**	**b**	**c**
	a	1	0	1	1	1	0	0
	b	1	1	0	1	0	1	0
	c	1	1	1	0	0	0	1
	ab	1	0	0	1	0	0	0
	ac	0	0	1	1	0	0	0
	bc	1	1	0	0	0	0	0

Fig. 3. The columns demonstrate all the possible isoforms that can be constructed with three contigs. The first three rows show the contigs *a*, *b*, *c* and the other three ones represent all the possible *welds* or junctions of these contigs. Now assume we want to set the value of first column of the matrix. Since *abc* is a combination of *a*, *b*, *c*, *ab* and *bc*, so the values of related rows are set to 1 and the value of fifth cell which is related to *ac* junction is set to 0.

3. Results and discussion

In order to evaluate the performance of SSP, we used simulated and real RNA-seq reads. Then the SSP results are compared with the results obtained from three other recent de novo assembly programs, named: Trinity, Oases and Mira. The length and quantification of reconstructed transcripts have also been investigated by these tools.

3.1. Simulated data

The transcriptome dataset of Rat including all alternative isoforms was retrieved from the alternative splicing database (http://www.ebi.ac.uk/asd/index.html). After specification of the number of isoforms for each gene, the genes were classified based on their number of alternatively spliced isoforms. In these classes, we replaced the genes with a group of all their transcripts. Moreover, due to the native of transcriptome, which involves variant expression values for each transcript, random abundance is assigned to each transcript. We observed that Rat genes have isoforms of size between 3 and 25 transcripts. Genes with more than 12 isoforms are rare. Hence, different datasets were made that each included 100 genes with equal number of transcript isoforms from 3 to 12. Moreover, we constructed some mixture datasets from random number of genes with different numbers of isoforms. We also assigned random expression level to transcripts in each dataset for a performance comparison. RNA-seq data was generated from transcripts to obtain simulated short reads with length of 45, using *dwgsim* package [41]. This popular simulator has various parameters to produce short reads like as real experiments. “The distance between two paired-end reads (insert size) has been set to 150 bp with a standard deviation of 10.”

3.1.1. Performance of transcriptome assembly

The main purpose of our algorithm is to improve de novo assembly of all the isoforms of each gene. Therefore, we devised the assessment process to specify whether it is successful in reaching this goal. We performed the SSP, Oases and Trinity for all the simulated datasets of the previous section. The reconstructed transcripts were aligned to the known ones using BLAT [42]. The transcripts that are reconstructed

