



HAL
open science

De Novo Clustering of Long Reads by Gene from Transcriptomics Data

Camille Marchet, Lolita Lecompte, Corinne da Silva, Corinne Cruaud,
Jean-Marc Aury, Jacques Nicolas, Pierre Peterlongo

► **To cite this version:**

Camille Marchet, Lolita Lecompte, Corinne da Silva, Corinne Cruaud, Jean-Marc Aury, et al.. De Novo Clustering of Long Reads by Gene from Transcriptomics Data. *Nucleic Acids Research*, inPress, pp.1-12. 10.1093/nar/gky834 . hal-01643156v2

HAL Id: hal-01643156

<https://hal.science/hal-01643156v2>

Submitted on 14 Sep 2018

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

De Novo Clustering of Long Reads by Gene from Transcriptomics Data

Camille Marchet¹, Lolita Lecompte¹, Corinne Da Silva², Corinne Cruaud², Jean-Marc Aury², Jacques Nicolas¹ and Pierre Peterlongo¹

¹Univ Rennes, Inria, CNRS, IRISA, F-35000 Rennes, France

²Commissariat à l'Énergie Atomique (CEA), Institut de Biologie François Jacob, Genoscope, 91000, Evry, France

Received XXX, 2018; Revised XXX, XXX; Accepted XXX, XXX

ABSTRACT

Long-read sequencing currently provides sequences of several thousand base pairs. It is therefore possible to obtain complete transcripts, offering an unprecedented vision of the cellular transcriptome.

However the literature lacks tools for de novo clustering of such data, in particular for Oxford Nanopore Technologies reads, because of the inherent high error rate compared to short reads.

Our goal is to process reads from whole transcriptome sequencing data accurately and without a reference genome in order to reliably group reads coming from the same gene. This de novo approach is therefore particularly suitable for non-model species, but can also serve as a useful pre-processing step to improve read mapping. Our contribution both proposes a new algorithm adapted to clustering of reads by gene and a practical and free access tool that allows to scale the complete processing of eukaryotic transcriptomes.

We sequenced a mouse RNA sample using the MinION device. This dataset is used to compare our solution to other algorithms used in the context of biological clustering. We demonstrate that it is the best approach for transcriptomics long reads. When a reference is available to enable mapping, we show that it stands as an alternative method that predicts complementary clusters.

INTRODUCTION

Massively parallel cDNA sequencing by Next Generation Sequencing (NGS) technologies (RNA-seq) has made it possible to take a big step forward in understanding the transcriptome of cells, by providing access to observations as diverse as the measurement of gene expression, the identification of alternative transcript isoforms, or the composition of different RNA populations (1). The main drawback of RNA-seq is that the reads are usually shorter than a full-length RNA transcript. There has been a recent explosion in databases accession records for transcripts obtained with short reads (2) but a laborious curation is needed to filter out false positive reconstructed variants

that do not have enough support. Long read sequencing technologies such as Pacific Biosciences (3) and Oxford Nanopore Technologies (4) are referred to as Third Generation Sequencing and make it possible to sequence full-length RNA molecules. In doing so, they remove the need for transcript reconstruction before studying complete RNA transcripts (5). The size of short reads is certainly a major limitation in the process of whole transcript reconstruction, because they may not carry enough information to enable the recovery of the full sequence. In addition, tools for *de novo* assembly of transcripts from short reads (5, 6) use heuristic approaches that cannot guarantee the retrieval of exact original transcripts. On the contrary long reads tend to cover full-length cDNA or RNA molecules, and can therefore provide information about the comprehensive exon combinations present in a dataset. This gain in length comes at the cost of a computationally challenging error rate (which varies significantly between protocols, up to over 15%, although RNA reads generally show lower rates, at around 9% or less (7, 8)).

Over the last few years, increasing number of studies have been focusing on the treatment of long read data generated via the Oxford Nanopore MinION, GridION or PromethION platforms, for transcriptome and full-length cDNA analysis (4, 9, 10, 11). International projects have been launched and the WGS nanopore consortium (<https://github.com/nanopore-wgs-consortium/NA12878/blob/master/RNA.md>) has for example sequenced the complete human transcriptome using the MinION and GridION nanopores. Besides Human and microbial sequencing, this technology has also proved useful for the *de novo* assembly of a wide variety of species including nematodes (12) and plants (13) or fishes (14). It seems clear that the reduced cost of sequencing and the portable and real-time nature of the equipment compared to the PacBio technology will encourage a wide dissemination of this technology the laboratories (see Schalamun et al., A comprehensive toolkit to enable MinION long-read sequencing in any laboratory, *bioRxiv*, 2018) and many authors point out the world of opportunities offered by nanopores (15). Variant catalogs and expression levels are starting to be extracted from these new resources (16, 17, 18, 19, 20), and isoform discovery was cited as a major application of nanopore reads by a recent review (21). However, the vast majority of these works concern species with a reference. In this study we propose supporting the *de novo* analysis

of Oxford Nanopore Technologies (ONT) RNA long read sequencing data. We introduce a clustering method that works at the gene level, without the help of a reference. This makes it possible to retrieve the transcripts expressed by a gene, grouped in a cluster. Such clustering may be the basis for a more comprehensive study that aims to describe alternative variants or gene expression patterns.

Problem statement

Within a long-read dataset, our goal is to identify the associated subset of Third Generation Sequencing reads for each expressed gene without mapping them onto a reference. In order to group RNA transcripts from a given gene using these long and spurious reads, we propose a novel clustering approach. The application context of this paper is non-trivial and specific for at least three reasons: 1/ in eukaryotes, it is common that alternative spliced and transcriptional variants with varying exon content (isoforms) occur for a given gene (22). The challenge is to automatically group alternative transcripts in the same cluster (Figure 1); 2/ long reads currently suffer from a high rate of difficult indel errors (7, 8); 3/ all genes are not expressed at the same level in the cell (23, 24, 25). This leads to a heterogeneous abundance of reads for the different transcripts present. Clusters of different sizes including small ones are expected, which is a hurdle for most algorithms, including the prevalent methods based on community detection (26).

Our method starts from a set of long reads and a graph of similarities between them. It performs an efficient and accurate clustering of the graph nodes to retrieve each group of a gene’s expressed transcripts (detailed in section “*MATERIALS AND METHODS*”). A second contribution of our work is an implementation of the clustering algorithm via a tool dubbed CARNAC-LR (**C**lustering coefficient-based **A**cquisition of **R**NA **C**ommunities in **L**ong **R**eads) inserted into a pipeline (see Results section). The input of this pipeline is a whole dataset of raw reads, with no prior filter or correction needed. The output is a set of clusters that groups reads by gene without the help of a reference genome or transcriptome.

Background

Early attempts to solve this problem can be traced back to before the age of NGS: in the NCBI UniGene database (27) Expressed Sequence Tags (ESTs) are partitioned into clusters that are very likely to represent distinct genes. In fact, clustering has been the basis for gene indexing in major gene catalogues like UniGene, HGI, STACK or the TIGR Gene Indices (28, 29). Moreover this problem has come up in many disciplines, taking different forms according to the application domain. Many studies on sequence clustering worked to find the most efficient way to compute similarity but remained quite basic in their clustering scheme (e.g. CD-HIT (30), SEED (31), Uclust (32), DNACLUSt (33)). They essentially used simple schemes to try to avoid all-versus-all pairwise comparison of sequences, which became a major issue with the advent of NGS and meta-transcriptomics. These approaches and the underlying similarity measures were designed for highly similar sequences, and are also popular for applications beyond the scope of this paper such

as clustering OTUs. For proteins (34), spectral clustering has been shown to provide meaningful clustering of families. It uses the Blast E-value as a raw distance between sequences and takes all of them into account to establish a global partition of protein sequences via simple K-means clustering. This type of work cannot easily be extended to the comparison of reads, which are much less structured than protein sequences. To our knowledge no article has been published so far using spectral clustering on RNA reads. For RNA, using Sanger reads then short reads, many approaches used simple single linkage transitive-closure algorithms (EST clustering such as (35, 36, 37)), i.e. searched for connected components in a graph of similar sequences. Single linkage clustering is often used for expression data as two similar sequences are meant to merge their clusters into a single one. A problem with simple search for clusters is that it can easily lead to chimeric clusters, especially because of repetitions.

