De novo Clustering of Gene Expressed Variants in Transcriptomic Long Reads Data Sets

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

De novo Clustering of Gene Expressed Variants in Transcriptomic Long Reads Data Sets

Camille Marchet[1], Lolita Lecompte[1], Corinne Da Silva[2], Corinne Cruaud[2], Jean-Marc Aury[2], Jacques Nicolas[1] and Pierre Peterlongo[1]

[1] IRISA - Inria Campus de Beaulieu, Rennes
[2] Commissariat à l'Energie Atomique (CEA), Institut de Biologie François-Jacob, Genoscope, Evry

Abstract

This work addresses the problem of grouping by genes long reads expressed in a whole transcriptome sequencing data set. Long read sequencing produces several thousands base-pair long sequences, although showing high error rate in comparison to short reads. Long reads can cover full-length RNA transcripts and thus are of high interest to complete references. However, the literature is lacking tools to cluster such data *de novo*, in particular for Oxford Nanopore Technologies reads. As a consequence, we propose a novel algorithm based on community detection and its implementation. Since solution is meant to be reference-free (*de novo*), it is especially well-tailored for non model species. We demonstrate it performs well on a real mouse data set. When a reference is available, we show that it stands as an alternative to mapping. In addition, we show that quick assessment of gene's expression is a straightforward use case of our solution.

1 Introduction

1.1 Motivation

Referred to as Third Generation Sequencing (TGS), long read sequencing technologies as Pacific Biosciences (PB) [16] and Oxford Nanopore Technologies (ONT) [44] have brought the opportunity to sequence full-length RNA molecules. In doing so, they relax the previous constraint of transcript reconstruction prior to study complete RNA transcripts [18]. The size of short reads constitutes indeed a major limitation in the process of whole transcript reconstitution, because they may not carry enough information to ensure the recovery of the full sequence. In addition, tools for *de novo* assembly of transcripts from short reads [18, 41] use heuristic approaches that do not guarantee an optimal solution. By avoiding these limitations and giving access to the whole trancript structure, long reads contribute to complement and improve transcriptome studies. This gain in length is at the cost of a computationally challenging error rate (up to 15%, although RNA reads generally show lower rates, around 9% or less [24, 22]). Yet, variant catalogs and expression levels start to be extracted from these new resources [6, 38, 3, 42, 19]. However, the vast majority of these works concern species with a reference. Methodological contributions that would enable to make the most of this promising data remain rare in particular for non model species [17, 26]. Moreover, while PB associates a dedicated protocol (Iso-Seq [16]) to numerous studies, ONT activity for RNA has just started in a few projects [8, 33]. In this work we propose to support the *de novo* analysis of RNA long read sequencing and show an application to ONT data. We introduce a clustering method that works at the gene level, without the help of a reference. This enables to retrieve the transcripts expressed by a gene, grouped in a cluster. Such clustering can be a component of a wider pipeline that aims at describing alternative variants or gene expression patterns. Such needs were already of concern in the past

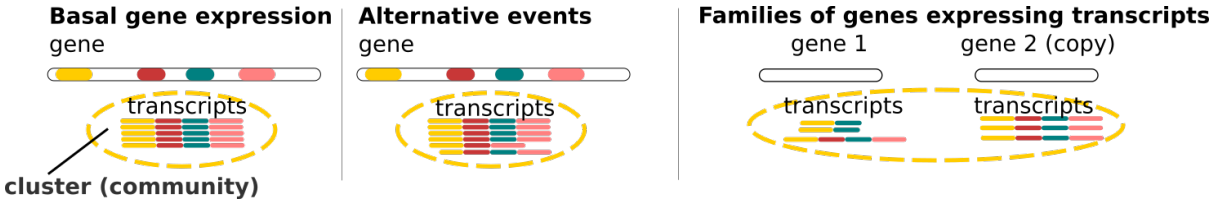

Figure 1: **Clustering on several example cases.** In eukaryotes, through transcription then splicing, exons (colored blocks) of genes are combined while introns are spliced to form RNA transcripts. Alternative events can produce variants with certain combination of exons, or part of exons. For basal gene expression as well as alternative events, all transcripts from a gene are expected to be found together in a cluster. In the complex case of families of genes, several copies of a paralog gene can express transcripts at the same time. If these transcripts share exon content and if the genes sequences have not diverged too much reads are in the same cluster. A similar scenario occurs for transcripts sharing genomic repeats.

long read literature [3, 26] and are even more acute when a mapping strategy cannot be taken into consideration.