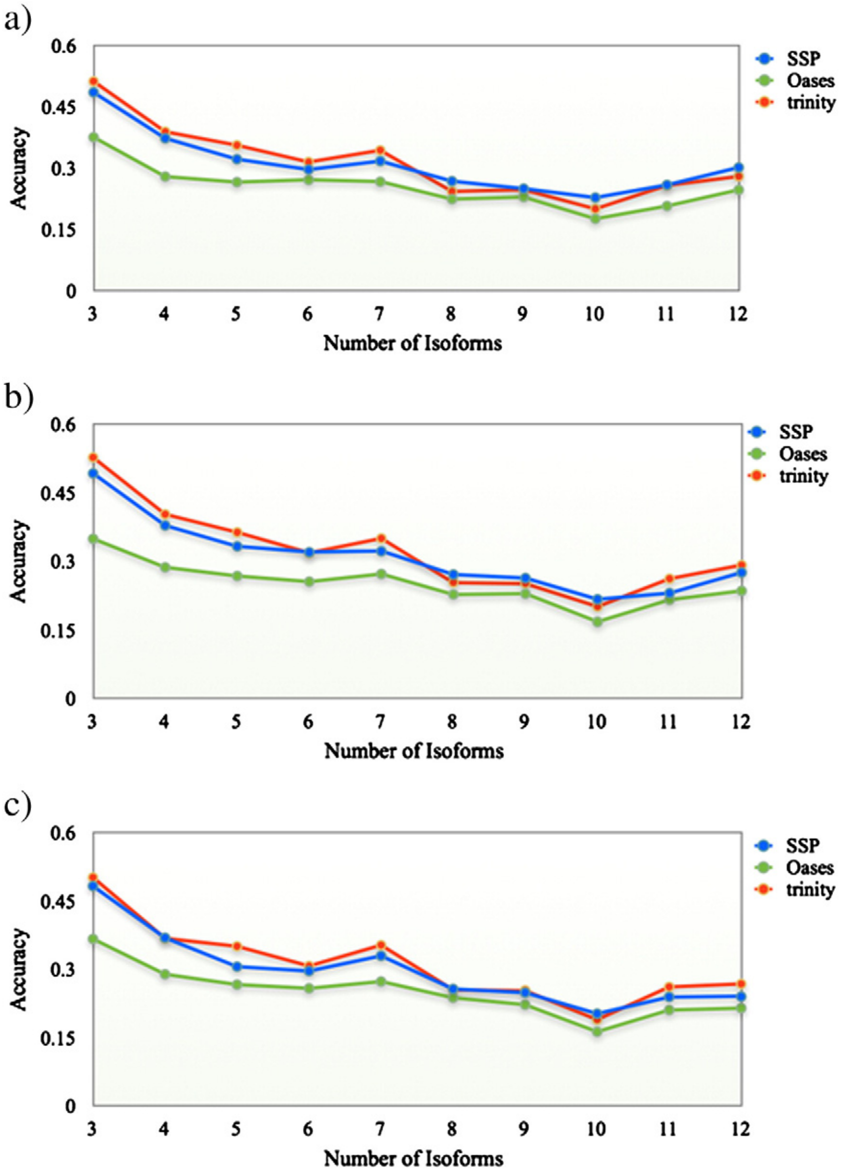


Fig. 4. The accuracy of de novo isoform reconstruction with respect to the isoform number of each gene. (a) The expression level of all the transcripts considered identical. (b) The variant expression level assigned to the longer transcripts is higher than that assigned to the shorter ones. (c) The variant expression level assigned to the shorter transcripts is higher than that assigned to the longer ones.

with 80% length and have 100% identity were identified and reported as a true positive (TP). Sensitivity and precision were calculated using Eqs. (2a) and (2b), respectively. We measured the accuracy using the combination of sensitivity and precision as in Eq. (2c).

$$Sensitivity = \frac{TP}{TP + FN} \tag{2a}$$

$$Precision = \frac{TP}{TP + FP} \tag{2b}$$

$$Accuracy = \sqrt{sensitivity \times precision}. \tag{2c}$$

3.1.2. Investigation of the impact of the isoform number

We classified Rat genes based on their isoforms and replaced genes with the group of isoforms that belong to them. Then, we selected at most 100 genes of each class with 3 to 12 isoforms. In the first experiment, we considered the expression level of all the identical transcripts and assigned a constant value as the abundance of all the transcripts. Fig. 4a shows the accuracy of SSP, Oases and Trinity on each dataset. As shown in Fig. 4a, the accuracy is decreased as the number of isoforms increased. The small growths in the end part of the diagram are due to the shortage of genes with 11 and 12 isoforms. Moreover, the performance of SSP and Trinity is almost similar and greater than that of Oases in all the cases.

3.1.3. Impact of the variant expression level and transcript length

To evaluate the effect of transcript abundance, we assigned a random expression level to transcripts in each dataset from 10 to 100 that approximately follows a log normal distribution [43]. The approach of the second experiment was to express longer transcripts more. Hence, we consider a relationship between the length of transcripts and their expression level. Fig. 4b shows the performance of SSP, Oases and Trinity for each dataset in this experiment. In contrast, in the third experiment, the expression level assigned to the shorter isoforms is higher than the longer ones. Fig. 4c shows the performance of SSP, Oases and Trinity. As shown in Figs. 4b and c, the accuracy of the programs is not effected by the diverse coverage of the isoforms and it is also independent of the isoform length.