More advanced clustering strategies have therefore been developed for graphs, which use the topological properties of the graph to select relevant classes. Roughly speaking, resolution strategies can be classified into two broad camps according to applications and the community of affiliation: a *graph clustering strategy* based on the search for minimum cuts in these graphs and a *community finding strategy* based on the search for dense subgraphs. Our own approach aims to combine the best of both worlds. The first approach generally searches for a partition into a fixed number of clusters by deleting a minimum number of links that are supposed to be incorrect in the graph. The second approach frequently uses a *modularity* criterion to measure the link density and decide whether overlapping clusters exist, without assumptions regarding the number of clusters. Given that it is difficult to decide on the right number of clusters and to form them solely on the basis of minimizing potentially erroneous links, the main findings and recent developments are based on the community finding strategy and we will focus our review on this approach. *Modularity* measures the difference between the fraction of edges within a single cluster and the fraction of edges that would be observed by chance given the degree of each node. In particular *modularity*-based partitioning of sequences (38) was applied for discovering protein homology (39) or repeat sequence clustering (40). Improved state-of-the-art methods consider either overlapping communities or hierarchical communities. A well-established method for overlapping communities is the Clique Percolation Method (CPM) (41). CPM came with applications such as identification of protein families (42, 43).

Finally recent studies (44) rely on the Louvain algorithm (45) that is also based on *modularity* and looks for a hierarchy of clusters, through multi-level optimization that merges the clusters initially reduced to one element as long as *modularity* increases. This algorithm is fast because it uses a greedy strategy and is quite popular for extracting communities from large networks. However, like the other algorithms based on *modularity*, it suffers from two drawbacks: it has difficulty dealing with small clusters and is unstable in that, depending on the order of application of merges, it can produce very different results that are difficult to compare (46). Clustering problems associated with the specifics of long reads start to emerge. Such needs were already a concern in past long read literature (18, 47) and are even more

acute when a mapping strategy cannot be taken into consideration. We place ourselves in the particular framework of *de novo* identification. While several studies based on long read mapping onto a reference have been produced, methodological contributions that would make it possible to benefit from this promising data remain rare in particular for non model species. To our knowledge, two contributions (47, 48) propose respectively *de novo* detection of alternative variants, and clustering and detection of isoforms in long reads transcriptome datasets. However these tools highly rely heavily on the high accuracy provided by Pacific Biosciences Consensus Circular Sequence (CCS) long reads, and therefore do not apply to ONT reads. The method we propose is much more robust to noise.

MATERIALS AND METHODS

Input similarity graph

We define a similarity graph as an undirected graph in which nodes are reads and there is an edge between two nodes if the computed similarity between these nodes exceeds a fixed threshold. In such a graph, reads from a single gene are expected to be connected with one another because they are likely to share exons. In the ideal scenario, all reads from a gene are connected with one another. It is therefore a clique. However, the spurious nature of data imposes the use of heuristics to detect read overlaps.

In addition to the presence of genomic repeats, this leads to the expectation of a graph with both missing edges (connection missed during the search for overlapping reads) and spurious edges (wrong connections between unrelated reads), which motivates the development of tailored clustering methods.

Clustering long reads

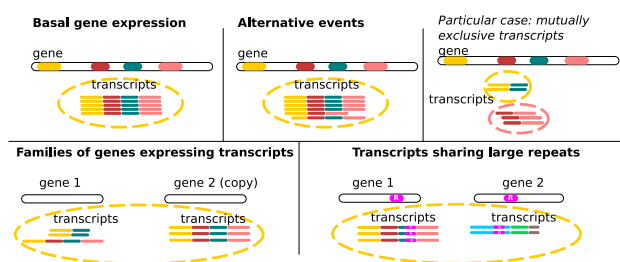


Figure 1. Clustering scenarios. In the case of basal gene expression and alternative events (described below), with the exception of mutually exclusive transcripts, it is expected that all transcripts of a gene will be grouped together in a single cluster. Very small exons or very long retained introns (not shown) can also be limitations according to the mapping tool strategies. In the more complex case of families of genes, two or more copies of paralogous genes can express transcripts at the same time. If these transcripts share a common exonic content and if the gene sequences have not diverged too much (to allow overlap detection), transcripts from this family of genes are clustered together, despite coming from different loci. Although this is an algorithmic limitation, it can be interesting to group these sequences together, as they likely share similar functions. A like scenario occurs for transcripts sharing genomic repeats (such as transposable elements).

Clustering issue and sketch of the algorithm

Problem formalization In what follows, we describe the clustering algorithm that is the main contribution of this paper. Our method makes no assumption regarding the number of expressed genes (i.e. clusters/communities), or on the size

distribution of such communities, and it needs no input parameter value in the sense that all necessary values are estimated on the data. Since we want to produce a partition of the graph, there are no intersecting communities (no read belongs to more than one gene family) and every node belongs to a community (each read is assigned to a gene). As mentioned previously, the expected subgraph signature of a gene in the graph of reads is a community, that is, a cluster of similar reads. Clustering seeks to maximize intra-cluster similarity and minimize inter-cluster similarity. To measure the density of a connected component, we use the clustering coefficient (*ClCo*) (49) rather than *modularity*. Indeed, we assume that a gene should be represented by a complete subgraph (clique) in a perfect similarity graph. The value of *ClCo* measures the concentration of triangles in a given subgraph (see “Selection of community founding node” section), and this coefficient is more directly connected to the notion of clique than *modularity*. Although we have designed a method that does not require parameter tuning, its foundation depends on two parameters, the number k of clusters and the cutoff θ on the *ClCo* value. Specifically, the original problem is formalized as follows:

DEFINITION 1. A community is a connected component in the graph of similarity having a clustering coefficient above a fixed cutoff θ . An optimal clustering in k communities is a minimum k -cut, that is, a partition of the graph nodes into k subsets, that minimizes the total number of edges between two different subsets (the cut-set).

We assume that the overlap detection procedure (section “First step: computing similarity between long reads”) has good specificity (it produces a low rate of spurious overlaps). This can be ensured by carefully tuning the parameters of this procedure. The logic behind the search for a minimum cut in the graph is that most of the edges of the initial graph should therefore be kept during clustering. This problem is known to be NP-hard for $k \geq 3$ (50). Another source of complexity is that we don’t know the number of communities in advance, so we have to guess the value of k . The k -cut should therefore be computed for each possible value between 1 and the maximum, which is the number of reads. Solving this problem is not feasible for the large number of reads that have to be managed. We are therefore looking for an approximation of the solution by using an efficient heuristic approach exploring a restricted space of values for k . Finally, the second parameter, the cutoff θ , is not known either. The algorithm thus has to loop over all possible values, that is, all *ClCo* values for a given connected component. In practice it is enough to sample a restricted space of possible k values.

Algorithm overview In brief, our community detection algorithm is composed of two main steps. The first one looks for an upper bound of the number of clusters k . To this aim, we relax the disjointed communities condition and look initially for star subgraphs (a read connected to all reads similar to it) having a clustering coefficient over a certain cutoff. This corresponds to detecting well-connected reads, called seed reads, using *ClCo* and node degrees (detailed in section “Selection of community founding nodes”). They form the basis of communities with their neighborhood.

The main challenge is then to refine the boundaries of each

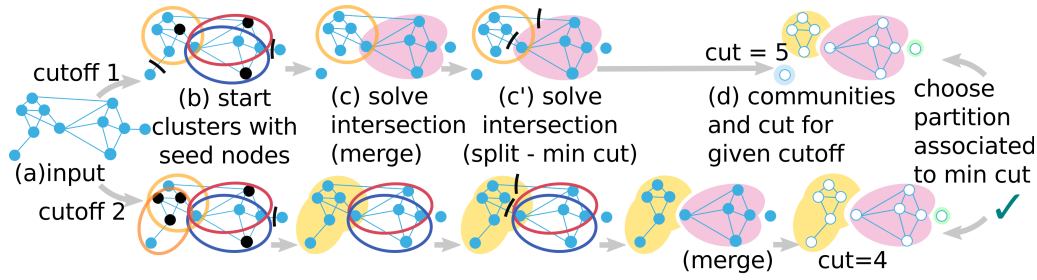


Figure 2. Summary of the algorithm. (a) All $CICo$ and degrees are computed. Each $CICo$ value is a cutoff. For a given cutoff, (b) different cutoffs yield different seed nodes (black stroke) that initiate clusters with their neighborhood (section “*Selection of community founding nodes*”). (c, c’) Boundaries of each cluster are then refined. Intersection between clusters are solved either by (c) merging them or by (c’) splitting (section “*Refinement of community boundaries*”). (d) The communities at different cutoffs evolve in different partitions. In the end we keep only the best partition according to our criterion, i.e. minimizing the cut.

community (section “*Refinement of community boundaries*”) in order to fulfill the partition condition. During this process, the value of k is progressively refined by potentially merging clusters whose combination produces a better community (greater $CICo$ value). The other possibility of refinement is to assign nodes to a community and remove them from another. If x edges between a node and its previous community are removed, the cut size of the partition is increased by x . This core algorithm is run for different cutoff values to obtain different partitions that we compare. We keep the partition associated with the minimal cut (i.e. number of edges removed when computing the partition). The pseudocode of the implemented algorithm is given in “*Supplementary material*”. We set out the different implementation steps in detail below.