1.2 Problem statement

Long reads from TGS give access to full-length RNA transcripts from many genes [6, 38]. Within a long reads data set, our goal is to identify for each expressed gene the associated subset of reads without mapping them on a reference. In order to group RNA transcripts from a given gene using these long and spurious reads, we propose a novel clustering approach. This problem can be computationally formalized as a community detection problem, where a community (also referred to as a cluster) is a population of reads coming from a same gene. Community detection is a broad field, rooted to the fundamental work in [15]. Among the popular methods, the Clique Percolation Method (CPM) has been applied for the detection of communities in a biological context [35, 4].
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 (called isoforms, see [28] for instance for a detailed depiction of alternative events) which differ in exon content, occur for a given gene. The issue is to automatically group alternative transcripts in a same cluster (Figure 1); 2/ long reads currently suffer from difficult indel errors at high rate [24, 22]; 3/ all genes are not expressed at the same level in the cell [20, 36, 37], which leads to an heterogeneous coverage in reads of the different genes, then to communities of different sizes including small ones, which is a hurdle for community detection [13].
Our proposal comes in two steps: first a graph of similarities between reads (see section 2.1) is computed with a third-party tool and then our clustering scheme proposal is applied to the graph. We perform clustering on the graph to retrieve each community, i.e. gene's expressed transcripts (detailed in section 2.2). In addition, we propose an implementation of the clustering algorithm in a tool dubbed CARNAC-LR (**C**lustering coefficient-based **A**cquisition of **RNA** **C**ommunities in **L**ong **R**eads) inserted into a pipeline. The input of this pipeline is a whole raw reads data set, 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.

Availability: CARNAC-LR is written in C++ and is available for Linux systems at `github.com/kamimrcht/CARNAC`.

2 Material and methods

2.1 First step: computing similarity between long reads

Given a raw set of long reads from a transcriptome sequencing, a graph of similarity is built prior to clustering, where a node represents a read and an edge a sequence similarity between two reads above a certain threshold. In such a graph, reads from a same gene are expected to be connected with one another because they are likely to share exons. For the pipeline we propose we chose the tool Minimap [25] for its efficiency and its very high level of precision on ONT and PB [10], among other recent methods that can compute similarity or overlaps between long reads despite their error rate [30, 7, 39, 9]. To generate the similarity graph for CARNAC-LR, Minimap version 0.2 was launched with parameters tuned to improve recall (`-Sw2 -L100 -t10`).
In the ideal case, a gene is easy to detect in the graph as all its reads are connected with one another. It is therefore a clique. However, the spurious nature of data imposes the use of heuristics to detect overlaps. This, in addition to the presence of genomic repeats leads to the expectation of a graph with both missing edges (connection missed during the search of overlapping reads) and spurious edges (wrong connections between unrelated reads).

2.2 Second step: clustering long reads

2.2.1 Clustering issue and sketch of the algorithm

Problem formalization In the following, we describe the clustering algorithm that is the main contribution of this paper. Our method makes no assumption on the number of expressed genes (i.e. communities), nor on the size distribution of such communities, and it needs no input parameters. In our case, we want the different communities to realize a partition of the graph, which means that there are no intersecting communities (no read belongs to several gene families); and that all nodes belong 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. As usual, the task of clustering is to maximize the intra-cluster similarity and minimize the inter-cluster similarity. We rely on the concept of clustering coefficient ($ClCo$) [31], to measure the similarity of a connected component. Although we have designed a parameter-free method, its foundation is a problem depending 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. Communities are disjoint sets. An optimal clustering in k communities is a minimal k-cut of the graph, that is, a set of k disjoint subsets of reads such that the set of edges between two different subsets (the cut-set) has minimal size.*

The rationale behind the search of a minimal cut in the graph is that the overlap detection procedure (section 2.1) has good specificity (it does not produce a lot of false positives). Thus, most of the edges in the initial graph have to be kept during the clustering. This problem is known to be NP-hard for $k \geq 3$ [11]. Another source of complexity is that we don't know in advance the number of communities, so we have to guess the value of k. One should thus compute the k-cut 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 thus looking for an approximation of the solution by using an efficient heuristic approach exploring a restricted space of k values. Finally, the second parameter, the cutoff θ, is not known either. The algorithm has thus to loop over all possible values, that is, all $ClCo$ values for a given connected component.

Implementation For space limitation reasons, in the following subsections we do not give a detailed description of the whole algorithm, but we give the main choices that make the approach feasible. Shortly, 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 condition of disjoint communities and look only for connected component having a clustering coefficient above a certain cutoff. Note that such connected components are not necessarily maximal. This corresponds to detecting well-connected nodes, called seed reads, using $ClCo$ and node degrees (detailed in section 2.2.3). They form the basis of communities with their neighborhood.
The main challenge is then to refine the boundaries of each community (section 2.2.4) in order to fulfill the partition condition. During this process, the value of k is progressively refined by possibly merging clusters whose combination produces a better community (greater $ClCo$ value). The other possibility of refinement is to assign nodes to a community and remove them from another. If x edges between the 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 that is associated to the minimal cut (i.e. number of edges removed when computing this partition).

2.2.2 Generation of partitions

In order to generate and compare different partitions for the graph, we define cutoffs that rule 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 C_lC_o are computed in the first place, and partitions are built for each non-zero $ClCo$ value as a cutoff. In the end, only one partition is retained, associated to the minimal cut (step (d) in Figure 2). However we have reduced the number of possible cutoff values for the sake of scalability (section 2.3). In the following, each step is described for a given cutoff value.

