Thierry Lecroq, Eric Rivals and Irena Rusu (Eds.)

Montpellier, France
November 4th and 5th, 2014

Table of Contents

With support from:

MASTODONS Défi SePhHaDe

Preface

The pluridisciplinary workshop SeqBio 2014 was held at the Campus Saint-Priest in Montpellier, France on November 2014, 4th and 5th. It gathered computer science and bioinformatic communities working on textual analysis methods and biologists and geneticists interested in sequence bioinformatics. The programme includes talks selected on submissions and two invited talks by Laurent Bréhélin (LIRMM, CNRS & Université Montpellier) and Sven Rahmann (University Essen, Germany).

Thanks to the financial support of GdR (working groups) BIM (BioInformatique Moléculaire) and IM (Informatique Mathématique) of the CNRS, and of the project MASTODONS SePhHaDe, the participation was completely free.

The problems tackled during SeqBio spread from combinatorics on words and text algorithmics to their applications to bioinformatics analysis of biological sequences. This includes, without being restricted to, the following topics:

- text algorithmics;
- indexing data structures;
- combinatorics and statistics on words;
- high performance or parallel algorithmics;
- text mining;
- compression;
- alignment and similarity search;
- pattern or repeat matching, extraction and inference;
- analysys of high throughput sequencing data (genomic, RNA-seq, Chip-seq, ...);
- genome annotation, gene prediction;
- haplotypes and polymorphisms;
- comparative genomics;
- control signals.

This meeting comes after the following previous editions:

- Montpellier, November 2013;
- Marne-la-Vallée, November 2012;
- Lille, December 2011;
- Rennes, January 2011;
- Montpellier, January 2010;
- Rouen, September 2008;
- Marne-la-Vallée, September 2007;
- Orsay, November 2005;
- Lille, December 2004;
- Nantes, May 2004;
- Montpellier, November 2003;
- Nancy, January 2003;
- Rouen, June 2002;
- Montpellier, March 2002.

Programme Committee

- Guillaume Blin, LaBRI, Univ. Bordeaux I
- Jérémie Bourdon, LINA, Univ. Nantes
- Annie Chateau, LIRMM, CNRS Univ. Montpellier 2
- Hélène Chiapello, INRA Toulouse
- Julien Clément, GREYC, Univ. Caen
- Éric Coissac, LECA, Univ. Grenoble 1
- Thomas Faraut, INRA Toulouse
- Gabriele Fici, DMI, Univ. Palermo, Italy
- Gregory Kucherov, LIGM, Univ. Paris Est Marne-la-Vallée
- Thierry Lecroq, LITIS, Univ. Rouen (chair)
- Claire Lemaitre, INRIA Rennes
- Laurent Mouchard, LITIS, Univ. Rouen
- Macha Nikolski, LABRI, Univ. Bordeaux I
- Gwenael Richomme, LIRMM, Univ. Montpellier 3
- Eric Rivals, LIRMM, CNRS Univ. Montpellier 2 (chair)
- Irena Rusu, LINA, Univ. Nantes (chair)
- Hélène Touzet, LIFL, Univ. Lille I
- Raluca Uricaru, LABRI, Univ. Bordeaux I

Organizing Committee

The local organization has been realized by:

- the team "Methods and Algorithms for Bioinformatics (MAB)" of the LIRMM (Lab. of Computer Science, Robotics and Microelectronics of Montpellier)
- the "Institut de Biologie Computationnelle (IBC)"
- the French CNRS (Centre National de la Recherche Scientifique)
- the University of Montpellier 2

Members:

- Caroline Benoist
- Manuel Binet
- Bastien Cazaux
- Elisabeth Gréverie
- Sylvain Milanesi
- Damien Paulet
- Eric Rivals
- Amal Zine El Aabidine

Recherche de motifs de régulation à l'aide de données d'expression

Laurent Bréhélin

Méthodes et Algorithmes pour la Bioinformatique (MAB)
Laboratoire d'Informatique Robotique et Microélectronique de Montpellier (LIRMM)
161 rue Ada
34095 Montpellier Cedex 5 - France

L'identification de motifs de régulation constitue l'une des plus anciennes problématiques de recherche en bioinformatique, et de nombreuses méthodes ont été proposées. Les méthodes les plus classiques utilisent un modèle de background probabiliste et cherchent des motifs sur-représentés au sens de ce modèle, dans un ensemble prédéterminé de séquences. Plus récemment, de nouvelles approches utilisant des mesures d'expression de gènes ont été proposées. Ces algorithmes cherchent des motifs dont la présence est corrélée à l'expression des gènes. Un des avantages est que l'on s'affranchit alors de la nécessité d'un modèle de background. Après un tour d'horizon général des approches proposées pour l'identification de motifs, je présenterai la méthode RED2 qui guide sa recherche en utilisant la notion de densités de motifs dans l'espace d'expression.

http://www.atgc-montpellier.fr/RED2/

References

[1] M. Lajoie, O. Gascuel, V. Lefort, and L. Bréhélin. Computational discovery of regulatory elements in a continuous expression space. *Genome Biol.*, 13(11) :R109, Nov 27 2012.

Reproducible High-Throughput Sequencing Data Analysis

Sven Rahmann

Genominformatik, Inst. für Humangenetik
Medizinische Fakultät
Universität Duisburg-Essen
Hufelandstr. 55
45122 Essen
Germany

and

Bioinformatik
Informatik XI
TU Dortmund
44221 Dortmund
Germany

The ever-increasing adoption of high-throughput sequencing technologies in many distinct medical, biological and bio-technological applications has lead to the development of many different and individualised data analysis processes ("pipelines") in different labs. Initially, developing a new process requires trial and error, including the evaluation of different tools and techniques and the optimisation of parameters, and a typical pipeline changes more rapidly than it can be described in a typical "Material and Methods" section of a publication. When a project is finished and a publication is accepted, a pipeline is often re-used in another project and developed further. In the interest of reproducible science, it is important that, ideally, there is a completely and formally specified process, from raw data such as FASTQ files, to finally published data, such as tables and figures. This process can ideally be re-executed (with the appropriate software environment) and reproduce the same results from the same raw data at any later time point. Several tools have been developed to create, describe, store and execute complex data analysis workflows, for example the popular Galaxy platform directed mainly at biologists. We found that bioinformaticians often require more powerful tools and developed a domain specific programming language called "Snakemake" that is a hybrid between Python and Make. It extends the Python language by data transformation rules from input files to output files. Snakemake supports both multi-core and cluster environments. It allows to specify resource constraints in the environment that are respected by the scheduler. In the talk I will present the motivation and design decisions behind Snakemake, its main features, several NGS workflow examples, and some details of the Python implementation. Snakemake has mainly been developed by Johannes Köster and is being used in many different labs around the world.

References

[1] J. Köster and S. Rahmann. Building and documenting workflows with Python-based Snakemake. In *German Conference on Bioinformatics (GCB)*, volume 26 of *OASICS*, pages 49–56, 2012.

[2] J. Köster and S. Rahmann. Snakemake: a scalable bioinformatics workflow engine. *Bioinformatics*, 28(19):2520–2522, 2012.

Transposable Elements Investigation Tool Applied to Prokaryotic Genomes

Huda Al-Nayyef[1,2], Christophe Guyeux[1] and Jacques M. Bahi[1]

[1] FEMTO-ST Institute, UMR 6174 CNRS, DISC Computer Science Department, Université de Franche-Comté, 16, Rue de Gray, 25000 Besançon, France
[2] Computer Science Department, University of Mustansiriyah, Iraq

huda.al-nayyef@univ-fcomte.fr
christophe.guyeux@univ-fcomte.fr
jacques.bahi@univ-fcomte.fr

Transposable elements (TEs), which are DNA segments that have the ability to insert or copy themselves into new chromosomal locations. In bacterial reign, only cut-and-paste mechanism of transposition can be found, These types of mobile genetic elements (MGEs) involved in such a move being the insertion sequences (ISs). Two main factors have big effects on IS discovery, namely: genes annotation and functionality prediction. The authors have designed a novel pipeline for ISs detection, which embeds the most recently tools, namely OASIS (Optimized Annotation System for Insertion Sequence) [1] and ISFinder database (an up-to-date repository of known ISs) [2].

OASIS identifies ISs in each genome by finding conserved regions surrounding already-annotated transposase genes. It takes as input NCBI genbank file with descriptive functionality. The main problem found in it solved in our pipeline by designing two modules based on OASIS which is called NOASIS and DOASIS. Our pipeline could be represented in the following steps:

Step 1: ORF identification. Our pipeline is currently compatible with any type of annotation tools, having either functionality capability or not, but for comparison we focus on (*BASys*, *Prokka*, and *Prodigal*).

Step 2: IS Prediction. Using either NOASIS or DOASIS for predicting IS elements. Notice that NOASIS requires information about gene functionality by depending not only on NCBI, while DOASIS works with or without gene functionality by modifying genbank files using the suggested methods:

1. **All-Tpase**: we consider that all the genes may potentially be a transposase. So all product fields are set to "transposase".
2. **Zigzag Odd**: we suggest that genes in odd positions are putative transposases and we update the genbank file adequately. Oddly, this new path will produce new candidates which are not detected during All-Tpase.
3. **Zigzag Even**: similar to Zigzag Odd, but on even positions.

Step 3: IS Validation. This step is realized by launching BLASTN on each predicted IS sequence with ISFinder. The e-value of the first hit is then checked: if it is 0.0, then the ORF within this sequence is a Real IS known by ISFinder. It will be considered as Partial IS if its e-value is lower than 10^{-10}. Both IS names of family and group are returned too.

A complete IS detection and classification pipeline has then been proposed and tested on a set of 23 complete genomes of *Pseudomonas aeruginosa*. This pipeline can also be used as an investigator of annotation tools performance, which has led us to conclude that Prodigal is the suitable annotation tool for IS prediction of prokaryotic. A deeper study regarding IS elements in *P.aeruginosa* has then been conducted, leading to the conclusion that close genomes inside this species have also a close numbers of IS families and groups.