Generation of partitions

In order to generate and compare different partitions for the graph, we define cutoffs that govern the generation and refinement of communities. The cutoffs can be seen as the level of connectivity at which a community can be generated ((a,b) steps and (c) merge step in Figure 2). In the basic algorithm, for each connected component, all $CICo$ are computed in the first place, and partitions are built for each non-zero $CICo$ value as a cutoff. In the end, only one partition is retained, associated with the minimum cut (step (d) in Figure 2). However we have reduced the number of possible cutoff values for the sake of scalability (see “*Implementation choices for scalability*” section in “*Supplementary material*”). Each step is described for a given cutoff value below.

Selection of community founding nodes

Let $\mathcal{G} = (\mathcal{N}, \mathcal{E})$ be an undirected graph of reads. Let n_i be a node (read) from \mathcal{N} and $N_i \subset \mathcal{N}$ its direct neighborhood. Let $deg(n_i)$ be the number of edges connecting n_i to its direct neighbors (similar reads), i.e. $deg(n_i) = |N_i|$. For each node $n_i \in \mathcal{N}$ with degree $deg(n_i) > 1$, first we compute the *local clustering coefficient*:

$$CICo_i = \frac{2 |\{(n_j, n_k) \in \mathcal{E} : n_j, n_k \in N_i\}|}{deg(n_i) \times (deg(n_i) - 1)} \quad (1)$$

Nodes of degree 0 and 1 have a $CICo$ of 1. This local coefficient represents the *cliqueness* of the $N_i \cup n_i$ set of nodes. The closer the coefficient is to 1, the more the set of nodes is inter-connected, which suggests a group a reads

that potentially come from the same gene. By contrast, the subgraph induced by a node with a $CICo$ of 0 and its neighbors is a star (i.e. a tree whose leaves are all the neighbours). If the coefficient is close to 0, the nodes are weakly connected and are unlikely to come from the same gene. In order to prevent unwanted star patterns, we add a statistical precaution to prevent star-like patterns (with a very low $CICo$ with respect to the degree of the seed node) from initiating communities. We state the following auxiliary condition for seeds:

$$\forall n_i, CICo_i \in]cutoff, \theta_2[\Rightarrow deg(n_i) \leq \theta_1 \quad (2)$$

θ_1 and θ_2 are values such that 1% of the observed degrees are greater than θ_1 and 1% of the observed $CICo$ are lower than θ_2 (1st and 99th percentiles). The selected seeds and their direct neighbors form the initial communities. At this point it is possible that two or more communities intersect.

Refinement of community boundaries

Community refinement aims at solving the conflicts of intersecting communities. Communities intersect because of spurious connections in the graph due to the creation of edges between unrelated reads in the first step.

The intersecting communities are looked up pairwise in order to assign nodes of the intersection to a single community. In fact two cases have to be differentiated. Either the edges between two communities are considered spurious and these communities must be seen separated (*split*, (c’) step in Figure 2 (the pseudocode for the *split* procedure is also given in “*Supplementary material*”), or the edges have sufficient support and the two communities have to be merged to obtain the full gene expression (*merge*, (c) step in Figure 2). In order to decide between the two, we again use the *cliqueness* notion. This time we introduce an *aggregated clustering coefficient* of the union of two nodes n_i and n_j :

$$CICo_{ij} = \frac{2 |\{(n_k, n_l) \in \mathcal{E} : n_k, n_l \in N_i \cup N_j\}|}{|N_i \cup N_j| \times (|N_i \cup N_j| - 1)} \quad (3)$$

If the value of $CICo_{ij}$ is greater than or equal to the current cutoff, we consider that there is a gain in connectivity when looking at the union of the two communities and they are merged. In the other case, the nodes of the intersection are reported to only one of the two communities. We remove the

edges connecting these nodes from one or the other cluster depending on which offers the minimum cut. In case of ties for the cut, the algorithm uses a second criterion. It chooses the cut that maximizes the differences of clustering coefficient values across communities. For two sets $N_1, N_{1'}, N_1 \subseteq N_{1'}$, this difference is defined as :

$$\Delta CC_{N_1, N_{1'}} = CC_{N_{1'}} - CC_{N_1}, \quad (4)$$

with CC calculated as in Equation 1, N_1 being the community before the merge and $N_{1'}$ being the community after the merge. The overall result depends on the order in which pairs of clusters are compared. This order is carefully designed. First, the communities associated with the two nodes of greatest degree (and secondly maximum $ClCo$) are chosen, the intersection is resolved and the first community is updated. Then, it is compared to the third best community that intersected if appropriate, and so on until all intersections are solved. This way, we start the comparison with the most promising communities that combine reliability (well-connected subgraphs) with a high potential for resolution (they likely to be the biggest communities, thereby solving intersections for many nodes). On the contrary, communities associated with small subgraphs and relatively low $ClCo$ are only resolved afterwards.

Complexity and Implementation choices

Our algorithm has a quadratic component in order to compare sets to generate clusters. In addition, it explores the whole space of clustering coefficients with fixed cutoffs. This results in a time complexity that could theoretically be cubic in the number of reads at worst, which is incompatible with processing large datasets.

In order to cope with noise in the input graph, we introduce features to simplify the graph (disconnect loosely connected nodes) and to control the space for looking for possible partitions. In practice these features are also key to reducing the complexity of our approach. Our experiments showed that the running time is reasonable, clustering millions of reads in a few hours. Two key ideas for obtaining this result have been reducing the number of cutoffs and disconnecting the articulation points (51) to reduce the size of connected components in the graph. Details are given in “Supplementary material”.

Indeed, the most costly phase involves processing of the largest connected components. In these components, many clustering coefficients values are very close and variation in them is mainly an result of noise. Introducing a rounding factor when computing the $ClCo$ results in a neat decrease in the number of different values observed, which drastically limits the number of iterations required for the main loop, while providing a very good approximation of the minimum cut. In addition, an upper bound is set on the number of sampled values (100 by default).

We also chose to disconnect the graph’s *articulation points* in order to remove nodes to be targeted as potential bridges between two correct clusters. These are nodes whose removal increases the number of connected components in the graph. Such nodes can be identified as problematic, since we do not expect a single read to be the only link between many others.

They can be detected with a DFS for the whole graph in linear time.

Our algorithm has been also carefully designed with respect to the features of long read clustering. To obtain a $O(n \cdot \log(n))$ complexity with respect to the number n of reads, we have made the following assumption: The degree of each node is bounded by a constant, i.e. there is a limited number of transcripts that share similar exons. This ensures that the clustering coefficient of all nodes is calculated in linear time. The most complex operation is the initial sorting of nodes, first by decreasing degree value, then by decreasing $ClCo$ value, which can be achieved in $O(n \cdot \log(n))$. Moreover, since each cluster is initially built on a seed read (see “Selection of community founding nodes” paragraph), it intersects with a bounded number of clusters. Since the loop for making a partition from overlapping clusters is based on a scan of intersections, it is achieved in linear time with respect to the number of reads.

Validation procedure

Production of validation material

RNA MinION sequencing cDNA were prepared from 4 aliquots (250ng each) of mouse commercial total RNA (brain, Clontech, Cat# 636601 and 636603), according to the ONT (UK) protocol “1D cDNA by ligation (SQK-LSK108)”. The data generated by MinION software (MinKNOWN, Metrichor) was stored and organized using a Hierarchical Data Format. FASTA reads were extracted from MinION HDF files using poretools (52). We obtained 1,256,967 nanopore 1D reads representing around 2GB of data with an average size of 1650 base pairs and a N50 of 1885 base pairs.