2.2.3 Selection of seed nodes and communities initiation

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 $n_i \in N$ with degree $deg(n_i) > 1$, we first compute the *local clustering coefficient*:

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

Nodes of degree 0 and 1 have a $ClCo$ of 1. This local coefficient represents the *cliqueness* of the $N_i \cup n_i$ set of nodes. The closer to 1, the more the set of nodes is inter-connected, which witnesses a group a reads that potentially come from the same gene. By contrast, if the coefficient is close to 0, the nodes are weakly connected and are less likely to come from the same gene. At a given cutoff value, the seed reads are primarily nodes which $ClCo$ is above or equal to this value. We add a statistical precaution to prevent star-like patterns (with a very low $ClCo$ with respect to the degree of the seed node) to initiate communities. We state the following auxiliary condition for seeds: $\forall n_i, ClCo_i \in]cutoff, \theta_2[\Rightarrow deg(n_i) \leq \theta_1$. θ_1 and θ_2 are the values such that 1% of the observed degrees are greater than θ_1 and 1% of the observed $ClCo$ are lower than θ_2 (1st and 99th percentiles). The selected seeds with their direct neighbors form the initial communities. At this point it is possible that two or more communities intersect.

2.2.4 Refinement of community boundaries

Community refinement aims at solving the conflicts of intersecting communities. Communities intersection happen 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 only one community. In fact two cases have to be distinguished. Either the edges between two communities are estimated spurious and these communities must be seen separated ((c') step in Figure 2), or the edges have sufficient support and the two communities have to be merged to obtain the full gene expression ((c) step in Figure 2). In order to decide between the two, we use again the *cliqueness* notion. This time we introduce an *aggregated clustering coefficient* of the union of two nodes n_i and n_j :

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

If the value of $ClCo_{ij}$ 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 to one or the other cluster according to which realizes the minimal cut. In case of ties for the cut, the algorithm uses a second criterion, the maximization of the sum over all communities of their clustering coefficient values.
The global result depends on the order in which pairs of clusters are compared. This order is carefully designed. First the communities associated to the two nodes of greatest degree (and secondly maximal $ClCo$) are chosen, the intersection is resolved and the first community is updated. Then it is compared to the third best community that intersected it if it exists, and so on until all intersections are solved. This way, we start the comparison by the most promising communities that combine reliability (they are well-connected subgraphs) with a high potential of resolution (they likely are the biggest communities, thus solve intersections for many nodes). On the contrary, communities associated to small subgraphs and relatively low $ClCo$ are only resolved afterwards.

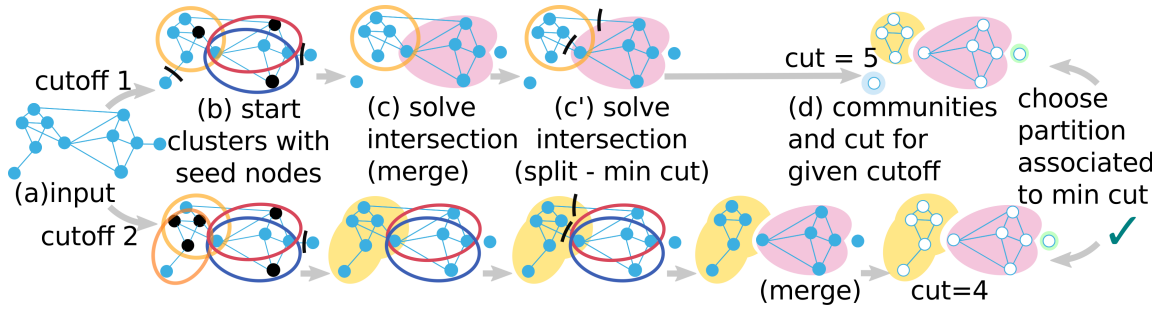

Figure 2: **Overview of the algorithm.** (a) All $ClCo$ and degrees are computed. Each $ClCo$ value will be a cutoff. Different cutoffs yield different seed nodes (black nodes). For a given cutoff, (b) seed nodes initiate clusters with their neighborhood (section 2.2.3). (c, c') Boundaries of each cluster are then refined. Intersection between clusters are solved either by (c) merging them or by (c') splitting (section 2.2.4) according to the cutoff. (d) The communities at different cutoffs evolve in different partitions. Finally, only the partition minimizing the cut is kept.