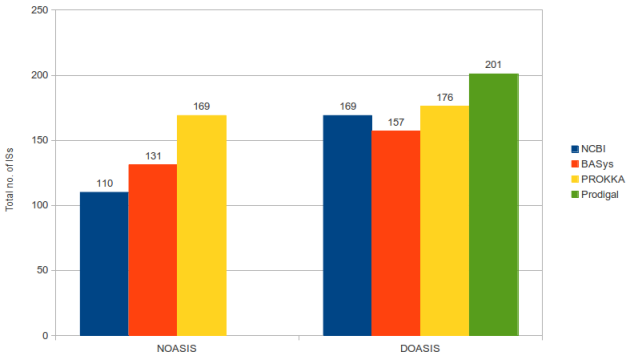

Figure 1: Comparison between NOASIS and DOASIS

References

[1] D. G. Robinson, M.-C. Lee, and C. J. Marx. Oasis: an automated program for global investigation of bacterial and archaeal insertion sequences. *Nucleic Acids Research*, 40(22):e174–e174, 2012.

[2] P. Siguier, J. Pérochon, L. Lestrade, J. Mahillon, and M. Chandler. ISfinder: the reference centre for bacterial insertion sequences. *Nucleic Acids Research*, 34(suppl 1):D32–D36, 2006.

Recherche de motifs dans des séquences similaires

Nadia Ben Nsira[1,2], Thierry Lecroq[2] and Mourad Elloumi[1],

[1] LaTICE, Université de Tunis El Manar, Tunisie
[2] LITIS EA 4108, NormaStic CNRS FR 3638, IRIB, Université de Rouen, Normandie Université, France

Avec l'arrivée des nouvelles technologies de séquençage à haut débit (*Next Generation Sequencing* en anglais), des quantités gigantesques de séquences génomiques d'individus de même espèce sont maintenant disponibles.

Ces séquences ne diffèrent que par de très petites quantités de variations et présentent un niveau de similarité très élevé. Elles peuvent donc être représentées par une séquence de référence et un ensemble de variations sur les autres séquences par rapport à cette séquence de référence. Il existe donc un fort besoin d'algorithmes efficaces pour indexer et effectuer des recherches rapides dans ces ensembles spécifiques de séquences dites fortement similaires. Dans certains cas on peut être amené à effectuer une recherche dans ces données sans pouvoir les indexer (en cas de manque d'espace par exemple). Ainsi il doit être possible d'effectuer une recherche incrémentale dans l'ensemble de ces séquences.

Nous avons conçu deux algorithmes de recherche exacte incrémentale d'un motif court dans un ensemble de séquences fortement similaires. Les solutions proposées sont restreintes et supposent que les séquences ne contiennent que des variations de type substitution. Les deux algorithmes utilisent une fenêtre glissante qui parcourt simultanément l'ensemble de séquences du début jusqu'à la fin. La fenêtre a la même longueur que le motif. Une tentative consiste à comparer le contenu de la fenêtre avec le motif.

Le premier algorithme suit une analyse étroite de l'algorithme de Morris-Pratt qui compare le contenu de la fenêtre de gauche à droite. Le décalage de la fenêtre après chaque tentative est calculé en considérant des bords du motif avec une distance de Hamming à 0 (décalage classique de Morris-Pratt) et des bords avec une distance de Hamming à 1 (de manière à tenir compte des variations). La difficulté provient du fait qu'un bord à distance 1 d'un bord à distance 1 peut être un bord à distance 2 et non pas 1 [2].

Le deuxième algorithme adapte une variante de l'algorithme de Boyer-Moore qui compare le contenu de la fenêtre de droite à gauche. Le décalage est calculé en fonction de ré-occurrences de suffixes du motif avec une distance de Hamming à 0 ou avec une distance de Hamming à 1 [1].

En pratique, nous avons comparé nos solutions avec des algorithmes de recherches exactes efficaces d'un motif dans une seule séquence. Ces résultats montrent que nos solutions sont efficaces.

References

[1] N. Ben Nsira, T. Lecroq, and M. Elloumi. A fast pattern matching algorithm for highly similar sequences. In *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBM 2014)*, 2014.

[2] N. Ben Nsira, T. Lecroq, and M. Elloumi. On-line string matching in highly similar DNA sequences. In C. S. Iliopoulos and A. Langiu, editors, *Proceedings of the 2nd International Conference on Algorithms for Big Data (ICABD 2014)*, volume 1146 of *CEUR Workshop Proceedings*, 2014.

Dynamic mappers of NGS reads

Karel Břinda, Valentina Boeva and Gregory Kucherov

Read mapping continuously belongs to hot topics in bioinformatics and the number of mappers released every year is enormous. Differences between them are often only of technical nature and there is, up to a few exceptions, a lack of conceptional algorithmical improvements. Eventually, only few well-developed and debugged mappers are usually used in practice (e.g., BWA with its three separate mapping algorithms [4, 5, 3], GEM [6], NovoAlign).

In spite of a great attention devoted to this topic, *dynamic mapping*, i.e., mapping with updating reference in accordance to already mapped reads, has not been well-studied yet. The only texts on this topic are [2, 7] and still, to the best of our knowledge, there has not been developed any real mapper supporting dynamic updates until today in spite of the fact that they can provide a notable improvement in accuracy and sensitivity. This situation is a consequence of the following facts.

Firstly, dynamic mappers bring several technical and algorithmic issues, e.g.:

1. Underlying data structures (FM indexes, hash-tables, etc.) must be dynamic. Updating them is a non-trivial task (for dynamic FM index, see [8]) and implementations cannot be as memory-efficient and cache-optimized as static versions of these data structures.
2. Expensive statistics about already mapped reads must be kept in memory during whole mapping.
3. Addressing in the reference must be somehow generalized because after every insertion and every deletion, the coordinates of mapped reads change. Also, CIGAR strings are evolving during dynamic mapping.
4. One must deal with remapping and unmapping already mapped reads.

Inevitably, these four points imply that the memory consumption will extensively increase while the performance will decrease, in comparison to static mappers.

Secondly, dynamic mapping might be undoubtedly contributive for a limited class of applications when speed can be sacrificed to sensitivity and selectivity (low number of reads, far reference, many hot spot regions, etc.).

Besides the above mentioned approaches, static mapping and dynamic mapping, there exists an intermediate approach, a so-called *iterative referencing* [1], which is based on iterative repetition of "static mapping of all reads" and "variant calling" until number of updates decreases below a given threshold.

In order to obtain data proving the superiority of dynamic mappers over static mappers, we developed a pipeline simulating dynamic updates. Selected state-of-the-art mappers are used to map reads in small batches followed by computation of statistics and update of the reference sequence. Dynamic mappers should be capable to perform updates after mapping of each new read, and with batches small enough, we can reach a good approximation of their behavior even with static mappers. The main difference from the iterative referencing approach (with updates after the mapping of all reads and computationally intensive algorithms for calling variants) resides in simplified algorithms for calling variants and small size of batches. Improvements of alignments are measured using our newly developed evaluation program LAVEnder for the DWGSim read simulator.

In our presentation, we will show first results obtained with this pipeline and state several conclusions about the usefulness of the dynamic index approach.

References

[1] A. Ghanayim and D. Geiger. Iterative referencing for improving the interpretation of DNA sequence data. Technical report, Technion, Israel, 2013. http://www.cs.technion.ac.il/users/wwwb/cgi-bin/tr-get.cgi/2013/CS/CS-2013-05.pdf.

[2] C. S. Iliopoulos, D. Kourie, L. Mouchard, T. K. Musombuka, S. P. Pissis, and C. de Ridder. An algorithm for mapping short reads to a dynamically changing genomic sequence. *Journal of Discrete Algorithms*, 10:15–22, 2012.

[3] H. Li. Aligning sequence reads, clone sequences and assembly contigs with BWA-MEM. Technical report, 2013. Preprint, arXiv:1303.3997.

[4] H. Li and R. Durbin. Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics*, 25(14):1754–1760, 2009.

[5] H. Li and R. Durbin. Fast and accurate long-read alignment with Burrows-Wheeler transform. *Bioinformatics*, 26(5):589–595, 2010.

[6] S. Marco-Sola, M. Sammeth, R. Guigó, and P. Ribeca. The GEM mapper: fast, accurate and versatile alignment by filtration. *Nature Methods*, 9(12):1185–1188, 2012.

[7] J. Pritt. Efficiently improving the reference genome for DNA read alignment, 2013. Retrieved January 11, 2014 from http://jacobpritt.com/projects/updating_reference_genome.pdf.

[8] M. Salson, T. Lecroq, M. Léonard, and L. Mouchard. Dynamic extended suffix arrays. *Journal of Discrete Algorithms*, 8(2):241–257, 2012.

From Indexing Data Structures to de Bruijn Graphs [1]

Bastien Cazaux[1], Thierry Lecroq[2] and Eric Rivals[1]

[1] L.I.R.M.M. & Institut Biologie Computationnelle, Université de Montpellier II, CNRS U.M.R. 5506, Montpellier, France
[2] LITIS EA 4108, NormaStic CNRS FR 3638, IRIB, Université de Rouen, Normandie Université, France

cazaux@lirmm.fr
thierry.lecroq@univ-rouen.fr
rivals@lirmm.fr

New technologies have tremendously increased sequencing throughput compared to traditional techniques, thereby complicating DNA assembly. Hence, assembly programs resort to de Bruijn graphs (dBG) of k-mers of short reads to compute a set of long contigs, each being a putative segment of the sequenced molecule. Other types of DNA sequence analysis, as well as preprocessing of the reads for assembly, use classical data structures to index all substrings of the reads. It is thus interesting to exhibit algorithms that directly build a dBG of order k from a pre-existing index, and especially a contracted version of the dBG, where non branching paths are condensed into single nodes. Here, we formalise the relationship between substringsx trees/arrays and dBGs, and exhibit linear time algorithms for constructing the full or contracted dBGs. Finally, we provide hints explaining why this bridge between indexes and dBGs enables to dynamically update the order k of the graph.

References

[1] B. Cazaux, T. Lecroq, and E. Rivals. From indexing data structures to de Bruijn graphs. In *Proceedings of the 25th Annual Symposium on Combinatorial Pattern Matching (CPM 2014)*, pages 89–99, Moscow, Russia, 2014.