Mapping to obtain reference clusters for validation We compute “ground truth” clusters for the purposes of validation, using a sensitive third-party protocol based on mapping onto a reference. Nanopore reads from the mouse brain transcriptome were aligned to the masked mouse genome assembly (version GRCh38) using BLAT (53) used for isoform identification with long reads in various studies (21). For each read, the best matches based on the BLAT score (with an identity percent greater than 90%) were selected. Then, those matches were realigned onto the unmasked version of the genome using Est2genome (54) that is dedicated to precise spliced-mapping onto reference genomes. Reads that corresponded to mitochondrial and ribosomal sequences were discarded. Next, nanopore reads were clustered according to their genomic positions: two reads were added to a given cluster if they shared at least 10 nucleotides in their exonic regions. For the whole data experiment, all reads that could be mapped on the reference were taken into account (501,787). Due to repeats (paralogy, transposable elements, etc), some reads were mapped at multiple loci on the reference. When a given read maps on several loci, such loci are gathered into a single expected cluster (12,596 expected clusters). This means that for instance reads from copies of paralog genes that have not diverged to much or reads containing a copy of a transposable element are expected to be in the same cluster.

Assessment metrics for cluster accuracy

To assess the results, we used recall and precision measures, which are standard measures for assessing the relevance of

biological sequence clustering (55). The recall and precision measures are based on reference clusters obtained by mapping for this validation. They are computed based on representative graphs (56). These measures had already been used to assess the relevance of biological sequence clustering (55). Let $\{C_1, \dots, C_i\}_{1 \leq i \leq L}$ be the set of clusters found by the clustering method tested, where L is the number of predicted clusters. Let $\{K_1, \dots, K_j\}_{1 \leq j \leq K}$ be the set of “ground truth” clusters, where K is the number of expected clusters. Let R_{ij} be the number of nodes from C_i that are in “ground truth” cluster K_j . We compute the recall and the precision such as:

$$Recall = \frac{\sum_{j=1}^K \max_i(R_{ij})}{\sum_{i=1}^L \sum_{j=1}^K R_{ij}} \quad (5)$$

$$Precision = \frac{\sum_{i=1}^L \max_j(R_{ij})}{\sum_{i=1}^L \sum_{j=1}^K R_{ij}} \quad (6)$$

Note that the “ground truth” is not really available and that it is estimated from mapping results onto the reference genome. The recall expresses the mean over all clusters of the fraction of relevant reads in a result cluster out of the expected read population of this cluster. It shows to what extent clusters are complete. The precision expresses the mean over all clusters of the fraction of relevant reads among the population of a result cluster. It shows the clusters’ purity. The F-measure is a summary measure computed as the weighted harmonic mean between precision and recall. Recall and precision are tailored to express the confidence that can be placed in the method, according to its ability to retrieve information and to be precise. We also assess the closeness of the computed clusters as compared to mapping approaches. Let \mathcal{X}_0 be the reference partition (set of clusters obtained by mapping), and \mathcal{X} the partition obtained using a given clustering method. Then a_{11} is the number of pairs of nodes that are placed in a single cluster in \mathcal{X}_0 and \mathcal{X} and a_{00} is the number of pairs for which nodes are placed in different clusters both in \mathcal{X}_0 and \mathcal{X} . a_{10} (resp. a_{01}) is the number of pairs of nodes placed in the same cluster in the reference \mathcal{X}_0 (resp. \mathcal{X}) but in different clusters in \mathcal{X} (resp. \mathcal{X}_0). On this basis, a metric such as the Jaccard index shows the match between the reference and computed partitions:

$$J(\mathcal{X}_0, \mathcal{X}) = \frac{a_{11}}{a_{11} + a_{10} + a_{01}} \quad (7)$$

The Jaccard index is between 0 and 1. The closer it is to 1, the more the set of clusters computed by a method is close to the “ground truth” set of clusters predicted.

RESULTS

All experiments were run on Linux distribution with 24 Intel Xeon 2.5 GHz processors, 40 threads and 200 GB

RAM available. First we present the tool we have developed and made available for large scale long-reads clustering. We demonstrate it performs well on a canonical example on which other clustering approaches were assessed. We compare our approach to well established community detection methods and demonstrate its relevance to long read application. Then we validate our method’s results by comparing them with independent clusters obtained by mapping a real size dataset. In these two parts (sections “*Comparison to state of the art clustering algorithms*” and “*Biological relevance*”), reads from the mouse brain transcriptome were used in order to access a “ground truth” via a reference. Then we show that our approach offers an alternative to the classical mapping approach even when a reference is available.

CARNAC-LR, a software for long read clustering

Input/Output

We implemented our novel algorithm presented in section “*MATERIALS AND METHODS*”, integrated into a pipeline called CARNAC-LR. It starts with the computation of long read similarities via a program called Minimap (57) and then produces the clusters. The pipeline’s input is a FASTA file of reads. The output is a text file with one line per cluster, each cluster containing the read indexes. Each read is represented by its index in the original FASTA file during CARNAC-LR computation. Then using indexes, each cluster can easily be converted to a FASTA file where each read’s sequence is retrieved from the original file (a script is provided for this task).

First step: computing similarity between long reads

We chose the Minimap tool for its efficiency and its very high level of precision on ONT and PB (58), with regard to other recent methods that can compute similarity or overlaps between long reads despite their error rate (59, 60, 61, 62). To generate the similarity graph for CARNAC-LR, Minimap version 0.2 was launched with parameters tuned to improve recall (`-Sw2 -L100`). It produces a file of read overlaps in .paf format.

Second step: clustering

Minimap’s output is converted into a graph of similarity, where each node represents a read and an edge a sequence similarity between two reads above a certain threshold (see (57)). This graph is then passed to CARNAC-LR that retrieves and outputs the gene clusters. CARNAC-LR benefits from parallelization. A thread can be assigned to the treatment of a single connected component, thus many connected component can be computed in parallel. Further results on scalability are provided in “*Supplementary material*”.

Method validation

Input graph

In order to compare different clustering methods, we generated an input graph from the mouse dataset. We ran all methods on the same input graph, pre-processed using the procedure described in the “*Complexity and Implementation choices*” section. For scaling purpose, we chose to perform the benchmark on a subset of 10K reads (9,609 mouse reads within 527 reference clusters determined by mapping,

section “*Production of validation material*”). This sampling shows the effect of high gene expression on clustering. We also checked on a second 10K sample from the whole dataset that further accentuates the low expression effect. Directly after Minimap, the graph has 701 connected components, then pre-processing is applied to obtain an input graph that has 1109 connected components. We present a binned distribution of the input graph degrees in Figure 3. Finally, the input graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ has the following properties: $|\mathcal{V}| = 8539$, $|\mathcal{E}| = 143317$, graph clustering coefficient: 0.049991, graph average geodesic distance: 8.812268 and graph diameter: 24.

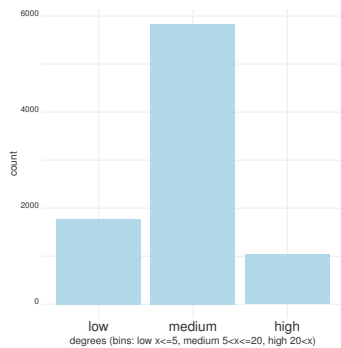


Figure 3. Binned distribution of nodes' degrees in the input graph.

Comparison to state of the art clustering algorithms

We show results of state of the art algorithms and compare them to our tool's results. We compared CARNAC-LR results to algorithms we identified as close to the solution we propose. We evaluated four state-of-the-art methods that have been previously applied to similar biological clustering problems: single linkage transitive-closure (35, 36, 37), *modularity* (39, 63, 64), Clique Percolation Method (42, 43) and Louvain (44, 65). The results are presented Table 1. Our method has the

Table 1. Comparison with state of the art methods. The benchmark was realized on a 10K reads dataset from the mouse chromosome 1. Recall precision and Jaccard Index are presented (see Equation 7, 5 and 6) to assess for the goodness of communities detection. CPM3 denotes the CPM algorithm using $k=3$. “Clusters” column shows the number of output clusters of size > 1 .

