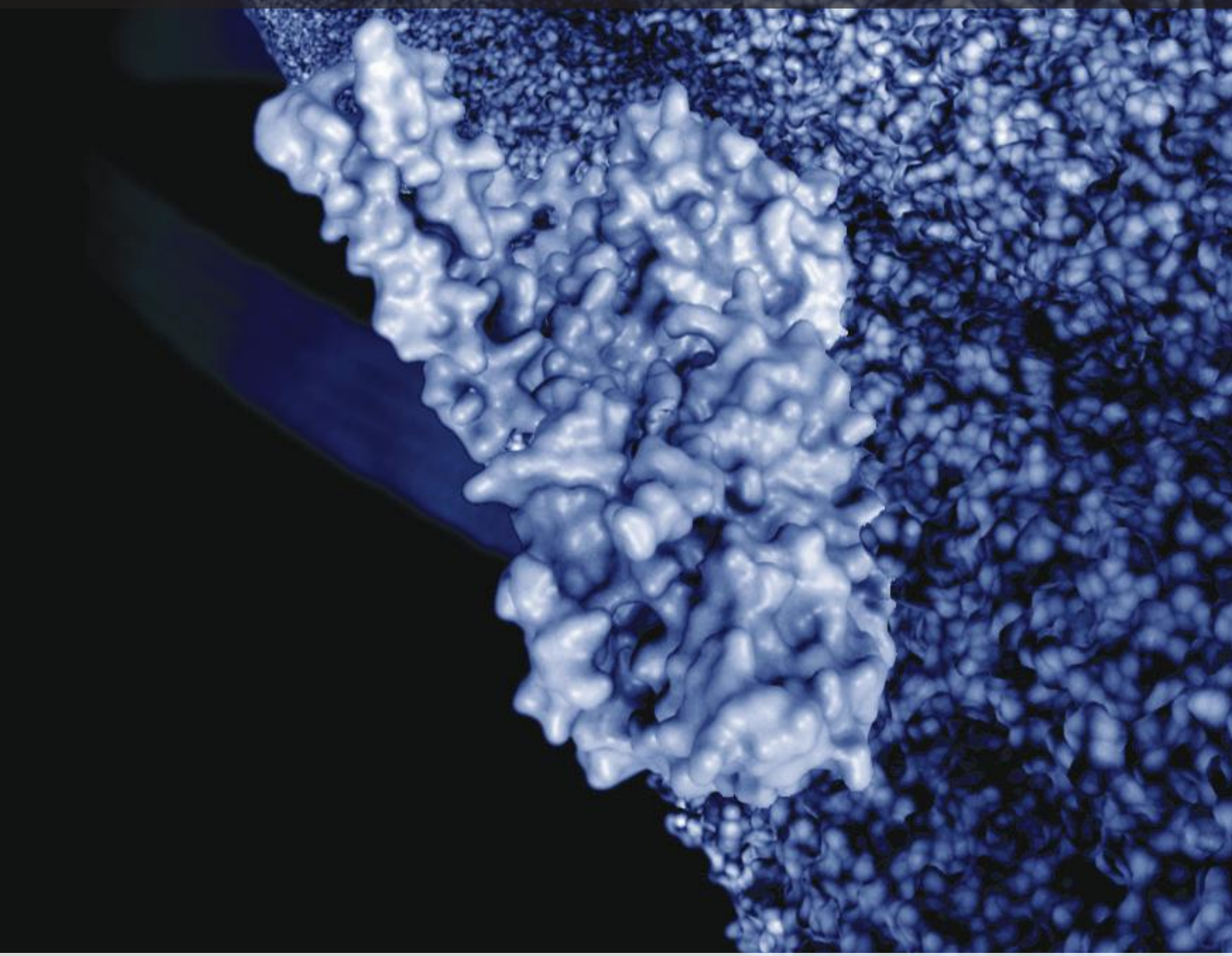
REPETITIVE STRUCTURES IN BIOLOGICAL SEQUENCES: ALGORITHMS AND APPLICATIONS

EDITED BY: Marco Pellegrini, Alberto Magi and Costas S. Iliopoulos

PUBLISHED IN: Frontiers in Bioengineering and Biotechnology



frontiers Research Topics

ISSN 1664-8714
ISBN 978-2-88945-018-3
DOI 10.3389/978-2-88945-018-3

About Frontiers

Frontiers is more than just an open-access publisher of scholarly articles: it is a pioneering approach to the world of academia, radically improving the way scholarly research is managed. The grand vision of Frontiers is a world where all people have an equal opportunity to seek, share and generate knowledge. Frontiers provides immediate and permanent online open access to all its publications, but this alone is not enough to realize our grand goals.

Frontiers Journal Series

The Frontiers Journal Series is a multi-tier and interdisciplinary set of open-access, online journals, promising a paradigm shift from the current review, selection and dissemination processes in academic publishing. All Frontiers journals are driven by researchers for researchers; therefore, they constitute a service to the scholarly community. At the same time, the Frontiers Journal Series operates on a revolutionary invention, the tiered publishing system, initially addressing specific communities of scholars, and gradually climbing up to broader public understanding, thus serving the interests of the lay society, too.

Dedication to quality

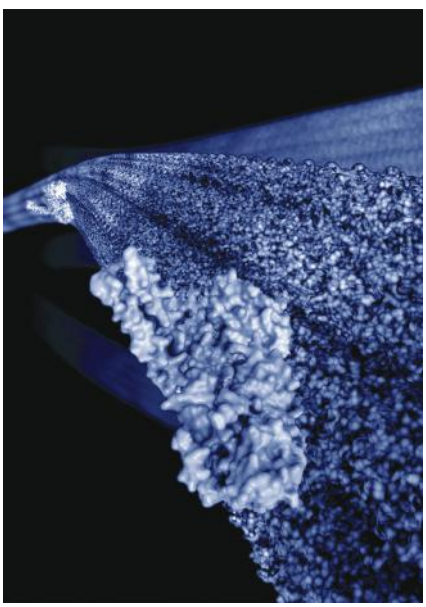
Each Frontiers article is a landmark of the highest quality, thanks to genuinely collaborative interactions between authors and review editors, who include some of the world's best academicians. Research must be certified by peers before entering a stream of knowledge that may eventually reach the public - and shape society; therefore, Frontiers only applies the most rigorous and unbiased reviews.
Frontiers revolutionizes research publishing by freely delivering the most outstanding research, evaluated with no bias from both the academic and social point of view.
By applying the most advanced information technologies, Frontiers is catapulting scholarly publishing into a new generation.

What are Frontiers Research Topics?

Frontiers Research Topics are very popular trademarks of the Frontiers Journals Series: they are collections of at least ten articles, all centered on a particular subject. With their unique mix of varied contributions from Original Research to Review Articles, Frontiers Research Topics unify the most influential researchers, the latest key findings and historical advances in a hot research area! Find out more on how to host your own Frontiers Research Topic or contribute to one as an author by contacting the Frontiers Editorial Office: **researchtopics@frontiersin.org**

REPETITIVE STRUCTURES IN BIOLOGICAL SEQUENCES: ALGORITHMS AND APPLICATIONS

Topic Editors:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy
Alberto Magi, University of Florence, Italy
Costas S. Iliopoulos, King's College London, UK



Computer generated image of microtubules with associated proteins.

Image created with BioBlender, software of the SciVis Lab, IFC-CNR, Pisa. Copyright by SciVis. Graphic Processing by Patrizia Andronico, IIT-CNR, Pisa

Repetitive structures in biological sequences are emerging as an active focus of research and the unifying concept of "repeatome" (the ensemble of knowledge associated with repeating structures in genomic/proteomic sequences) has been recently proposed in order to highlight several converging trends.

One main trend is the ongoing discovery that genomic repetitions are linked to many biological significant events and functions. Diseases (e.g. Huntington's disease) have been causally linked with abnormal expansion of certain repeating sequences in the human genome. Deletions or multiple copy duplications of genes (Copy Number Variations) are important in the aetiology of cancer, Alzheimer, and Parkinson diseases.

A second converging trend has been the emergence of many different models and algorithms for detecting non-obvious repeating patterns in strings with applications to in genomic data.

Borrowing methodologies from combinatorial pattern, matching, string algorithms, data structures, data mining and machine learning these new approaches break the limitations of the current approaches and offer a new way to design better trans-disciplinary research.

The articles collected in this book provides a glance into the rich emerging area of repeatome research, addressing some of its pressing challenges. We believe that these contributions are valuable resources for repeatome research and will stimulate further research from bioinformatic, statistical, and biological points of view.

Citation: Pellegrini, M., Magi, A., Iliopoulos, C. S., eds. (2016). Repetitive Structures in Biological Sequences: Algorithms and Applications. Lausanne: Frontiers Media. doi: 10.3389/978-2-88945-018-3

Table of Contents

05 ***Editorial: Repetitive Structures in Biological Sequences: Algorithms and Applications***

Marco Pellegrini, Alberto Magi and Costas S. Iliopoulos

Repeats in next generation sequencing data

07 ***Detection of Genomic Structural Variants from Next-Generation Sequencing Data***

Lorenzo Tattini, Romina D'Aurizio and Alberto Magi

15 ***The Challenge of Small-Scale Repeats for Indel Discovery***

Giuseppe Narzisi and Michael C. Schatz

20 ***G-CNV: A GPU-Based Tool for Preparing Data to Detect CNVs with Read-Depth Methods***

Andrea Manconi, Emanuele Manca, Marco Moscatelli, Matteo Gnocchi, Alessandro Orro, Giuliano Armano and Luciano Milanesi

Models for repeats

34 ***Statistical Approaches to Detecting and Analyzing Tandem Repeats in Genomic Sequences***

Maria Anisimova, Jūlija Pečerska and Elke Schaper

40 ***Accurate Prediction of the Statistics of Repetitions in Random Sequences: A Case Study in Archaea Genomes***

Mireille Régnier and Philippe Chassignet

50 ***Transposable Element Insertions in Long Intergenic Non-Coding RNA Genes***

Sivakumar Kannan, Diana Chernikova, Igor B. Rogozin, Eugenia Poliakov, David Managadze, Eugene V. Koonin and Luciano Milanesi

Algorithms and systems

59 ***SPECTRA: An Integrated Knowledge Base for Comparing Tissue and Tumor-Specific PPI Networks in Human***

Giovanni Micale, Alfredo Ferro, Alfredo Pulvirenti and Rosalba Giugno

74 ***Searching and Indexing Genomic Databases Via Kernelization***

Travis Gagie and Simon J. Puglisi

78 ***Tandem Repeats in Proteins: Prediction Algorithms and Biological Role***

Marco Pellegrini

86 ***Knowledge in the Investigation of A-to-I RNA Editing Signals***

Giovanni Nigita, Salvatore Alaimo, Alfredo Ferro, Rosalba Giugno and Alfredo Pulvirenti

frontiers in Bioengineering and Biotechnology

EDITORIAL
published: 04 August 2016
doi: 10.3389/fbioe.2016.00066

Editorial: Repetitive Structures in Biological Sequences: Algorithms and Applications

Marco Pellegrini[1,2], Alberto Magi[3] and Costas S. Iliopoulos[4]*

[1] Laboratory for Integrative Systems Medicine (LISM), Istituto di Informatica e Telematica, Consiglio Nazionale delle Ricerche, Pisa, Italy, [2] Laboratory for Integrative Systems Medicine (LISM), Istituto di Fisiologia Clinica, Consiglio Nazionale delle Ricerche, Pisa, Italy, [3] Department of Clinical and Experimental Medicine, University of Florence, Florence, Italy, [4] Department of Informatics, King's College London, London, UK

Keywords: repetitive structures, algorithms, tandem repeats, next generation sequencing, transposable elements

The Editorial on the Research Topic

Repetitive Structures in Biological Sequences: Algorithms and Applications

Repetitive structures in biological sequences are emerging as an active focus of research and the unifying concept of "repeatome" (the ensemble of knowledge associated with repeating structures in genomic/proteomic data) has been recently proposed in order to highlight several converging trends.

One main trend is the ongoing discovery that genomic repetitions are often linked to biologically significant events and functions. For example, an abnormal number of tandem repeating units both in coding and regulatory parts of the genome have been found to cause a series of diseases, including Huntington disease (MacDonald et al., 1993). There are indications of a link between tandem repeat expansion and certain forms of Amyotrophic Lateral Sclerosis (Renton et al., 2011).

Copy Number Variations and alterations (CNV/CNA), not necessarily in tandem, have been demonstrated to be one of the main sources of genomic variation in humans. These participate to phenotypic variation and adaptation and contribute to causing various diseases, including cancer, cardiovascular diseases, HIV acquisition and progression, autoimmune diseases, and Alzheimer's and Parkinson's diseases (Zhang et al., 2009).

Genome-wide identification of CNVs can be performed with array-based comparative genomic hybridization (aCGH), SNP arrays, and next generation sequencing (NGS). Although the experimental nature of these technologies is very different, the genomic profiles that they generate for CNVs identification are mathematically very similar. Several computational methods have been published in the last 10 years for segmenting these genomic profiles; however, much work still needs to be done, in particular for discovering CNV in low frequency subclones of cancer samples.

Intragenic tandem repeats polymorphisms may be involved in mis-regulations leading to protein toxicity through multiple pathways. Tandem repeats and CNV in Next Generation Sequencing (NGS) data are, however, difficult to detect and analyze, and devising effective detection algorithms is still a very open area of research (Treangen and Salzberg, 2012).

Repeating structures abound also in human proteins and they are a possible key to exploring sequence, structure, and function relationships. Inverted repeats are fingerprints of DNA hairpins and have been shown to contribute to chromosomal fragility in the human genome.

A second converging trend has been the emergence of many different models and algorithms for detecting non-obvious repeating patterns in strings with applications to genomic data collected in Hight Throughput assays (e.g., reads from NGS sequencing, or assembled genomes). A challenging aspect still to be explored is the full impact of evolutionary sequence divergence, and evolutionary

OPEN ACCESS

Edited and Reviewed by:
Richard D. Emes, University of Nottingham, UK

****Correspondence:***
Marco Pellegrini
marco.pellegrini@iit.cnr.it

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *27 June 2016*
Accepted: *25 July 2016*
Published: *04 August 2016*

Citation:
Pellegrini M, Magi A and Iliopoulos CS (2016) Editorial: Repetitive Structures in Biological Sequences: Algorithms and Applications. Front. Bioeng. Biotechnol. 4:66. doi: 10.3389/fbioe.2016.00066

selection over the origin and functional significance of repeating substructure. High divergence repetitions are harder to detect from the genomic background; however, they may give us more insight into the evolution of functional units in the genome. New modeling and algorithmic schemes are emerging to tackle these issues, focusing on the computational characterization of the individual entities involved in the repeatome. Borrowing methodologies from combinatorial pattern matching, string algorithms, data structures, data mining, machine learning, probability, and statistics, these new approaches overcome the limitations of the current approaches and offer an example of trans-disciplinary research.

In this Research Topic, we have collected four original research articles and six reviews spanning the full scope of the Topic.

NGS data are a common theme of three of the contributions. Tattini et al. give an overview of the challenges and the several approaches in the literature for detecting structural variants in the human genome using whole genome and whole exome sequencing data, pointing at major advantages and drawbacks of each approach. Narzisi and Schatz analyze the impact of small-scale repetitive sequences, in particular near-tandem repeats, on the discovery of DNA structural variations with the micro-assembly approach. Manconi et al. describe a GPU-based efficient pipeline for filtering reads obtained from Next Generation sequencing, in conjunction with read depth CNV detection methods.

Repetitive sequences both within a single genome and across multiple genomes cause several problems in building effective genomic databases that support efficient data mining on genomic data. Gagie and Puglisi survey advances in algorithmic techniques for taking advantage of repetitive sequences in indexing and searching genomic databases.

The study of tandem repeats in DNA sequences has been a very active area of research in the last decade. Anisimova et al. survey both computational and statistical approaches for TR detection and their application to sequence alignment, phylogenetic analysis, and benchmarking. Régnier and Chassignet develop new models for predicting the statistics of repetitions and show that the proposed model fits nicely data from a biological case study. Pellegrini gives an overview on the multi-faceted aspects of research on protein tandem repeats (PTR), including prediction algorithms, databases, early classification efforts, mechanisms of PTR formation and evolution, and synthetic PTR design, embracing both sequence and 3-dimensional structural aspects.

Transposable Elements (TE) are DNA subsequences that can replicate themselves via a series of biochemical mechanisms and are particularly abundant in mammalian genomes. Kannan et al. investigate the correlations between TE and long intergenic non-coding RNA genes (lincRNA), corroborating the hypothesis that TE have substantially contributed to the origin, evolution, and functional diversification of lincRNA genes.

Nigita et al. investigate computational aspects of RNA editing, which is a post-transcriptional alteration of expressed RNA sequences eventually affecting protein and ncRNA structure and function. This phenomenon is mostly associated with repetitive regions of RNA sequences.

Besides sequence and 3-dimensional structures, biological data are increasingly available in graphical form. Micale et al. describe a web-based tool (SPECTRA) to build and analyze PPI networks that capture tumor and tissue-specific interactions via integration of a variety of heterogeneous data repositories, thus allowing the comparative exploration of similarities/differences in tissue-specific processes.

This series of papers provides a glance into the rich emerging area of repeatome research, addressing some of its pressing challenges. We believe that these contributions are valuable resources for repeatome research and will stimulate further research from bioinformatic, statistical, and biological points of view.

AUTHOR CONTRIBUTIONS

The authors contributed equally to this work.

FUNDING

Work supported by Italian Ministry of Education, Universities and Research (MIUR) and by the National Research Council of Italy (CNR) within the Flagship Project InterOmics PB.P05.

REFERENCES

MacDonald, M. E., Ambrose, C. M., Duyao, M. P., Myers, R. H., Lin, C., Srinidhi, L., et al. (1993). A novel gene containing a trinucleotide repeat that is expanded and unstable on Huntington's disease chromosomes. *Cell* 72, 971–983. doi:10.1016/0092-8674(93)90585-E

Renton, A., Majounie, E., Waite, A., Simon-Sanchez, J., Rollinson, S., Gibbs, J., et al. (2011). A hexanucleotide repeat expansion in C9ORF72 is the cause of chromosome 9p21-linked ALS-FTD. *Neuron* 72, 257–268. doi:10.1016/j.neuron.2011.09.010

Treangen, T. J., and Salzberg, S. L. (2012). Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.* 13, 36–46. doi:10.1038/nrg3117

Zhang, F., Gu, W., Hurles, M. E., and Lupski, J. R. (2009). Copy number variation in human health, disease, and evolution. *Annu. Rev. Genomics Hum. Genet.* 10, 451. doi:10.1146/annurev.genom.9.081307.164217

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

MINI REVIEW
published: 25 June 2015
doi: 10.3389/fbioe.2015.00092

Detection of genomic structural variants from next-generation sequencing data

Lorenzo Tattini[1]*, *Romina D'Aurizio*[2] and *Alberto Magi*[3]

[1] Department of Neurosciences, Psychology, Pharmacology and Child Health, University of Florence, Florence, Italy, [2] Laboratory of Integrative Systems Medicine (LISM), Institute of Informatics and Telematics and Institute of Clinical Physiology, National Research Council, Pisa, Italy, [3] Department of Clinical and Experimental Medicine, University of Florence, Florence, Italy

Structural variants are genomic rearrangements larger than 50 bp accounting for around 1% of the variation among human genomes. They impact on phenotypic diversity and play a role in various diseases including neurological/neurocognitive disorders and cancer development and progression. Dissecting structural variants from next-generation sequencing data presents several challenges and a number of approaches have been proposed in the literature. In this mini review, we describe and summarize the latest tools – and their underlying algorithms – designed for the analysis of whole-genome sequencing, whole-exome sequencing, custom captures, and amplicon sequencing data, pointing out the major advantages/drawbacks. We also report a summary of the most recent applications of third-generation sequencing platforms. This assessment provides a guided indication – with particular emphasis on human genetics and copy number variants – for researchers involved in the investigation of these genomic events.