2.3 Implementation choices

We implemented the previously described algorithm in a tool called CARNAC-LR. Its input is the overlap file in .paf format provided by Minimap. The output of CARNAC-LR is a text

file with one line per cluster, each cluster containing the read indexes. Then each cluster can easily be converted to a FASTA file. Our algorithm has a quadratic component to compare sets in order to obtain clusters and in addition it explores the whole space of clustering coefficients to fix cutoffs. It results in a time complexity that can be theoretically cubic in the number of reads in the worst case. In practice we introduce key features to reduce efficiently the complexity of our approach, and our experiments rather showed that the running time is reasonable, clustering millions of reads in a few hours. Two key ideas to obtain this result have been to reduce the number of cutoffs by rounding the *ClCo* values and to disconnect the articulation points [21] to reduce the size of connected components in the graph. More details regarding the implementation are given in the Appendix.

2.4 Validation procedure

Since CARNAC-LR does not rely on a reference to compute the clusters, we used third-party mapping as a way of validation. In order to obtain a ground truth to validate the results of clustering, we used a real data set from the sequencing of the mouse brain transcriptome. Independent ground truth clusters were inferred using BLAT [23] for mapping on the reference genome and Est2genome [29].
To assess the results, we used recall and precision measures, which are standard measures to assess the relevance of biological sequence clustering [43]. For a given cluster, recall R expresses the fraction of relevant reads in this cluster cluster out of the expected read population of this cluster. Precision P shows the fraction of relevant reads in this cluster among the population of this cluster. Presented recall and precision are the mean values computed on all clusters. They are not absolute indicators, as they are computed comparatively to mapping results. The F-measure is a summary measure computed as the weighted harmonic mean between precision and recall. As a complementary measure, we assess the ability of the different algorithms to retrieve the correctness of the communities structure by adding the Jaccard index measure. The Jaccard index is between 0 and 1. The closer to 1, the more the *de novo* and mapping partitions agree on how the clusters are defined, consequently the more they depict the same groups of reads per gene. Details of the clustering by mapping procedure and metrics computation are given in Appendix.

3 Results

All experiments were run on a Linux distribution with 12 Intel Xeon 2.50GHz processors and with 200 GB RAM available. First we compare our approach to well established community detection methods and demonstrate its relevance on long read application. Then we validate our method's results by comparing them with independent clusters obtained by mapping on a real size data set. In these two parts, reads from the brain mouse transcriptome were used in order to access a ground truth via a reference. Then we show that our approach can offer an alternative to the classical mapping approach even when a reference is available.

3.1 Comparison to state of the art methods

We show results of state of the art algorithms and compare them to our tool's results. For scaling purpose, we chose to perform the benchmark on a subset of 10K reads (10,183 mouse reads within 207 reference clusters determined by mapping, section 2.4). Such sampling can accentuate the low expression effect in the subset. We have thus checked on a second 10K sample from chromosome 1 only to also account for highly expressed genes that results have the same trend than those presented (not shown). We compared CARNAC-LR+Minimap results to two classical methods for community detection (*modularity*-based [2] that was already used for

community detection in biological sequences [32], and CPM [1]). We also included a *transitive-closure* algorithm that partitions the graph in its connected components, such as used in EST-clustering [12]. Results are presented in Table 1. Our method has the best precision and the

	Recall (%)	Precision (%)	F-measure (%)	*size ratio*	Jaccard index
Connected comp.	75.74	5.614	13.62	1.7	0.0726
Modularity	60.70	71.16	65.51	1.9	0.0972
CPM5	**79.00**	69.35	73.86	**0.92**	0.353
CPM50	49.21	89.92	63.60	0.27	0.0757
CARNAC-LR+Minimap	65.0	**98.41**	**86.62**	3.8	**0.791**

Table 1: **Comparison with state of the art methods.** Size ratio divides the number of predicted clusters by the expected number of clusters and shows potential over/sub clustering effects. CPM5 (resp. CPM50) designates the CPM algorithm using $k = 5$(resp. $k = 50$). The F-measure shows that our approach is able to find a good tradeoff between recall and precision.

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. The *modularity*-based approach fails to find good clusters for this graph, with both low recall and precision. The *transitive-closure* approach suffers from low precision. CPM was tested with values for input parameter k ranging from 3 to 50 (no community found for greater values). Results are presented for k=5 and k=50 and summarize the behavior of this approach on our input graph. For low values of k, CPM outputs more clusters than for high values and shows the best recall. However its precision is globally low. For higher values of k, the results are strongly enhanced but represent only a small fraction of the input. As CARNAC-LR is conceived for general pipelines making the complete analysis of gene variants, it is important that is does not mix two unrelated genes in a same cluster. Thus our approach is more conservative than CPM, and it shows comparatively good results in any case, and furthermore needs no input parameter.