	Recall (%)	Precision (%)	F-measure (%)	Jaccard index	#clusters
Transitive closure	63.86	87.20	73.72	$6.7E^{-1}$	731
Modularity	60.70	71.16	65.51	$2.6E^{-1}$	733
CPM3	63.03	87.17	73.16	$6.7E^{-1}$	536
Louvain	81.01	14.71	28.89	$3.6E^{-2}$	53
CARNAC-LR	60.16	98.04	74.57	$7.1E^{-1}$	748

best precision and the best overall trade-off between precision and recall as shown by the F-measure. It also has the highest Jaccard index among all tested approaches. Louvain's results were tested for each of its iterations, we present here the best result (other results are given in “Supplementary Material”). Despite showing the best recall, Louvain's precision is too low to reach a high F-measure or Jaccard index. The *modularity*-based method achieves average recall and precision values, but one of the lowest Jaccard indices. The transitive closure approach and the CPM are the two other methods that show good results on this instance. The CPM was tested with values for input parameter k ranging from 3 to 145 (no community found for greater values). Results are presented for $k=3$ and show the method's best performance. For higher values of k , the precision increases up to more than 98%, however

the recall is dragged down to less than 15% (details shown in “Supplementary Material”). Both the CPM and transitive closure present a precision which is more than 10% inferior to our method. As CARNAC-LR is designed for general pipelines providing a complete analysis of gene variants, it is important that it does not mix two unrelated genes in a single cluster. Our approach is therefore more conservative than CPM and searching for connected components, and it shows comparatively good results in any case. Furthermore it needs no input parameter. Results on the other sample present the same trend as than those presented (shown in “Supplementary material”) and demonstrate that CARNAC-LR also deals better with shallow coverage.

Comparison to other nucleic acid sequence clustering tools

We have simply situated the CARNAC-LR algorithm in relation to existing general cluster detection methods, but we still have to compare our pipeline to other tools dedicated to the comparison of nucleotide sequences that have been developed for the same clustering task. We started with one of the most powerful tools currently available, Starcode (66), which was designed for reads correction and offers a benchmark for the most widely used clustering tools, which we have adopted. This includes CD-HIT (30), SEED (31) and Rainbow (67). It should be noted that none of these tools have been designed to work with ONT reads. They were created before the full development of long reads technology, they have proven not surprisingly completely ill-suited to clustering these long reads. For this test, we used the same mouse dataset as in the previous section. The methods stumble over two features of the data: the error rate and the length of the sequences. SEED for instance is designed to create clusters with sequences that show a maximum of 3 mismatches, and so finds no clusters. Starcode is not adapted to the size of ONT sequences and terminates with an error message. We tried to increase the maximum size allowed for sequences (initially set at 1024) but the memory consumed continued to grow rapidly and reasonable capacities (200GB) were quickly exceeded. We then tried to perform the calculation by rejecting the longest reads but like SEEDS, Starcode authorizes a limited distance between pairs of sequences (a maximum Levenshtein distance of 8) which is far too small for ONT reads, resulting in singleton clusters. Rainbow only accepts paired reads such as those sequenced in RAD-seq short reads experiments and cannot be adapted to our problem. Finally the most flexible tool, CD-HIT, was the only one to give results. Its “EST” version was used. We tested different values for sequence identity threshold ($-c$), that can be decreased down to 0.8. We report only the best result, for $-c = 0.8$. It is a long way below the result obtained by CARNAC-LR (F-measure up to 41.96% due to low recall, compared with 86.62% for CARNAC-LR). In addition, our pipeline is substantially faster with memory consumption in the same range (within a factor of 2). In view of these results, we added Tofu (48), the only other *de novo* clustering tool that, to our knowledge, is designed to work with long reads, to the benchmark. Unfortunately, Tofu relies heavily on the specificity of Pacific Bioscience RNA protocol (Isoseq) sequences, and cannot be run with ONT reads. Incidentally, the aim of Tofu differs from CARNAC-LR as it is expected to retrieve one cluster per isoform rather than one cluster per expressed gene. A detailed summary of this

benchmark result is presented in “Supplementary materials”. Once again, another sampling on mouse chromosome 1 was used to perform a second benchmark that presents same conclusions, as also shown in “Supplementary material”.

Biological relevance

Validation on a real size dataset

Clusters quality In this experiment we demonstrate the quality of *de novo* clusters obtained using CARNAC-LR. We used the subset of reads that could be mapped onto the mouse genome reference (501,787 reads) as a means of comparison for assessing the biological relevance of our clusters. CARNAC-LR’s results were computed using 43 GB RAM and took 18 minutes.

The mean recall for CARNAC-LR was 75.38% and the mean precision was 79.62%. In other words, retrieved clusters are on average 75.38% complete, and on average 79.62% of the clusters are composed of unmixed reads from the same gene. In order to assess whether our method’s recall and precision is consistent regardless of the gene expression levels, we computed expression bins. For a given gene, we use the number of reads of the “ground truth” cluster to approximate an expression. Any “ground truth” cluster with 5 or less reads is placed in the first bin, and so on for 5-10, 10-50 and ≥ 50 reads categories. Each of the four bins represent quartiles of expression, which means there is an equal number of clusters in each bin. Figure 4 presents the recalls obtained for binned expression levels and shows our approach’s recall and precision remain consistent despite the heterogeneous coverage in reads.

Furthermore, we can deduce from this plot that small clusters do not bias the presented mean recall and precision, as even for big clusters (i.e. ≥ 50 expression bin) that are more prone to lose some reads, these metrics remain high.

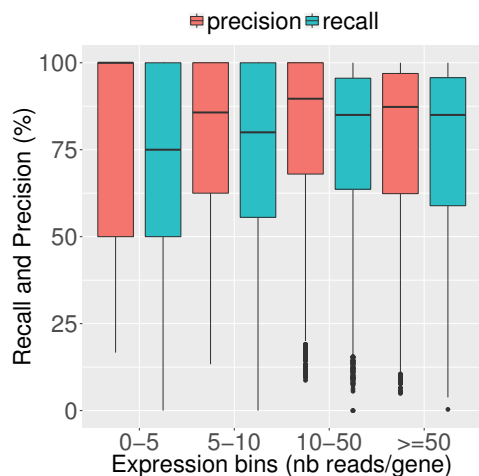


Figure 4. Assessed mean recall and precision of CARNAC-LR+Minimap. They were computed on mouse reads using clusters found by mapping on a reference as a “ground truth” (see Equations 5 and 6). Expression bins are computed from quartiles of expression predicted by mapping and represent the number of mapped reads by gene. Mean precision and recall over all clusters falling in these bins were then calculated.

Output excerpt Once CARNAC-LR has been run, FASTA files can be extracted for each cluster. We selected the sequences

contained in a cluster after CARNAC-LR’s pass on the mouse transcriptome. We used a genome browser to graphically show the reads that were grouped by our approach. (Figure 5). We selected a cluster of sufficient size to be able to present a variety of isoforms. This corresponds to a gene for which mapping retrieved 120 reads. In this example, our approach recovered 93% of the predicted gene’s reads while including no unrelated read in the cluster. Two types of missed reads can be distinguished: 1) A group of black reads that share no genomic sequence with the majority of the gene’s transcript, because they come from an intronic region. These reads could not be linked to the others by Minimap and therefore cannot be clustered with them, as shown in the particular case described in Figure 1. 2) Two other reads for which local connectivity was not detected by Minimap were discarded from the cluster. The image shows different exon usage in transcripts, which reveals alternative splicing in this cluster. Different alternative isoforms were therefore gathered in a single cluster as expected (see Figure 1).

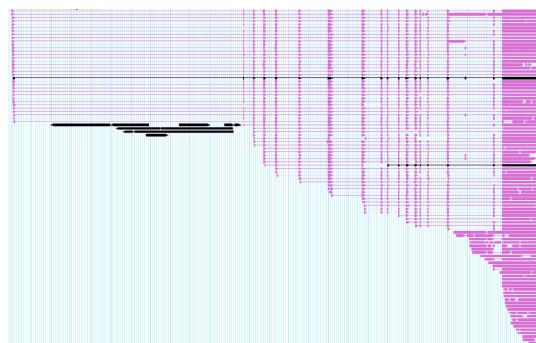


Figure 5. Example of CARNAC-LR’s output cluster in mouse. The output of CARNAC-LR is a text file with one line per cluster, each cluster containing the read indexes. We selected a predicted cluster made of 112 reads (purple). For validation purpose these reads were mapped with BLAST on an in-house igv (68) version for mouse genome. Reads are spliced-mapped, bold parts are the mapped sequences from the reads and thin parts represent the gaps between the different mapped parts of the reads. Despite the staircase effect observed in the data, this allows to notice that several types of variants were gathered. They could all be assigned to gene *Pip5k1c* (chr 10:81,293,181-81,319,812), which shows no false positive was present in this cluster. 8 reads (black) present in the data are missed in this cluster. The group of 6 black reads on the left represent intronic sequences and share no sequence similarity with the others and thus could not appear in the same cluster.