Keywords: next generation sequencing, structural variants, copy number variants, statistical methods, whole-exome sequencing, whole-genome sequencing, amplicon sequencing

OPEN ACCESS

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Christian Cole, University of Dundee, UK
Alexander Schönhuth, Centrum Wiskunde & Informatica, Netherlands

***Correspondence:**
Lorenzo Tattini, Department of Neurosciences, Psychology, Pharmacology and Child Health, University of Florence, Viale Pieraccini, 6, Florence 50139, Italy
lorenzo.tattini@unifi.it

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *10 December 2014*
Accepted: *10 June 2015*
Published: *25 June 2015*

Citation:
Tattini L, D'Aurizio R and Magi A (2015) Detection of genomic structural variants from next-generation sequencing data. Front. Bioeng. Biotechnol. 3:92. doi: 10.3389/fbioe.2015.00092

Introduction

Structural variants (SVs) are genomic rearrangements affecting more then 50 bp. The average SV size detected by the 1000 Genomes Project is 8 kbp (1000 Genomes Project Consortium et al., 2010), whereas a study based on tiling CGH array (Conrad et al., 2010) reports a four times larger value. SVs comprise balanced as well as unbalanced events, namely, variants altering the total number of base pairs in a genome. Thus, SVs include deletions, insertions, inversions, mobile-element transpositions, translocations, tandem repeats, and copy number variants (CNVs).

Several databases – e.g., the Database of Genomic Variants archive which reports structural variation identified in healthy control samples (DGVa[1]) – have been created for the collection of SVs data (Lappalainen et al., 2013). Public data resources have been developed with the purpose of supporting the interpretation of clinically relevant variants, e.g., dbVar[2], or collecting known disease genes (OMIM[3]) hit by SVs.

Structural variants account for 1.2% of the variation among human genomes while single nucleotide polymorphisms (SNPs) represent 0.1% (Pang et al., 2010). Notably, unbalanced events

[1] http://www.ebi.ac.uk/dgva
[2] http://www.ncbi.nlm.nih.gov/dbvar
[3] http://www.omim.org

provide 99.8% of the entries reported in dbVar (Lin et al., 2014). CNVs may result in benign polymorphic variations or clinical phenotypes due to gene dosage alteration or gene disruption (Zhang et al., 2009). Though the impact of SVs in human genomics was first recognized by their presence in healthy individuals (Zhao et al., 2013), two models account for their association to human disease. Rare large events (<1%, hundreds kbp) have been related to neurological and neurocognitive disorders (Sebat et al., 2007; Girirajan et al., 2013), whereas multicopy gene families, which are commonly copy number variable, contribute to disease susceptibility.

Next-generation sequencing technologies (NGS) have been revolutionizing genome research [for a survey of NGS tools from quality check to variant annotation and visualization, see Pabinger et al. (2014)] as well as the study of CNVs (Duan et al., 2013; Zhao et al., 2013; Samarakoon et al., 2014; Tan et al., 2014; Alkodsi et al., 2015; Kadalayil et al., 2015) and SVs on the whole (Alkan et al., 2011a), replacing microarrays as the leading platform for the investigation of genomic rearrangement (Pinkel et al., 1998; Snijders et al., 2001; Iafrate et al., 2004; Sebat et al., 2007). NGS platforms are based on various implementations of cyclic-array sequencing (Shendure and Ji, 2008; Shendure et al., 2011). They allow for the sequencing of millions of short (few hundreds bp) DNA fragments (reads) simultaneously and may process a whole human genome in three days at 500-fold less cost than previous methods (Voelkerding et al., 2009; Metzker, 2010).

The 1000 Genomes Project applied methods based on all of the four approaches available for the detection of SVs, reporting false discovery rates ranging from 10 to 89%, remarkable differences in terms of genomic regions discovered, size range, and breakpoint precision (Mills et al., 2011; Teo et al., 2012).

Overview of the Approaches

Four strategies for the detection of SV signatures that are diagnostic of different rearrangements have been reported in the literature (**Figure 1**; **Table 1**).

Read-pair (RP) methods are based on the evaluation of the span and orientation of paired-end reads. Discordant pairs in which the mapping span and/or orientation of the read pairs are inconsistent with the expected insert size are collected. Several classes of SVs can be investigated by means of this approach. Read pairs mapping too far apart are associated to deletions while those found closer than expected are indicative of insertions. Furthermore, orientation inconsistencies can represent inversions and a specific class of tandem duplications.

Read-depth (or read count, RC) approaches assume a random (Poisson or modified Poisson) distribution in mapping depth and investigate the divergence from this distribution to highlight duplications and deletions (Magi et al., 2012). Sequencing of duplicated/amplified regions results in higher read depth while deleted regions show reduced read depth when compared to normal (e.g., diploid) regions.

Split-read (SR) methods allow for the detection of SVs with single base-pair resolution. The presence of a SV breakpoint is investigated on the basis of a split sequence-read signature breaking

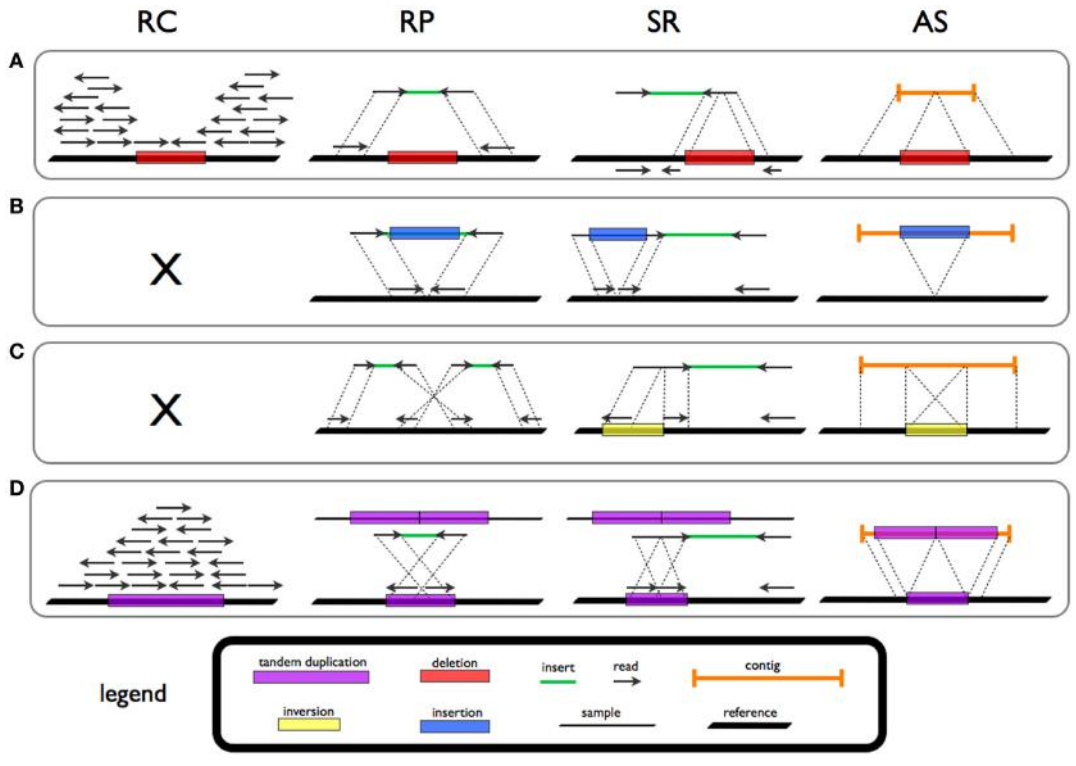


FIGURE 1 | Signatures and patterns of SVs for deletion (A), novel sequence insertion (B), inversion (C), and tandem duplication (D) in read count (RC), read-pair (RP), split-read (SR), and *de novo* assembly (AS) methods.

TABLE 1 | A non-exhaustive summary of the tools/algorithms for the investigation of SVs, their input data (WGS, whole-genome sequencing; WES, whole-exome sequencing; CC, custom capture; AMS, amplicon sequencing), and their underling approach.

Tool/algorithm	Input data	Method	Reference
EXCAVATOR	WES	RC	Magi et al. (2013)
ExomeCNV	WES	RC	Sathirapongsasuti et al. (2011)
CoNIFER	WES	RC	Krumm et al. (2012)
CODEX	WES	RC	Jiang et al. (2015)
XHMM	WES	RC	Fromer et al. (2012)
–	WES/CC	RC	Bansal et al. (2014)
ONCOCNV	AMS	RC	Boeva et al. (2014)
CNVnator	WGS	RC	Abyzov et al. (2011)
SegSeq	WGS	RC	Chiang et al. (2009)
CNAnorm	WGS	RC	Gusnanto et al. (2012)
CNAseg	WGS	RC	Ivakhno et al. (2010)
rSW-seq	WGS	RC	Kim et al. (2010)
cn.MOPS	WGS	RC	Klambauer et al. (2012)
JointSLM	WGS	RC	Magi et al. (2011)
ReadDepth	WGS	RC	Miller et al. (2011)
BIC-seq	WGS	RC	Xi et al. (2011)
PSCC	WGS	RC	Li et al. (2014)
CNV-seq	WGS	RC	Xie and Tammi (2009)
CLEVER	WGS	RP	Marschall et al. (2012)
BreakDancer	WGS	RP	Chen et al. (2009)
VariationHunter	WGS	RP	Hormozdiari et al. (2011)
PEMer	WGS	RP	Korbel et al. (2009)
MoDIL	WGS	RP	Lee et al. (2009)
Gustaf	WGS	SR	Trappe et al. (2014)
Socrates	WGS	SR	Schröder et al. (2014)
Splitread	WGS/WES	SR	Karakoc et al. (2012)
Cortex	WGS	AS	Iqbal et al. (2012)
Magnolya	WGS	AS	Nijkamp et al. (2012)
Tea	WGS	DC	Lee et al. (2012)
RetroSeq	WGS	DC	Keane et al. (2013)
Tangram	WGS	DC	Wu et al. (2014)
Mobster	WGS/WES	DC	Keane et al. (2013)
SVDetect	WGS	RC + RP	Zeitouni et al. (2010)
GASVpro	WGS	RC + RP	Sindi et al. (2012)
CNVer	WGS	RC + RP	Medvedev et al. (2010)
inGAP-sv	WGS	RC + RP	Qi and Zhao (2011)
Pindel	WGS	RP + SR	Ye et al. (2009)
LUMPY	WGS	RP + SR	Layer et al. (2014)
DELLY	WGS	RP + SR	Rausch et al. (2012)
PRISM	WGS	RP + SR	Jiang et al. (2012)
MATE-CLEVER	WGS	RP + SR	Marschall et al. (2013)
NovelSeq	WGS	RP + AS	Hajirasouliha et al. (2010)
HYDRA	WGS	RP + AS	Quinlan et al. (2010)
CREST	WGS	SR + AS	Wang et al. (2011)
SVseq	WGS	RC + SR	Zhang and Wu (2011)
SoftSearch	WGS/WES/CC	RP + SR	Hart et al. (2013)
Genome STRiP	WGS	RP + SR + RC	Handsaker et al. (2011)

Methods designed using WGS data can, in principle, be used with WES data, though with limitations due to the intrinsic sparseness of WES data.

the alignment to the reference. A gap in the read is a marker of a deletion while stretches in the reference reflect insertions.

Theoretically, all forms of structural variation could be investigated by means of *de novo* assembly (AS) methods. *De novo* assembly refers to merging and ordering short fragments to reassemble the original sequence from which the short fragments were sampled (Earl et al., 2011). NGS data intrinsic characteristics, such as (short) read length, limit the use of AS approaches for variant investigation.

Moreover, a specific class of SV, mobile elements (ME) insertions, can be investigated exploiting discordant and clipped (DC) read information.

Read Count Methods

Read count is suitable for the investigation of CNVs. RC methods comprise four steps: RC data preparation, data normalization, SV regions identification, and copy number estimation. Reads mapping to windows/bins of fixed size are counted (Yoon et al., 2009; Magi et al., 2011) and the results are normalized for the mitigation of local GC content and mappability effects.

The correlation between local GC content and read coverage has been detected through the analysis of data from several platforms (Harismendy et al., 2009). Mappability bias is due to repetitive regions within the human genome (Miller et al., 2011).

A segmentation step is necessary to split RC signal into segments characterized by a constant DNA copy number. Algorithms conceived for aCGH data such as the circular binary segmentation (CBS) algorithm (Campbell et al., 2008; Miller et al., 2011) and those based on hidden Markov models (HMM) (Magi et al., 2010) are used with this scope.

Copy number estimation can be tackled by means of two strategies. Both assume that the sequencing process is uniform. Thus, the number of reads mapping to a genomic region is expected to be proportional to the number of times the regions appears in the DNA sample. Three methods (Campbell et al., 2008; Yoon et al., 2009; Magi et al., 2011) estimate DNA copy number of all the detected regions rounding the median RCs (normalized to copy number 2) to the nearest integer, while CNVnator (Abyzov et al., 2011) uses RC signal normalized to the genomic average for the regions of the same length.

A considerable number of methods for the detection of CNV in whole-genome sequencing (WGS) data have been reported in the literature, including CNVnator, CNAnorm, CNAseg, rSW-seq, cn.MOPS, JointSLM, ReadDepth, and BIC-seq (Ivakhno et al., 2010; Kim et al., 2010; Abyzov et al., 2011; Magi et al., 2011; Miller et al., 2011; Xi et al., 2011; Gusnanto et al., 2012; Klambauer et al., 2012). Recently, PSCC (Li et al., 2014) has been compared with SegSeq (Chiang et al., 2009) and ReadDepth (Miller et al., 2011).

CNV Detection from Whole-Exome Data

Due to the costs associated to WGS, the investigation of CNVs using whole-exome sequencing (WES) data is definitely an attractive perspective. Nevertheless, the sparse nature of the target and the non-uniform read-depth among captured regions make CNV detection from WES data awkward with respect to WGS [in particular, regarding the segmentation step as reported in Magi et al. (2013)].

Several tools have been reported in the literature for this purpose including ExomeCNV (Sathirapongsasuti et al., 2011), CoNIFER (Krumm et al., 2012), CNV-seq (Xie and Tammi, 2009), XHMM (Fromer et al., 2012), and recently EXCAVATOR (Magi et al., 2013) and CODEX (Jiang et al., 2015). Notably, the method developed by Bansal and co-workers (Bansal et al., 2014) allows for the analysis of NGS data generated from small subsets of the exome, namely custom capture (CC) data.

Amplicon Sequencing Data

Amplicon sequencing (AMS) techniques have been reported in the literature in particular for clinical applications (Desai and Jere, 2012; Beadling et al., 2013).

Amplicon sequencing data show different biases in respect of WES data (Boeva et al., 2014). Data normalization can be less effective due to the limited number of target regions. Furthermore, protocols involved in the preparation of amplicon libraries result in high depth of coverage at the expense of coverage homogeneity.

The first method designed for the investigation of CNV from AMS data is ONCOCNV. Duplicate sequences are not removed, while RC is performed assigning "each read to only one amplicon region, the one with which the read alignment has the maximum overlap" (Boeva et al., 2014).

Data are then normalized with respect to library size assuming a similar efficiency of PCR amplification for all the targeted regions. GC content and amplicon length biases are corrected by means of a local polynomial regression fitting. Principal component analysis (PCA) is employed to construct a baseline reflecting the technological bias in control samples. The baseline is calculated by means of the first three principal components (calculated from control samples data). In order to define a significant threshold to call a copy number change, the standard deviation of the normalized RCs for each amplicon region is calculated.

This procedure is applied to data from test samples keeping the residuals of the linear regression of normalized RCs over the baseline calculated for the control samples.

Segmentation of the resulting signal profile is performed with CBS method (Venkatraman and Olshen, 2007). A segmentation and clustering approach (SCA) is used to define the copy number state (neutral, gain, or loss) of the segmented regions.

Read-Pair Algorithms

As already mentioned, RP methods, as well as SR approaches, are suitable for the detection of several classes of SV including insertions of novel sequences and inversions. Notably, RP algorithms cannot detect the signatures of novel sequence insertions larger than the average insert size. Several tools based on the detection of SV signatures from *clusters* of read-pairs have been reported in the literature including BreakDancer, VariationHunter, PEMer, and GASV (Chen et al., 2009; Hormozdiari et al., 2009, 2011; Korbel et al., 2009; Sindi et al., 2009). Remarkably, PEMer can be exploited for the identification of linked insertions (Medvedev et al., 2010).

Clusters can be defined according to two strategies. The standard clustering strategy relies on two parameters: the minimum number of pairs with similar signature and the maximum value of the mean insert size standard deviation for a pair to be considered concordant. The maximum standard deviation value is fixed and events spanning the same locus, resulting in a small value of the insert size standard deviation, may be missed.

Distribution-based approaches, e.g., MoDIL (Lee et al., 2009), exploit the local distribution of all the mappings spanning a particular location on the genome. A read cluster is generated when the local distribution is shifted in respect to the typical insert size distribution. This approach allows for the detection of smaller events (e.g., compared with VariationHunter). The presence of two superimposed insert size distributions can be also detected, thus allowing for the discrimination of homozygous and heterozygous variants.

In the first implementations of the approach, e.g., BreakDancer (Chen et al., 2009), reads with multiple mappings were discarded. Thus, repetitive regions of the genome (including segmental duplications and copy-number amplifications) could not be investigated. Notably, BreakDancer allows for the identification of inter- and intra-chromosomal translocations. Tools such as MoDIL and VariationHunter or, more recently, CLEVER (Marschall et al., 2012) deal with multiple mapping reads [aligned, for instance, with mrFast (Alkan et al., 2009), Mosaik (Lee et al., 2014), BWA (Li and Durbin, 2010), or Bowtie (Langmead et al., 2009)]. CLEVER uses an insert size-based approach to build a graph with all reads and evaluates SV from maximal cliques. It is particularly well-tuned for the investigation of insertions and deletions of 50–100 bp.

Split-Read Approaches

Though SR methods were conceived for Sanger sequencing reads (Mills et al., 2006), algorithms such as Pindel, Splitread, and Gustaf (Ye et al., 2009; Karakoc et al., 2012; Trappe et al., 2014) use paired-end NGS reads to identify SVs (or indel) events. SR approaches take advantage of one-end anchored reads, namely those pairs in which "one end is anchored to the reference genome and the other end maps imprecisely owing to the presence of an underlying structural variant or indel breakpoint" (Karakoc et al., 2012). SR-based tools can be applied solely to unique reference regions.

Pindel uses pattern growth for optimal matching in target regions, exploiting reads mapped with SSAHA2 [Sequence Search and Alignment by Hashing Algorithm, Ning et al. (2001)], BWA, or Mosaik. It must be stressed that the latest version of Pindel integrates RP to the SR information (Lin et al., 2014). Splitread searches for clusters of split reads using balanced splits as seeds. Splitread can detect, at least in theory, deletions without size limitation, while for insertions the size spectrum depends on the sequencing library. Insertions shorter than the read length can be accurately identified but larger insertions can only be approximately characterized within the insert size (Karakoc et al., 2012). Splitread is suitable for WGS/WES reads aligned using mrsFAST (Hach et al., 2010) to discover indels, SVs, *de novo* events, and pseudogenes.