Finally, we constructed 50 different datasets by random mixture of transcripts of several genes with different numbers of alternative isoforms. For each of them we applied the three abovementioned scenarios of assigning expression level to transcripts. Table 1 shows the average results of the implementation of all the programs for these simulated transcriptomes with regard to the applied expression level scenario. As represented in Table 1, accuracy of de novo assembly of transcriptome tools is independent of the coverage of transcripts. Furthermore, SSP and Trinity have the same average accuracy for all the cases.

Analysis of the results of all the above experiments also shows that, Trinity generally assembles longer isoforms in comparison to SSP and Oases. In other words, the length of the most of the true positive isoforms assembled by SSP and Oases is shorter than the average length of the isoforms in each dataset.

Table 1
The average accuracy of applying the methods for 50 datasets constructed of random mixture of Rat transcripts. First the expression of the entire transcripts considered identical and then variant expression level assigned to them.

Expression level situation	SSP	Oases	Trinity
Identical expression level	0.3253	0.2781	0.3292
Variant expression level (longer expressed more)	0.3306	0.2763	0.3318
Variant expression level (longer expressed less)	0.3205	0.2766	0.3209

3.1.4. Expression level estimation

The expression level of transcripts was calculated using Eq. (3) as illustrated by Mortazavi et al. [20].

$$RPKM = \#MappedReads \times \frac{1000\ bases \times 10^6}{length\ of\ transcript \times total\ number\ of\ mapped\ reads}. \tag{3}$$

The Pearson's correlation coefficient [44] between the expression level of the true positive reconstructed transcripts and the abundance of primary ones can be calculated via Eq. (4).

$$r = \frac{n\left(\sum xy\right) - \left(\sum x\right)\left(\sum y\right)}{\sqrt{\left[n\sum x^2 - \left(\sum x\right)^2\right]\left[n\sum y^2 - \left(\sum y\right)^2\right]}}. \tag{4}$$

In the case of having datasets of transcripts with identical expression levels, calculation of the Pearson's correlation coefficient of expression level of transcripts is meaningless. This is because the covariance values of the primary transcripts would be zero. Hence, the correlation coefficient amounts of transcript abundance are computed for all the above stated experiments with variant number of expression levels in the simulated transcriptome. Fig. 5 shows the results for different datasets in histograms. The first ten bars show the results for classes of genes with 3 to 12 isoforms, respectively and the last bar shows the average of the correlation coefficient values for different datasets made by random mixtures of Rat transcripts. As shown in Fig. 5, SSP outperforms the other methods on average, even if Oases or Trinity performances are better in some cases.

3.2. Real data

We evaluated SSP on data from two well-annotated species, human and baker's yeast. A real RNA-seq dataset (SRP000698) of poly(A) selected RNA from primary CD + T cells of human individual [45] and another dataset (SRP002790) mRNA from WT cells of *Saccharomyces cerevisiae* [46] were downloaded via www.ebi.ac.uk/ena. Ensembl database version 73 was used to obtain the annotated genes of *Homo sapiens* and *S. cerevisiae*. SSR, Oases, Trinity and an overlap-based method, Mira [47], were run on Paired-end reads from three samples for human (SRR027872, SRR027876, SRR027878) and three samples for yeast (SRR059167, SRR059168, SRR059169). The reconstructed transcripts of each method were aligned to the sequences of Ensembl annotated genes using BLAT [42]. Table 2 presents the mean and standard deviation of sensitivity and specificity for reconstructed transcripts longer than 100 bp which entire length of them matched uniquely to the reference genes. In fact one-to-one mappings between

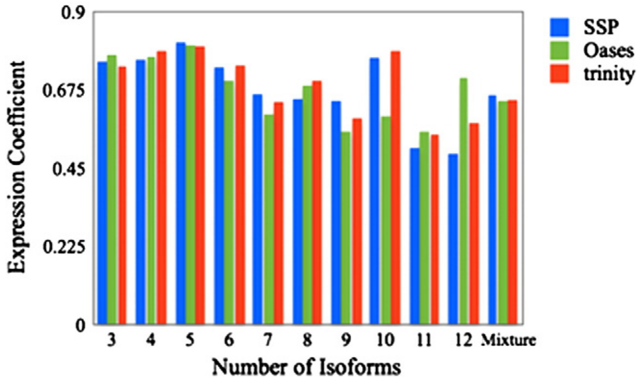


Fig. 5. Comparison of the Pearson's correlation coefficient of expression level for different datasets. Mixture shows the correlation coefficient of expression values of all the fifteen random datasets.