Complementarity of *de novo* and reference-based approaches

Intersection and difference with the set of mapping clusters

Since it does not rely on any reference information, our approach putatively yields different results than classical mapping approaches. In this section, we investigate the differences between the two approaches and demonstrate the usefulness of CARNAC-LR even if a reference is available. We ran it on the full mouse brain transcriptome dataset (1,256,967 reads). We compared the intersections and differences of the results of our approach and mapping. The CARNAC-LR+Minimap pipeline took less than three hours (using 40 threads). In comparison, the “ground truth” clusters took 15 days to be computed (using up to 40 threads). Our approach was able to place 67,422 additional reads that were absent in the mapping procedure, resulting in 39,662 clusters.

These clusters fall in two categories (i) *common clusters* with a mix of reads processed by our approach and/or by mapping, or (ii) *novel clusters* that contain reads processed exclusively by our approach or mapping. Each approach performed differently on these categories.

Common clusters For category (i), mapping complemented many common clusters with small amounts of reads left aside by our approach. As some reads are processed by mapping, a recall and precision can still be computed using mapping as ground truth. We computed recall and precision based on the read fraction of clusters that could be compared with mapping. They are quite similar to the values obtained in the previous section (mean recall 75.26% and mean precision 79.30%). This demonstrates that CARNAC-LR efficiently used the supplementary connectivity information despite the addition of potentially noisy reads.

Novel clusters Conversely CARNAC-LR shows a better ability to group reads unprocessed by mapping into novel clusters (Figure 6). CARNAC-LR produced 824 novel clusters (17,189 reads) of category (ii) containing at least 5 reads. In order to assess the relevance of these novel clusters, we remapped reads *a posteriori*, separately for each cluster, onto the reference genome using a sensible approach (GMAP (69) version 2015-09-29). This operation took approximately 10 hours (using 4 threads). 19.68% of mapped reads were assigned to the MT chromosome, then chromosome 11 represented 10.85% of the reads, and other chromosomes less than 10% of mapped reads each. A third of the reads were multi-mapped (36.7%). However, on average, for each cluster 98.89% of the reads shared a common genomic locus. This is consistent with the expected results of the clustering for reads containing repeats or paralog regions (Figure 1). Finally, 5.7% of the clusters exclusively contained reads mapped at a single locus. All of them could be assigned to an annotated gene. So even if a reference was available, our approach was able to retrieve *de novo* expressed variants of the genes that were unassigned by the mapping.

Correlation of expression levels Another way to look at these results is two consider the number of reads predicted by each method as the gene’s expression, and to compare expression levels predicted by our approach and by mapping. We showed that, despite the previously described differences, they are highly and linearly correlated, with a Pearson correlation coefficient of 0.80 (Figure 7).

DISCUSSION

CARNAC-LR is well-suited to transcriptome instances

We demonstrated that our approach can compete with the state-of-the-art algorithms to detect communities. With just a fairly small example, state of the art approaches at least show a lack of precision in comparison to our approach. We showed that a *modularity*-based algorithms such as Louvain algorithm are not well-tailored to this problem, probably because of the heterogeneous size distribution of the clusters, and because of overlapping effects due to the repeats. Among tested state-of-the-art approaches, only the CPM qualifies for retrieving clusters in our input graphs. However, by concentrating its results in a small subset of clusters, it obtains a poor recall and not all its predicted clusters can be trusted. On the other hand

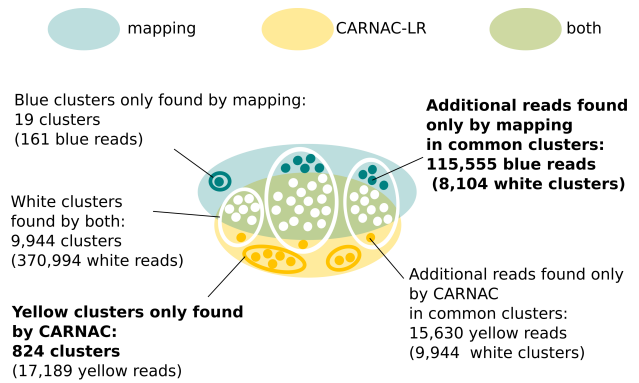


Figure 6. Complementarity of CARNAC-LR and mapping approaches. Only clusters of size ≥ 5 are represented. Mapping complemented common clusters with a mean 13 reads per cluster in 90% of clusters. CARNAC-LR’s supply was tenfold lower with a mean 1,3 read added to 100% of common clusters. On the other hand, CARNAC-LR retrieved 15 fold more novel cluster than mapping.



Figure 7. Comparison of clustering and mapping approaches. Comparison and correlation of expressions levels. Gene’s expression can be inferred by counting the number of reads by gene. For each gene we counted the number of reads retrieved by mapping and we compared it to the number of reads reported by our pipeline and validated by mapping. We computed the Pearson correlation coefficient between the two (in green). Density is the number of points counted in a colored region. Despite a few outliers, we can see a strong linear correlation between the two expression levels estimations (plotted in black). 7 outliers above 750 on Y axis (up to 3327) are not shown.

our approach shows a good consistency. We supplemented these results with a comparison with tools extensively used for clustering nucleotide sequences, including developments used for EST clustering such as CD-HIT EST. We have shown that no published tool is currently capable of producing quality clusters from ONT RNA reads. We validated CARNAC-LR’s results using mouse transcriptome ONT reads, showing we can compute high confidence clusters for many genes. We underlined that the mapping procedure used for producing reference clusters for validation has its own limitations. Thus the “ground truth” we refer to for the sake of clarity is in fact only partial.

CARNAC-LR can complement mapping approaches with respect to data with reference

Long reads make it possible to skip the transcript reconstruction step that is necessary with short reads, although this is particularly difficult when it involves assembly. Therefore, long reads constitute an interesting and novel way of obtaining reference transcripts. However, only a fraction of long reads are processed by mappers and downstream analysis is made difficult because of the error rates. In this context, our approach is shown to be an alternative approach to mapping for the identification of gene transcripts. We have shown that our pipeline could be a complementary procedure when reads can be mapped to a reference. It tends to recover some clusters missed by mapping and allows a more efficient use of data. We have demonstrated a straightforward use case for our pipeline as a good proxy for accessing expression levels by gene. ONT sequences have been shown to qualify for transcript quantification in Oikonomopoulos et al., (9). In a long read sequencing experiment, it is likely that some reads contain too many errors to be mapped onto a genome. CARNAC-LR can help identifying the origin gene of such reads, if they are clustered with other mapped reads. Moreover CARNAC-LR provides structured information that can be a sound input for other applications. For instance, a read correction step can be performed on each cluster instead of processing all the data, in order to obtain high quality reference transcripts.

CARNAC-LR applies on non-model species and ONT data

Non model species require *de novo* approaches, and two bioinformatics tools dedicated to them have emerged so far (47, 48). Both comprise a pipeline conceived to process Pacific Biosciences Isoseq (3) reads only and require high quality long reads. Thus they could not be used on the data presented here. On the other hand CARNAC-LR is a generic approach that is designed to be used regardless of Third Generation Sequencing error profile and protocol. As a consequence it is the first method to perform *de novo* clustering on RNA reads from ONT.

Paralogy and repeats

The clustering of sequences from transcriptome reads is made difficult by the existence of multiple repeats. This first attempt to cluster RNA reads by gene is not designed to precisely assign reads from paralog genes to their original locus. We argue that specific instances such as paralog genes constitute research themes on their own and the clustering provides first-approximation results in these cases. We can imagine a second clustering pass with the development of an adapted similarity calculation. CARNAC-LR gathers all reads from a gene family together, provided the different copies have not diverged too much and can therefore be seen as a useful pre-processing step for the analysis of paralogs. A related research axis would be to describe how repeats like transposable elements that can be found in exons or retained introns are treated by the clustering procedure.

CONCLUSION

We propose a method for clustering long reads obtained from transcriptome sequencing in groups of expressed genes. New algorithmic challenge arises with the combination of a high error rate in the data (7, 8), a high heterogeneity of coverage typical of expression data and a large volume of data. In this, our issue differs from EST clustering problems for instance. We demonstrated our method’s relevance for this application, in comparison to literature approaches. It takes reads early after their generation, without correction or filtering. The expressed variants of each gene are obtained from the clusters, and related transcripts are identified, even when no reference is available. To make our solution practical for users, we provide an implementation called CARNAC-LR that, combined with Minimap, scales and is able to quickly process real data instances, as demonstrated by the processing of the whole mouse brain transcriptome.