Recently, Socrates (a SR method designed for cancer genomics) was compared to several tools (Schröder et al., 2014), including BreakDancer, CLEVER, CREST (Wang et al., 2011), DELLY (Rausch et al., 2012), Pindel, and PRISM (Jiang et al., 2012).

Assembly Based Tools

De novo assembly allows – at least in principle – for the detection of all the forms of structural variation but the application of this approach is still challenging due to the limited length of NGS reads (Alkan et al., 2011a; O'Rawe et al., 2015).

AS methods were first exploited for Sanger sequencing data (characterized by read length between 300 and 1000 bp). The original *string graph approach* has been extended to de Bruijn graphs. The Assemblathon competition (Earl et al., 2011) produced a detailed comparison among *de novo* assemblers, including

Phusion2 (Mullikin and Ning, 2003), SGA (Simpson and Durbin, 2010, 2012), Quake (Kelley et al., 2010), the first implementation of SOAPdenovo (Li et al., 2010; Luo et al., 2012), and ALLPATHS-LG (Gnerre et al., 2011), based on simulated data.

Two AS based callers have been reported in the literature for the investigation of SVs. Magnolya (Nijkamp et al., 2012) uses a Poisson mixture model (PMM) for CNV detection from contigs co-assembled from NGS sequencing data. The authors use an overlap-layout-consensus assembler to generate a contig string graph. Contig string graphs are characterized by nodes representing reads and edges representing an overlap. The final form of the graph is produced by transitive reduction – which removes redundant edges – and by unitigging (i.e., collapsing simple paths without branches) (Myers, 2005). In the resulting contig string graph, each node represents a collapsed set of reads called *contig*. Finally, the PMM approach for modeling read count is introduced to estimate the copy number of a contig. Once the model has been corrected for the presence of repetitive regions in the genome and prior knowledge on ploidy has been included, the model with the optimal number of Poisson distributions is selected by means of the lowest Bayesian information criterion. Integer copy numbers can be thus inferred by maximum *a posteriori* estimation. Remarkably, the method can be applied when no reference is available but – as already stressed – it is limited by the short read length typical of NGS platforms.

Cortex uses colored de Bruijn graphs with colors of both edges and nodes representing different samples and, possibly, reference sequences or known variants to assemble NGS reads. "The graph consists of a set of nodes representing words of length k (k-mers). Directed edges join k-mers seen consecutively in the input" (Iqbal et al., 2012). The package includes four algorithms for variant discovery. For example, the *bubble calling* algorithm may be exploited for the detection of variant bubbles in a colored de Bruijn graph from a single diploid individual. It must be stressed that using a reference genome aids the identification of variants while it is indispensable for the investigation of homozygous variant sites. Nevertheless, the sensitivity of the method decreases with the size of the variant. The tool has been extensively tested on human data.

Combined Methods

None of the aforementioned approaches is capable of capturing the full spectrum of SV events with high sensitivity and specificity. RC methods can accurately predict absolute copy numbers but the breakpoint resolution is often inadequate and events such as inversions and novel sequence insertions cannot be detected. On the other hand, RP and SR approaches show low sensitivity in repetitive regions. Several packages combining different approaches for the investigation of SVs have been reported.

Combining RC for the detection of large events and RP for accurate identification of breakpoints can reduce the number of false positive calls [SVDetect (Zeitouni et al., 2010), CNVer (Medvedev et al., 2010), GASVPro (Sindi et al., 2012), and inGAP-sv (Qi and Zhao, 2011)]. Genome STRiP (Handsaker et al., 2011) exploits RP, RC, SR, and population-scale patterns to detect genome structural polymorphisms.

Packages implementing RP and (local) AS have been also reported [NovelSeq (Hajirasouliha et al., 2010), HYDRA (Quinlan et al., 2010)] as well as tools exploiting SR and RC/RP such as SVseq, MATE-CLEVER, and PRISM (Zhang and Wu, 2011; Jiang et al., 2012; Marschall et al., 2013). PRISM was tested on simulated data and compared with Pindel, SVseq, Splitread, and CREST. Notably, DELLY is suitable for detecting copy-number variable deletion and tandem duplication events as well as balanced rearrangements such as inversions or reciprocal translocations (Rausch et al., 2012), while SoftSearch (Hart et al., 2013) is designed for WGS, WES, and CC data. Recently, LUMPY has been shown to be "especially pronounced when evidence is scarce, either due to low coverage data or low variant allele frequency" (Layer et al., 2014). LUMPY is designed to integrate signals rather then refining primary signal with a secondary one. Furthermore, the tool combines different types of evidence from multiple samples.

Detection of Mobile Elements

Mobile elements are repetitive DNA sequences that can change position within the genome (Lander et al., 2001). Due to this intrinsic characteristic, their detection is challenging. Latest estimates suggest that more than half of the human genome is repetitive or repeat-derived (de Koning et al., 2011). Though the DC approach can be ascribed to RP and SR methods, "the mates of the anchoring reads are then mapped to a custom but configurable library of known active ME consensus sequences" (Thung et al., 2014).

Among WGS tools, Tangram (Wu et al., 2014), a tool developed using Mosaik (Lee et al., 2014) alignments (though it may use alignments produced by other mappers), Next-Generation VariationHunter (Hormozdiari et al., 2010), Tea (Lee et al., 2012), RetroSeq eKeane:2013kq, and Mobster (Thung et al., 2014) have been reported in the literature.

Conclusion

Overall, all the approaches discussed are fairly limited with respect to repeated regions of the reference genome (Alkan et al., 2009, 2011b). The complete range of structural DNA variation cannot be investigated with a single tool (Mills et al., 2011), though combined methods may aid the discovery of SV. Three pipelines integrating different tools exploiting WGS data have been reported in the literature (Wong et al., 2010; Lam et al., 2012; Mimori et al., 2013). WES data can be exploited for the investigation of SVs by means of RC, SR, and RP methods – though with limitations due to the intrinsic sparseness of exomic data.

Each method for the detection of SVs shows advantages/drawback. RC methods are particularly well-suited for the investigation of a particular class of SV, namely CNV. Notably, RC can be used to predict absolute copy number. A major drawback of RC tools is the poor breakpoint resolution. Furthermore, they cannot distinguish tandem from interspersed duplications. SR algorithms can accurately predict SV breakpoint (down to single-base resolution) as well as AS methods. Finally, the RP and SR approaches can be applied for the investigation of the widest range of SV classes (i.e., deletions, inversions, novel sequence insertions, tandem duplications), though both cannot be exploited for the calculation of absolute copy number.

The advent of third-generation sequencing (TGS) technology may contribute to overcome these issues (Schadt et al., 2010; Niedringhaus et al., 2011; Pareek et al., 2011; Venkatesan and Bashir, 2011). TGS single-end reads, characterized by read length up to thousands base pairs, may boost AS methods and the application of mapping algorithms allowing for split alignment such as BWA (Li and Durbin, 2010), LAST (Kiełbasa et al., 2011) and BLASR (Chaisson and Tesler, 2012). Though TGS platforms rely on different chemistry, reads produced by platforms, such as PacBio RS (Kim et al., 2014) and Oxford Nanopore MinION (Bayley, 2015), show similar read length and base-calling accuracy (~85%) (Quail et al., 2012; Quick et al., 2014; Ashton et al., 2015; Chaisson et al., 2015). Recent works have demonstrated that these technologies allow for the investigation of complex repetitive regions of the human genome (Chaisson et al., 2015) as well as the structure of complex antibiotic resistance islands in *Salmonella typhi* (Ashton et al., 2015) and tandem repeats in human bacterial artificial chromosome (Jain et al., 2015).

References

1000 Genomes Project Consortium, Abecasis, G. R., Altshuler, D., Auton, A., Brooks, L. D., Durbin, R. M., et al. (2010). A map of human genome variation from population-scale sequencing. *Nature* 467, 1061–1073. doi:10.1038/nature09534

Abyzov, A., Urban, A. E., Snyder, M., and Gerstein, M. (2011). Cnvnator: an approach to discover, genotype, and characterize typical and atypical cnvs from family and population genome sequencing. *Genome Res.* 21, 974–984. doi:10.1101/gr.114876.110

Alkan, C., Coe, B. P., and Eichler, E. E. (2011a). Genome structural variation discovery and genotyping. *Nat. Rev. Genet.* 12, 363–376. doi:10.1038/nrg2958

Alkan, C., Sajjadian, S., and Eichler, E. E. (2011b). Limitations of next-generation genome sequence assembly. *Nat. Methods* 8, 61–65. doi:10.1038/nmeth.1527

Alkan, C., Kidd, J. M., Marques-Bonet, T., Aksay, G., Antonacci, F., Hormozdiari, F., et al. (2009). Personalized copy number and segmental duplication maps using next-generation sequencing. *Nat. Genet.* 41, 1061–1067. doi:10.1038/ng.437

Alkodsi, A., Louhimo, R., and Hautaniemi, S. (2015). Comparative analysis of methods for identifying somatic copy number alterations from deep sequencing data. *Brief. Bioinform.* 16, 242–254. doi:10.1093/bib/bbu004

Ashton, P. M., Nair, S., Dallman, T., Rubino, S., Rabsch, W., Mwaigwisya, S., et al. (2015). MinION nanopore sequencing identifies the position and structure of a bacterial antibiotic resistance island. *Nat. Biotechnol.* 33, 296–300. doi:10.1038/nbt.3103

Bansal, V., Dorn, C., Grunert, M., Klaassen, S., Hetzer, R., Berger, F., et al. (2014). Outlier-based identification of copy number variations using targeted resequencing in a small cohort of patients with tetralogy of fallot. *PLoS ONE* 9:e85375. doi:10.1371/journal.pone.0085375

Bayley, H. (2015). Nanopore sequencing: from imagination to reality. *Clin. Chem.* 61, 25–31. doi:10.1373/clinchem.2014.223016

Beadling, C., Neff, T. L., Heinrich, M. C., Rhodes, K., Thornton, M., Leamon, J., et al. (2013). Combining highly multiplexed pcr with semiconductor-based sequencing for rapid cancer genotyping. *J. Mol. Diagn.* 15, 171–176. doi:10.1016/j.jmoldx.2012.09.003

Boeva, V., Popova, T., Lienard, M., Toffoli, S., Kamal, M., Le Tourneau, C., et al. (2014). Multi-factor data normalization enables the detection of copy number aberrations in amplicon sequencing data. *Bioinformatics* 30, 3443–3450. doi:10.1093/bioinformatics/btu436

Campbell, P. J., Stephens, P. J., Pleasance, E. D., O'Meara, S., Li, H., Santarius, T., et al. (2008). Identification of somatically acquired rearrangements in cancer using genome-wide massively parallel paired-end sequencing. *Nat. Genet.* 40, 722–729. doi:10.1038/ng.128

Chaisson, M. J., and Tesler, G. (2012). Mapping single molecule sequencing reads using basic local alignment with successive refinement (blasr): application and theory. *BMC Bioinformatics* 13:238. doi:10.1186/1471-2105-13-238

Chaisson, M. J. P., Huddleston, J., Dennis, M. Y., Sudmant, P. H., Malig, M., Hormozdiari, F., et al. (2015). Resolving the complexity of the human genome using single-molecule sequencing. *Nature* 517, 608–611. doi:10.1038/nature13907

Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., Pohl, C. S., et al. (2009). Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods* 6, 677–681. doi:10.1038/nmeth.1363

Chiang, D. Y., Getz, G., Jaffe, D. B., O'Kelly, M. J. T., Zhao, X., Carter, S. L., et al. (2009). High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat. Methods* 6, 99–103. doi:10.1038/nmeth.1276

Conrad, D. F., Pinto, D., Redon, R., Feuk, L., Gokcumen, O., Zhang, Y., et al. (2010). Origins and functional impact of copy number variation in the human genome. *Nature* 464, 704–712. doi:10.1038/nature08516

de Koning, A. P. J., Gu, W., Castoe, T. A., Batzer, M. A., and Pollock, D. D. (2011). Repetitive elements may comprise over two-thirds of the human genome. *PLoS Genet.* 7:e1002384. doi:10.1371/journal.pgen.1002384

Desai, A. N., and Jere, A. (2012). Next-generation sequencing: ready for the clinics? *Clin. Genet.* 81, 503–510. doi:10.1111/j.1399-0004.2012.01865.x

Duan, J., Zhang, J.-G., Deng, H.-W., and Wang, Y.-P. (2013). Comparative studies of copy number variation detection methods for next-generation sequencing technologies. *PLoS ONE* 8:e59128. doi:10.1371/journal.pone.0059128

Earl, D., Bradnam, K., St John, J., Darling, A., Lin, D., Fass, J., et al. (2011). Assemblathon 1: a competitive assessment of de novo short read assembly methods. *Genome Res.* 21, 2224–2241. doi:10.1101/gr.126599.111

Fromer, M., Moran, J. L., Chambert, K., Banks, E., Bergen, S. E., Ruderfer, D. M., et al. (2012). Discovery and statistical genotyping of copy-number variation from whole-exome sequencing depth. *Am. J. Hum. Genet.* 91, 597–607. doi:10.1016/j.ajhg.2012.08.005

Girirajan, S., Dennis, M. Y., Baker, C., Malig, M., Coe, B. P., Campbell, C. D., et al. (2013). Refinement and discovery of new hotspots of copy-number variation associated with autism spectrum disorder. *Am. J. Hum. Genet.* 92, 221–237. doi:10.1016/j.ajhg.2012.12.016

Gnerre, S., Maccallum, I., Przybylski, D., Ribeiro, F. J., Burton, J. N., Walker, B. J., et al. (2011). High-quality draft assemblies of mammalian genomes from massively parallel sequence data. *Proc. Natl. Acad. Sci. U.S.A.* 108, 1513–1518. doi:10.1073/pnas.1017351108

Gusnanto, A., Wood, H. M., Pawitan, Y., Rabbitts, P., and Berri, S. (2012). Correcting for cancer genome size and tumour cell content enables better estimation of copy number alterations from next-generation sequence data. *Bioinformatics* 28, 40–47. doi:10.1093/bioinformatics/btr593

Hach, F., Hormozdiari, F., Alkan, C., Hormozdiari, F., Birol, I., Eichler, E. E., et al. (2010). mrsfast: a cache-oblivious algorithm for short-read mapping. *Nat. Methods* 7, 576–577. doi:10.1038/nmeth0810-576

Hajirasouliha, I., Hormozdiari, F., Alkan, C., Kidd, J. M., Birol, I., Eichler, E. E., et al. (2010). Detection and characterization of novel sequence insertions using paired-end next-generation sequencing. *Bioinformatics* 26, 1277–1283. doi:10.1093/bioinformatics/btq152

Handsaker, R. E., Korn, J. M., Nemesh, J., and McCarroll, S. A. (2011). Discovery and genotyping of genome structural polymorphism by sequencing on a population scale. *Nat. Genet.* 43, 269–276. doi:10.1038/ng.768

Harismendy, O., Ng, P. C., Strausberg, R. L., Wang, X., Stockwell, T. B., Beeson, K. Y., et al. (2009). Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol.* 10, R32. doi:10.1186/gb-2009-10-3-r32

Hart, S. N., Sarangi, V., Moore, R., Baheti, S., Bhavsar, J. D., Couch, F. J., et al. (2013). Softsearch: integration of multiple sequence features to identify breakpoints of structural variations. *PLoS ONE* 8:e83356. doi:10.1371/journal.pone.0083356

Hormozdiari, F., Alkan, C., Eichler, E. E., and Sahinalp, S. C. (2009). Combinatorial algorithms for structural variation detection in high-throughput sequenced genomes. *Genome Res.* 19, 1270–1278. doi:10.1101/gr.088633.108

Hormozdiari, F., Hajirasouliha, I., Dao, P., Hach, F., Yorukoglu, D., Alkan, C., et al. (2010). Next-generation variationhunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics* 26, i350–i357. doi:10.1093/bioinformatics/btq216

Hormozdiari, F., Hajirasouliha, I., McPherson, A., Eichler, E. E., and Sahinalp, S. C. (2011). Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res.* 21, 2203–2212. doi:10.1101/gr.120501.111

Iafrate, A. J., Feuk, L., Rivera, M. N., Listewnik, M. L., Donahoe, P. K., Qi, Y., et al. (2004). Detection of large-scale variation in the human genome. *Nat. Genet.* 36, 949–951. doi:10.1038/ng1416

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored de bruijn graphs. *Nat. Genet.* 44, 226–232. doi:10.1038/ng.1028

Ivakhno, S., Royce, T., Cox, A. J., Evers, D. J., Cheetham, R. K., and Tavaré, S. (2010). Cnaseg – a novel framework for identification of copy number changes in cancer from second-generation sequencing data. *Bioinformatics* 26, 3051–3058. doi:10.1093/bioinformatics/btq587

Jain, M., Fiddes, I. T., Miga, K. H., Olsen, H. E., Paten, B., and Akeson, M. (2015). Improved data analysis for the minion nanopore sequencer. *Nat. Methods* 12, 351–356. doi:10.1038/nmeth.3290

Jiang, Y., Oldridge, D. A., Diskin, S. J., and Zhang, N. R. (2015). Codex: a normalization and copy number variation detection method for whole exome sequencing. *Nucleic Acids Res.* 43, e39. doi:10.1093/nar/gku1363

Jiang, Y., Wang, Y., and Brudno, M. (2012). Prism: pair-read informed split-read mapping for base-pair level detection of insertion, deletion and structural variants. *Bioinformatics* 28, 2576–2583. doi:10.1093/bioinformatics/bts484

Kadalayil, L., Rafiq, S., Rose-Zerilli, M. J. J., Pengelly, R. J., Parker, H., Oscier, D., et al. (2015). Exome sequence read depth methods for identifying copy number changes. *Brief. Bioinform.* 16, 380–392. doi:10.1093/bib/bbu027

Karakoc, E., Alkan, C., O'Roak, B. J., Dennis, M. Y., Vives, L., Mark, K., et al. (2012). Detection of structural variants and indels within exome data. *Nat. Methods* 9, 176–178. doi:10.1038/nmeth.1810

Keane, T. M., Wong, K., and Adams, D. J. (2013). Retroseq: transposable element discovery from next-generation sequencing data. *Bioinformatics* 29, 389–390. doi:10.1093/bioinformatics/bts697

Kelley, D. R., Schatz, M. C., and Salzberg, S. L. (2010). Quake: quality-aware detection and correction of sequencing errors. *Genome Biol.* 11, R116. doi:10.1186/gb-2010-11-11-r116

Kiełbasa, S. M., Wan, R., Sato, K., Horton, P., and Frith, M. C. (2011). Adaptive seeds tame genomic sequence comparison. *Genome Res.* 21, 487–493. doi:10.1101/gr.113985.110

Kim, K. E., Peluso, P., Babayan, P., Yeadon, P. J., Yu, C., Fisher, W. W., et al. (2014). Long-read, whole-genome shotgun sequence data for five model organisms. *Sci. Data* 1, 140045. doi:10.1038/sdata.2014.45

Kim, T.-M., Luquette, L. J., Xi, R., and Park, P. J. (2010). rsw-seq: algorithm for detection of copy number alterations in deep sequencing data. *BMC Bioinformatics* 11:432. doi:10.1186/1471-2105-11-432