1. This work is supported by ANR Colib'read (ANR-12-BS02-0008) and Défi MASTODONS SePhHaDe from CNRS

Functional annotation of polymorphisms identified by NGS approaches in *P.falciparum*

Ankit Dwivedi[1], Emmanuel Cornillot[1], Christelle Reynes[2], Nimol Khim[3], Didier Menard[3], Roger Frutos[4], Eric Rivals[5], Maxime Hebrard[5] and Sylvain Milanesi[6],

[1] CPBS, IBC, UM1/UM2
[2] Laboratoire de Physique Industrielle et traitement de l'Information - UM1
[3] Institut Pasteur du Cambodge (IPC) - Phnom Penh - Cambodge
[4] CPBS, UM2
[5] IBC, LIRMM, UM1/UM2
[6] IBC, UM1/UM2

ankit.dwivedi@cpbs.cnrs.fr
emmanuel.cornillot@univ-montp1.fr
christelle.reynes@univ-montp1.fr
knimol@pasteur-kh.org
dmenard@pasteur-kh.org
frutossmt@gmail.com
rivals@lirmm.fr
Maxime.Hebrard@lirmm.fr
sylvain.milanesi@lirmm.fr

Malaria is one of the most widespread parasitic infections in the world. The ongoing WHO Malaria elimination program has resulted in decreased cases. These encouraging results are the issue of public health policies and development of artemisinin based therapies. These approaches are now threatened by the emergence of artemisinin resistant parasites. The development of resistant assay (RSA test, [3]) and genetic markers (Kelch gene, [1]) enable us to better evaluate the prevalence of artemisinin resistant isolates in Cambodia. *Plasmodium falciparum* is one the major causative agent of malaria in Cambodia. The focus of this project is to identify drug resistant genes in the malaria parasite *P.falciparum*. It aims to identify these genes using genome polymorphisms. We use a large datasets to analyse the distribution of parasite population over the country. The set is based on NGS genome sequences available in ENA database. We recover 167 genomes originating from four different localities in Cambodia. We describe a reliable SNP variant calling pipeline from around 200 NGS genome sequences based on quantitative parameters provided in the VCF files. SNPs were extracted and filtered after comparison with 3D7 reference genome. Different tools like R, Perl and Artemis were used for the analysis. The major steps involved in the pipeline are, a) The quantitative parameters provided in the variant calling format (VCF) files were analysed to define a threshold to select good quality SNPs, b) SNPs were filtered based on MQ which represents the mapping quality and DA ($\sum$ ALT / $\sum$ DP4) which represents the percentage of high quality ALT reads, c) SNPs with low frequency and SNPs with uncertain ALT bases were not considered, d) Mapping was done to different genome version and annotation information was provided for each SNP. These SNPs were then characterized into three categories: non coding region, synonymous and non-synonymous. We differentiate SNPs associated to the coding core and to the sub-telomeric regions of the genome. The large number of samples indeed improves SNP extraction. The dataset obtained with the variant calling pipeline was compared to the other published datasets and validated with the presence of marker SNPs. Recent studies provide evidence that sub-populations of parasites are present in Cambodia [2]. We probe this hypothesis using SNP dataset extracted with pipeline as described above. Different set of SNPs were tested to evaluate the robustness of the sub-population including mutations in the Kelch gene that are being associated to the resistance to artemisinin derivatives. This genetic marker is found in large numbers in the region of Pailin, where drug resistance was first described. We provide genetic evidence for acquisition and transmission of artemisinin

resistance in Cambodian parasite sub-populations. These results question the origin and the persistence of these sub-populations. Fragmentation of the *P. falciparum* is important information that must be taken into account for further statistical analysis of SNP distribution. Different approaches using bioinformatics resources and SNP data will be established to identify features providing functional annotation for proteins, pathways, isolates and sub-populations. These steps are essential to identify parasite sub-populations that could be more susceptible to acquire and to transmit drug resistance in Cambodia.

References

[1] F. Ariey *et al.* *Nature*, 505(7481):50–55, 2014.

[2] O. Miotto *et al.* *Nature Genetics*, 45(6):648–655, 2013.

[3] B. Witkowski *et al.* *Lancet Infect Dis.*, 13(12):1043–1049, 2013.

Post-Alignment Visualization and Exploration of High-Throughput Sequencing Experiments

Bernd Jagla[1], Manuel Holtgrewe[2] and Jean-Yves Coppeé[1]

[1] Plate-forme 2 Transcriptome et Epigénome, Département Génomes & Génétique, Institut Pasteur, Paris, France
[2] Algorithmische Bioinformatik, Institut für Informatik, Freie Universität Berlin, Berlin, Germany

Abstract: The visual exploration of high-throughput sequencing experiments is largely limited to the use of genome-browsers. These browsers present the sequencing profiles in the context of the reference genome. We have decoupled the expression profiles from this constrain and present tools that allow for the visual inspection and exploration of "regions of interest" outside of the context of the reference sequence. In the context of quality control we can visually inspect RNA-seq experiments within seconds even for larger genomes. In addition, the visual exploration of NGS data proved to being useful in small-RNA-seq experiments and others. We show how to apply this technology to quality control, miRNA analysis, and transcription start site annotation/analyses. C++ version, R, and Galaxy integrations are available through http://www.seqan.de/projects/ngs-roi/ and links therein. Development version can be found at git-hub: https://github.com/baj12/seqan_apps/. And the KNIME integration can be found here http://tech.knime.org/community/next-generationsequencing and through the official update-site of KNIME.

1 Introduction

High throughput sequencing (HTS) platforms such as those from Illumina produce hundreds of millions of reads. The quality of raw HTS data can be evaluated using basic statistics such as nucleotide or base quality distributions (FASTQC, http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc). In many experiments, the HTS data is then aligned to a reference. For biological interpretation, data is then summarized as counts per region (with a region being a gene, exon, transcript, etc) [4]. In addition to these tools specialized in basic statistics, fitgcp [9] uses genome coverage profiles for metagenomic analyses. And [10] have shown that coverage profiles can be used for comparison in genomic regions. RSeQC [15], for example, use coverage uniformity over the gene body as quality control measurement for transcript expression. All these publications show the importance coverage profiles. The visual exploration though is largely limited to genome browsers such as JBrowse [14] or IGV [12]. The tools presented here will aid in visualizing and utilizing coverage profiles.

2 Region of Interest Analysis

Our analysis tools called, Regions Of Interest (ROI) Analysis, decouple the mapping results from a strict linear ordering on the genome. Instead, regions on the genome are identified and considered as independent objects for further analysis. In a first step, intervals of overlapping read alignments are considered as a region of interest. The region is annotated with its locus, length, and the coverage over that region. The regions can later be modified and redefined (e.g. by splicing intronic ROIs together to their transcript or mapping to annotations from GFF formatted files), sorted, and filtered. We also provide software for visualization. We define regions resulting from these processes as regions of interest (ROIs).
The concepts described here allow for the visualization and analysis of coverage profiles in regions of interest. The ROIs are usually vectors of different lengths, thus they cannot be compared directly. To alleviate this limitation we introduce metrics. Some of the more obvious metrics describe the length of an expression profile or the coverage (mean, min, max, etc.). The distribution of these simple metrics can be used in the context

of quality control to identify outliers. Figure 1a shows an example distribution of the properties "length" and "normalized average coverage" (nac(region) = average(coverage(region))/max(coverage(region)), plotted in log space for all ROIs.

3 Application

Profiles corresponding to high coverage may derive from either systematic errors or biologically relevant processes. NGS experiments suffer from various biases due to the sequencing device [6, 13] or experimental conditions (e.g. library construction, adapter sequences). Some of these artifacts can be identified and removed by using existing software [5] or applying normalization procedures [4]. Yet, other errors can be introduced during alignment. For example, [3] report biases towards the reference allele. A second type of error comes from the fact that a sequence read can align to more than one region in the reference with similar alignment scores. This can lead to sharp peaks where a short region of a highly expressed gene occurs somewhere else in the reference. A third type of error can produce a peak in a profile on either end of an otherwise poorly expressed gene in a non-directional RNA-Seq experiment. This could result from two overlapping genes that are located on opposite strands with very different expression levels. The tools presented here help identifying such regions by grouping them together and allowing for a visual and algorithmic analysis.

When visualized using a genome browser, the coverage is only available in the genomic context. When a certain feature or problem is identified in the coverage profiles of the human genome, it is practically impossible to search the 3 billion sequence positions to identify similar profiles. Given a metric that describes features they, too, can be analyzed with the given tools.

The concept of ROIs can also be used in circumstances where biological processes produce patterns. One such example is the biogenesis of miRNAs [2]. miRNAs are translocated as pre-miRNA from the nucleus into the cytosol where they are further processed into a miRNA/miRNA* duplex and then separated into miRNA and miRNA*, of which the miRNA* will be degraded. This process can be seen in the profile of an RNA-Seq experiment. Figure 1b shows such a profile. In case the profile comes from a non-annotated region (this can be easily achieved by applying the appropriate annotation to the ROIs) this would indicate a potential novel miRNA. A link is associated with this plot that opens a defined genome browser at that particular location of the genome (Fig. 1c). Given the output of a miRNA prediction tool the potential miRNAs could sorted by relevance and visualized with the associated expression pattern.

Other alignment patterns may rise from special experimental conditions like the ones used in transcription start site mapping experiments or CHiP-Seq experiments [11]. The difficulty then is to define the correct metrics that capture the essence of the question. The visual representation in form of a HTML web-page allows evaluating such metrics in an efficient way. Once these metrics are established, unknown regions can be easily identified using the tools at hand. This also opens the door for applying supervised and unsupervised machine learning technologies for the discovery of new features. We are currently investigating such possibilities for quality control purposes and new biological interpretation of the data.

In principle, ROIs are not limited to coverage data. They could also hold information like GC content, or any other numerical value that can be assigned to a sequence position within the given reference genome.

Tools provided

bam2roi: create ROI file from sam/bam formatted alignment file

roi_feature_projection: map ROIs onto features from a BED/GFF formatted annotation file.

roi_plot_thumbnails.py: create a HTLM overview page showing a PNG image with a collection of ROIs.

roi_table.py: create a HTML file with a table containing a picture of an ROI and the associated metrics.

roi_report.py: creates a HTML file with summary graphs of the metrics and their summary statistics.

Libraries for importing/exporting ROIs using R, Python, Galaxy, KNIME.