As a result of the quick development of TGS, the sequencing field is frequently upgraded with new types of sequences. For instance, recent long read technology ONT RNA-direct could unlock amplification biases issues in RNA sequencing and is therefore promising for gene expression studies (see Garalde et al., Highly parallel direct RNA sequencing on an array of nanopores, *bioRxiv*, 2016). But it shows higher error rates, at least compared to reads presented in this study, according to unpublished works. By proposing a tool tailored to ONT, we wish to promote and encourage a broader use of these long reads for transcriptome analysis.

Data availability and Implementation CARNAC-LR is written in C++, open source and available for Linux systems at github.com/kamimrcht/CARNAC-LR under the Affero GPL license.

The nanopore reads from the mouse RNA sample are available from the ENA repository under the following study : ERP107503.

ACKNOWLEDGEMENTS

The study has been partially supported by ANR ASTER, contract ANR-16-CE23-0001, which provided the data. Authors would like to thank all members of ASTER as well as GenScale team members for their support and fruitful discussions, and Gaëtan Benoit and Antoine Limasset for their precious help on the implementation. Computations were made possible thanks to the resources of the GenOuest infrastructures.

Conflict of interest statement. None declared.

REFERENCES

1. Kimberly R Kukurba and Stephen B Montgomery. Rna sequencing and analysis. *Cold Spring Harbor protocols*, 2015(11):pdb-top084970, 2015.
2. Nuala A O’Leary, Mathew W Wright, J Rodney Brister, Stacy Ciuffo, Diana Haddad, Rich McVeigh, Bhanu Rajput, Barbara Robbertse, Brian Smith-White, Danso Ako-Adjei, et al. Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. *Nucleic acids research*, 44(D1):D733–D745, 2015.
3. Manuel L Gonzalez-Garay. Introduction to isoform sequencing using pacific biosciences technology (iso-seq). In *Transcriptomics and Gene Regulation*, pages 141–160. Springer, 2016.
4. Jason L Weirather, Mariateresa de Cesare, Yunhao Wang, Paolo Piazza, Vittorio Sebastiano, Xiu-Jie Wang, David Buck, and Kin Fai Au. Comprehensive comparison of pacific biosciences and oxford nanopore technologies and their applications to transcriptome analysis. *F1000Research*, 6, 2017.
5. Manfred G Grabherr, Brian J Haas, Moran Yassour, Joshua Z Levin, Dawn A Thompson, Ido Amit, Xian Adiconis, Lin Fan, Raktima Raychowdhury, Qiandong Zeng, et al. Trinity: reconstructing a full-length transcriptome without a genome from rna-seq data. *Nature biotechnology*, 29(7):644, 2011.
6. Cole Trapnell, Adam Roberts, Loyal Goff, Geo Pertea, Daehwan Kim, David R Kelley, Harold Pimentel, Steven L Salzberg, John L Rinn, and Lior Pachter. Differential gene and transcript expression analysis of rna-seq experiments with tophat and cufflinks. *Nature protocols*, 7(3):562–578, 2012.
7. David Laehnemann, Arndt Borkhardt, and Alice Carolyn McHardy. Denoising dna deep sequencing data – high-throughput sequencing errors and their correction. *Briefings in bioinformatics*, 17(1):154–179, 2015.
8. Camilla LC Ip, Matthew Loose, John R Tyson, Mariateresa de Cesare, Bonnie L Brown, Miten Jain, Richard M Leggett, David A Eccles, Vadim Zalunin, John M Urban, et al. Minion analysis and reference consortium: Phase 1 data release and analysis. *F1000Research*, 4, 2015.
9. Spyros Oikonomopoulos, Yu Chang Wang, Haig Djambazian, Dunarel Badescu, and Jiannis Ragoussis. Benchmarking of the oxford nanopore minion sequencing for quantitative and qualitative assessment of cdna populations. *Scientific reports*, 6:31602, 2016.
10. Mohan T Bolisetty, Gopinath Rajadinakaran, and Brenton R Graveley. Determining exon connectivity in complex mRNAs by nanopore sequencing. *Genome biology*, 16(1):204, 2015.
11. Adam D Hargreaves and John F Mulley. Assessing the utility of the oxford nanopore minion for snake venom gland cdna sequencing. *PeerJ*, 3:e1441, 2015.
12. David Eccles, Jodie Chandler, Mali Camberis, Bernard Henrissat, Sergey Koren, Graham Le Gros, and Jonathan J. Ewbank. De novo assembly of the complex genome of nipponstrongylus brasiliensis using minion long reads. *BMC Biology*, 16(1):6, Jan 2018.
13. Maximilian H.-W. Schmidt, Alexander Vogel, Alisandra K. Denton, Benjamin Istace, Alexandra Wormit, Henri van de Geest, Marie E. Bolger, Saleh Alseekh, Janina Maß, Christian Pfaff, Ulrich Schurr, Roger Chetelat, Florian Maumus, Jean-Marc Aury, Sergey Koren, Alisdair R. Fernie, Dani Zamir, Anthony M. Bolger, and Björn Usadel. De novo assembly of a new solanum pennellii accession using nanopore sequencing. *The Plant Cell*, 29(10):2336–2348, 2017.
14. Hans J Jansen, Michael Liem, Susanne A Jong-Raadsen, Sylvie Dufour, Finn-Arne Weltzien, William Swinkels, Alex Koelewijn, Arjan P Palstra, Bernd Pelster, Herman P Spaink, et al. Rapid de novo assembly of the european eel genome from nanopore sequencing reads. *Scientific Reports*, 7(1):7213, 2017.
15. Richard M Leggett and Matthew D Clark. A world of opportunities with nanopore sequencing. *Journal of Experimental Botany*, 68(20):5419–5429, 2017.
16. Kin Fai Au, Vittorio Sebastiano, Pegah Tootoonchi Afshar, Jens Durruthy Durruthy, Lawrence Lee, Brian A Williams, Harm van Bakel, Eric E Schadt, Renee A Reijo-Pera, Jason G Underwood, et al. Characterization of the human esc transcriptome by hybrid sequencing. *Proceedings of the National Academy of Sciences*, 110(50):E4821–E4830, 2013.
17. Donald Sharon, Hagen Tilgner, Fabian Grubert, and Michael Snyder. A single-molecule long-read survey of the human transcriptome. *Nature biotechnology*, 31(11):1009–1014, 2013.
18. Salah E Abdel-Ghany, Michael Hamilton, Jennifer L Jacobi, Peter Ngam, Nicholas Devitt, Faye Schilkey, Asa Ben-Hur, and Anireddy SN Reddy. A survey of the sorghum transcriptome using single-molecule long reads. *Nature communications*, 7, 2016.
19. Bo Wang, Elizabeth Tseng, Michael Regulski, Tyson A Clark, Ting Hon, Yiping Jiao, Zhenyuan Lu, Andrew Olson, Joshua C Stein, and Doreen Ware. Unveiling the complexity of the maize transcriptome by single-molecule long-read sequencing. *Nature communications*, 7:11708, 2016.
20. Nam V Hoang, Agnelo Furtado, Patrick J Mason, Annelie Marquardt, Lakshmi Kasirajan, Prathima P Thirugnanasambandam, Frederik C Botha, and Robert J Henry. A survey of the complex transcriptome from the highly polyploid sugarcane genome using full-length isoform sequencing and de novo assembly from short read sequencing. *BMC genomics*, 18(1):395, 2017.
21. Fritz J Sedlazeck, Hayan Lee, Charlotte A Darby, and Michael C Schatz. Piercing the dark matter: bioinformatics of long-range sequencing and mapping. *Nature Reviews Genetics*, 16(1):204, 2018.
22. Barmak Modrek and Christopher Lee. A genomic view of alternative splicing. *Nature genetics*, 30(1):13–19, 2002.
23. Neal S Holter, Madhusmita Mitra, Amos Maritan, Marek Cieplak, Jayanth R Banavar, and Nina V Fedoroff. Fundamental patterns underlying gene expression profiles: simplicity from complexity. *Proceedings of the National Academy of Sciences*, 97(15):8409–8414, 2000.
24. Graham EJ Rodwell, Rebecca Sonu, Jacob M Zahn, James Lund, Julie Wilhelmy, Lingli Wang, Wenzhong Xiao, Michael Mindrinos, Emily Crane, Eran Segal, et al. A transcriptional profile of aging in the human kidney. *PLoS biology*, 2(12):e427, 2004.
25. Eric E Schadt, Stephanie A Monks, Thomas A Drake, Aldons J Lusis, et al. Genetics of gene expression surveyed in maize, mouse and man. *Nature*, 422(6929):297, 2003.
26. Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.
27. Gregory D Schuler. Pieces of the puzzle: expressed sequence tags and the catalog of human genes. *Journal of Molecular Medicine*, 75(10):694–698, 1997.
28. John Bouck, Wei Yu, Richard Gibbs, and Kim Worley. Comparison of gene indexing databases. *Trends in Genetics*, 15(4):159–162, 1999.
29. John Quackenbush, Feng Liang, Ingeborg Holt, Geo Pertea, and Jonathan Upton. The tigr gene indices: reconstruction and representation of expressed gene sequences. *Nucleic Acids Research*, 28(1):141–145, 2000.
30. Weizhong Li and Adam Godzik. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics*, 22(13):1658–1659, 2006.
31. Ergude Bao, Tao Jiang, Isgouhi Kaloshian, and Thomas Girke. Seed: efficient clustering of next-generation sequences. *Bioinformatics*, 27(18):2502–2509, 2011.
32. Robert C Edgar. Search and clustering orders of magnitude faster than blast. *Bioinformatics*, 26(19):2460–2461, 2010.
33. Mohammadreza Ghodsi, Bo Liu, and Mihai Pop. Dnaclust: accurate and efficient clustering of phylogenetic marker genes. *BMC bioinformatics*, 12(1):271, 2011.
34. Alberto Paccanaro, James A. Casbon, and Mansoor A. S. Saqi. Spectral clustering of protein sequences. *Nucleic Acids Research*, 34(5):1571–1580, 2006.
35. Banu Dost, Chunlei Wu, Andrew Su, and Vineet Bafna. Tcclust: A fast method for clustering genome-scale expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(3):808–818, 2011.
36. John Burke, Dan Davison, and Winston Hide. d2-cluster: a validated method for clustering est and full-length cdna sequences. *Genome research*, 9(11):1135–1142, 1999.
37. Alan Christoffels, Antoine van Gelder, Gary Greyling, Robert Miller, Tania Hide, and Winston Hide. STACK: sequence tag alignment and consensus knowledgebase. *Nucleic Acids Research*, 29(1):234–238, 2001.
38. Michelle Girvan and Mark EJ Newman. Community structure in social and biological networks. *Proceedings of the national academy of sciences*, 99(12):7821–7826, 2002.
39. Juan Mei, Ji Zhao, Xiaojian Yang, and Weican Zhou. Remote protein homology detection using a modularity-based approach. In *Information Science and Technology (ICIST), 2011 International Conference on*, pages 1287–1291. IEEE, 2011.
40. Petr Novák, Pavel Neumann, and Jiří Macas. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC bioinformatics*, 11(1):378, 2010.
41. G Palla, AL Barabási, and T Vicsek. Quantifying social group evolution.