Klambauer, G., Schwarzbauer, K., Mayr, A., Clevert, D.-A., Mitterecker, A., Bodenhofer, U., et al. (2012). cn.mops: mixture of poissons for discovering copy number variations in next-generation sequencing data with a low false discovery rate. *Nucleic Acids Res.* 40, e69. doi:10.1093/nar/gks003

Korbel, J. O., Abyzov, A., Mu, X. J., Carriero, N., Cayting, P., Zhang, Z., et al. (2009). Pemer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.* 10, R23. doi:10.1186/gb-2009-10-2-r23

Krumm, N., Sudmant, P. H., Ko, A., O'Roak, B. J., Malig, M., Coe, B. P., et al. (2012). Copy number variation detection and genotyping from exome sequence data. *Genome Res.* 22, 1525–1532. doi:10.1101/gr.138115.112

Lam, H. Y. K., Pan, C., Clark, M. J., Lacroute, P., Chen, R., Haraksingh, R., et al. (2012). Detecting and annotating genetic variations using the hugeseq pipeline. *Nat. Biotechnol.* 30, 226–229. doi:10.1038/nbt.2134

Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., et al. (2001). Initial sequencing and analysis of the human genome. *Nature* 409, 860–921. doi:10.1038/35057062

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10, R25. doi:10.1186/gb-2009-10-3-r25

Lappalainen, I., Lopez, J., Skipper, L., Hefferon, T., Spalding, J. D., Garner, J., et al. (2013). Dbvar and dgva: public archives for genomic structural variation. *Nucleic Acids Res.* 41, D936–D941. doi:10.1093/nar/gks1213

Layer, R. M., Chiang, C., Quinlan, A. R., and Hall, I. M. (2014). Lumpy: a probabilistic framework for structural variant discovery. *Genome Biol.* 15, R84. doi:10.1186/gb-2014-15-6-r84

Lee, E., Iskow, R., Yang, L., Gokcumen, O., Haseley, P., Luquette, L. J. III, et al. (2012). Landscape of somatic retrotransposition in human cancers. *Science* 337, 967–971. doi:10.1126/science.1222077

Lee, S., Hormozdiari, F., Alkan, C., and Brudno, M. (2009). Modil: detecting small indels from clone-end sequencing with mixtures of distributions. *Nat. Methods* 6, 473–474. doi:10.1038/nmeth.f.256

Lee, W.-P., Stromberg, M. P., Ward, A., Stewart, C., Garrison, E. P., and Marth, G. T. (2014). Mosaik: a hash-based algorithm for accurate next-generation sequencing short-read mapping. *PLoS ONE* 9:e90581. doi:10.1371/journal.pone.0090581

Li, H., and Durbin, R. (2010). Fast and accurate long-read alignment with burrows-wheeler transform. *Bioinformatics* 26, 589–595. doi:10.1093/bioinformatics/btp698

Li, R., Fan, W., Tian, G., Zhu, H., He, L., Cai, J., et al. (2010). The sequence and de novo assembly of the giant panda genome. *Nature* 463, 311–317. doi:10.1038/nature08696

Li, X., Chen, S., Xie, W., Vogel, I., Choy, K. W., Chen, F., et al. (2014). Pscc: sensitive and reliable population-scale copy number variation detection method based on low coverage sequencing. *PLoS ONE* 9:e85096. doi:10.1371/journal.pone.0085096

Lin, K., Smit, S., Bonnema, G., Sanchez-Perez, G., and de Ridder, D. (2014). Making the difference: integrating structural variation detection tools. *Brief. Bioinform.* doi:10.1093/bib/bbu047

Luo, R., Liu, B., Xie, Y., Li, Z., Huang, W., Yuan, J., et al. (2012). Soapdenovo2: an empirically improved memory-efficient short-read de novo assembler. *Gigascience* 1, 18. doi:10.1186/2047-217X-1-18

Magi, A., Benelli, M., Marseglia, G., Nannetti, G., Scordo, M. R., and Torricelli, F. (2010). A shifting level model algorithm that identifies aberrations in array-cgh data. *Biostatistics* 11, 265–280. doi:10.1093/biostatistics/kxp051

Magi, A., Benelli, M., Yoon, S., Roviello, F., and Torricelli, F. (2011). Detecting common copy number variants in high-throughput sequencing data by using jointslm algorithm. *Nucleic Acids Res.* 39, e65. doi:10.1093/nar/gkr068

Magi, A., Tattini, L., Cifola, I., D'Aurizio, R., Benelli, M., Mangano, E., et al. (2013). Excavator: detecting copy number variants from whole-exome sequencing data. *Genome Biol.* 14, R120. doi:10.1186/gb-2013-14-10-r120

Magi, A., Tattini, L., Pippucci, T., Torricelli, F., and Benelli, M. (2012). Read count approach for dna copy number variants detection. *Bioinformatics* 28, 470–478. doi:10.1093/bioinformatics/btr707

Marschall, T., Costa, I. G., Canzar, S., Bauer, M., Klau, G. W., Schliep, A., et al. (2012). Clever: clique-enumerating variant finder. *Bioinformatics* 28, 2875–2882. doi:10.1093/bioinformatics/bts566

Marschall, T., Hajirasouliha, I., and Schönhuth, A. (2013). Mate-clever: Mendelian-inheritance-aware discovery and genotyping of midsize and long indels. *Bioinformatics* 29, 3143–3150. doi:10.1093/bioinformatics/btt556

Medvedev, P., Fiume, M., Dzamba, M., Smith, T., and Brudno, M. (2010). Detecting copy number variation with mated short reads. *Genome Res.* 20, 1613–1622. doi:10.1101/gr.106344.110

Metzker, M. L. (2010). Sequencing technologies – the next generation. *Nat. Rev. Genet.* 11, 31–46. doi:10.1038/nrg2626

Miller, C. A., Hampton, O., Coarfa, C., and Milosavljevic, A. (2011). Readdepth: a parallel r package for detecting copy number alterations from short sequencing reads. *PLoS ONE* 6:e16327. doi:10.1371/journal.pone.0016327

Mills, R. E., Luttig, C. T., Larkins, C. E., Beauchamp, A., Tsui, C., Pittard, W. S., et al. (2006). An initial map of insertion and deletion (indel) variation in the human genome. *Genome Res.* 16, 1182–1190. doi:10.1101/gr.4565806

Mills, R. E., Walter, K., Stewart, C., Handsaker, R. E., Chen, K., Alkan, C., et al. (2011). Mapping copy number variation by population-scale genome sequencing. *Nature* 470, 59–65. doi:10.1038/nature09708

Mimori, T., Nariai, N., Kojima, K., Takahashi, M., Ono, A., Sato, Y., et al. (2013). isvp: an integrated structural variant calling pipeline from high-throughput sequencing data. *BMC Syst. Biol.* 7(Suppl. 6):S8. doi:10.1186/1752-0509-7-S6-S8

Mullikin, J. C., and Ning, Z. (2003). The phusion assembler. *Genome Res.* 13, 81–90. doi:10.1101/gr.731003

Myers, E. W. (2005). The fragment assembly string graph. *Bioinformatics* 21(Suppl. 2), ii79–ii85. doi:10.1093/bioinformatics/bti1114

Niedringhaus, T. P., Milanova, D., Kerby, M. B., Snyder, M. P., and Barron, A. E. (2011). Landscape of next-generation sequencing technologies. *Anal. Chem.* 83, 4327–4341. doi:10.1021/ac2010857

Nijkamp, J. F., van den Broek, M. A., Geertman, J.-M. A., Reinders, M. J. T., Daran, J.-M. G., and de Ridder, D. (2012). De novo detection of copy number variation by co-assembly. *Bioinformatics* 28, 3195–3202. doi:10.1093/bioinformatics/bts601

Ning, Z., Cox, A. J., and Mullikin, J. C. (2001). Ssaha: a fast search method for large dna databases. *Genome Res.* 11, 1725–1729. doi:10.1101/gr.194201

O'Rawe, J. A., Ferson, S., and Lyon, G. J. (2015). Accounting for uncertainty in dna sequencing data. *Trends Genet.* 31, 61–66. doi:10.1016/j.tig.2014.12.002

Pabinger, S., Dander, A., Fischer, M., Snajder, R., Sperk, M., Efremova, M., et al. (2014). A survey of tools for variant analysis of next-generation genome sequencing data. *Brief. Bioinformatics* 15, 256–278. doi:10.1093/bib/bbs086

Pang, A. W., MacDonald, J. R., Pinto, D., Wei, J., Rafiq, M. A., Conrad, D. F., et al. (2010). Towards a comprehensive structural variation map of an individual human genome. *Genome Biol.* 11, R52. doi:10.1186/gb-2010-11-5-r52

Pareek, C. S., Smoczynski, R., and Tretyn, A. (2011). Sequencing technologies and genome sequencing. *J. Appl. Genet.* 52, 413–435. doi:10.1007/s13353-011-0057-x

Pinkel, D., Segraves, R., Sudar, D., Clark, S., Poole, I., Kowbel, D., et al. (1998). High resolution analysis of dna copy number variation using comparative genomic hybridization to microarrays. *Nat. Genet.* 20, 207–211. doi:10.1038/2524

Qi, J., and Zhao, F. (2011). ingap-sv: a novel scheme to identify and visualize structural variation from paired end mapping data. *Nucleic Acids Res.* 39, W567–W575. doi:10.1093/nar/gkr506

Quail, M. A., Smith, M., Coupland, P., Otto, T. D., Harris, S. R., Connor, T. R., et al. (2012). A tale of three next generation sequencing platforms: comparison of ion torrent, pacific biosciences and illumina miseq sequencers. *BMC Genomics* 13:341. doi:10.1186/1471-2164-13-341

Quick, J., Quinlan, A. R., and Loman, N. J. (2014). A reference bacterial genome dataset generated on the minion™ portable single-molecule nanopore sequencer. *Gigascience* 3, 22. doi:10.1186/2047-217X-3-22

Quinlan, A. R., Clark, R. A., Sokolova, S., Leibowitz, M. L., Zhang, Y., Hurles, M. E., et al. (2010). Genome-wide mapping and assembly of structural variant breakpoints in the mouse genome. *Genome Res.* 20, 623–635. doi:10.1101/gr.102970.109

Rausch, T., Zichner, T., Schlattl, A., Stütz, A. M., Benes, V., and Korbel, J. O. (2012). Delly: structural variant discovery by integrated paired-end and split-read analysis. *Bioinformatics* 28, i333–i339. doi:10.1093/bioinformatics/bts378

Samarakoon, P. S., Sorte, H. S., Kristiansen, B. E., Skodje, T., Sheng, Y., Tjønnfjord, G. E., et al. (2014). Identification of copy number variants from exome sequence data. *BMC Genomics* 15:661. doi:10.1186/1471-2164-15-661

Sathirapongsasuti, J. F., Lee, H., Horst, B. A. J., Brunner, G., Cochran, A. J., Binder, S., et al. (2011). Exome sequencing-based copy-number variation and loss of heterozygosity detection: Exomecnv. *Bioinformatics* 27, 2648–2654. doi:10.1093/bioinformatics/btr462

Schadt, E. E., Turner, S., and Kasarskis, A. (2010). A window into third-generation sequencing. *Hum. Mol. Genet.* 19, R227–R240. doi:10.1093/hmg/ddq416

Schröder, J., Hsu, A., Boyle, S. E., Macintyre, G., Cmero, M., Tothill, R. W., et al. (2014). Socrates: identification of genomic rearrangements in tumour genomes by re-aligning soft clipped reads. *Bioinformatics*. 30, 1064–1072. doi:10.1093/bioinformatics/btt767

Sebat, J., Lakshmi, B., Malhotra, D., Troge, J., Lese-Martin, C., Walsh, T., et al. (2007). Strong association of de novo copy number mutations with autism. *Science* 316, 445–449. doi:10.1126/science.1138659

Shendure, J., and Ji, H. (2008). Next-generation dna sequencing. *Nat. Biotechnol.* 26, 1135–1145. doi:10.1038/nbt1486

Shendure, J. A., Porreca, G. J., Church, G. M., Gardner, A. F., Hendrickson, C. L., Kieleczawa, J., et al. (2011). Overview of dna sequencing strategies. *Curr. Protoc. Mol. Biol.* Chapter 7, Unit7.1. doi:10.1002/0471142727.mb0701s96

Simpson, J. T., and Durbin, R. (2010). Efficient construction of an assembly string graph using the fm-index. *Bioinformatics* 26, i367–i373. doi:10.1093/bioinformatics/btq217

Simpson, J. T., and Durbin, R. (2012). Efficient de novo assembly of large genomes using compressed data structures. *Genome Res.* 22, 549–556. doi:10.1101/gr.126953.111

Sindi, S., Helman, E., Bashir, A., and Raphael, B. J. (2009). A geometric approach for classification and comparison of structural variants. *Bioinformatics* 25, i222–i230. doi:10.1093/bioinformatics/btp208

Sindi, S. S., Onal, S., Peng, L. C., Wu, H.-T., and Raphael, B. J. (2012). An integrative probabilistic model for identification of structural variation in sequencing data. *Genome Biol.* 13, R22. doi:10.1186/gb-2012-13-3-r22

Snijders, A. M., Nowak, N., Segraves, R., Blackwood, S., Brown, N., Conroy, J., et al. (2001). Assembly of microarrays for genome-wide measurement of dna copy number. *Nat. Genet.* 29, 263–264. doi:10.1038/ng754

Tan, R., Wang, Y., Kleinstein, S. E., Liu, Y., Zhu, X., Guo, H., et al. (2014). An evaluation of copy number variation detection tools from whole-exome sequencing data. *Hum. Mutat.* 35, 899–907. doi:10.1002/humu.22537

Teo, S. M., Pawitan, Y., Ku, C. S., Chia, K. S., and Salim, A. (2012). Statistical challenges associated with detecting copy number variations with next-generation sequencing. *Bioinformatics* 28, 2711–2718. doi:10.1093/bioinformatics/bts535

Thung, D., de Ligt, J., Vissers, L., Steehouwer, M., Kroon, M., de Vries, P., et al. (2014). Mobster: accurate detection of mobile element insertions in next generation sequencing data. *Genome Biol.* 15, 488. doi:10.1186/s13059-014-0488-x

Trappe, K., Emde, A.-K., Ehrlich, H.-C., and Reinert, K. (2014). Gustaf: detecting and correctly classifying SVs in the NGS twilight zone. *Bioinformatics* 30, 3484–3490. doi:10.1093/bioinformatics/btu431

Venkatesan, B. M., and Bashir, R. (2011). Nanopore sensors for nucleic acid analysis. *Nat. Nanotechnol.* 6, 615–624. doi:10.1038/nnano.2011.129

Venkatraman, E. S., and Olshen, A. B. (2007). A faster circular binary segmentation algorithm for the analysis of array cgh data. *Bioinformatics* 23, 657–663. doi:10.1093/bioinformatics/btl646

Voelkerding, K. V., Dames, S. A., and Durtschi, J. D. (2009). Next-generation sequencing: from basic research to diagnostics. *Clin. Chem.* 55, 641–658. doi:10.1373/clinchem.2008.112789

Wang, J., Mullighan, C. G., Easton, J., Roberts, S., Heatley, S. L., Ma, J., et al. (2011). Crest maps somatic structural variation in cancer genomes with base-pair resolution. *Nat. Methods* 8, 652–654. doi:10.1038/nmeth.1628

Wong, K., Keane, T. M., Stalker, J., and Adams, D. J. (2010). Enhanced structural variant and breakpoint detection using svmerge by integration of multiple detection methods and local assembly. *Genome Biol.* 11, R128. doi:10.1186/gb-2010-11-12-r128

Wu, J., Lee, W.-P., Ward, A., Walker, J. A., Konkel, M. K., Batzer, M. A., et al. (2014). Tangram: a comprehensive toolbox for mobile element insertion detection. *BMC Genomics* 15:795. doi:10.1186/1471-2164-15-795

Xi, R., Hadjipanayis, A. G., Luquette, L. J., Kim, T.-M., Lee, E., Zhang, J., et al. (2011). Copy number variation detection in whole-genome sequencing data using the bayesian information criterion. *Proc. Natl. Acad. Sci. U.S.A.* 108, E1128–E1136. doi:10.1073/pnas.1110574108

Xie, C., and Tammi, M. T. (2009). Cnv-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics* 10:80. doi:10.1186/1471-2105-10-80

Ye, K., Schulz, M. H., Long, Q., Apweiler, R., and Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25, 2865–2871. doi:10.1093/bioinformatics/btp394

Yoon, S., Xuan, Z., Makarov, V., Ye, K., and Sebat, J. (2009). Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.* 19, 1586–1592. doi:10.1101/gr.092981.109

Zeitouni, B., Boeva, V., Janoueix-Lerosey, I., Loeillet, S., Legoix-né, P., Nicolas, A., et al. (2010). Svdetect: a tool to identify genomic structural variations from paired-end and mate-pair sequencing data. *Bioinformatics* 26, 1895–1896. doi:10.1093/bioinformatics/btq293

Zhang, F., Gu, W., Hurles, M. E., and Lupski, J. R. (2009). Copy number variation in human health, disease, and evolution. *Annu. Rev. Genomics Hum. Genet.* 10, 451–481. doi:10.1146/annurev.genom.9.081307.164217

Zhang, J., and Wu, Y. (2011). Svseq: an approach for detecting exact breakpoints of deletions with low-coverage sequence data. *Bioinformatics* 27, 3228–3234. doi:10.1093/bioinformatics/btr563

Zhao, M., Wang, Q., Wang, Q., Jia, P., and Zhao, Z. (2013). Computational tools for copy number variation (CNV) detection using next-generation sequencing data: features and perspectives. *BMC Bioinformatics* 14(Suppl. 11):S1. doi:10.1186/1471-2105-14-S11-S1

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in
BIOENGINEERING AND BIOTECHNOLOGY

MINI REVIEW ARTICLE
published: 26 January 2015
doi: 10.3389/fbioe.2015.00008

The challenge of small-scale repeats for indel discovery

***Giuseppe Narzisi*[1]* and *Michael C. Schatz*[2]**

[1] *New York Genome Center, New York, NY, USA*
[2] *Cold Spring Harbor Laboratory, Simons Center for Quantitative Biology, Cold Spring Harbor, New York, NY, USA*

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Francesco Vezzi, SciLifeLab, Sweden
Lisle Elliott Mose, University of North Carolina at Chapel Hill, USA
Pierre Peterlongo, Inria, France

***Correspondence:**
Giuseppe Narzisi, New York Genome Center, 101 Avenue of the Americas, New York, NY 10013, USA
e-mail: gnarzisi@nygenome.org

Repetitive sequences are abundant in the human genome. Different classes of repetitive DNA sequences, including simple repeats, tandem repeats, segmental duplications, interspersed repeats, and other elements, collectively span more than 50% of the genome. Because repeat sequences occur in the genome at different scales they can cause various types of sequence analysis errors, including in alignment, *de novo* assembly, and annotation, among others. This mini-review highlights the challenges introduced by small-scale repeat sequences, especially near-identical tandem or closely located repeats and short tandem repeats, for discovering DNA insertion and deletion (indel) mutations from next-generation sequencing data. We also discuss the de Bruijn graph sequence assembly paradigm that is emerging as the most popular and promising approach for detecting indels. The human exome is taken as an example and highlights how these repetitive elements can obscure or introduce errors while detecting these types of mutations.