3.2 Validation on a real size data set

In this experiment we demonstrated we output high quality *de novo* clusters. We used the subset of reads that could be mapped to the mouse genome reference (501,787 reads) as a way of comparison to assess the biological relevance of our clusters. CARNAC-LR's results were

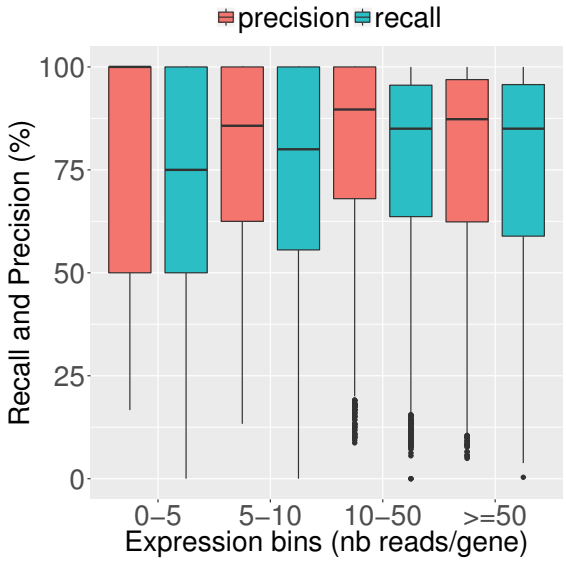

Figure 3: **Assessed mean recall and precision of CARNAC-LR+Minimap.** They were computed on mouse reads using clusters found by mapping on a reference as ground truth. Expression bins are computed from quartiles of expression predicted by mapping and represent the number of mapped reads by gene. Mean precision and recall over all clusters falling in theses bins were then calculated.

computed using 43GB RAM and 18 minutes using 10 threads.
The global recall for CARNAC-LR+Minimap pipeline was of 75.38% and the global precision

was 79.62%. Figure 3 presents the recalls obtained for binned expression levels and shows our approach's recall and precision remain consistent despite the heterogeneous coverage in reads. In order to present a visual example of the output, we used a genome browser to display reads grouped by our approach (Figure 4): in this example, our approach retrieved 93% of the predicted gene's reads while including no unrelated read in the cluster. Different alternative isoforms were gathered as expected (see Figure 1).

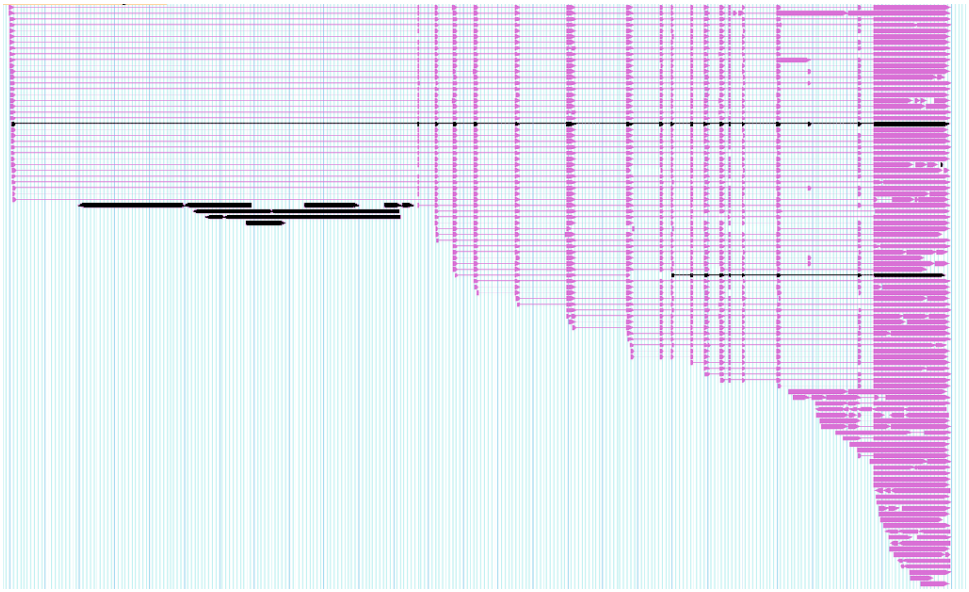

Figure 4: **Visual example of a CARNAC-LR's cluster.** We selected an example of 112 reads (purple) from a cluster output by CARNAC-LR. For validation purpose these reads were mapped with BLAST on mouse genome (using Genoscope's GGB [5, 40]). Reads are spliced-mapped, bold parts are the mapped sequences from the reads and thin parts represent the gaps between the different mapped parts of the reads. Ignoring the staircase effect observed in reads, it can be noticed 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. Eight reads (black) though present in the data are missed in this cluster. The group of six 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.

3.3 Complementarity of *de novo* and reference-based approaches

To demonstrate the interest of CARNAC-LR even if a reference is available, we ran it on the full mouse brain transcriptome data set (1,256,967 reads). We compared the intersection and difference of results of our approach and mapping. CARNAC-LR+Minimap pipeline took less than three hours (using 10 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. It resulted into 39,662 clusters. These clusters either contain (i) a mix of reads treated by our approach and/or processed by mapping, or (ii) reads treated by our approach exclusively. Each approach performed differently on these categories. Mapping complemented many clusters with small amounts of reads left aside by our approach. Conversely CARNAC-LR shows a better ability to group reads unprocessed by mapping into novel clusters (Figure 5).