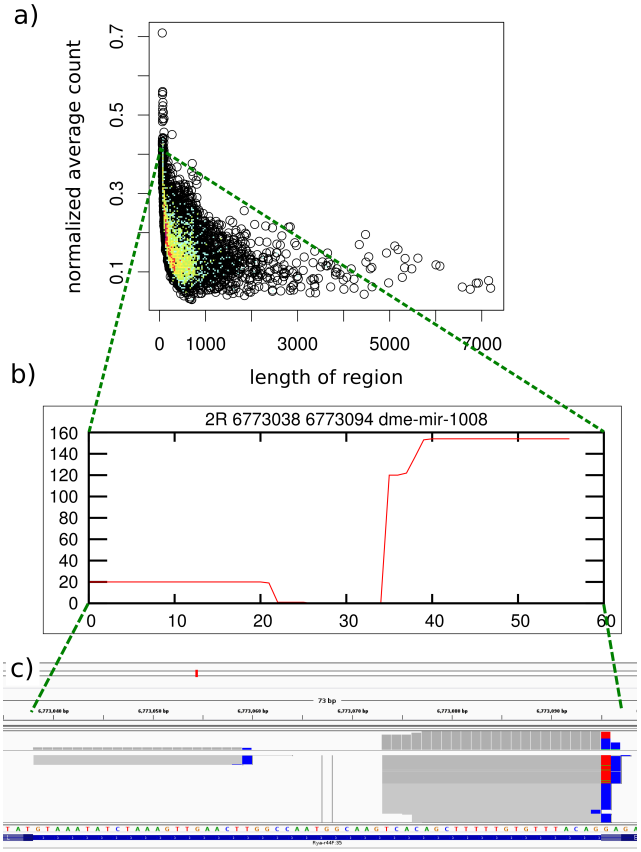

Figure 2: Properties of ROIs. (a,b,c) ROI information of a small RNA experiment from the European Nucleotide Archive (SRR618933). (a) The properties "length" and "normalized average coverage (nac) are plotted in log space for all ROIs; the density of points is represented by color, where dark red/violet represents large amounts and light cyan represent low counts. (b) Example ROIs of a miRNA (FBgn00262412) (b). (c) Hyperlinks give access to further information on a genome browser. Green lines indicate where the different images correspond to each other.

4 Acknowledgements

We thank N. Aulner, T. Melia, B. Schwikowski, and K. Smith for helpful discussions and reading the manuscript.

References

[1] M. Berthold *et al.* In C. Preisach and et al., editors, *Data Analysis, Machine Learning and Applications: Studies in Classification, Data Analysis, and Knowledge Organization*, volume V, pages 319–326, 2008.

[2] G. Chawla and N. Sokol. MicroRNAs in Drosophila development. *Int. Rev. Cell Mol. Biol.*, 286:1–65, 2011.

[3] J. Degner *et al.* Effect of read-mapping biases on detecting allele-specific expression from RNA-sequencing data. *Bioinformatics*, 25:3207–3212, 2009.

[4] M. Dillies *et al.* A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief. Bioinform.*, Sep 17, 2012.

[5] M. Dodt *et al.* FLEXBAR—flexible barcode and adapter processing for next-generation sequencing platforms. *Biology*, 1:895–905, 2012.

[6] J. Dohm *et al.* Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.*, 36:e105, 2008.

[7] A. Döring *et al.* SeqAn an efficient, generic C++ library for sequence analysis. *BMC Bioinformatics*, 9:11, 2008.

[8] J. Goecks *et al.* Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.*, 11:R86., 2011.

[9] M. Lindner *et al.* Analyzing genome coverage profiles with applications to quality control in metagenomics. *Bioinformatics*, 29:1260–1267, 2013.

[10] M. Okoniewski *et al.* Preferred analysis methods for single genomic regions in RNA sequencing revealed by processing the shape of coverage. *Nucleic Acids Res.*, 40:e363, 2012.

[11] P. Park. ChIP-seq: advantages and challenges of a maturing technology. *Nat. Rev. Genet.*, 10:669–680, 2009.

[12] J. Robinson *et al.* Integrative genomics viewer. *Nat. Biotechnol*, 29:24–26, 2011.

[13] M. Ross *et al.* Characterizing and measuring bias in sequence data. *Genome Biol.*, 14:R51, 2013.

[14] M. Skinner *et al.* JBrowse: a next-generation genome browser. *Genome Res.*, 19:1630–1638, 2009.

[15] L. Wang *et al.* RSeQC: Quality control of RNA-seq experiments. *Bioinformatics*, 28:2184–2185, 2012.

Approximate String Matching using a Bidirectional Index [1]

Gregory Kucherov[1,2], Kamil Salikhov[3] and Dekel Tsur[1]

[1] CNRS/LIGM, Université Paris-Est Marne-la-Vallée, France
[2] Department of Computer Science, Ben-Gurion University of the Negev, Israel
[3] Mechanics and Mathematics Department, Lomonosov Moscow State University, Russia

We study strategies of approximate pattern matching that exploit bidirectional text indexes, extending and generalizing ideas of [2]. We introduce a formalism, called search schemes, to specify search strategies of this type, then develop a probabilistic measure for the efficiency of a search scheme, prove several combinatorial results on efficient search schemes, and finally, provide experimental computations supporting the superiority of our strategies.

Consider approximate string matching, where k letter mismatches are allowed between a pattern P and a text T. Both *forward* and *backward* searches can be extended to approximate search in a straightforward way, by exploring all possible mismatches along the search, as long as their number does not exceed k and the current pattern still occurs in the text. Our current results have been obtained under the assumption of Hamming distance, but our techniques can be extended to the edit distance, which is the subject of ongoing work.

Lam et al. [2] gave a new search algorithm, called *bidirectional search*, that utilizes the bidirectional property of the index. The idea is to partition the pattern P on $k+1$ parts of almost equal size, and perform a sequence of searches, that cover all possible distributions of mismatches among the parts.

Our contribution consists of the following. First, we provide a general framework for working with search schemes and a theoretical tool to measure their efficiency. Second, we prove several facts about properties of optimal partitions. Third, we suggest two ideas that improve search schemes performance: using *uneven* partitions instead of partitions of equal-sized parts, and partition the pattern into *more* than $k+1$ parts ($k+2$, $k+3$, etc).

We demonstrate the superiority of our search strategies, for many practical parameter ranges, by both comparative analytical estimations based on our probabilistic analysis, and by large-scale experiments on real genomic data.

References

[1] P. Ferragina and G. Manzini. Opportunistic data structures with applications. In *Proc. 41st Symposium on Foundation of Computer Science (FOCS)*, pages 390–398, 2000.

[2] T. W. Lam, R. Li, A. Tam, S. C. K. Wong, E. Wu, and S.-M. Yiu. High throughput short read alignment via bi-directional BWT. In *Proc. IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 31–36, 2009.

1. The work has been presented in June 2014 to the Combinatorial Pattern Matching Conference

MoBiDiCC : un nouvel outil pour rechercher des motifs HLA associés à une maladie dans des données cas-témoins

Sébastien Letort[1], Marie-Claude Babron[2] and Emmanuelle Génin[3]

[1] CHRU Brest, INSERM UMR1078, Brest, France
[2] INSERM UMR946, Université Diderot, Paris, France
[3] CHRU Brest, INSERM UMR1078, Université de Bretagne Occidentale, Brest, France

sebastien.letort@inserm.fr
marie-claude.babron@inserm.fr
emmanuelle.genin@inserm.fr

Dans les protéines de type récepteur seuls quelques acides aminés déterminent le site actif. La protéine HLA-DRB1 est associée à plusieurs maladies auto-immunes dont la polyarthrite rhumatoïde (PR). Parce que certains des allèles de cette protéine sont plus associés que d'autres, on pense qu'un ligand se fixe sur eux déclenchant la maladie. Pour déterminer sans a priori le motif en acides aminés du site actif de HLA-DRB1, plusieurs méthodes existent et reposent sur l'utilisation de données cas-témoins. Une première méthode consiste à générer une séquence consensus d'acides aminés à partir des allèles HLA-DRB1 associés à la maladie. C'est par une telle méthode et en se focalisant sur la troisième région hypervariable des molécules HLA-DRB1 que l'hypothèse de l'épitope partagé dans la PR a été proposée par Gregersen et coll. (1987). Une autre méthode s'appuie sur des modèles de régression logistique pas à pas et appliquée à la PR a permis de mettre en évidence l'implication possible d'autres acides aminées que ceux de la troisième région hypervariable dans la susceptibilité à la PR (Raychaudury et al., 2012). Enfin, dans un article plus récent, Zhao et Wang (2013) ont proposé une méthode de regroupement récursif des allèles HLA-DRB1 basée sur les similarités de séquences et leurs différences de distribution entre cas et témoins qui permet d'identifier ce qu'ils appellent des SSV ou « Super Sequence Variants » associés à la maladie. Nous présentons une nouvelle méthode de recherche de motifs protéiques basée sur des données cas-témoins que nous avons baptisée MoBiDiCC pour « Motif Binding Discovery in Case-Control data ». Cette méthode repose sur plusieurs étapes. Dans la première étape, nous appliquons une approche par force brute pour identifier à partir de tous les allèles HLA-DRB1 présents dans l'échantillons tous les motifs possibles, c'est à dire toutes les combinaisons des séquences des allèles. Puis, nous utilisons les données cas-témoins pour attribuer un score à chacun des motifs qui dépend des différences de fréquences entre malades et témoins. Ensuite, nous construisons un graphe de dépendance entre ces motifs et nous identifions des scores maximums locaux, révélateurs des motifs potentiellement actifs. Nous avons procédé à des simulations faisant varier les motifs, la prévalence de la maladie et le risque associé à chaque motif et le nombre de motifs actifs afin de valider notre approche et de la comparer aux deux autres méthodes proposées dans la littérature. Nous montrons que notre approche permet de trouver plus souvent le bon motif que les autres méthodes. Lorsque plusieurs motifs sont associés à la maladie, MoBiDiCC est la seule des trois méthodes qui permet de les identifier. Enfin, si nous nous sommes intéressés dans un premier temps aux molécules HLA-DRB1, la méthode MoBiDiCC est implémentée dans un programme flexible qui pourra à terme être étendu à l'étude d'autres systèmes protéiques.