Table 2
Comparison of different de novo methods on three samples of human and yeast datasets. Mean ± standard deviation of the number of reconstructed transcripts longer than 100 bp, specificity and sensitivity as well as the number of transcripts with at least 80% length of Ensembl transcripts are shown.

Data	Method	Reconstructed transcripts ≥ 100 bp	Sensitivity %	Specificity %	Reconstructed transcripts (80% length)
Human	SSP	23944 ± 4995	10 ± 1.9	82 ± 3	721 ± 324
	Oases	22491 ± 5872	9 ± 1.8	83 ± 4	860 ± 292
	Trinity	14368 ± 3249	4 ± 0.8	85 ± 3	940 ± 254
	Mira	6686 ± 2238	3 ± 1	95 ± 1	129 ± 49
Yeast	SSP	4109 ± 564	6.7 ± 0.9	14 ± 2	19 ± 0.7
	Oases	3609 ± 649	4.4 ± 0.7	16 ± 3	83 ± 0.1
	Trinity	2107 ± 288	5.6 ± 0.1	23 ± 3	60 ± 1.5
	Mira	562 ± 79	1.5 ± 0.02	57 ± 5	3.5 ± 0.7

reconstructed transcripts and reference transcripts were considered. The specificity is calculated using Eq. (5).

$$Specificity = \frac{TN}{TN + FP}. \tag{5}$$

Moreover, the mean numbers of reconstructed transcripts which cover 80% length of Ensembl transcripts with at least 90% sequence identity are shown in Table 2. Although the sensitivity of SSP is better than the other methods, but Mira reconstructs more specific assemblies. Moreover, the average length of reconstructed transcripts by SSP is longer than that by Mira and shorter than that by Oases and Trinity.

3.2.1. Runtime

All the mentioned methods were run on a high-performance Linux-based system, which consists of two Intel Xeon X5650 processors, 24 GB of RAM, and 32 GB of swap memory. The execution time of various methods to assemble the short reads of *Human* and *Yeast* datasets is presented in Table 3. As seen in Table 3, Mira took the longest runtime whereas, Oases is the fastest one. This has also been mentioned in another comparison of different transcriptome assembly methods [48].

4. Conclusion

De novo assembly of RNA-seq short reads has many informatics difficulties. Indentifying all the alternative isoforms separately is one of the biggest challenges of this task to deal with. In this article, we present SSP with an innovative interval integer linear programming system to resolve alternatively spliced isoforms and perform a comprehensive transcriptome assembly. Moreover, SSP estimates the expression level of reconstructed transcripts almost precisely.

Evaluation of the accuracy of SSP in comparison to the other de novo transcript assemblers in literature reveals that it is a powerful tool specially for determining different isoforms. Although, both the SSP and the Oases exploit the contigs produced by the *velvet* to detect all the possible transcripts, SSP results are more accurate than Oases for almost all the simulated datasets. Furthermore, Trinity has a better performance in most of the cases, on cost of a longer runtime. We observed that there is a significant correlation between the abundance of reconstructed isoforms by SSP and the primary expression level of transcripts. Further analysis suggests that the results obtained by SSP are independent of the coverage of transcripts.

Table 3
The running time of various methods for de novo transcriptome assembly in minutes.

	SSP	Oases	Trinity	Mira
Human	25	15	48	307
Yeast	5	4.5	28	39

Acknowledgment

This work is in part supported by the Institute for Research in Fundamental Sciences (IPM) (No. CS 1391-0-01). The authors would like to thank HPC laboratory and Biomath research group at IPM.