12 *Nucleic Acids Research*, 2018, Vol. X, No. X

- Nature*, 446(7136):664–667, 2007.
42. Pall F Jonsson, Tamara Cavanna, Daniel Zicha, and Paul A Bates. Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. *BMC bioinformatics*, 7(1):2, 2006.
 43. Balázs Adamcsek, Gergely Palla, Illés J Farkas, Imre Derényi, and Tamás Vicsek. Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics*, 22(8):1021–1023, 2006.
 44. Dominik Forster, Lucie Bittner, Slim Karkar, Micah Dunthorn, Sarah Romac, Stéphane Audic, Philippe Lopez, Thorsten Stoeck, and Eric Bapteste. Testing ecological theories with sequence similarity networks: marine ciliates exhibit similar geographic dispersal patterns as multicellular organisms. *BMC biology*, 13(1):16, 2015.
 45. Vincent D Blondel, Jean-Loup Guillaume, Renaud Lambiotte, and Etienne Lefebvre. Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment*, 2008(10):P10008, 2008.
 46. Benjamin H Good, Yves-Alexandre de Montjoye, and Aaron Clauset. Performance of modularity maximization in practical contexts. *Physical Review E*, 81(4):046106, 2010.
 47. Xiaoxian Liu, Wenbin Mei, Pamela S Soltis, Douglas E Soltis, and W Brad Barbazuk. Detecting alternatively spliced transcript isoforms from single-molecule long-read sequences without a reference genome. *Molecular Ecology Resources*, 2017.
 48. Sean P Gordon, Elizabeth Tseng, Asaf Salamov, Jiwei Zhang, Xiandong Meng, Zhiying Zhao, Dongwan Kang, Jason Underwood, Igor V Grigoriev, Melania Figueroa, et al. Widespread polycistronic transcripts in fungi revealed by single-molecule mrna sequencing. *PLoS one*, 10(7):e0132628, 2015.
 49. Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.
 50. Elias Dahlhaus, David S. Johnson, Christos H. Papadimitriou, Paul D. Seymour, and Mihalis Yannakakis. The complexity of multiterminal cuts. *SIAM Journal on Computing*, 23(4):864–894, 1994.
 51. John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.
 52. Nicholas J Loman and Aaron R Quinlan. Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, 30(23):3399–3401, 2014.
 53. W James Kent. Blat – the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.
 54. Richard Mott. Est.genome: a program to align spliced dna sequences to unspliced genomic dna. *Bioinformatics*, 13(4):477–478, 1997.
 55. Yi Wang, Henry C.M. Leung, S.M. Yiu, and Francis Y.L. Chin. Metacluster 5.0: a two-round binning approach for metagenomic data for low-abundance species in a noisy sample. *Bioinformatics*, 28(18):i356–i362, 2012.
 56. James K Senior. Partitions and their representative graphs. *American Journal of Mathematics*, 73(3):663–689, 1951.
 57. Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016.
 58. Justin Chu, Hamid Mohamadi, René L Warren, Chen Yang, and Inanc Birol. Innovations and challenges in detecting long read overlaps: an evaluation of the state-of-the-art. *Bioinformatics*, 33(8):1261–1270, 2016.
 59. Gene Myers. Efficient local alignment discovery amongst noisy long reads. In *International Workshop on Algorithms in Bioinformatics*, pages 52–67. Springer, 2014.
 60. Konstantin Berlin, Sergey Koren, Chen-Shan Chin, James P Drake, Jane M Landolin, and Adam M Phillippy. Assembling large genomes with single-molecule sequencing and locality-sensitive hashing. *Nature biotechnology*, 33(6):623–630, 2015.
 61. Ivan Sović, Mile Šikić, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjana Nagarajan. Fast and sensitive mapping of nanopore sequencing reads with graphmap. *Nature communications*, 7:11307, 2016.
 62. Mark J Chaisson and Glenn Tesler. Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC bioinformatics*, 13(1):238, 2012.
 63. Ze-Gang Wei, Shao-Wu Zhang, and Yi-Zhai Zhang. Dmclust, a density-based modularity method for accurate otu picking of 16s rna sequences. *Molecular informatics*, 36(12), 2017.
 64. Petr Novák, Pavel Neumann, and Jiří Macas. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC bioinformatics*, 11(1):378, 2010.
 65. Vincent Miele, Simon Penel, Vincent Daubin, Franck Picard, Daniel Kahn, and Laurent Duret. High-quality sequence clustering guided by network topology and multiple alignment likelihood. *Bioinformatics*, 28(8):1078–1085, 2012.
 66. Eduard Zorita, Pol Cusco, and Guillaume J Filion. Starcode: sequence clustering based on all-pairs search. *Bioinformatics*, 31(12):1913–1919, 2015.
 67. Zechen Chong, Jue Ruan, and Chung-I Wu. Rainbow: an integrated tool for efficient clustering and assembling rad-seq reads. *Bioinformatics*, 28(21):2732–2737, 2012.
 68. James T Robinson, Helga Thorvaldsdóttir, Wendy Winckler, Mitchell Guttman, Eric S Lander, Gad Getz, and Jill P Mesirov. Integrative genomics viewer. *Nature biotechnology*, 29(1):24, 2011.
 69. Thomas D Wu and Colin K Watanabe. Gmap: a genomic mapping and alignment program for mrna and est sequences. *Bioinformatics*, 21(9):1859–1875, 2005.