For category (i) we computed recall and precision based on the read fraction of clusters that could be compared with mapping. They are quite similar compared to the values obtained in the previous section (75.26% and 79.30%). This demonstrates that CARNAC-LR efficiently used

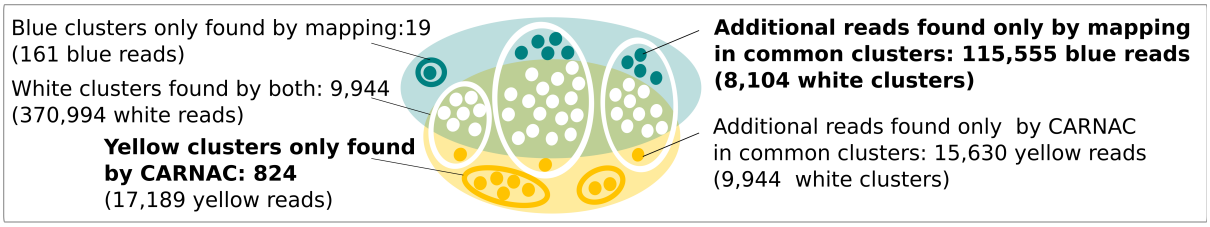

Figure 5: **Complementarity of CARNAC-LR and mapping approaches.** Only clusters of size ≥ 5 are represented. Mapping complemented common clusters with 13 reads on average per cluster in 90% of clusters. In contrast, CARNAC-LR's supply was low with a mean 1,3 read added to 100% of common clusters. On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

the supplementary connectivity information despite the addition of potentially noisy reads. CARNAC-LR output 824 novel clusters (17,189 reads) of category (ii) containing at least 5 reads. In order to evaluate the relevance of these novel clusters, we remapped reads *a posteriori*, separately for each cluster, on the reference genome using a sensible approach (GMAP [45] 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 each less than 10% of mapped reads. 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 contained exclusively reads mapped at a single locus. All of them could be assigned to an annotated gene. Thus even if a reference was available, our approach was able to retrieve *de novo* expressed variants of genes that were completely missed by the mapping. We also compared the genes expression levels computed by the two approaches, and shown they are highly correlated (Figure 6).

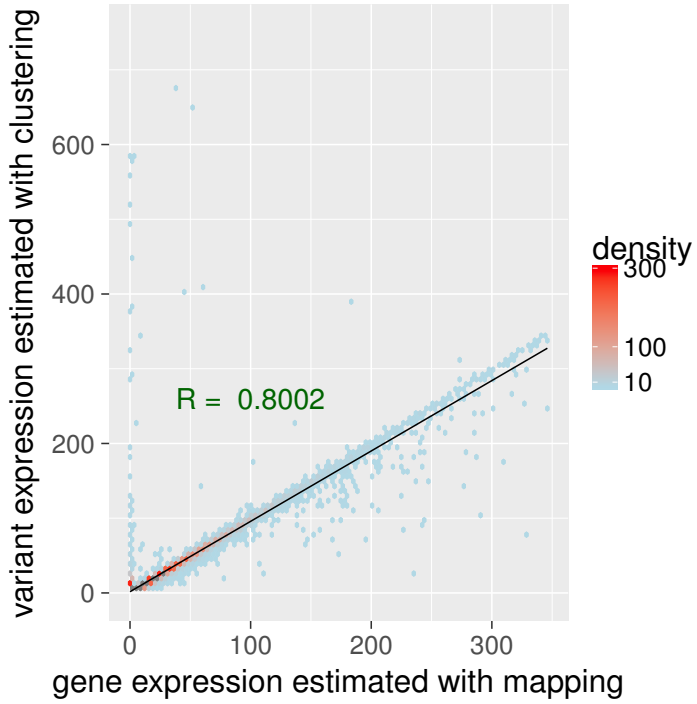

Figure 6: **Comparison and correlation of expressions levels.** Gene's expression can be inferred by counting the number of reads by gene. For each gene we counted the number of reads retrieved by mapping and we compared it to the number of reads reported by our pipeline and validated by mapping. We 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.

4 Discussion and perspectives

We propose a method designed for clustering long reads obtained from transcriptome sequencing in groups of expressed genes. New algorithmic challenge arises of the combination of a high error rate in data [24, 22], a high heterogeneity of coverage typical from expression data and an important volume of data. To this extent our question differs from EST clustering problems for instance. We demonstrated our method's relevance for this application, in comparison to literature approaches. To make our solution practical for users, we provide an implementation called CARNAC-LR that, combined to Minimap, scales and is able to process quickly real data instances, as demonstrated by the processing of the whole mouse brain transcriptome. We validated its results using mouse transcriptome ONT reads, showing we can compute high confidence clusters for many genes. We highlight 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.