Keywords: next-generation sequencing, sequence assembly, sequence analysis, variant detection, indel mutation, repetitive sequences, nucleic acid

INTRODUCTION

Enormous advances made over the last decade in next-generation sequencing technologies and computational variation analysis have made it feasible to study human genetics in unprecedented detail. These technologies have enabled the sequencing of many thousands of human genomes to examine the genetics of healthy and diseased human populations. This has included sequencing thousands of healthy individuals of different ancestries from around the world (The 1000 Genomes Project Consortium, 2010; Khurana et al., 2013), along with detailed studies of cancer[1], autism (Iossifov et al., 2014), and schizophrenia (Schizophrenia Working Group of the Psychiatric Genomics Consortium, 2014), among many other projects. While historically genomic studies have focused on single nucleotide polymorphisms (SNPs) due to their prevalence and relative technical simplicity, a recent trend has been to study the role of insertion and deletion (indel) mutations. Already these projects have discovered indels to be ubiquitous in genomes, occurring nearly as frequently as SNPs, but with great diversity in size ranging from single base indels through larger events covering much larger regions (Montgomery et al., 2013). Indel mutations are especially important because they have been implicated in dozens of diseases through small frameshift mutations as well as larger indels that radically alter genes, change splicing and binding sites, or disrupt other important genomic sequences.

Most of the commonly used approaches for finding mutations from next-generation sequence data align one read at a time to the reference genome and then scan the alignments to identify any mutations (DePristo et al., 2011). This analytical framework works well for identifying simple mutations, as reads with a few mutated bases can generally be correctly aligned to a genome across the mutation. However, for indel analysis, this process becomes less and less effective. In the case of a larger insertion reads supporting the mutation will contain fewer and fewer bases matching the reference and therefore increasingly fail to map. A large deletion also leads to mapping complications, because even though the read consists of bases from the reference, there may not be enough bases to unambiguously map to both sides of the deletion forcing the aligner to instead trim or "soft clip" the reads. Consequently, insertions or deletions of more than a few bases are challenging to discover using standard alignment-based methods, and recent approaches have instead focused on assembly techniques to recover them instead.

Repetitive sequences in the genome also significantly complicate both sequencing and analysis accuracy (Treangen and Salzberg, 2011). Repeats of all classes complicate the mapping process as they introduce ambiguity into the true position of a read potentially reducing the sensitivity of our ability to discover indels or other mutations. Repeats, if not analyzed properly, can also introduce false positives by suggesting the presence of an artificial indels between repetitive elements and decrease the specificity of variant calling methods. Simple tandem repeats (STRs) are especially challenging genomic sequences to sequence and analyze, as they have a substantially greater sequencing error rate than other sequences and are prone to polymerase slippage that can artificially extend or contract the length of the repetitive element (Ellegren, 2004). For example, if a locus should consist of 10 adenines, during the sequencing process reads may be generated with just 9 or even 11. Downstream algorithms examining the sequencing data around STRs may misinterpret the sequencing error as an indel polymorphism in the genome if they are not aware of such effects.

Finally, detection of *de novo* and somatic mutations, although conceptually simple tasks, pose additional challenges within

[1] http://cancergenome.nih.gov/

small-scale repeats due to technical and algorithmic problems that can easily introduce false-negative variants. For example, strand biases at the sequencing stage can introduce allele imbalance favoring the reference allele over the mutation at a specific locus in the normal sample with the negative effect of introducing false-positive somatic calls in the tumor. Similarly, calling *de novo* mutations within repeat structures, in particular STRs, is complicated by noise introduced by sequencing errors which can mask (by chance) a true *de novo* mutation by generating the same mutation signature in the parent. Consequently, many large-scale studies currently avoid calling *de novo* mutations at those highly variable sites.

INDEL DISCOVERY – THE ERA OF MICRO-ASSEMBLY

In an effort to extend the power of detectable mutation using short reads from next-generation sequencing technologies (e.g., Illumina), assembly based variant detection techniques are now becoming a popular solution. The strategy employed by these methods is to perform localized sequence assembly, micro-assembly, of the reads mapping around the location of the candidate mutation. Recently developed tools that use this strategy include Scalpel (Narzisi et al., 2014), GATK HaplotypeCaller[2], SOAPindel (Li et al., 2012), Platypus (Rimmer et al., 2014), ABRA (Mose et al., 2014), TIGRA (Chen et al., 2014), DISCOVAR (Weisenfeld et al., 2014), and Bubbleparse (Leggett et al., 2013). **Figure 1A** illustrates the general workflow followed by tools that employ micro-assembly. The first step consists in performing a fast alignment of the reads to the reference genome, typically using BWA (Li and Durbin, 2009); however, these alignments are not directly used to identify mutations but instead the purpose is to localize the analysis by identifying all the reads that have similarity to the given locus. Once a region of interest has been selected, the reads are extracted, including soft-clipped reads and reads that fail to map but are anchored by their mate. The de Bruijn graph of those reads is then constructed by decomposing the reads into overlapping *k*-mers, and then explored to select candidate paths that contain mutations. Finally, the assembled sequences are aligned back to the reference to detect the correct signature for the mutation.

Although all the above-mentioned tools follow this paradigm, they differ in many important aspects. Two key differences are the selection of the *k*-mer size used for assembly and the way repeat sequences are handled. For example, SOAPindel tries to reconnect a broken path in low-coverage regions by searching for unused reads with gradually shorter *k*-mers until a path is formed or the lower bound on *k*-mer length has been reached. Similarly in TIGRA, the user can specify the list of *k*-mer sizes to use; however, this tool has been tailored for breakpoint detection without reporting the indel sequence. GATK HaplotypeCaller by default attempts to build two separate graphs, using *k*-mers of 10 and 25 bases in size; however, other *k*-mer sizes can be specified from the command line. Scalpel instead employs a self-tuning *k*-mer strategy that is coupled with a meticulous repeat composition analysis in order to reduce errors in highly repetitive regions. In contrast

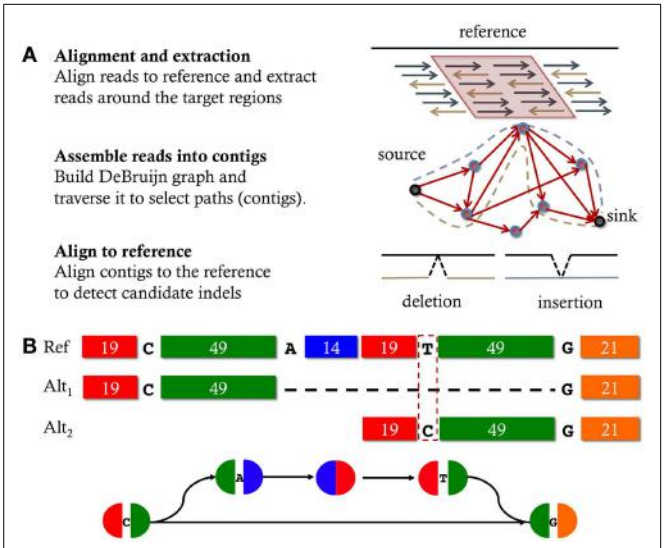


FIGURE 1 | Micro-assembly. (A) typical workflow of the micro-assembly strategy: given a specific region of interest, the reads, previously aligned to the reference, are extracted, assembled, and then contigs are aligned back to the reference to discover mutations. **(B)** Schematic representation of a bubble in a De Bruijn graph induced by a heterozygous mutation within a repetitive sequence composed of two near-identical copies (with 1 bp mismatch) that are 15 bp apart. Each block represents a sequence; sequences that have the same base pair composition have the same color; the length of each sequence (in base pairs) is reported inside each block. Each node of the graph is colored according to the order and sequence composition contained in it. Simple (non-branching) paths are represented in a single node. Alt1 and Alt2 are two alternative alignments of the sequence contained in the bottom side of the bubble, where the dashed line represents the alignment gap. The top side of the bubble matches the reference. Any *k*-mer longer than the longest identical repeat (>49 bp) and shorter than the longest near-identical repeat (<69 bp) would create a bubble like the one depicted **(B)** where a jump is allowed from the first copy of the near-identical repeat to the second copy.

to SOAPindel that uses a decreasing size of *k*-mer values, Scalpel starts with a small value (default $k = 25$) and if a repeat structure (either perfect or near-perfect up to a few mismatches) is detected, the graph is discarded and a larger *k*-mer is selected. This process continues until a "repeat-free" graph is constructed or a maximum *k*-mer length is reached, and in the latter case, the region is discarded as undetectable. GATK HaplotypeCaller uses a similar iterative strategy to Scalpel to avoid false-positives indels within repeats by trying a larger *k*-mer when a cycle is detected in the graph. However, only perfect repeats (cycles in the graph) are checked by the GATK HaplotypeCaller, while nearly identical repeats can still mislead the algorithm to generate false-positive calls. ABRA also performs a localized assembly of the reads for genomic regions of size $\leq$2 kb. Similarly to Scalpel and GATK HaplotypeCaller, all non-cyclic paths through the graph are traversed and, in case a cycle is detected, the region is iteratively reassembled using increasing *k*-mer sizes until the cycle non-longer exists. Platypus integrates the colored de Bruijn graph methods initially developed for Cortex (Iqbal et al., 2012) to also perform a local assembly in small regions (by default 1.5 kb) using a fixed *k*-mer (15 by default). A revised DFS traversal of the graph is used in

[2]http://www.broadinstitute.org/gatk/guide/article?id=4146

Platypus to avoid loops and to generate only non-self-intersecting paths. DISCOVAR also involves initial alignment of reads to the genomic regions followed by careful local assembly. Similarly to Platypus, DISCOVAR combines the detection of SNPs and indels into a unified framework. Using a combination of longer reads and improved error-correction algorithms, DISCOVAR demonstrates increased power compared to GATK and Cortex to detect challenging variants located in low-complexity sequences and segmental duplications. However, DISCOVAR is designed to work only with PCR free 250 bp paired-end reads, which are not commonly available. Finally, Bubbleparse, although it is not exactly a micro-assembly method, also attempts to identify SNPs and indels independent of a reference genome using the de Bruijn graph implementation in the Cortex framework, but it does not specifically evaluate the repetitive content surrounding a candidate indel, and was reported to have a high false-positive rate for indels (Leggett et al., 2013).

By assembling longer stretches of DNA sequence around the mutation, micro-assembly techniques allow more accurate alignments and interpretation of the detected mutations and extend the power of detectable indels $\geq$30 bp. However, like alignment-based methods, these techniques are also susceptible to errors when calling mutation within small-scale repeat sequences, specifically short tandem repeats (STRs) and localized near-identical repeats. For example, a comparative assessment (Narzisi et al., 2014) demonstrated through a large-scale re-sequencing experiment that SOAPindel has a high error rate within repeat structures. In this review, we start by discussing some classes of repeats that can be found in the human exome. We then show examples of the type of errors introduced by these repetitive structures and we provide recommendation on how to reduce or avoid the errors.

REPETITIVE STRUCTURES IN THE HUMAN EXOME

Repeats are the most difficult sequences to assemble and the specificity of any indel detection method is correlated to its ability to detect and analyze repetitive sequences correctly. Although the exome sequence composition is generally assumed to be relatively simple compared to the rest of the human genome, 30% of exons have a perfect 10 bp or larger repeat (Narzisi et al., 2014). More significantly, the number of near-identical repeats (sequences which differ from each other by just a few bases) increases substantially if more mismatches are permitted. **Figure 2** shows the percent of locally repetitive human exons as a function of different k-mer values and maximum number of mismatches. Each exon target is exhaustively analyzed to check for the presence of an identical or near-identical repeat (up to a maximum of three mismatches) in the same region defining the exon. The y-axis reports the percentage of those exons that have been found to contain a repeat of size k (x-axis). Since the presence of repeats is confined only inside each exon, this analysis demonstrates a substantial level of locally repetitive sequences in the human exome. The two main classes of repeat structures that contribute to this plot are near-identical repeats and STRs. Given the generally low error rate of the Illumina sequencing technology, allowing 3-mismatches for a 10-mer (30% error rate) would seem to not be realistic for a sequencing study. However, this repeat analysis must be examined in the context of performing sequence assembly using de Bruijn graph method, where a perfect match is required for two overlapping k-mers.

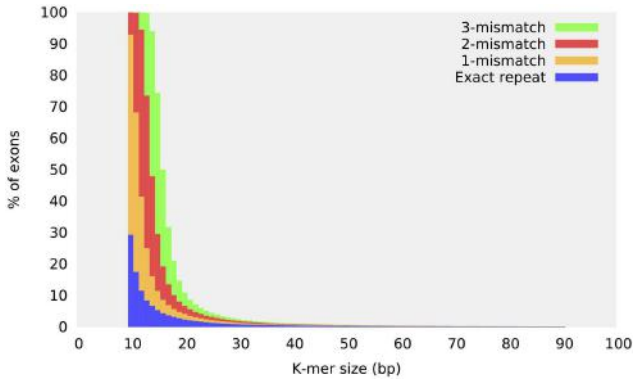


FIGURE 2 | Repeat content in the human exome. Repeat content distribution in the human exome target regions as a function of the k-mer size. The sequence of each target exon is analyzed to check for the presence of a repeat structure within the same region defining the exon. The y-axis reports the percentage of those exons that have been found to contain an identical or near-identical repeat of size k (up to three mismatches).

NEAR-IDENTICAL CLOSELY LOCATED REPEATS

The first major class of repeats that can confound indel discovery techniques is near-identical repetitive sequences that are localized within an exon or other small spans. This type of structures can introduce artifacts in the assembly graph that mislead such methods to make false-positive calls. **Figure 1B** shows an example of a near-identical repeat that can be misinterpreted as a large deletion. The key observation is that the beginning of this sequence is a nearly identical 69 bp repeat with just 1 bp different between the two copies that are 15 bp apart. The sequence is segmented as 19-C-49-A-14-19-T-49-G-21 where 19 and 49 are 19 and 49 bp identical repeats, separated by a 15 bp unique sequence (A, C, T, G are the typical bases). Since the longest exact repeat is 49 bp long, one would expect that using k-mer $= 55$ should be large enough to correctly assemble reads sampled from this sequence. However, if the sequencing data also contains reads with sequence 19-C-49-G because of a single base A to G change from the expected 19-C-49-A from sequencing error or true mutation, it can be wrongly interpreted as an 84 bp deletion of the A-14-19-T-49 internal segment.

The reason for this ambiguity and other false positives is that the de Bruijn graph is constructed using perfect matches of length $k-1$. So any k-mer longer than the longest identical repeat (>49 bp) and shorter than the longest near-identical repeat (<69 bp) would create a bubble like the one depicted **Figure 1B** where a jump is allowed from the first copy of the near-identical repeat to the second copy. When aligned end-to-end to the reference, the sequence associated to the branch can be aligned in two different ways, one showing a large (false-positive) 84 bp deletion and the other one showing a single nucleotide variation. To reduce the chance to make false-positive indel calls in these regions, it is essential to evaluate the presence of near-identical repeats in either the reference or the assembled sequences. Then, if this type

of ambiguity is detected, a decision must be taken to discard the region, flag the candidate mutations as having low quality, or use a *k*-mer size (if any) that avoids the creation of the false-bubble due to the near-identical repeat (e.g., Scalpel).

SHORT TANDEM REPEATS

Short tandem repeats, also known as microsatellites, are highly mutable genetic elements that consist of multiple repeating copies of elements composed of 1–6 nucleotides. Indels that alter the length of the repeat motif have been linked to more than 40 neurological diseases (Pearson et al., 2005). Despite recent advances in sequencing technology, STR variation pose remarkable challenges to variant detection methods compared to other classes of mutations, such as single nucleotide and copy number variation. Discovery of genetic STR variation with short-read sequence data is confounded by (1) the difficulty of uniquely mapping short, low-complexity reads, (2) the high rate of STR amplification error (e.g., homopolymers) due to replication slippage events and which result in high variability in the number of repeat elements (Mirkin, 2007), and (3) the fact that the spontaneous mutation rate of STRs can reach 1/500 mutations per locus per generation (Ballantyne et al., 2010). Due to these effects, distinguishing between sequencing errors and true mutations within STRs is the major challenge faced by indel detection methods. Moreover, even after a candidate locus for an STR mutation has been identified, the associated indel haplotype description can still have an ambiguous position. For example, if there is a 1 bp deletion in a long homopolymer (. . .AAAAAA. . .), deleting any A will give rise to the same haplotype but just with a different position. A more complex example which gives rise to two logically equivalent 3 bp deletions is

ref: AAACTGGAGGTTGC
alt_1: AAACT−−−GGTTGC
alt_2: AAACTGG−−−TTGC

Note that two different 3 bp sequences can be deleted (GGA or AGG) at two different locations generating the same alternative sequence. Since different methods might report different signatures for the same indel, these examples show how essential is to normalize the signature (typically left-normalization) when comparing indels.

Relatively few computational tools have been developed to specifically deal with the complexity of calling in STR regions. RepeatSeq (Highnam et al., 2013) and lobSTR (Gymrek et al., 2012) are the two most recent ones. In order to reduce the error rate at STR loci both methods use statistical modeling to empirically derive the sequencing error model. Comparisons with standard aligners such as BWA, Bowtie, and Novoalign, demonstrate that these aligners are biased toward the detection of the reference allele and specialized tools are required. Moreover most of the highly polymorphic STRs have length of 20 bp or longer, and unfortunately these sizes are the most prone to polymerase slippage and alignment artifacts (Gymrek et al., 2012). The major obstacle for STR profiling is the limited read length of current widely used sequencing technologies. A read must span the complete STR sequence in order to be detected with alignment. Micro-assembly is a promising approach to extend even further the spectrum of detectable STRs mutations and we expect micro-assembly tools specialized for STRs profiling to be developed in the near future.

CONCLUSION

The first major consideration for correctly identifying indel mutations is the use of assembly based approaches over alignment-based approaches. Assembly based methods afford the best sensitivity for detecting indel mutations, especially long indels, as they avoid any expectation or dependencies of the reads aligning end-to-end to the reference genome. This is especially important for reads sequencing long indels (>30 bp), as most read mapping algorithms typically treat these as soft-clipped reads or fail to map them at all. The next most important consideration is the presence of repeats in the genome, especially near-identical repeats within close proximity to each other and STR sequences that may increase the false-positive rate. Our analysis shows near-identical repeats are widespread in the genome, and if not carefully detected may introduce "false-bubbles" where the reads or assembled contigs incorrectly align to the repetitive sequences. Simple tandem repeats (STRs) are also very challenging, because of their especially high indel error rates and also especially high true mutation rates that can obscure true indels.

Considering the widespread interest to sequence genomes and identify indels for medical and other purposes, it is virtually certain that we will see the rise of new algorithmic and experimental approaches for indel detection in the near future. Algorithmically, we anticipate the development of more specialized methods for detecting indels within complex samples, such as somatic mutations in heterozygous cancer populations. For these samples, new scoring metrics will need to be developed that can accurately recognize low-coverage indels present in only a fraction of the cells. We also anticipate the rise of algorithms that can utilize very large populations of samples, especially to augment the standard reference genome with indels commonly found in the population to improve initial mapping efficiencies and also to flag problematic regions with unusual rates of mutations. Experimentally, we anticipate the rise of new sequencing technologies that can produce longer reads that will improve both micro-assembly and resolving repetitive elements. Already the widely used Illumina chemistry is available to produce ~250 bp reads, or even ~500 bp reads by merging paired-end reads together, and new single molecule approaches (PacBio and Oxford Nanopore) can generate substantially longer reads, although requiring new algorithms to tolerate the high error rate of the technologies. As these and other technologies improve we anticipate our ability to discover indel mutations will improve leading to the discovery of many additional indel-related diseases and phenotypes.