Statistical estimation of genomic alterations of tumors

Yi Liu[1], Christine Keribin[1,2], Tatiana Popova[3] and Yves, Rozenholc[2,4]

[1] Laboratoire de Mathématique, Université Paris Sud
[2] INRIA Equipe Select
[3] INSERM U830 – Institut Curie
[4] MAP5 UFR Math-Info Université Paris Descartes

Abstract: Recent research reveals that personalized medicine is one major way to treat cancer. In order to develop personalized medicine, characterizing the genomic alterations is a vital component. Several methods have been proposed to this end. One of the first methods is the Genome Alteration Print (GAP) by Popova *et al.*. We follow this approach and develop a parametric probabilistic model for GAP, based on a preliminary segmentation of SNP measurements obtained from microarray experiments. Moreover we implement the expectation-maximization (EM) algorithm to estimate the parameters of our model that characterize the genomic alterations. Finally, the model is tested on simulated data and real data.
Keywords: cancer, EM algorithm, genomic alterations, GAP, SNP

Recent research reveals that personalized medicine is arguably one major way to treat cancer because of, for example, the immense diversity of underlying genomic alterations. In order to develop personalized medicine, characterizing the genomic alterations is a vital component. One way to characterize this alteration is to use a Single Nucleotide Polymorphism (SNP) microarray. A SNP is a nucleotide showing variability in the population. In theory, there are four possible variations, however, in practice, only two variations are observed which are called A-allele and B-allele, one being common in more than 90% of the population. Since the chromosomes in human come in pairs, it is possible for a SNP to have the genotype AA, BB, AB, or BA. The two former cases are called homozygous SNP, and the two latter, which are indistinguishable, are called heterozygous SNP. If in the population, the proportion of the major allele for a SNP is h, the probability for this SNP to be homozygote is $q = 1 - 2h(1 - h)$.

Using microarrays one can detect genomic alterations such as copy-number variation and allele-imbalance. Having at hand two microarrays, one for the tumor, the other for the normal tissue, one can get rid of the unknown proportion p of normal tissues in the tumor sample which acts as a confusing parameter in the tumoral alteration characterization. However, clinicians are expecting to retrieve this information from only a single tumor sample microarray. Several methods have already been developed for this goal. GenoCNA[4], OncoSNP[6], and GPHMM[1] employ a Hidden Markov Model (HMM) integrating both segmentation and mutation characterization in a single step. GAP[3] and ASCAT[5] adopt a two-step approach in which the data are first segmented and then the mutation types are estimated. Both methods are based on an optimization step with respect to p of a deterministic quality criterion. Taking into account allelic imbalance and copy number aberration, the criterion used in ASCAT[5] measures a weighted discrepancy based on several heuristics. Noticing that, for a given p, the possible mutations are precisely localized in the plane (BAF, LRR), the GAP[3] criterion is defined as the number of segmented observations that are close to these locations within a predefined proximity value. In [2], a comparison of these methods has been performed, showing that, the two-step approaches have better performance.

Using the mutation localization in the plane (BAF, LRR) as introduced in [3], we develop a probabilistic model to estimate statistically the parameters and the mutation types of each segment. Our method uses an optimization with respect to p of a criterion which can easily be understood from a probabilistic point-of-view as it is the likelihood of our model, together with the estimation of the other parameters such as the variances of the observations. Moreover, our approach does not use any heuristic or any given tuning parameter. We expect our strategy to be not only satisfying from a mathematical point-of-view but also bring to the clinicians the expected probabilistic model for mutations.

For a SNP, microarray measures the intensities I_A and I_B of the two alleles which are proportional to the number of copies n_A and n_B of the two alleles ($I_A = kn_A$, $I_B = kn_B$). From the two intensities, it is possible to derive two variables characterizing the copy-number and the allele imbalance of the SNP

$$lrr = \log_2\left(\frac{I_A + I_B}{I_{Ref}}\right)^{\alpha} = \alpha \log_2(CN) + \beta, \quad baf = \frac{I_B}{I_A + I_B} = \frac{n_B}{n_A + n_B},$$

where $CN = n_A + n_B$ is the copy-number, α the contraction factor due to experimental techniques and $\beta = \alpha \log_2(k/I_{Ref})$. By definition, baf is bounded between 0 and 1. Assume that the proportion of normal tissues is p in the biopsy and that the tumor cells have $(n_A = k, n_B = l)$ as mutation genotype, then for heterozygous SNP we have

$$lrr = \alpha \log_2(2p + (1-p)(k+l)) + \beta, \quad baf = (p + l(1-p))/(2p + (k+l)(1-p)).$$

Measurement values are noisy and we observe on SNP m

$$LRR_m = lrr_m + \eta\xi_m, \quad BAF_m = baf_m + \sigma\varepsilon_m$$

where ξ_m and ε_m are independent standard random variables that we will assume Gaussian, and η and σ two positive real numbers. We assume that σ and η do not depend on the SNP. In the following, we assume that $\alpha = 1$ and $\beta = 0$ as a first approximation. Genomic alterations occuring on intervals of the genome, the two distributions of baf_m and lrr_m can be considered piecewise constant as m varies. Hence mutation characterization can be realized from a proper segmentation of the two distributions. Following [3, 5], we assume these segmentations already realized and we focus on the characterization part of the mutations. On each segment, the BAF values are mirrored around 0.5, so we confine ourselves to the range of $[0.5, 1]$ by symmetry. On segment i of lengths I_i, we observe

$$BAF_i^0 = baf_k^0 + \varepsilon_i^0 \frac{\sigma}{\sqrt{I_i(1-q)}}, \quad BAF_i^1 = baf_k^1 + \varepsilon_i^1 \frac{\sigma}{\sqrt{I_i q}}, \quad LRR_i = lrr_k + \xi_i \frac{\eta}{\sqrt{I_i}},$$

obtained by averaging over the segment. Here BAF_i^0 is the heterozygous BAF with a relative weight of $1-q$, BAF_i^1 the homozygous BAF with a relative weight of q, and LRR_i the LRR of the segment. The integer k is the class label indicating the underlying mutation type of the segment. ε_i^0, ε_i^1 and ξ_i are independent standard Gaussian random variables. Taking into account the heterozygosity and homozygosity, we split the interval observation into two weighted sub-observations (BAF_i^0, LRR_i) with weight $1-q$ and (BAF_i^1, LRR_i) with weight q. These split observations can be mapped into the plane of (baf, lrr) where (baf_k^0, lrr_k) and (baf_k^1, lrr_k) have fixed positions only defined by p and the underlying mutation.

We define a mixture model for the split observations and introduce the component indicators z_{ik} with $i = 1, 2, \ldots, n$ (n the number of segments) and $k = 1, 2, \ldots, K$ (K number of considered mutations). Its value is 1 if the observation is emitted from the underlying mutation k. Our model is a Gaussian mixture model in the plane of (BAF, LRR) with observations (BAF_i, LRR_i) and known parameters I_i and q, the latter assumed to be known and constant for all segments. The parameters to estimate are: p the proportion of normal tissues, η the standard deviation of LRR, and σ the standard deviation of BAF. The parameter p is tricky to infer simultaneously with the other parameters, hence we use a two-level strategy: for a given p, we implement an EM algorithm to estimate the other parameters and then use gradient descent method to find the optimal value of p. We tested our implementation on simulated data where the estimation agreed well with the parameters used to generate the data. As a result, we can retrieve the mutation of any given interval as the most probable. Moreover, one has at hand the probability distribution of the mutations for each interval. Possible extensions of our strategy include the direct use of un-split observations.

References

[1] A. Li *et al.* GPHMM: an integrated hidden markov model for identification of copy number alteration and loss of heterozygosity in complex tumor samples using whole genome SNP arrays. *Nucl. Acids Res.*, 2011.

[2] D. Mosén-Ansorena *et al.* Comparison of methods to detect copy number alterations in cancer using simulated and real genotyping data. *BMC Bioinfo.*, 2012.

[3] T. Popova *et al.* Genome alteration print (GAP): a tool to visualize and mine complex cancer genomic profiles obtained by SNP arrays. *Genome Biology*, 2010.

[4] W. Sun *et al.* Integrated study of copy number states and genotype calls using high-density SNP arrays. *Nucl. Acids Res.*, 2009.

[5] P. Van Loo *et al.* Allele-specific copy number analysis of tumors. *Proc. of the Nat. Acad. of Sci.*, 2010.

[6] C. Yau *et al.* A statistical approach for detecting genomic aberrations in heterogeneous tumor samples from single nucleotide polymorphism genotyping data. *Genome Biology*, 2010.

A coverage criterion for spaced seeds and its applications to SVM string-kernels and k-mer distances

Laurent Noé[1] and Donald E. K. Martin[2]

[1] LIFL (UMR 8022 Lille 1/CNRS) - Inria Lille, Villeneuve d'Ascq, France
[2] Department of Statistics, North Carolina State University, Raleigh, NC, USA

laurent.noe@univ-lille1.fr
martin@stat.ncsu.edu

Spaced seeds have been recently shown to not only detect more alignments, but also to give a more accurate measure of phylogenetic distances [2, 4, 3], and to provide a lower misclassification rate when used with Support Vector Machines (SVMs) [7], We confirm by independent experiments these two results, and propose to use a *coverage criterion* [1, 5, 6], to measure the seed efficiency in both cases in order to design better seed patterns. We show first how this *coverage criterion* can be directly measured by a full automaton-based approach. We then illustrate how this criterion performs when compared with two other criteria frequently used, namely the *single-hit* and *multiple-hit criteria*, through correlation coefficients with the correct classification/the true distance. At the end, for alignment-free distances, we propose an extension by adopting the *coverage criterion*, show how it performs, and indicate how it can be efficiently computed.

More details can be found at http://bioinfo.lifl.fr/yass/iedera.php#iedera_coverage

References

[1] G. Benson and D. Y. Mak. Exact distribution of a spaced seed statistic for DNA homology detection. In *Proceedings of the International Symposium on String Processing and Information Retrieval (SPIRE)*, volume 5280 of *LNCS*, pages 282–293, 2008.

[2] M. Boden, M. Schöneich, S. Horwege, S. Lindner, C. Leimeister, and B. Morgenstern. Alignment-free sequence comparison with spaced k-mers. In *Proceedings of the German Conference on Bioinformatics (GCB)*, volume 34 of *OpenAccess Series in Informatics (OASIcs)*, pages 24–34, 2013.

[3] S. Horwege, S. Lindner, M. Boden, K. Hatje, M. Kollmar, C.-A. Leimeister, and B. Morgenstern. Spaced words and kmacs: Fast alignment-free sequence comparison based on inexact word matches. *Nucleic Acids Research*, 42(W1):W7–W11, 2014.