The growth of accession records in databases recently burst for transcripts obtained with short reads [34] but a laborious curation is needed to filter out false positive reconstructed variants that do not have enough support. This illustrates the need for new methods to access novel transcripts with full-length reads. Long reads enable to skip the transcript reconstruction step that is necessary with short reads, though difficult in particular when it involves assembly. Therefore, long reads constitute an interesting novel way to obtain 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 genes' transcripts. Non model species require such *de novo* approaches, thus two bioinformatics tools dedicated to them have emerged so far [17, 26]. Both comprise a pipeline conceived to process Pacific Biosciences Isoseq [16] 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 TGS error profile and protocol. As a consequence it is the first method to perform *de novo* clustering on RNA reads from Oxford Nanopore. It takes reads early after their generation, without correction or filter. From the clusters, the expressed variants of each gene are obtained and related transcripts are identified, even when no reference is available.

We have also shown that our pipeline could be a complementary procedure when reads can be mapped to a reference. Thus it tends to retrieve some clusters missed by mapping and allows a more efficient use of data. We have demonstrated a straightforward use case of our pipeline as a good proxy to access the expression levels by gene. ONT sequences have been shown to qualify for transcript quantification in [33]. Moreover CARNAC-LR provides structured information that can be a sound input to other applications. For instance, a read correction step can be performed on each cluster instead of processing the whole data, in order to obtain high quality reference transcripts.

We argue that particular instances such as paralog genes constitute research themes on their own and require specific developments to untangle each paralog contribution to the observed variants expression. Our clustering already provides a first insight in this case, by allowing to access the whole population of transcripts a family of genes can produce.

As a consequence of the quick evolution of TGS, the sequencing field is frequently upgraded with new types of sequences. For instance, recent long read technology ONT RNA-direct [14] could unlock amplification biases issues in RNA sequencing and thus is promising for gene expression studies. But it shows higher error rates, at least comparatively to reads presented in this study, according to unpublished works. By proposing a generic tool that is tailored to these technologies, we wish to promote and encourage a broader use of long reads for transcriptome analysis.

Acknowledgments

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 the support and fruitful discussions, and Gaëtan Benoit and Antoine Limasset for their precious help on the implementation. Computations have been made possible thanks to the resources of the GenOuest infrastructures.

References

[1] http://igraph.org/.

[2] https://sites.google.com/site/cliqueperccomp/.

[3] 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.

[4] 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.

[5] Stephen F Altschul, Warren Gish, Webb Miller, Eugene W Myers, and David J Lipman. Basic local alignment search tool. *Journal of molecular biology*, 215(3):403–410, 1990.

[6] 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.

[7] 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.

[8] Mohan T Bolisetty, Gopinath Rajadinakaran, and Brenton R Graveley. Determining exon connectivity in complex mrnas by nanopore sequencing. *Genome biology*, 16(1):204, 2015.

[9] 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.

[10] 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.

[11] 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.

[12] Banu Dost, Chunlei Wu, Andrew Su, and Vineet Bafna. Tclust: A fast method for clustering genome-scale expression data. *IEEE/ACM Transactions on Computational Biology and Bioinformatics (TCBB)*, 8(3):808–818, 2011.

[13] Santo Fortunato. Community detection in graphs. *Physics reports*, 486(3):75–174, 2010.

[14] Daniel R Garalde, Elizabeth A Snell, Daniel Jachimowicz, Andrew J Heron, Mark Bruce, Joseph Lloyd, Anthony Warland, Nadia Pantic, Tigist Admassu, Jonah Ciccone, Sabrina Serra, Jemma Keenan, Samuel Martin, Luke McNeill, Jayne Wallace, Lakmal Jayasinghe, Chris Wright, Javier Blasco, Botond Sipos, Stephen Young, Sissel Juul, James Clarke, and Daniel J Turner. Highly parallel direct rna sequencing on an array of nanopores. *bioRxiv*, 2016.

[15] 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.

[16] Manuel L Gonzalez-Garay. Introduction to isoform sequencing using pacific biosciences technology (iso-seq). In *Transcriptomics and Gene Regulation*, pages 141–160. Springer, 2016.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] John Hopcroft and Robert Tarjan. Algorithm 447: efficient algorithms for graph manipulation. *Communications of the ACM*, 16(6):372–378, 1973.

[22] 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.

[23] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[24] 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.

[25] Heng Li. Minimap and miniasm: fast mapping and de novo assembly for noisy long sequences. *Bioinformatics*, 32(14):2103–2110, 2016.

[26] 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.

[27] Nicholas J Loman and Aaron R Quinlan. Poretools: a toolkit for analyzing nanopore sequence data. *Bioinformatics*, 30(23):3399–3401, 2014.

[28] Barmak Modrek and Christopher Lee. A genomic view of alternative splicing. *Nature genetics*, 30(1):13–19, 2002.

[29] Richard Mott. Est_genome: a program to align spliced dna sequences to unspliced genomic dna. *Bioinformatics*, 13(4):477–478, 1997.

[30] Gene Myers. Efficient local alignment discovery amongst noisy long reads. In *International Workshop on Algorithms in Bioinformatics*, pages 52–67. Springer, 2014.