References

[1] M. Marra, et al., An encyclopedia of mouse genes, Nat. Genet. 21 (1999) 191–194.

[2] P. Carninci, et al., Targeting a complex transcriptome: the construction of the mouse full-length cDNA encyclopedia, Genome Res. 13 (2003) 1273–1289.

[3] S.J. Souza, et al., Identification of human chromosome 22 transcribed sequences with ORF expressed sequence tags, Proc. Natl. Acad. Sci. U. S. A. 97 (2000) 12690–12693.

[4] D.A. Hahn, et al., Gene discovery using massively parallel pyrosequencing to develop ESTs for the flesh fly *Sarcophaga crassipalpis*, BMC Genomics 10 (2009) 234.

[5] J. Hughes, et al., Dense taxonomic EST sampling and its applications for molecular systematics of the Coleoptera (beetles), Mol. Biol. Evol. 23 (2006) 268–278.

[6] M. Garber1, et al., Computational methods for transcriptome annotation and quantification using RNA-seq, Nat. Methods 8 (2011) 469–478.

[7] M.L. Metzker, Sequencing technologies — the next generation, Nat. Rev. Genet. 11 (2010) 31–46.

[8] B.J. Haas, M.C. Zody, Advancing RNA-Seq analysis, Nat. Biotechnol. 28 (2010) 421–423.

[9] S. Wilkening, et al., An efficient method for genome-wide polyadenylation site mapping and RNA quantification, Nucleic Acids Res. 41 (2013) e65.

[10] Z. Wang, M. Gerstein, M. Snyder, RNA-seq: a revolutionary tool for transcriptomics, Nat. Rev. Genet. 10 (2009) 57–63.

[11] B.T. Wilhelm, J.R. Landry, RNA-seq-quantitative measurement of expression through massively parallel RNA-sequencing, Methods 48 (2009) 249–257.

[12] D.C. Christodoulou, et al., Construction of normalized RNA-seq libraries for next-generation sequencing using the crab duplex-specific nuclease, Curr. Protoc. Mol. Biol. 12 (2011)(Chapter 4, Unit4).

[13] J.A. Martin, Z. Wang, Next-generation transcriptome assembly, Nat. Rev. Genet. 12 (2011) 671–682.

[14] F. Ozsolak, P.M. Milos, RNA sequencing: advances, challenges and opportunities, Nat. Rev. Genet. 12 (2011) 87–98.

[15] S. Marguerat, J. Bahler, RNA-seq: from technology to biology, Cell. Mol. Life Sci. 67 (2010) 569–579.

[16] Y. Surget-Grobac, J.I. Montoya-Burgos, Optimization of de novo transcriptome assembly from next-generation sequencing data, Genome Res. 20 (2010) 1432–1440.

[17] C. Trapnell, et al., Transcript assembly and quantification by RNA-seq reveals unannotated transcripts and isoform switching during cell differentiation, Nat. Biotechnol. 28 (2010) 511–515.

[18] M. Guttman, et al., Ab initio reconstruction of cell type-specific transcriptomes in mouse reveals the conserved multi-exonic structure of lincRNAs, Nat. Biotechnol. 28 (2010) 503–510.

[19] M. Yassour, et al., Ab initio construction of a eukaryotic transcriptome by massively parallel mRNA sequencing, Proc. Natl. Acad. Sci. U. S. A. 106 (2009) 3264–3269.

[20] A. Mortazavi, et al., Mapping and quantifying mammalian transcriptomes by RNA-seq, Nat. Methods 5 (2008) 621–628.

[21] Y. Lin, et al., CLIIQ: accurate comparative detection and quantification of expressed isoforms in a population. Algorithms in bioinformatics, Lect. Notes Comput. Sci 7534 (2012) 178–189.

[22] W. Li, T. Jiang, Transcriptome assembly and isoform expression level estimation from biased ANA-seq reads, Bioinformatics 28 (2012) 2914–2921.