ACKNOWLEDGMENTS

We would like to thank Han Fang, Gholson Lyon, Ivan Iossofov, Michael Ronemus, Dan Levy, and Michael Wigler for their helpful discussions. This work has been supported in part, by National Institutes of Health award (R01-HG006677) and by National Science Foundation award (DBI-1350041) to Michael C. Schatz.

REFERENCES

Ballantyne, K. N., Goedbloed, M., Fang, R., Schaap, O., Lao, O., Wollstein, A., et al. (2010). Mutability of Y-chromosomal microsatellites: rates, characteristics, molecular bases, and forensic implications. *Am. J. Hum. Genet.* 87, 341–353. doi:10.1016/j.ajhg.2010.08.006

Chen, K., Chen, L., Fan, X., Wallis, J., Ding, L., and Weinstock, G. (2014). TIGRA: a targeted iterative graph routing assembler for breakpoint assembly. *Genome Res.* 24, 310–317. doi:10.1101/gr.162883.113

DePristo, M., Banks, E., Poplin, R., Garimella, K., Maguire, J., Hartl, C., et al. (2011). A framework for variation discovery and genotyping using next-generation DNA sequencing data. *Nat. Genet.* 43, 491–498. doi:10.1038/ng.806

Ellegren, H. (2004). Microsatellites: simple sequences with complex evolution. *Nat. Rev. Genet.* 5, 435–445. doi:10.1038/nrg1348

Gymrek, M., Golan, D., Rosset, S., and Erlich, Y. (2012). lobSTR: a short tandem repeat profiler for personal genomes. *Genome Res.* 22, 1154–1162. doi:10.1101/gr.135780.111

Highnam, G., Franck, C., Martin, A., Stephens, C., Puthige, A., and Mittelman, D. (2013). Accurate human microsatellite genotypes from high-throughput resequencing data using informed error profiles. *Nucleic Acids Res.* 41, e32. doi:10.1093/nar/gks981

Iossifov, I., O'Roak, B. J., Sanders, S. J., Ronemus, M., Krumm, N., Levy, D., et al. (2014). The contribution of de novo coding mutations to autism spectrum disorder. *Nature* 515, 216–221. doi:10.1038/nature13908

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored de Bruijn graphs. *Nat. Genet.* 44, 226–232. doi:10.1038/ng.1028

Khurana, E., Fu, Y., Colonna, V., Mu, X. J., Kang, H. M., Lappalainen, T., et al. (2013). Integrative annotation of variants from 1092 humans: application to cancer genomics. *Science* 342, 1235587. doi:10.1126/science.1235587

Leggett, R. M., Ramirez-Gonzalez, R. H., Verweij, W., Kawashima, C. G., Iqbal, Z., Jones, J. D., et al. (2013). Identifying and classifying trait linked polymorphisms in non-reference species by walking coloured de Bruijn graphs. *PLoS One* 8(3):e60058. doi:10.1371/journal.pone.0060058

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with Burrows–Wheeler transform. *Bioinformatics* 25, 1754–1760. doi:10.1093/bioinformatics/btp324

Li, S., Li, R., Li, H., Lu, J., Li, Y., Bolund, L., et al. (2012). SOAPindel: efficient identification of indels from short paired reads. *Genome Res.* 23, 195–200. doi:10.1101/gr.132480.111

Mirkin, S. M. (2007). Expandable DNA repeats and human disease. *Nature* 447, 932–940. doi:10.1038/nature05977

Montgomery, S. B., Goode, D. L., Kvikstad, E., Albers, C. A., Zhang, Z. D., Mu, X. J., et al. (2013). The origin, evolution, and functional impact of short insertion–deletion variants identified in 179 human genomes. *Genome Res.* 23, 749–761. doi:10.1101/gr.148718.112

Mose, L. E., Wilkerson, M. D., Hayes, D. N., Perou, C. M., and Parker, J. S. (2014). ABRA: improved coding indel detection via assembly-based realignment. *Bioinformatics* 30, 2813–2815. doi:10.1093/bioinformatics/btu376

Narzisi, G., O'Rawe, J. A., Iossifov, I., Fang, H., Lee, Y., Wang, Z., et al. (2014). Accurate de novo and transmitted indel detection in exome-capture data using microassembly. *Nat. Methods* 11, 1033–1036. doi:10.1038/nmeth.3069

Pearson, C. E., Edamura, N. K., and Cleary, J. D. (2005). Repeat instability: mechanisms of dynamic mutations. *Nat. Rev. Genet.* 6, 729–742. doi:10.1038/nrg1689

Rimmer, A., Phan, H., Mathieson, I., Iqbal, Z., Twigg, S. R. F.; WGS500 Consortium, et al. (2014). Integrating mapping-, assembly- and haplotype-based approaches for calling variants in clinical sequencing applications. *Nat. Genet.* 46, 912–918. doi:10.1038/ng.3036

Schizophrenia Working Group of the Psychiatric Genomics Consortium. (2014). Biological insights from 108 schizophrenia-associated genetic loci. *Nature* 511, 421–427. doi:10.1038/nature13595

The 1000 Genomes Project Consortium. (2010). A map of human genome variation from population-scale sequencing. *Nature* 467, 1061–1073. doi:10.1038/nature09534

Treangen, T. J., and Salzberg, S. L. (2011). Repetitive DNA and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.* 13, 36–46. doi:10.1038/nrg3117

Weisenfeld, N. I., Yin, S., Sharpe, T., Lau, B., Hegarty, R., Holmes, L., et al. (2014). Comprehensive variation discovery in single human genomes. *Nat. Genet.* 46, 1350–1355. doi:10.1038/ng.3121

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 28 October 2014; accepted: 07 January 2015; published online: 26 January 2015.

Citation: Narzisi G and Schatz MC (2015) The challenge of small-scale repeats for indel discovery. Front. Bioeng. Biotechnol. ***3****:8. doi: 10.3389/fbioe.2015.00008*

This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology.

frontiers in
BIOENGINEERING AND BIOTECHNOLOGY

ORIGINAL RESEARCH ARTICLE
published: 10 March 2015
doi: 10.3389/fbioe.2015.00028

G-CNV: a GPU-based tool for preparing data to detect CNVs with read-depth methods

Andrea Manconi[1]*, Emanuele Manca[2], Marco Moscatelli[1], Matteo Gnocchi[1], Alessandro Orro[1], Giuliano Armano[2] and Luciano Milanesi[1]

[1] *Institute for Biomedical Technologies, National Research Council, Milan, Italy*
[2] *Department of Electrical and Electronic Engineering, University of Cagliari, Cagliari, Italy*

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Jian Ren, Sun Yat-sen University, China
Sandra Gesing, University of Notre Dame, USA

****Correspondence:***
Andrea Manconi, Institute for Biomedical Technologies, National Research Council, Via F.lli Cervi, 93, Segrate, Milan 20090, Italy
e-mail: andrea.manconi@itb.cnr.it

Copy number variations (CNVs) are the most prevalent types of structural variations (SVs) in the human genome and are involved in a wide range of common human diseases. Different computational methods have been devised to detect this type of SVs and to study how they are implicated in human diseases. Recently, computational methods based on high-throughput sequencing (HTS) are increasingly used. The majority of these methods focus on mapping short-read sequences generated from a donor against a reference genome to detect signatures distinctive of CNVs. In particular, read-depth based methods detect CNVs by analyzing genomic regions with significantly different read-depth from the other ones. The pipeline analysis of these methods consists of four main stages: (i) data preparation, (ii) data normalization, (iii) CNV regions identification, and (iv) copy number estimation. However, available tools do not support most of the operations required at the first two stages of this pipeline. Typically, they start the analysis by building the read-depth signal from pre-processed alignments. Therefore, third-party tools must be used to perform most of the preliminary operations required to build the read-depth signal. These data-intensive operations can be efficiently parallelized on graphics processing units (GPUs). In this article, we present G-CNV, a GPU-based tool devised to perform the common operations required at the first two stages of the analysis pipeline. G-CNV is able to filter low-quality read sequences, to mask low-quality nucleotides, to remove adapter sequences, to remove duplicated read sequences, to map the short-reads, to resolve multiple mapping ambiguities, to build the read-depth signal, and to normalize it. G-CNV can be efficiently used as a third-party tool able to prepare data for the subsequent read-depth signal generation and analysis. Moreover, it can also be integrated in CNV detection tools to generate read-depth signals.

Keywords: CNV, GPU, HTS, read-depth, parallel

1. INTRODUCTION

SVs in the human genome can influence phenotype and predispose to or cause diseases (Feuk et al., 2006a,b). Single nucleotide polymorphisms (SNPs) were initially thought to represent the main source of human genomic variation (Sachidanandam et al., 2001). However, following the advances in technologies to analyze genome, it is now acknowledged that different types of SVs contribute to the genetic makeup of an individual. SV is a term generally used to refer different types of genetic variants that alter chromosomal structure as inversions, translocations, insertions, and deletions (Hurles et al., 2008). SVs such as insertions and deletions are also referred as CNVs. CNVs are the most prevalent types of SVs in the human genome and are implicated in a wide range of common human diseases including neurodevelopmental disorders (Merikangas et al., 2009), schizophrenia (Stefansson et al., 2008), and obesity (Bochukova et al., 2009). Studies based on microarray technology demonstrated that as much as 12% of the human genome is variable in copy number (Perry et al., 2006), and this genomic diversity is potentially related to phenotypic variation and to the predisposition to common diseases. Hence, it is essential to have effective tools able to detect CNVs and to study how they are implicated in human diseases.

Hybridization-based microarray approaches as array-comparative genomic hybridization (a-CGH) and SNP microarrays have been successfully used to identify CNVs (Carter, 2007). The low cost of a-CGH and SNP platforms promoted the use of microarray approaches. However, as pointed out in Alkan et al. (2011), microarrays (i) have limitations in the task of detecting copy number differences, (ii) provide no information on the location of duplicated copies, and (iii) are generally unable to resolve breakpoints at the single-base-pair level. Recently, computational methods for discovering SVs with HTS (Kircher and Kelso, 2010) have also been proposed (Medvedev et al., 2009). These methods can be categorized into alignment-free (i.e., *de novo* assembly) and alignment-based (i.e., paired-end mapping, split read, and read-depth) approaches (Zhao et al., 2013). The former (Iqbal et al., 2012; Nijkamp et al., 2012) focus on reconstruct DNA fragments by assembling overlapping

short-reads. CNVs are detected by comparing the assembled contigs to the reference genome. The latter focus on mapping short-read sequences generated from a donor against the reference genome with the aim of detecting signatures that are distinctive of different classes of SVs. Mapping data hide useful information that can be used to detect different SVs. Different methods that analyze different mapping information have been devised.

Paired-end mapping (PEM) methods (Chen et al., 2009; Korbel et al., 2009; Sindi et al., 2009; Hormozdiari et al., 2010, 2011; Mills et al., 2011) identify SVs/CNVs by detecting and analyzing paired-end reads generated from a donor that are discordantly mapped against the reference genome. These methods allow to detect different types of SVs (i.e., insertions, deletions, mobile element insertions, inversions, and tandem duplications), but they do not allow to detect insertions larger than the average insert size of the library preparations.

Split read (SR) methods (Ye et al., 2009; Abel et al., 2010; Abyzov and Gerstein, 2011; Zhang et al., 2011) are also based on paired-end reads. Unlike PEM methods that analyze discordant mappings, SR methods analyze unmapped or partially mapped reads as they potentially provide accurate breaking points at the single-base-pair level for SVs/CNVs.

Read-depth (RD) methods (Chiang et al., 2008; Xie and Tammi, 2009; Yoon et al., 2009; Ivakhno et al., 2010; Xi et al., 2010; Abyzov et al., 2011; Miller et al., 2011) are based on the assumption that the RD in a genomic region depends on the copy number of that region. In fact, as the sequencing process is uniform, the number of reads aligning to a region follows a Poisson distribution with mean directly proportional to the size of the region and to the copy number [see **Figure 1** and Chiang et al. (2008)]. These methods analyze the RD of a genome sequence through non-overlapping windows, with the aim of detecting those regions that exhibit a RD significantly different from the other ones. A duplicated region will differ from the other ones for a higher number of reads mapping on it, and then for a higher RD. Conversely, a deleted region will differ from the other ones for a lower number of reads mapping on it, and then for a lower RD. Basically, the analysis pipeline implemented in RD methods consists of four fundamental stages (Magi et al., 2012): (i) data preparation; (ii) data normalization; (iii) CNV regions identification; and (iv) copy number estimation (see **Figure 2**).

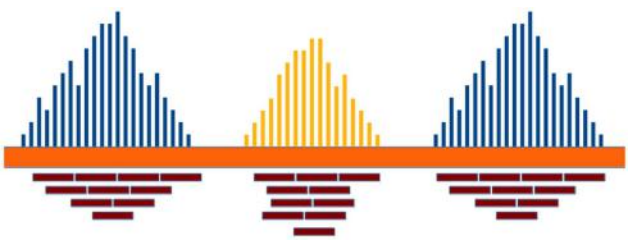


FIGURE 1 | The RD in a genomic region depends on the copy number of that region and follows a Poisson distribution. Duplicated and deleted regions are characterized by a RD signal different from that of the other ones.

Data preparation consists of different tasks aimed at assessing the quality of the read sequences, mapping the reads against the reference genome, removing low mapping quality sequences, and sizing the observing window used to calculate the RD signal. Data normalization is aimed at correcting the effect of two sources of bias that affect the detection of CNVs. In particular, it has been proved that correlation exists between RD and the GC-content (Dohm et al., 2008; Hillier et al., 2008; Harismendy et al., 2009); the RD increases with the GC-content of the underlying genomic region. Moreover, there exists a mappability bias due to

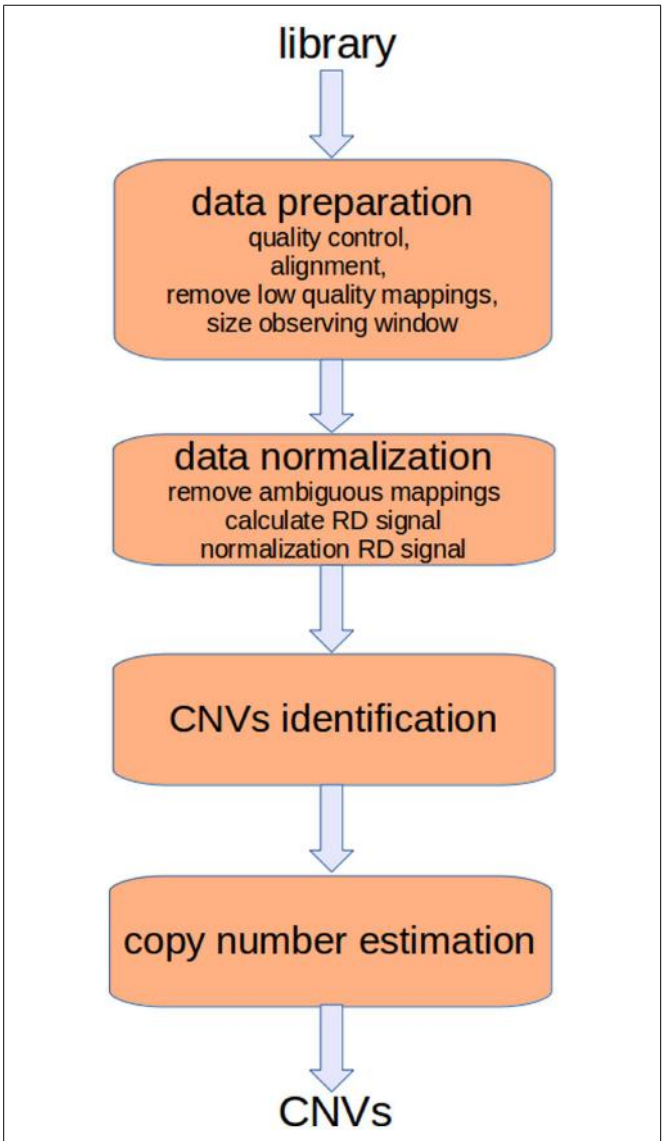


FIGURE 2 | The analysis pipeline of RD-based methods consists of four main stages. The first two stages consist of preparatory operations aimed at generating the RD signal. Sequencing produces artifacts that affect the alignment and consequently the RD signal. Different filtering operators can be applied to reduce these errors. Moreover, alignments must be post-processed to remove those of low quality and to resolve ambiguities. Finally, the RD signal is calculated taking into account the bias related with the GC-content.

the repetitive regions in a genome. A read can be mapped to different positions so that ambiguous mappings must be dealt with. After normalization, RD data are analyzed to detect the boundaries of regions characterized by changed copy number. Finally, DNA copy number of each region within breakpoints is estimated.

The first two stages of the analysis pipeline consist of common operations, whereas the last two consist of specific operations for each method. However, it should be pointed out that available tools do not implement most of the operations required at the first and second stage. Typically, these tools start the analysis by building the RD signal from the post-processed alignments. All preparatory operations must be performed by the researchers using third-party tools. Moreover, other tools as ReadDepth (Miller et al., 2011) require annotation files with information about the GC-content that are pre-computed only for some reference genome builds. Only some tools provide limited functionalities to pre-process alignments. For instance, RDXplorer (Yoon et al., 2009) and CNV-seq (Xie and Tammi, 2009) use the samtools (Li et al., 2009a) to remove low-quality mappings and to select the best hit location for each mapped read sequence, respectively.

Most of these operations are data-intensive and can be parallelized to be efficiently run on GPUs to save computing time. GPUs are hardware accelerators that are increasingly used to deal with computationally intensive algorithms. Recently, GPU-based solutions have been proposed to cope with different bioinformatics problems (e.g., Manavski and Valle, 2008; Liu et al., 2010; Shi et al., 2010; Yung et al., 2011; Manconi et al., 2014a,b; Zhao and Chu, 2014).

In this work, we present GPU-copy number variation (G-CNV), a GPU-based tool aimed at performing the preparatory operations required at the first two stages of the analysis pipeline for RD-based methods. G-CNV can be used to (i) filter low-quality sequences, (ii) mask low-quality nucleotides, (iii) remove adapter sequences, (iv) remove duplicated reads, (v) map read sequences, (vi) remove ambiguous mappings, (vii) build the RD signal, and (viii) normalize it. Apart the task of removing adapter sequences, all the other tasks are implemented on GPU. G-CNV can be used as a third-party tool to prepare the input for available RD-based detection tools or can be integrated in other tools to efficiently build the RD signal.

G-CNV is freely available for non-commercial use. The current release can be downloaded at the following address http://www.itb.cnr.it/web/bioinformatics/gcnv

2. MATERIALS AND METHODS

Data preparation and data normalization are crucial operations to properly detect CNVs. It is widely known that sequencing is a process subject to errors. These errors can affect the alignments; hence both the RD signal and the accuracy of the identified CNVs can be affected as well. G-CNV implements filtering operators aimed at correcting some errors related to the sequencing process. In particular, G-CNV is able to analyze the read sequences to filter those read sequences that do not satisfy a quality constraint, to mask low-quality nucleotides with an a*N*y symbol, to remove adapter sequences, and to remove duplicated read sequences. G-CNV uses *cutadapt* (Martin, 2011) to remove adapter sequences. As for the alignment, G-CNV uses the GPU-based short-read mapping tool SOAP3-dp (Luo et al., 2013). Low-quality alignments are filtered out, while ambiguous mappings can be treated according to different strategies. To build the RD signal, G-CNV builds a RD signal according to a fixed-size observing window. Then, this raw RD signal is corrected according to the GC-content of the observed windows.