[31] Mark EJ Newman. The structure and function of complex networks. *SIAM review*, 45(2):167–256, 2003.

[32] 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.

[33] 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.

[34] Nuala A O'Leary, Mathew W Wright, J Rodney Brister, Stacy Ciufo, 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.

[35] G Palla, AL Barabási, and T Vicsek. Quantifying social group evolution. *Nature*, 446(7136):664–667, 2007.

[36] 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.

[37] 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.

[38] 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.

[39] Ivan Sović, Mile Šikić, Andreas Wilm, Shannon Nicole Fenlon, Swaine Chen, and Niranjan Nagarajan. Fast and sensitive mapping of nanopore sequencing reads with graphmap. *Nature communications*, 7:11307, 2016.

[40] Lincoln D Stein, Christopher Mungall, ShengQiang Shu, Michael Caudy, Marco Mangone, Allen Day, Elizabeth Nickerson, Jason E Stajich, Todd W Harris, Adrian Arva, et al. The generic genome browser: a building block for a model organism system database. *Genome research*, 12(10):1599–1610, 2002.

[41] 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.

[42] Bo Wang, Elizabeth Tseng, Michael Regulski, Tyson A Clark, Ting Hon, Yinping 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, 2016.

[43] 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.

[44] 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.

[45] 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.

5 Appendix

5.1 Implementation choices

In large connected components, many clustering coefficients values are very close. Introducing a rounding factor in when computing the $ClCo$ results in a neat decrease of the number of different values observed, and thus restrains drastically the number of iterations necessary for the main loop. As a consequence, the optimization only computes an upper bound of the minimal cut, an acceptable compromise since it has no impact on the precision level (supplementary tests not shown).

The most costly phase relies on the treatment of the largest connected components. We chose to disconnect the *articulation points* of the graph to remove nodes that can 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 spotted as problematic as we do not expect a single read to be the only link between many others. Their detection can be done with a DFS in time complexity of $\mathcal{O}(\mathcal{N} + \mathcal{E})$ for the whole graph. Simulations shown that the best results occur when a first removal is performed on the whole graph and a second time within each connected component.

5.2 Reference material for validation

5.2.1 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 Oxford Nanopore Technologies (UK) protocol "1D cDNA by ligation (SQK-LSK108)". The data generated by MinION software (MinKNOWN, Metrichor) were stored and organized using a Hierarchical Data Format. FASTA reads were extracted from MinION HDF files using poretools [27].

5.2.2 Mapping to obtain reference clusters for validation

To obtain those reference for the validation of clustering, Nanopore reads from the mouse brain transcriptome were aligned to the masked mouse genome assembly (version GRCm38) using BLAT [23]. For each read, the best matches based on BLAT score (with an identity percent greater than 90%) were selected. Then, those matches were realigned on the unmasked version of the genome using Est2genome [29]. Reads that mapped onto the mitochondrial and ribosomal sequences were discarded. Moreover, one region on chromosome 1 that corresponds to an unprocessed pseudogene was excluded as it harbors a high number of Nanopore reads (>10k). Next, Nanopore reads were clustered according to their genomic positions: two reads were added in a given cluster if sharing at least 10nt 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, ...), some reads were mapped at multiple loci on the reference. When a given read maps on several loci, such loci are gathered in 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 elements are expected to be in the same cluster.

5.3 Validation metrics

Let $\mathcal{X}_0$ be the reference partition (here a 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 same cluster in $\mathcal{X}_0$ and $\mathcal{X}$. a_{10} (resp. a_{01}) is the number of pairs of nodes placed in the same community in the reference $\mathcal{X}_0$ (resp. $\mathcal{X}$) but in different clusters in $\mathcal{X}$

(resp. $\mathcal{X}_0$). Based on those, metrics show the adequation between the reference and computed partitions described, such as the Jaccard index:

$$J(\mathcal{X}_0, \mathcal{X}) = \frac{a_{11}}{a_{11} + a_{10} + a_{01}} \tag{3}$$

Let L be the number of predicted clusters by CARNAC-LR with $\{\mathcal{C}_1, \dots \mathcal{C}_i\}_{1 \leq i \leq L}$ the set of clusters. Let K be the number of expected clusters with the set $\{\mathcal{K}_1, \dots \mathcal{K}_j\}_{1 \leq j \leq K}$ of ground truth clusters. Let R_{ij} be the number of nodes from $\mathcal{C}_i$ that are in ground truth cluster $\mathcal{K}_j$. We compute a recall R and a precision P such as:

$$R = \frac{\sum_{j=1}^{K} max_i(R_{ij})}{\sum_{i=1}^{L}\sum_{j=1}^{K} R_{ij}} \quad P = \frac{\sum_{i=1}^{L} max_j(R_{ij})}{\sum_{i=1}^{L}\sum_{j=1}^{K} R_{ij}} \tag{4}$$