[4] C.-A. Leimeister, M. Boden, S. Horwege, S. Lindner, and B. Morgenstern. Fast alignment-free sequence comparison using spaced-word frequencies. *Bioinformatics*, 30(14):1991–1999, 2014.

[5] D. E. K. Martin. Coverage of spaced seeds as a measure of clumping. In *JSM Proceedings, Statistical Computing Section*, Alexandria, Virginia, 2013. American Statistical Association.

[6] D. E. K. Martin and L. Noé. Faster exact probabilities for statistics of overlapping pattern occurrences. *Submitted to the Annals of the Institute of Statistical Mathematics (AISM)*, 2014.

[7] T. Onodera and T. Shibuya. The gapped spectrum kernel for support vector machines. In *Proceedings of the International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)*, volume 7988 of *LNCS*, pages 1–15, 2013.

Navigating in a Sea of Repeats in RNA-seq Without Drowning (Extended abstract)

Gustavo Sacomoto[1,2], Blerina Sinaimeri[1,2], Camille Marchet[1,2], Vincent Miele[2], Marie-France Sagot[1,2] and Vincent Lacroix[1,2]
[1] INRIA Grenoble Rhône-Alpes, France
[2] UMR CNRS 5558 - LBBE, Université Lyon 1, France

Abstract: The main challenge in *de novo* assembly of NGS data is certainly to deal with repeats that are longer than the reads. This is particularly true for RNA-seq data, since coverage information cannot be used to flag repeated sequences, of which transposable elements are one of the main examples. Most transcriptome assemblers are based on de Bruijn graphs and have no clear and explicit model for repeats in RNA-seq data, relying instead on heuristics to deal with them. The results of this work are twofold. First, we introduce a formal model for representing high copy-number repeats in RNA-seq data and exploit its properties to infer a combinatorial characteristic of repeat-associated subgraphs. We show that the problem of identifying in a de Bruijn graph a subgraph with this characteristic is NP-complete. In a second step, we show that in the specific case of a local assembly of alternative splicing (AS) events, using our combinatorial characterization we can *implicitly* avoid such subgraphs. In particular, we designed and implemented an algorithm to efficiently identify AS events that are not included in repeated regions. Finally, we validate our results using synthetic data. We also give an indication of the usefulness of our method on real data. [2]

1 Introduction

Transcriptomes can now be studied through sequencing. However, in the absence of a reference genome, de novo assembly remains a challenging task. The main difficulty certainly comes from the fact that sequencing reads are short, and repeated sequences within transcriptomes could be longer than the reads. This short read / long repeat issue is of course not specific to transcriptome sequencing. It is an old problem that has been around since the first algorithms for genome assembly. In this latter case, the problem is somehow easier because coverage can be used to discriminate contigs that correspond to repeats, *e.g.* using Myer's A-statistics [4] or [5]. In transcriptome assembly, this idea does not apply, since the coverage of a gene does not only reflect its copy-number in the genome, but also and mostly its expression level.

Initially, it was thought that repeats would not be a major issue in RNA-seq, since they are mostly in introns and intergenic regions. However, the truth is that many regions which are thought to be intergenic are transcribed [1] and introns are not always already spliced out when mRNA is collected to be sequenced. Repeats, especially transposable elements, are therefore very present in real samples and cause major problems in transcriptome assembly.

In the method we developed, KisSplice, which is a local transcriptome assembler [6], repeats may be less problematic, since the goal is not to assemble full-length transcripts. KisSplice instead aims at finding variations expressed at the transcriptome level (SNPs, indels and alternative splicings). However, as we previously reported in [6], there are certain complex regions in the graph, likely containing repeat-associated subgraphs but also real AS events [6], where KisSplice takes a huge amount of time. The enumeration of AS events is therefore halted after a given timeout. The AS-events *drowned* (or trapped) inside these regions are thus missed by KisSplice.

Here, we try and achieve two goals: (i) give a clear formalization of the notion of repeats with high copy-number in RNA-seq data, and (ii) based on it, give a practical way to enumerate bubbles that are lost because of such repeats. Recall that we are in a *de novo* context, so we assume that neither a reference genome/transcriptome nor a database of known repeats, are available.

2. The results presented in this extended abstract have been published in WABI 2014, LNCS vol 8701, 82-96.

2 Repeats in de Bruijn graphs

A *k-mer* is a sequence $s \in \{A, C, T, G\}^k$. Given a set of sequences (reads) R and an integer k, the directed de Bruijn graph $G_k(R) = (V, A)$ is such that V and A are the set of all distinct k-mers and $k+1$-mers respectively, that appear as a substring in R. An arc $(u, v) \in A$ is called *compressible* if the out-degree u and the in-degree of v are equal to 1.

Simple uniform model for repeats: Our model consists of several "similar" sequences, each generated by uniformly mutating a fixed initial sequence. The model has then the following parameters: the length n of the repeat, the number m of copies of the repeat, an integer k, and the mutation rate, α, *i.e.* the probability that a mutation happens in a particular position. We first choose uniformly at random a sequence s_0 of length n. At step $i \leq m$, we create a sequence s_i by mutating each position of s_0 with probability $1 - \alpha$. Repeating this process we create a set $S(m, n, \alpha)$ of m such sequences from s_0. The generated sequences thus have an expected Hamming distance of αn from s_0.

This model is a simple one but is realistic enough in some real cases. In particular, it enables to model well recent invasions of transposable elements which often involve high copy-number and low divergence rate.

The next result shows that the number of compressible arcs is a good parameter for characterizing a repeat-associated subgraph.

Theorem 1 *Given integers k, n, m with $k < n$ and a real number $0 \leq \alpha \leq 3/4$, consider $S(m, n, \alpha)$ and let R be a set of m sequences randomly chosen then:*

- *The expected number of compressible arcs in $G_k(S(m, n, \alpha))$ is $\emptyset(mn)$.*
- *The expected number of compressible arcs in $G_k(R)$ is $\Theta(mn)$*

Based on this, a natural formulation to the repeat identification problem in RNA-seq data is to search for large enough subgraphs that do not contain many compressible arcs. Unfortunately, the next result shows that an efficient algorithm for the repeat identification problem based on this formulation is unlikely.

Theorem 2 *Given a directed graph G and two positive integers m, t, it is NP-complete to decide whether there exists a connected subgraph $G' = (V', E')$, with $|V'| \geq m$ and having at most t compressible arcs. It remains NP-complete even for subgraphs of de Bruijn graphs on 4 symbols.*

3 Bubbles "drowned" in repeats

KisSplice [6] is a method for *de novo* calling of AS events through the enumeration of so-called *bubbles*, that correspond to pairs of vertex-disjoint paths in a de Bruijn graph. However, we showed in [6] that some bubbles were missed by KisSplice because they were "drown" in a complex region of the graph. See Fig. 3 for an example of a complex region with a bubble corresponding to an AS event. We saw that the repeat-associated subgraphs are characterized by the presence of few compressible arcs. This suggests that in order to avoid repeat-associated subgraphs, we should restrict the search to bubbles containing many compressible arcs. Equivalently, in a compressed de Bruijn graph, we should restrict the search to bubbles with few branching vertices. Indeed, in a compressed de Bruijn graph, given a fixed sequence length, the number of branching vertices in a path is inversely proportional to the number of compressible arcs of the corresponding path in the non-compressed de Bruijn graph. We thus modify the definition of $(s, t, \alpha_1, \alpha_2)$-bubbles in compressed de Bruijn graphs (Def. 1 in [7]) by adding the extra constraint that each path should have at most b branching vertices. By modifying the algorithm of [7] we have the following:

Theorem 3 *The $(s, *, \alpha_1, \alpha_2, b)$-bubbles can be enumerated in $O(b|V|^3|E||\mathcal{B}_s(G)|)$ time. Moreover, the time elapsed between the output of any two consecutive solutions (i.e. the delay) is $O(b|V|^3|E|)$.*

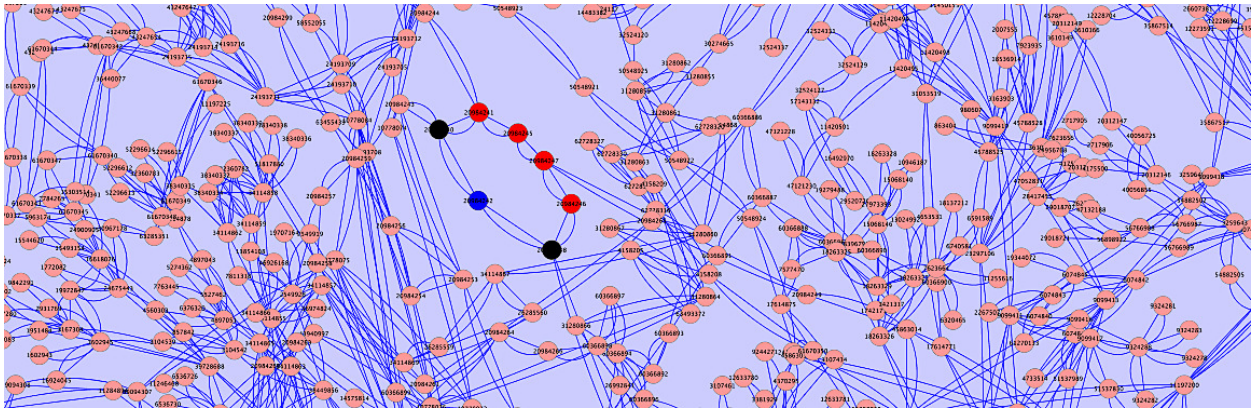

Figure 3: An alternative splicing event in the SCN5A gene (human) trapped inside a complex region, likely containing repeat-associated subgraphs, in a de Bruijn graph. The alternative isoforms correspond to a pair of paths shown in red and blue.

4 Experimental results

To evaluate the performance of our method, we simulated RNA-seq data using the FluxSimulator version 1.2.1 [3]. We generated 100 million reads of 75 bp using the default error model. We used the RefSeq annotated Human transcriptome (hg19 coordinates) as a reference and we performed a two-step pipeline to obtain a mixture of mRNA and pre-mRNA (*i.e.* with introns not yet spliced). We tested two values: 5% and 15% for the proportion of reads from pre-mRNAs. Those values were chosen so as to correspond to realistic ones as observed in a cytoplasmic mRNA extraction (5%) and a total (cytoplasmic + nuclear) mRNA extraction (15%) [8].