[23] M.G. Grabherr, et al., Full-length transcriptome assembly from RNA-seq data without a reference genome, Nat. Biotechnol. 29 (2011) 644–652.

[24] M.H. Schulz, et al., Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels, Bioinformatics 28 (2012) 1086–1092.

[25] J. Martin, et al., Rnnotator: an automated de novo transcriptome assembly pipeline from stranded RNA-seq reads, BMC Genomics 11 (2010) 663.

[26] G. Robertson, et al., De novo assembly and analysis of RNA-seq data, Nat. Methods 7 (2010) 909–912.

[27] G.A. Sacomoto, et al., KISSPLICE: de-novo calling alternative splicing events from RNA-seq data, BMC Bioinforma. 13 (Suppl. 6) (2012) S5.

[28] N.G. De Bruijn, A combinatorial problem, Proc. K. Ned.Akad. Wet. 49 (1946) 758–764.

[29] P.A. Pevzner, et al., An Eulerian path approach to DNA fragment assembly, Proc. Natl. Acad. Sci. U. S. A. 98 (2001) 9748–9753.

[30] D.R. Zerbino, E. Birney, *Velvet*: algorithms for de novo short read assembly using de Bruijn graphs, Genome Res. 18 (2008) 821–829.

[31] J.T. Simpson, et al., ABySS: a parallel assembler for short read sequence data, Genome Res. 19 (2009) 1117–1123.

[32] J. Butler, et al., ALLPATHS: de novo assembly of whole-genome shotgun microreads, Genome Res. 18 (2008) 810–820.

[33] P.N. Ariyaratne, W.K. Sung, PE-Assembler: de novo assembler using short paired-end reads, Bioinformatics 27 (2011) 167–174.

[34] http://hannonlab.cshl.edu/fastx_toolkit/index.html.

[35] J. Falgueras, et al., SeqTrim: a high-throughput pipeline for pre-processing any type of sequence read, BMC Bioinforma. 11 (2010) 38.

[36] D.R. Kelley, et al., Quake: quality-aware detection and correction of sequencing errors, Genome Biol. 11 (2010) R116.

[37] T. Lassmann, et al., TagDust—a program to eliminate artifacts from next generation sequencing data, Bioinformatics 25 (2009) 2839–2840.

[38] M. Bastian, et al., Gephi: an open source software for exploring and manipulating networks, International AAAI Conference on Weblogs and Social Media, 2009.
[39] G. Bernd, J. Matoušek, Understanding and Using Linear Programming, Springer, Berlin, ISBN: 3-540-30697-8, 2006.
[40] M. Berkelaar et al., lpsolve: Open source (Mixed-Integer) Linear Programming system. http://lpsolve.sourcefourge.net/5.5/.
[41] Whole genome simulation, http://sourceforge.net/apps/mediawiki/dnaa/index.php?title=Whole_Genome_Simulation.
[42] W. James Kent, BLAT—the BLAST-like alignment tool, Genome Res. 12 (2002) 656–664.
[43] M.D. Alter, et al., Variation in the large-scale organization of gene expression levels in the hippocampus relates to stable epigenetic variability in behavior, PLoS One 3 (2008) e3344.
[44] S.M. Stigler, Francis Galton's account of the invention of correlation, Stat. Sci. 4 (1989) 73–79.
[45] G.A. Heap, et al., Genome-wide analysis of allelic expression imbalance in human primary cells by high-throughput transcriptome resequencing, Hum. Mol. Genet. 19 (2009) 122–134.
[46] J.Z. Levin, et al., Comprehensive comparative analysis of strand-specific RNA sequencing methods, Nat. Methods 7 (2010) 709–715.
[47] B. Chevreux, et al., Using the miraEST assembler for reliable and automated mRNA transcript assembly and SNP detection in sequenced ESTs, Genome Res. 47 (2004) 1147–1159.
[48] Z. Qiong-Yi, et al., Optimizing de novo transcriptome assembly from short-read RNA-Seq data: a comparative study, BMC Bioinforma. 12 (Suppl. 14) (2011) S2.