In this section, we first give a short introduction to GPUs. Then, we present the strategies adopted to cope with the tasks implemented by G-CNV. Finally, we briefly recall the hardware and software equipment required to use G-CNV.

2.1. GPU

GPUs are hardware accelerators that are increasingly used to deal with computationally intensive algorithms. From an architectural perspective, GPUs are very different from traditional CPUs. Indeed, the latter are devices composed of few cores with lots of cache memory able to handle a few software threads at a time. Conversely, the former are devices equipped with hundreds of cores able to handle thousands of threads simultaneously, so that a very high level of parallelism can be reached (see **Figure 3**). Apart from the high level of parallelism, there may be other advantages to use the GPU technology. In particular, the low cost for accessing to the GPUs (if compared with the cost to equip a laboratory with a CPU-cluster) is promoting the technology. Moreover, GPUs are inherently more energy efficient than other ways of computation because they are optimized for throughput and performance per watt and not absolute performance. The main disadvantage of adopting the GPU technology is related with the effort required to code algorithms. GPUs can run certain algorithms very faster than CPUs. However, gaining this speed-up can require a notably effort to properly code the algorithms for GPU. Algorithms must be coded to reflect the GPU architecture. To do this can mean to dive into the code and make significant changes to several parts of the algorithm. For the sake of completeness, it should be pointed out that depending on the algorithm may be more advantageous to parallelize it on CPUs rather than on GPUs. Due to their significantly different architectures, CPUs and GPUs can be suited to address different tasks. Therefore, both CPU and GPU parallelism offer particular advantages for particular problems.

The GPU computing model uses both a CPU and a GPU in a heterogeneous co-processing computing model. As a CPU is more effective than a GPU for serial processing, it is used to run the sequential parts of an algorithm, whereas computationally intensive parts are accelerated by the GPU (see **Figure 4**). It should be pointed out that the task of the CPU is not limited to just control the GPU execution. Hybrid CPU/GPU parallelization can also be implemented depending on the algorithm. As for GPU programing, Compute Unified Device Architecture (CUDA) (Nvidia, 2007) and Open Computing Language (OpenCL) (Munshi et al., 2009) offer two different interfaces for programing GPU. OpenCL is an open standard that can be used to program CPUs, GPUs, and other devices from different vendors. CUDA is specific to NVIDIA GPUs.

In the NVIDIA GPU-based architecture, parallelization is obtained through the execution of tasks in a number of stream processors or CUDA cores. Cores are grouped in multiprocessors that execute in parallel. A CUDA core executes a floating point or

integer instruction per clock cycle for a thread and all cores in a streaming multiprocessor execute in a Single Instruction Multiple Thread (SIMT) fashion. All cores in the same group execute the same instruction at the same time. SIMT can be considered an extension of the Single Instruction Multiple Data (SIMD) paradigm: basically, the SIMD paradigm describes how instructions are executed whereas the SIMT paradigm also describes how threads are executed. The code is executed in groups of threads called warps. Device memory access takes a very long time due to the very long memory latency. The parallel programing model of the CUDA architecture provides a set of API that allows programmers to access the underlying hardware infrastructure and to exploit the fine-grained and coarse-grained parallelism of data and tasks. Summarizing, the CUDA execution model (see **Figure 5**) can be described as follow: the GPU creates an instance of the kernel program that is made of a set of threads grouped in blocks in a grid. Each thread has a unique ID within its block and a private memory and registers, and runs in parallel with others threads of the same block. All threads in a block execute concurrently and cooperatively by sharing memory and exchanging data. A block, identified by a unique ID within the block grid, can execute the same kernel program with different data that are read/written from a global shared memory. Each block in the grid is assigned to a streaming multiprocessor in a cyclical manner.

2.2. QUALITY CONTROL

The sequencing technology has been notably improved. Modern sequencers are able to generate hundreds of millions of reads in a single run and the sequencing cost is rapidly decreasing. Despite this improvement, sequencing data are affected by artifacts of different nature that may strongly influence the results of the research. Hence, the ability to assess the quality of read sequences and to properly filter them are major factors that determine the success of a sequencing project. In particular, as for RD methods, both low quality and duplicated read sequences affect the RD signal and consequently the identification of CNV regions.

Different tools have been proposed for quality control of sequencing data such as NGS QC Toolkit (Patel and Jain, 2012),

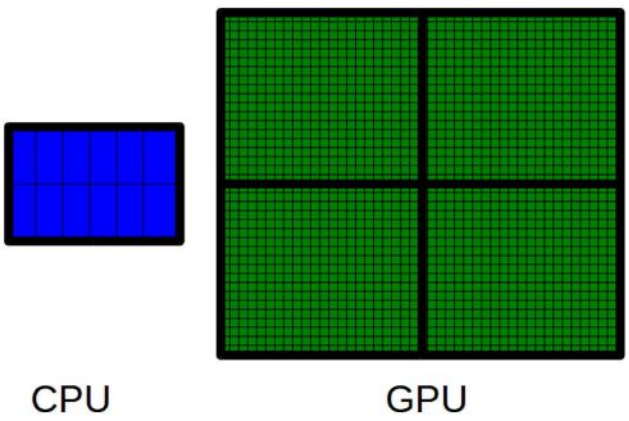


FIGURE 3 | The main difference between CPUs and GPUs is related to the number of available cores. A CPU is a multi-core processor that consists of a few cores. Conversely, a GPU is a many-core processor that consists of thousands of available cores.

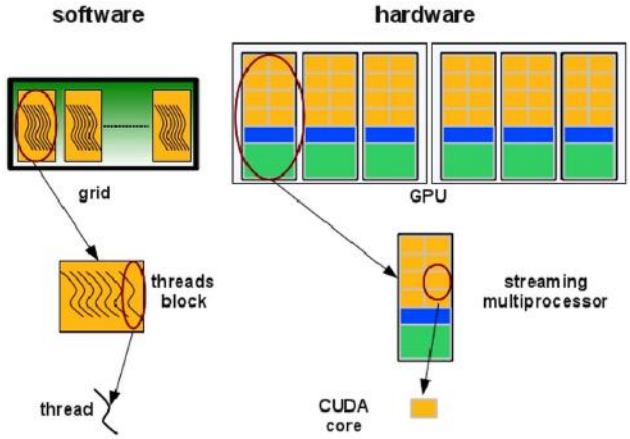


FIGURE 5 | Threads are grouped in blocks in a grid. Each thread has a private memory and runs in parallel with the others in the same block (Manconi et al., 2014a).

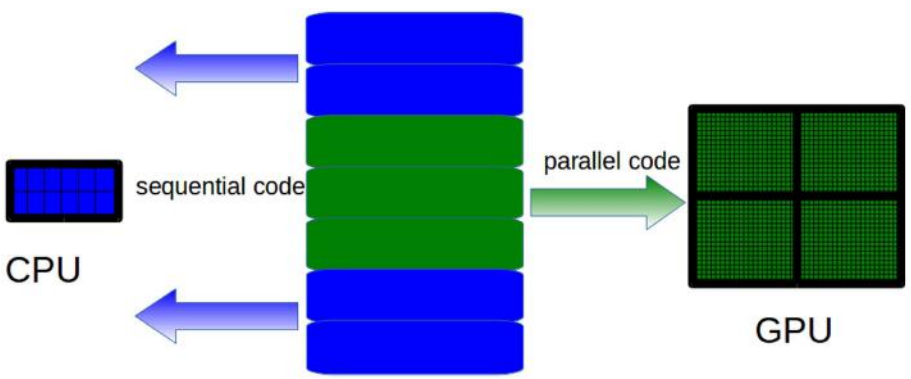


FIGURE 4 | CPUs are more effective than GPUs to run sequential code. Therefore, only the computationally intensive parts of the algorithms must be run on the GPU, whereas sequential parts must be run on CPUs.

HTQC (Yang et al., 2013), FASTX-Toolkit[1], FASTQC[2], and Picard[3]. Most of these tools support both Illumina and 454 platforms, while only some of them support CPU parallelization. It should be pointed out that the artifacts generated during the sequencing process and the massive amount of generated reads make quality control tasks difficult and computationally intensive. The massive parallelization that can be provided by GPUs can be used to deal with these computational tasks. Starting from this assumption, we integrated G-CNV with GPU-based operators to filter low-quality sequences, to mask low-quality nucleotides, and to detect and remove duplicated read sequences. Only the removing of adapter sequences has not yet been implemented on GPU. Currently, these operators are specialized for short-read sequences generated with Illumina platforms.

2.2.1. Filtering low-quality sequences

FASTQ files report quality values for each sequence. Basically, G-CNV parses these files to identify low-quality nucleotides. Nucleotides are classified as of low quality if their quality value is lower than a user-defined threshold. FASTQ files represent quality values using an ASCII encoding. Different encodings are used depending on the Illumina platform. Illumina 1.0 format encodes quality scores from −5 to 62 using ASCII 59 to 126. From Illumina 1.3 and before Illumina 1.8, quality scores ranges from 0 to 62 and are encoded using ASCII 64–126. Starting in Illumina 1.8, quality scores range from 0 to 93 and are encoded using ASCII 33–126.

G-CNV performs filtering in three steps. The first step is performed on CPU, whereas the last two steps are massively parallelized on a single GPU. As for the first step, G-CNV analyses the FASTQ files to detect the Illumina format. Then, the quality values of sequences are decoded according to the detected Illumina format. Finally, G-CNV removes those read sequences that exhibit a percentage of low-quality nucleotides that exceed a user-defined threshold. As a final result, a new FASTQ file is created with the filtered sequences so that the original FASTQ file is preserved.

2.2.2. Masking low-quality nucleotides

G-CNV can also be used to mask low-quality nucleotides. Similarly that for the filtering of low-quality sequences, G-CNV performs masking in three steps. The first step is performed on CPU and it is aimed at detecting the Illumina format. Conversely, the last two steps are massively parallelized on a single GPU and are aimed at decoding the quality values sequences according to the Illumina format, and at masking with an a*N*y symbol those nucleotides with a quality score lower than a user-defined threshold. Then, a new FASTQ file is created with the masked nucleotides.

2.2.3. Removing adapter sequences

In the current release, G-CNV uses *cutadapt* to remove adapter sequences. *Cutadapt* can be used to look for adapter sequences in reads generated with Illumina, 454, and SOLiD HTS machines. Basically, *cutadapt* is able to look for multiple adapters in the 5′ and 3′ ends according to different constraints (e.g., mismatches, indels, minimum overlap between the read and adapter). It can be used to trim or discard reads in which an adapter occurs. Moreover, it allows to automatically discard those reads that after the trimming are shorter than a given user-defined length. All features of *cutadapt* were wrapped in G-CNV.

It should be pointed out that the current release of *cutadapt* is not parallelized. In order to speed up the removing of the adapters, G-CNV splits the original FASTQ files in chunks and runs in parallel an instance of *cutadapt* on each of these chunks. Finally, the output files provided by each instance of *cutadapt* are merged together in a new FASTQ file.

2.2.4. Removing duplicated read sequences

Duplicate reads are one of the most problematic artifacts. These artifacts are generated during the PCR amplification. Ideally, duplicates should have identical nucleotide sequences. However, due to the sequencing errors, they could be nearly identical (Gomez-Alvarez et al., 2009). Alignment-based [e.g., NGS QC Toolkit, SEAL (Pireddu et al., 2011), and Picard MarkDuplicates] and alignment-free [e.g., FastUniq (Xu et al., 2012), Fulcrum (Burriesci et al., 2012), CD-HIT (Li and Godzik, 2006; Fu et al., 2012)] methods have been proposed in the literature to remove duplicated read sequences. Basically, alignment-based methods start from the assumption that duplicated reads will be mapped into a reference genome in the same position. Therefore, in these methods, read sequences are aligned against a reference genome and those reads with identical alignment positions are classified as duplicates. It should be pointed out that the final result is affected by both the alignment constraints and the accuracy of the aligner. In alignment-free methods, read sequences are compared among them according to a similarity measure. The reads with a similarity score lower than a given threshold are classified as duplicated.

G-CNV implements an alignment-free method to remove duplicated read sequences from single-end libraries. Like other tools, it implements a prefix–suffix comparison approach. The algorithm has been devised taking into account the per-base error rates of Illumina platforms. Analysis of short-read datasets obtained with Illumina highlighted a very low rate of indel errors (<0.01%) while the number of occurrences of wrong bases increases with the base position (Dohm et al., 2008). Therefore, G-CNV does not take into account indels and considers as potentially duplicated read sequences those with an identical prefix. Potential duplicated sequences are clustered together (see **Figure 6**), and for each cluster G-CNV compares the suffixes of its sequences. The first sequence of a cluster is taken as a seed and its suffix is compared with those of the other sequences in that cluster. Those sequences identical or very similar to the seed are considered duplicated. Duplicated sequences will be condensed in a new sequence and will be removed from the cluster (see **Figure 7**). Then, the process is iterated for the remaining sequences in the cluster (if any), until the cluster is empty or contains only a read sequence.

In G-CNV, clustering is performed sorting the prefixes of the read sequences. Sorting is performed on a GPU with our CUDA-Quicksort[4,5]. Experimental results show that CUDA-Quicksort

[1] http://hannonlab.cshl.edu/fastx_toolkit/
[2] http://www.bioinformatics.babraham.ac.uk/projects/fastqc/
[3] http://broadinstitute.github.io/picard/

[4] http://sourceforge.net/projects/cuda-quicksort/
[5] Submitted to Concurrency and Computation: Practice and Experience: manuscript CPE-14-0292 entitled "CUDA-Quicksort: an improved GPU-based implementation of Quicksort"

is faster than other available GPU-based implementations of the quicksort. In particular, it results be up to 4 times faster than GPU-Quicksort of Cederman and Tsigas (2008) and up to 3 times faster than the NVIDIA CDP-Quicksort (CUDA toolkit 6.0). As CUDA-Quicksort sorts numerical values, the prefixes must necessarily be subject to a numerical encoding. We devised the encoding with the aim to maximize the length of the prefixes that can be compared. In doing this, read sequence prefixes are subject to a dual numerical encoding. Initially, we encoded the prefixes using a base-5 encoding by replacing each nucleotide with a numerical value ranging from 0 to 4 (i.e., $A \rightarrow 0$, $C \rightarrow 1$, $G \rightarrow 2$, $T \rightarrow 3$, $N \rightarrow 4$). Using CUDA-Quicksort to sort items represented with *64 bit unsigned long long int* data type, prefixes of up to 19 nucleotides can be sorted. A longer prefix will exceed the limit for this type of data. However, it is possible to exceed this constraint using a different numerical base to represent the prefixes. In particular, using the base-10, it is possible to represent a number consisting of 27 digits with a *64 bit unsigned long long int* (see **Figure 8**). Therefore, G-CNV applies this second encoding to maximize the length of the prefixes used for clustering.

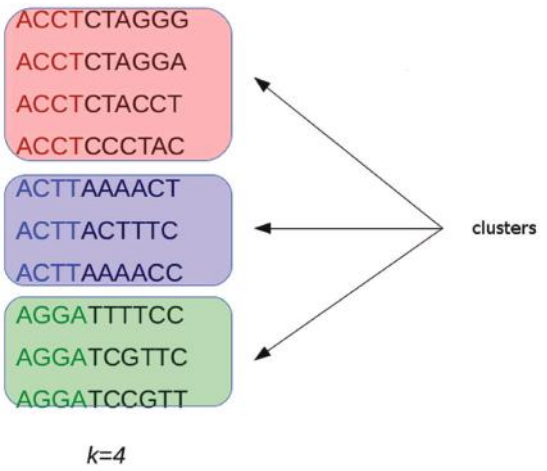


FIGURE 6 | Short-reads with an identical prefix (of fixed length *k*) are clustered together as potential duplicated sequences. This approach takes into account the error rates of Illumina platforms. Analysis performed on short-reads generated with these sequencing platforms highlighted that the number of wrong bases increases with the base position.

After that the reads have been clustered G-CNV compares their suffixes. This step requires a base-per-base comparison of the nucleotides of the seed read sequence with those of the other reads in a cluster. This approach can require a very high number of base–base comparisons. Let N be the length of the suffixes, and let m be the allowed number of mismatches. In the best case, m comparisons must be performed to classify two sequences as not duplicated. In the worst case, $N–m$ comparisons must be performed to classify two sequences as duplicated. Apart from the high number of comparisons required, this approach is not adapted to be efficiently implemented on GPUs. As GPUs adopt the SIMT paradigm, each thread in a block must perform the same operation on different data. Then, G-CNV implements a different comparison method. Suffixes are split into fixed length chunks. Subsequence of each chunk is subjected to the same dual numerical encoding used to represent the prefixes for clustering. Then for each cluster, the numerical difference between the i-th chunk of the seed and the related chunk of the other suffixes in a cluster is calculated (see **Figure 9**). The order of magnitude of the difference provides information about the position of the leftmost different nucleotides. Then, the subsequences are cut corresponding to the mismatch position. The rightmost parts of the mismatch position are maintained and the process is re-iterated.

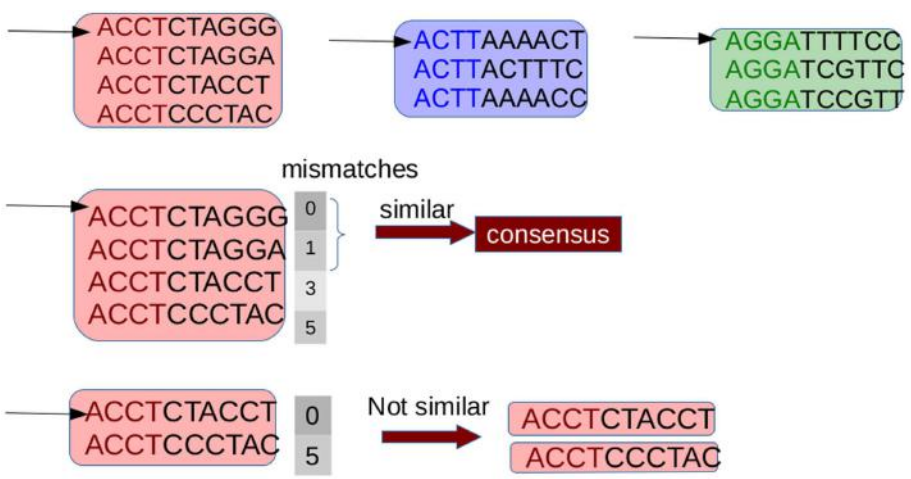


FIGURE 7 | Suffixes of sequences in a cluster are compared to identify the duplicates. The first read is taken as a seed and its suffix is compared with those of the other ones. Sequences with a number of mismatches lower than a given threshold are considered duplicates of the seed. These sequences are removed from the cluster and are represented with a consensus sequences. Then the process is repeated until the cluster is empty or consists of a single sequence.