On these simulated datasets, we ran KisSplice [6] versions 2.1.0 (KsOld) and 2.2.0 (KsNew, with a maximum number of branching vertices set to 5) and the full-length transcriptome assembler Trinity version r2013_08_14 [2].

We showed that KsNew improves by a factor of up to 2 the sensitivity of the previous version of KisSplice, while also improving its precision. Concerning the comparison with Trinity we showed that for the specific tasks of calling AS events, our algorithm is more sensitive, by a factor of 2, while also being slightly more precise (see Fig. 4).

Finally, we gave an indication of the usefulness of our method on real data where we have examples of AS-events not found by KsOld.

References

[1] S. Djebali, C. Davis, A. Merkel, and A. Dobin *et al.* Landscape of transcription in human cells. *Nature*, 2012.

[2] M. Grabherr, B. Haas, M. Yassour, and J. Levin *et al.* Full-length transcriptome assembly from RNA-Seq data without a reference genome. *Nat. Biot.*, 2011.

[3] T. Griebel, B. Zacher, P. Ribeca, and E. Raineri *et al.* Modelling and simulating generic RNA-Seq experiments with the flux simulator. *Nucleic Acids Res.*, 2012.

[4] E. Myers, G. Sutton, A. Delcher, and I. Dew *et al.* A whole-genome assembly of drosophila. *Science*, 287(5461):2196–204, 2000.

[5] P. Novák, P. Neumann, and J. Macas. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC Bioinf.*, 2010.

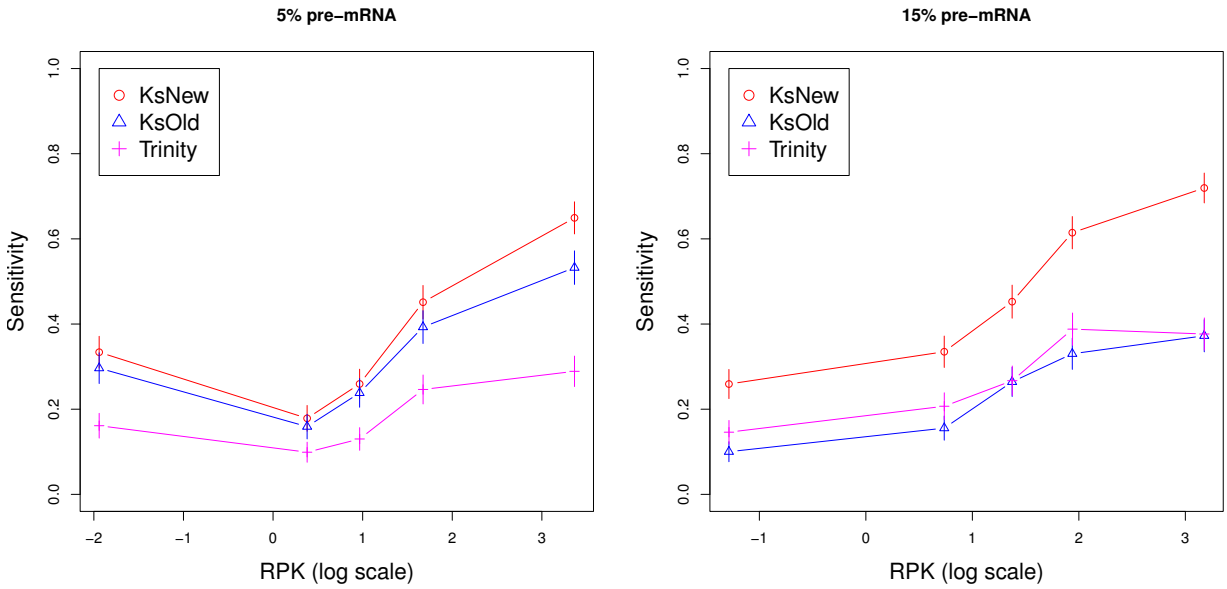

Figure 4: Sensitivity of KsNew, KsOld and Trinity.

[6] G. Sacomoto, J. Kielbassa, R. Chikhi, and R. Uricaru *et al.* KISSPLICE: de-novo calling alternative splicing events from RNA-seq data. *BMC Bioinformatics*, 13(Suppl 6):S5, 2012.

[7] G. Sacomoto, V. Lacroix, and M.-F. Sagot. A polynomial delay algorithm for the enumeration of bubbles with length constraints in directed graphs and its application to the detection of alternative splicing in rna-seq data. In *WABI*, pages 99–111, 2013.

[8] H. Tilgner, D. Knowles, R. Johnson, and C. Davis *et al.* Deep sequencing of subcellular RNA fractions shows splicing to be predominantly co-transcriptional in the human genome but inefficient for lncRNAs. *Genome Res.*, 2012.

An efficient method for correcting long PacBio reads

Leena Salmela[1] and Eric Rivals[2]

[1] University of Helsinki, Finland
[2] LIRMM - UMR 5506 CNRS UM2, France

Leena.Salmela@cs.helsinki.fi
rivals@lirmm.fr

PacBio Single Molecule, Real Time sequencing is a third generation sequencing technique producing long reads with comparatively lower throughput and higher error rate. Errors include numerous indels and complicate downstream analysis like mapping or de novo assembly. A hybrid strategy that takes advantage of the high accuracy of Second Generation short reads has been proposed for correcting long reads. Mapping of short reads on long reads provides sufficient coverage to eliminate up to 99% of errors, however at the expense of prohibitive running times and considerable amounts of disk and memory space. We present LoRDEC, a hybrid error correction method that builds a succinct de Bruijn graph representing the short reads, and seeks a corrective sequence for each erroneous region in the long reads by traversing chosen paths in the graph. In comparisons, LoRDEC is at least six times faster and requires at least 93% less memory or disk space than available tools, while achieving comparable accuracy. LoRDEC software is available on the ATGC platform (http://www.atgc-montpellier.fr/lordec). A more complete description of this work is available in [1].

References

[1] L. Salmela and E. Rivals. LoRDEC: accurate and efficient long read error correction. *Bioinformatics*, August 26 2014. On line doi:10.1093/bioinformatics/btu538.

A seeding framework for lossless filtering for the approximate pattern matching problem [1]

Christophe Vroland[1,2,3], Mikaël Salson[1,2] and Hélène Touzet[1,2]

[1] LIFL (UMR CNRS 8022, Université Lille 1)
[2] Inria Lille Nord-Europe, France
[3] GEPV (UMR CNRS 8198, Université Lille1)

christophe.vroland@lifl.fr
mikael.salson@lifl.fr
helene.touzet@lifl.fr

Introduction

Let A be a finite alphabet. Given two strings u and v of A^*, define $lev(u, v)$ to be the Levenshtein distance between u and v. This is the minimum number of operations needed to transform u into v, where the only allowed operations are substitution of a single character and deletion or insertion of a single character. Each such operation is also called an *error*. From now on, we assume that a given natural number k corresponds to a maximum number of errors.

The *approximate pattern matching problem* is to find all the locations where a pattern matches a text with at most k errors. This problem has been extensively studied in the literature. Navarro *et al* distinguish three main approaches [3]: *neighborhood generation*, *partitioning approach* and *hybrid method*. Here, we present a method that lies in the third case, and that uses a new kind of lossless approximate seeds.

01^*0 seeds

The main idea is as follows. Let P be a pattern over A. Using the pigeonhole principle, it is well-known that if P is partitioned into $k+1$ parts, then every string U, such that $lev(P, U) \leq k$, contains at least one of these parts. Similarly, if P is partitioned into $k+2$ parts, denoted $P_1, \ldots, P_{k+2}$, then U should contain at least two disjoint parts of P. The following lemma allows to push the analysis further. It is indeed possible to request that these two parts be separated by parts with exactly one error.

Lemma 1 *Let U be a string of A^* such that $lev(P, U) \leq k$. Then there exists i, j, $1 \leq i < j \leq k+2$, and $U_1, \ldots, U_{j-i-1}$ of A^* such that*

1. *$P_i U_1 \ldots U_{j-i-1} P_j$ is a substring of U, and*
2. *When $j > i+1$, for each ℓ, $1 \leq \ell \leq j-i-1$, $lev(P_{i+\ell-1}, U_\ell) = 1$.*

As a consequence of Lemma 1, we can design a seeding framework for lossless filtering for the approximate pattern matching problem with k errors. To this end, we introduce the 01^*0 seeds defined as follows.

Definition 1 Let $P = P_1 \ldots P_{k+2}$ be a pattern divided into $k+2$ parts. Then the 01^*0 *seed* for P and k is the regular expression

$$\cup_{i=1}^{k+1} \cup_{j=i+1}^{k+2} \; P_i \; lev^1(P_{i+1}) \ldots lev^1(P_{j-1}) \; P_j$$

where $lev^1(u)$ denotes the set of strings whose Levenshtein distance with u is 1.

1. This work was partly supported by the Mastodons project (CNRS).

Empirical measurements show that the 01^*0 seed is significantly more selective than exact seeds, such as q-grams. Of course, this higher selectivity comes at the price of some additional work to locate seeds in the text. However, the fact that errors are not randomly distributed within the seed drastically reduces the combinatorics.

Application to approximate pattern matching

We implemented our strategy using a full-text index (namely a FM-index [1]). For details on the search method using a FM-index, the reader should report to the full paper [5].

We measured the performance of our implementation (called Bwolo [2]) to a selection of tools chosen for their complementarity (indexed or not, using different approximate string matching approaches): Exonerate [4], RazerS3 [6], Bowtie2 [2]. We searched for short patterns in random texts and in DNA sequences with two or three errors at full sensitivity for each software. On our tests, Bwolo is faster by an order-of-magnitude than its counterparts, but uses a similar amount of memory.

Conclusion

The new 01^*0 seeds we introduced achieve a good balance between the filtration step and the verification effort. We have developed a full application that implements the approximate pattern matching problem with an index for the text. It could also be interesting to preprocess the patterns when dealing with a large number of them. On the contrary, we could also get rid of the index and make the filtration algorithm online. Finally, albeit having been beyond the scope of this paper, an important aspect to thoroughly analyze would be the average case of our algorithm.

References

[1] P. Ferragina and G. Manzini. Indexing compressed text. *Journal of the ACM (JACM)*, 52(4):552–581, 2005.

[2] B. Langmead and S. L. Salzberg. Fast gapped-read alignment with Bowtie 2. *Nature methods*, 9(4):357–359, 2012.

[3] G. Navarro, E. Sutinen, J. Tanninen, and J. Tarhio. Indexing text with approximate q-grams. In *Combinatorial Pattern Matching*, pages 350–363. Springer, 2000.

[4] G. S. C. Slater and E. Birney. Automated generation of heuristics for biological sequence comparison. *BMC Bioinformatics*, 6:1–11, 2005.

[5] C. Vroland, M. Salson, and T. Hélène. Lossless seeds for searching short patterns with high error rates. In *IWOCA*, 2014.

[6] D. Weese, M. Holtgrewe, and K. Reinert. RazerS 3: Faster, fully sensitive read mapping. *Bioinformatics*, 28(20):2592–2599, 2012.

2. http://bioinfo.lifl.fr/bwolo

Tedna: a Transposable Element De Novo Assembler

Matthias Zytnicki

INRA – MIAT

Abstract: Recent technological advances are allowing many laboratories to sequence their research organisms. Available de novo assemblers leave repetitive portions of the genome poorly assembled. Some genomes contain very high proportions of transposable elements, and transposable elements appear to be a major force behind diversity and adaptation. Few de novo assemblers for transposable elements exist, and most have either been designed for small genomes or 454 reads.

In this paper, we present a new transposable element de novo assembler, Tedna, which assembles a set of transposable elements directly from the reads. Tedna uses Illumina paired-end reads, the most widely used sequencing technology for de novo assembly, and forms full length transposable elements.

Tedna is available from http://urgi.versailles.inra.fr/Tools/Tedna, under the GPLv3 license. It is written in C++11 and only requires the sparsehash package, freely available under the New BSD License. Tedna can be used on standard computers with limited RAM resources, although it may also use large memory for better results. Most of the code is parallelized, and thus ready for large infrastructures.

Introduction

Most laboratories can now afford sequencing a genome and several de novo whole genome assemblers, such as Velvet [10], are available to assemble even large genomes. However, these assemblers usually do not assemble the transposable elements: current algorithms cannot correctly assemble highly repeated sequences. On the other hand, transposable elements can comprise to more than 90% of a genome, and their role in the evolution of the host genome has been often underlined [1].

Whereas assembling the *copies* (i.e. the traces of transposable elements, scattered along the genome, often strongly mutated) is usually impossible, assembling the *transposable elements* (a model of the first element that invaded the genome, reconstructed from the observable copies [3]) is a simpler task. Several de novo transposable elements assemblers have been presented so far, but most have been designed for small genomes [8], or assume that the 454 technology have been used [5, 6]. Only RepeatExplorer [7] currently exploits Illumina reads. Here, we present new a tool that reads Illumina paired-end reads, arguably the most widely used sequencing technology for de novo assembly, and provides a list of repeated elements.

A transposable element de novo assembler is somewhat different from a genome assembler: it assembles sequences from multiple copies, which have evolved through time. A transposable elements assembler should thus correctly handle polymorphism, including long insertions and deletions. Basically, every copy gives a hint about what the transposable element should be, but only the comparison with other copies —and the construction of a consensus— may help building the transposable element. As such, de Bruijn graphs, used by many assemblers, seem well-fitted for this purpose: a k-mer of a copy may be part of the consensus transposable element, whereas other parts of the copy may not. Transposable elements may thus be assembled as a set of highly repeated k-mers. Tedna is the first tool that uses a de Bruijn graph for transposable element assembly.

1 Results and Conclusion

We compared Tedna with the transposable element assembler RepeatExplorer, and the two other widely used assemblers: Velvet, for genome assembly, and Oases, for transcriptome assembly [9].

We first benchmarked the tools on the 5143bp long *copia* element, present in *Drosophila melanogaster*. We extracted all the copies annotated as *copia* by the REPET pipe-line [2], cut them into reads, and mixed

data set	tool	# seq.	sen.	spe.	avg. max. size
wheat	RepeatExplorer	982	35%	78%	6%
	Velvet	836	2%	6%	1%
	Oases	6505	33%	37%	13%
	Tedna	1365	38%	66%	11%
A. thaliana	RepeatExplorer	160	3%	14%	2%
	Velvet	67,615	73%	2%	42%
	Oases	1963	41%	38%	30%
	Tedna	1263	24%	26%	17%

Table 1: Comparison of the tools (sens.: sensitivity, spe.: specificity).

them with random reads. We thus produced 100,000 paired-ends reads, of size 2 × 100. *Copia* is an LTR-retrotransposon, and thus has long LTRs. With the best parameters of Velvet, we had a 4709bp long element, where the low complexity region, in the center of the element, is poorly assembled. Moreover, the predicted element is the concatenation of the 3' end of the element, one LTR, and the 5' end. The longest element of Oases is 5857bp long, longer than the actual element because Oases duplicates both LTRs. With other parameters, Oases predicts a 4912bp long element, similar to the element predicted by Velvet. Tedna correctly predicts an element with 99% identity when compared to the known element. The predicted element is somewhat shorter (5049 *vs* 5143bp) because the predicted LTRs are slightly too short, and the low complexity region is not accurately assembled. The three tools needed less than a minute to produce the assembly. RepeatExplorer produced an internal error, probably because it expects more reads. This shows that a transposable element can be reconstructed from its copies with a de Bruijn graph on the most frequent k-mers.

We then tested Tedna by assembling sequence data generated for the wheat genome, 90% of which is comprised of repetitive elements. We used the unassembled reads (size 2 × 100bp) produced by the International Wheat Genome Sequencing Consortium for wheat chromosome arm 3AL (to be published), and a manually curated library of 335 wheat transposable elements (Josquin Daron, to be published). We finally compared Tedna on an *Arabidopsis thaliana* resequencing project, available from SRA under code SRR616966, which contains 2 × 100 paired-end reads and an insert size of 500bp. We used the *A. thaliana* RepBase data [4] as reference, which contains 390 transposable elements.

For the two latter data sets, we gave the number of putative transposable elements given by each tool, as well as their sensitivity and specificity in Table 1. For each reference transposable element, we computed the size of the longest predicted fragment, and expressed it as a ratio of the reference transposable element size (100% would be a predicted full length element). The average ratio size is given in the last column. The wheat data set shows that Tedna has the best sensitivity, and a good specificity (although not as good as RepeatExplorer). When ranked by size of fragments, Tedna is the second best. Oases performs well because it also contains dedicated algorithms for merging contigs into full length transcripts, and it handles read coverage better than Velvet (which expects uniform coverage). The *A. thaliana* data set is clearly favorable to Oases, which gives almost everywhere the best results. The only exception is the fragment size, where Velvet performs better. This is a usual trade-off between specificity, which is very low for Velvet, and accuracy. This results suggest that Tedna performs better when used on genomes with high transposable element density, which is observed in most higher eukaryotes.

We then detailed the results given by Tedna for each major transposable element class of *A. thaliana*. Results vary greatly, and there is no clear reason why the DNA transposons are better assembled than other elements. Up to now, Tedna only assembles repeated sequences, and cannot discriminate transposable elements. In the *A. thaliana* data set, we located the sequences assembled by Tedna that did not match the RepBase sequences. Among them, we had 1618 matches in genes. Most map to protein of unknown function; some of them could be misannotated transposable elements. The other fragments map to known duplicated genes (Agamous-like proteins, cellulose synthase, expansins, etc.). 305 fragments map to inserted copies,

and have been thus missed in our classification protocol, most likely because they significantly diverged from the consensus. 26 fragments mapped the *gypsy* element, 19 MuDR, 16 *copia*.

In a future version, we would like to provide an annotation of the output of Tedna, as RepeatExplorer does, that would classify transposable elements and possibly discriminate genes or other repeated elements.

Acknowledgement

The authors wish to thank the URGI lab, and especially F. Maumus for his helpful comments, the International Wheat Genome Sequencing Consortium for providing unpublished sequencing data, J. Daron for sharing his manually curated transposable element library, and R. Chikhi for sharing his knowledge on assembly.

References

[1] N. Fedoroff. Transposable elements, epigenetics, and genome evolution. *Science*, 338:758–767, 2012.

[2] T. Flutre, E. Duprat, C. Feuillet, and H. Quesneville. Considering transposable element diversification in *de novo* annotation approaches. *PLoS ONE*, 6:e16526, 2011.

[3] T. Flutre, E. Permal, and H. Quesneville. In search of lost trajectories: Recovering the diversification of transposable elements. *Mobile Genetic Elements*, 1:151–154, 2011.

[4] J. Jurka, V. V. Kapitonov, A. Pavlicek, P. Klonowski, O. Kohany, and J. Walichiewicz. Repbase update, a database of eukaryotic repetitive elements. *Cytogenetic and Genome Research*, 110:462–7, 2005.

[5] R. Li, J. Ye, S. Li, J. Wang, Y. Han, C. Ye, J. Wang, H. Yang, J. Yu, G. K.-S. Wong, and J. Wang. ReAS: Recovery of ancestral sequences for transposable elements from the unassembled reads of a whole genome shotgun. *PLoS Computational Biology*, 1:e43, 2005.

[6] P. Novák, P. Neumann, and J. Macas. Graph-based clustering and characterization of repetitive sequences in next-generation sequencing data. *BMC Bioinformatics*, 11:378, 2010.

[7] P. Novák, P. Neumann, J. Pech, J. Steinhaisl, and J. Macas. RepeatExplorer: a Galaxy-based web server for genome-wide characterization of eukaryotic repetitive elements from next-generation sequence reads. *Bioinfomatics*, 29:792–793, 2013.

[8] P. A. Pevzner, H. Tang, and G. Tesler. De novo repeat classification and fragment assembly. *Genome Research*, 14:1786–1796, 2004.

[9] M. H. Schulz, D. R. Zerbino, M. Vingron, and E. Birney. Oases: robust de novo RNA-seq assembly across the dynamic range of expression levels. *Bioinfomatics*, 28:1086–92, 2012.

[10] D. R. Zerbino and E. Birney. Velvet: Algorithms for de novo short read assembly using de Bruijn graphs. *Genome Research*, 18:821–829, 2008.