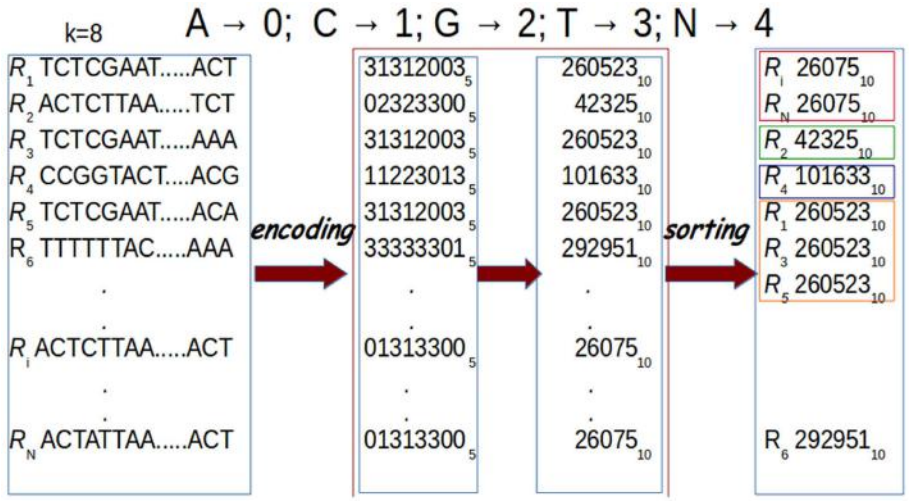


FIGURE 8 | Prefixes are subject to a dual encoding. As for the first encoding, each nucleotide in a prefix is represented with a numerical value from 0 to 4 ($A \rightarrow 0$, $C \rightarrow 1$, $G \rightarrow 2$, $T \rightarrow 3$, $N \rightarrow 4$). Then, these numerical representations are encoded using base-10. Finally, sorting is performed for clustering. In the figure, prefixes of length $k = 8$ are represented.

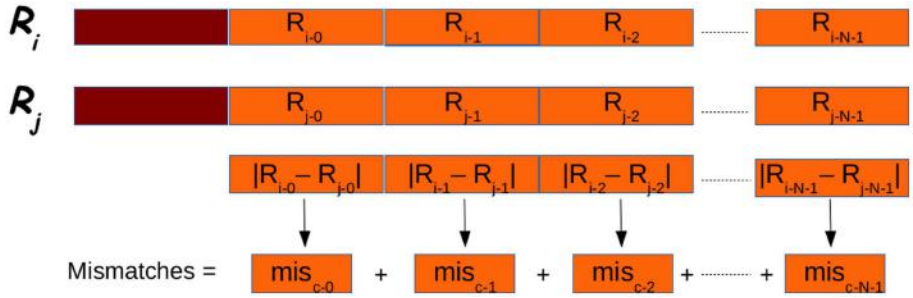


FIGURE 9 | Suffixes (in orange in the figure) are analyzed in chunks. Each chunk is subject to the dual encoding used for prefixes (in red in the figure). The overall number of mismatches if obtained summing the partial number of mismatches obtained for each chunk.

2.3. MAPPING

It is widely known that mapping of short-read sequences is computationally onerous. Several tools have been devised to deal with short-read mappings. Without claiming to be exhaustive, let us cite some of the most popular solutions, i.e., MAQ (Li et al., 2008), RMAP (Smith et al., 2008, 2009), Bowtie (Langmead et al., 2009), BWA (Li and Durbin, 2009), CloudBurst (Schatz, 2009), SOAP2 (Li et al., 2009b), and SHRiMP (Rumble et al., 2009; David et al., 2011). A comparative study aimed at assessing the accuracy and the runtime performance of different cutting-edge next-generation sequencing read alignment tools highlighted that among all, SOAP2 was the one that showed the higher accuracy (Ruffalo et al., 2011). Exhaustive review of the tools cited above can be found in Bao et al. (2011).

In general, the mentioned solutions exploit some heuristics to find a good compromise between accuracy and running time. Recently, the GPU-based short-read mapping tools Barracuda (Klus et al., 2012), CUSHAW (Liu et al., 2012b), SOAP3 (Liu et al., 2012a), and SOAP3-dp have been successfully proposed to the scientific community. In particular, SOAP3-dp aligns the read sequences in two steps. As for the first step, it looks for ungapped alignments with up to four mismatches without using heuristics. As for the second step, it uses the dynamic programing to look for gapped alignments. Compared with BWA, Bowtie2 (Langmead and Salzberg, 2012), SeqAlto (Mu et al., 2012), GEM (Marco-Sola et al., 2012), and the previously mentioned GPU-based aligners, SOAP3-dp is two to tens of times faster, while maintaining the highest sensitivity and lowest false discovery rate on Illumina reads with different lengths.

Starting from the previous analysis, we decided to use SOAP3-dp to support read mapping in G-CNV. G-CNV allows to set different parameters of SOAP3-dp that can be useful to properly generate alignments for RD methods. Apart from the constraints on the allowed mismatches, G-CNV allows to set SOAP3-dp parameters able to filter out alignments that are not of interest for the specific RD method. In particular, as different methods presented in the literature filter alignments using different quality mapping scores, G-CNV allows to set a quality mapping threshold on the

alignments that must be reported. To set these constraints, G-CNV needs to be able to access the SOAP3-dp files to change the initialization file. Moreover, a short-read may be uniquely aligned or can be aligned to multiple positions onto a genome. Multiple mappings can be related to the alignment constraints or to the nature of the sequenced read. A read sequence can be aligned to multiple positions, as it has been sequenced from repetitive regions or regions of segmental duplication (Abyzov et al., 2011). In the former case, alignments are characterized by different alignment scores, whereas in the latter case, they are expected to have equal or very similar scores. A common approach to take into account multiple mappings is to randomly select a best alignment. G-CNV allows to report only unique best alignments or a random best alignment.

2.4. RD SIGNAL

The RD signal depends on the size of the observing window. As methods proposed in the literature suggest different approaches to estimate the window size, G-CNV does not impose it. In G-CNV, the window size is a parameter that must be set by the user.

G-CNV builds the RD signal in two steps. Initially, G-CNV analyses the genome sequences to build a GC-content signal according to the fixed window size. A GC-signal for each genome sequence will be built. Then, G-CNV splits the mapping for each chromosome sequences, identifies the window where the mappings fall, and calculates a raw RD signal. By default, the window related to each alignment is identified considering the center of the read. Finally, G-CNV corrects the RD signal with the same approach proposed in Yoon et al. (2009) that adjust the RD by using the observed deviation of RD for a given GC percentage according to the following equation:

$$RD'_{w_i} = \frac{\overline{RD}}{\overline{RD}_{GC_{w_i}}} \cdot RD_{w_i} \quad (1)$$

where RD_{wi} is the RD for the i-th window to be corrected, $\overline{RD}$ is the average RD signal, $\overline{RD}_{GC_{w_i}}$ is the average RD signal calculated on the windows with the GC-content found in the i-th window, and RD'_{w_i} is the corrected RD for the i-th window.

2.5. HARDWARE AND SOFTWARE REQUIREMENTS

G-CNV has been designed to work with NVIDIA GPU cards based on the most recent Kepler architecture. G-CNV works on Linux-based systems equipped with CUDA (release $\geq$6.0). We tested it on the NVIDIA Kepler architecture-based k20c card. Experiments have been carried out using the last release of *soap3-dp* (rel. 2.3.177) and of *cutadapt* (rel. 1.7.1).

3. RESULTS

We performed different experiments aimed at assessing the performance of G-CNV. In particular, we assessed its performance when used to filter low-quality sequences, to mask low-quality nucleotides, to remove adapter sequences, to remove duplicated reads, and to calculate the RD signal. Since G-CNV performs the alignments running *SOAP3-dp*, we deemed not relevant to assess the performance of G-CNV in this task. We invite the readers to refer the *SOAP3-dp* manuscript for an in-depth analysis of the performance of the aligner. Similarly, as G-CNV uses the well-known tool *cutadapt* to remove adapter sequences, we did not perform tests aimed at assessing its reliability in this task. However, we performed experiments aimed at assessing the benefits of the parallelization of *cutadapt* provided with G-CNV.

Experiments have been carried out on both synthetic and real-life libraries. Synthetic reads have been used to assess and compare with other tools the reliability of G-CNV, whereas real-life data to assess and compare its performance in terms of both computing time and memory consumption.

Synthetic reads have been generated from the build 37.3 of the human genome using the *Sherman* simulator[6]. *Sherman* has been devised to simulate HTS datasets for both bisulfite sequencing and standard experiments. To mimic real data, it generates synthetic data using an error rate curve that follows an exponential decay model. We used *Sherman* to generate a single-end synthetic library consisting of 1 millions of 100 bp reads. Library has been generated simulating a sequencing error of 2% and contaminating the reads with the Illumina single-end adapter 1 (i.e., ACACTCTTTC CCTACACGACGCTGTTCCATCT). The contamination has been simulated with a normal distribution of fragment sizes. Moreover, since *Sherman* generates identical quality scores for all reads, we modified them to generate a 3% of low-quality nucleotides (PHRED value $\leq$20) and a 9% of low-quality sequences. In the following of the manuscript, we will refer to this dataset as the *S1* library.

Since *Sherman* does not permit to control the percentage of duplicates, we modified the simulated reads in *S1* to generate a new synthetic library (*S2*) consisting of 30% of duplicated sequences. Read sequences have been duplicated simulating a sequencing error of 2%. The library *S1* has been used to assess the reliability of G-CNV in the task of filtering low-quality sequences and masking low-quality nucleotides, whereas *S2* has been used to assess the reliability of G-CNV in the task of removing duplicated sequences.

As for real-life data, experiments have been performed on different libraries generated with Illumina platforms: (i) SRR001220 consisting of 3.3 millions of 94 bp reads; (ii) SRR001205 consisting of 9.7 millions of 47 bp reads; (iii) SRR005720 consisting of 26.2 millions of 36 bp reads; and (iv) SRR921889 consisting of 50 millions of 100 bp reads (see **Table 1**).

Moreover, with the aim to simulate a CNV pre-processing detection analysis, we simulated two high coverage (30x) whole genome sequencing experiments. The first experiment have been simulated generating 37 synthetic libraries consisting of 25 millions of 100 bp reads, and the second generating 9 synthetic libraries consisting of 100 millions of 100 bp reads. All libraries have been generated according to the same constraints used to generate *S1*. In the following of the manuscript, we will refer to these datasets as *HCS1* and *HCS2*.

In the following of this section, we describe the different experiments and present results. Finally, we briefly resume the hardware and software configuration used for experiments.

[6]http://www.bioinformatics.babraham.ac.uk/projects/sherman/

Table 1 | Real-life datasets.

Dataset	Library layout	Reads (M)	Read size (bp)	Organism	Instrument
SRR001220	Single	3.3	94	*Homo sapiens*	Illumina Genome Analyzer II
SRR001205	Single	9.7	47	*Homo sapiens*	Illumina Genome Analyzer II
SRR005720	Paired	26.2	36	*Homo sapiens*	Illumina Genome Analyzer
SRR921889	Single	50.0	100	*Mus musculus*	Illumina HiSeq 2000

The first column reports the name of the dataset. The second column reports the library layout. The third and fourth columns report the size of the dataset and the length of the reads, respectively. Organism and sequencing instrument are reported in column fifth and sixth.

Table 2 | Tools settings used to filter low-quality sequences.

Tool	
G-CNV	−mf 20 −pf 90
FASTX-Toolkit	-Q33 -q 20 -p 90
NGS QC Toolkit[a]	N A -l 90 -s 20
NGS QC Toolkit[b]	N A -l 90 -s 20 -c 12

Both FASTX-Toolkit and NGS QC Toolkit consist of different commands. As for FASTX-Toolkit, experiments have been performed using the fastq_quality_filter command, whereas the IlluQC_PRLL has been used for NGS QC Toolkit. The table reports the settings used to run NGS QC Toolkit without exploiting parallelization (NGS QC Toolkit[a]) and parallelized on 12 CPU cores (NGS QC Toolkit[b]).

3.1. FILTERING LOW-QUALITY SEQUENCES

To assess G-CNV in the task of filtering low-quality read sequences, we compared its performance with those of FASTX-Toolkit and NGS QC Toolkit. Experiments have been performed setting parameters with the aim to filter those sequences with a percentage of low-quality (PHRED score <20) bases >10% (see **Table 2**).

A first experiment has been performed on the *S1* synthetic library aimed at assessing and comparing the reliability of G-CNV with the other tools. As expected, all tools have been able to filter all low-quality sequences. The same experiment has been performed on the real-life libraries aimed at assessing the performance of G-CNV in terms of both computing time and memory consumption. It should be pointed out that FASTX-Toolkit does not support parallelization whereas in NGS QC Toolkit parallelization has been implemented in multiprocessing and multithreaded ways. Multiprocessing parallelization was implemented to process multiple files in parallel whereas multithreading parallelization to process in parallel a single file. The FASTQ file is split into chunks, processed in parallel, and results are merged at the end. With the aim to provide an in-depth comparison among all tools and to assess as NGS QC Toolkit can scale increasing the CPU cores, we initially run the experiments without using parallelization, then experiments have been performed parallelizing the computation on 12 CPU cores.

It should be pointed out that FASTX-Toolkit does not provide support for paired-end libraries. Therefore, it has not been possible to test it with the *SRR005720* dataset. Experimental results show that G-CNV is most effective than the other tools in terms of computing time. **Table 3** reports computing time and peak of memory required by G-CNV, FASTX-Toolkit, and NGS QC Toolkit to analyze the different datasets. G-CNV has been 12.4x/7.8x/NA/21.4x faster than FASTX-Toolkit and 24x/21x/26.5x/28.3x faster than NGS QC Toolkit parallelized on 12 CPU cores to filter the read sequences of the SRR001220/SRR001205/SRR005720/SRR921889 dataset. Obviously, the performance of G-CNV improves notably when compared with those of NGS QC Toolkit executed without parallelization. In this case, G-CNV has been 154x/120x/125x/175x faster than NGS QC Toolkit to analyze the SRR001220/SRR001205/SRR005720/SRR921889 dataset.

For the sake of completeness, it should be pointed out that NGS QC Toolkit automatically also generates statistics for quality check. Therefore, the computing time reported from NGS QC Toolkit takes into account also the time required to perform these operations.

As for the memory consumption, FASTX-Toolkit is undoubtedly the most effective tool. Conversely, G-CNV requires more memory than the other tools. Its performance is only comparable with those of NGS QC Toolkit executed in parallel for the *SRR001220* and *SRR001205* datasets. Experimental results show that the memory required by G-CNV increases with the size of the analyzed library. This is mainly due to the fact that to massively parallelize the computation G-CNV loads into the memory as many as possible read sequences to maximize the occupancy of the grid of the GPU.

Finally, we used G-CNV to filter the low-quality sequences of the *HCS1* and *HCS2* datasets. Filtering has been performed in ~20 min for the *HCS1* and in ~34 min for *HCS2*. As for the memory consumption, G-CNV required 5.7 GB to analyze *HCS1* and 20.5 GB for *HCS2*.

3.2. MASKING LOW-QUALITY NUCLEOTIDES

The performance of G-CNV in the task of masking low-quality nucleotides have only been compared with those of FASTX-Toolkit. NGS QC Toolkit does not provide support for this operator. G-CNV and FASTX-Toolkit have been run to mask with a a*N*y symbol the nucleotides with a PHRED quality score <20 (see **Table 4**). Experiments performed on the *S1* synthetic library shown that both tools have been able to mask all low-quality sequences. Experiments performed on real-life libraries show that G-CNV outperforms notably FASTX-Toolkit in terms of computing time. Results reported in **Table 5** show that G-CNV has been 12x/6.8x/5x/13.8x faster than FASTX-Toolkit to analyze the SRR001220/SRR001205/SRR005720/SRR921889 dataset. As previously highlighted, FASTX-Toolkit does not support paired-end reads. However, as for the task of masking low-quality nucleotides, it can be separately used on both the forward and the reverse read sequences. Then, as for the *SRR005720* dataset **Table 5** reports the overall computing time required by FASTX-Toolkit to analyze both files.

Table 3 | Performance evaluation to filter low-quality sequences.

Tool	Dataset	Filtered seq. (%)	Time	Memory
G-CNV	SRR001220	95.3	5 s	0.9 GB
	SRR001205	98.3	11 s	1.4 GB
	SRR005720	74.7	48 s	4.5 GB
	SRR921889	7.9	1 min 10 s	10.5 GB
FASTX-Toolkit	SRR001220	95.3	1 min 2 s	256 KB
	SRR001205	98.3	1 min 19 s	256 KB
	SRR005720	–	–	–
	SRR921889	7.9	17 min 10 s	256 KB
NGS QC Toolkit[a]	SRR001220	95.3	12 min 52 s	0.21 GB
	SRR001205	98.3	22 min	0.18 GB
	SRR005720	74.7	1 h 40 min	0.26 GB
	SRR921889	7.9	3 h 25 min	0.22 GB
NGS QC Toolkit[b]	SRR001220	95.3	2 min	1.4 GB
	SRR001205	98.3	3 min 52 s	1.4 GB
	SRR005720	74.7	21 min	1.3 GB
	SRR921889	7.9	33 min	1.9 GB

The first and the second column of the table report the tool and the analyzed library, respectively. The third column the percentage of filtered reads. Column fourth reports the computing time required to analyze the different libraries. The fifth column the peak of memory required to perform the analysis.

Table 4 | Tools settings used to mask low-quality nucleotides.

Tool	
G-CNV	-m 20
FASTX-Toolkit	-Q33 -q 20 -r N

As for FASTX-Toolkit experiments have been performed using the fastq_quality_masker command.

Table 5 | Performance evaluation to mask low-quality nucleotides.

Tool	Dataset	Masked nucl. (%)	Time	Memory
G-CNV	SRR001220	24.2	5 s	0.94 GB
	SRR001205	43.6	10 s	1.38 GB
	SRR005720	21.8	52 s	3.88 GB
	SRR921889	3	1 min 15 s	12 GB
FASTX-Toolkit	SRR001220	24.2	1 min	256 KB
	SRR001205	43.6	1 min 8 s	256 KB
	SRR005720	21.8	4 min 22 s	256 KB
	SRR921889	3	17 min 20 s	256 KB

The first and the second column of the table report the tool and the analyzed library, respectively. The third column the percentage of masked nucleotides. Column fourth reports the computing time required to analyze the different libraries. The fifth column the peak of memory required to perform the analysis.

Table 6 | Tools settings used to remove adapter sequences.

Tool	
G-CNV[a]	–ca-a ACACTCTTTCCCTACACGACGCTGTTCCATCT
G-CNV[b]	–ca-a ACACTCTTTCCCTACACGACGCTGTTCCATCT –ca-t 12
FASTX-Toolkit	–Q33 -a ACACTCTTTCCCTACACGACGCTGTTCCATCT
NGS QC Toolkit[a]	≪ ADAPTER FILE≫ A
NGS QC Toolkit[b]	≪ ADAPTER FILE≫ A -c 12

As for FASTX-Toolkit experiments have been performed using the fastx_clipper command, whereas the IlluQC_PRLL has been used for NGS QC Toolkit. In the table have been reported the settings used to run G-CNV[a] and NGS QC Toolkit[a] without exploiting the parallelization and in multithreading way (G-CNV[b] and NGS QC Toolkit[b]). The table shows the settings used to remove the Illumina Single-End Adapter 1. As for the SRR005720 dataset, settings have been modified to remove the Illumina paired-end adapters.

As for the high coverage-simulated sequencing experiments, G-CNV masked the low-quality nucleotides of *HCS1* in ~23 min using 7 GB of memory, whereas required ~39 min and 21.9 GB of memory for *HCS2*.

3.3. REMOVING ADAPTER SEQUENCES

As for the task of removing adapter sequences, G-CNV has been compared with both FASTX-Toolkit and NGS QC Toolkit. To assess the advantages of the implemented parallelization of *cutadapt*, we initially performed experiments running G-CNV without exploiting the parallelization, subsequently parallelizing the computation on 12 CPU cores. Tool settings used to perform these experiments are reported in **Table 6**.

Table 7 reports results obtained analyzing the real-life libraries. Results show that the performance of G-CNV improves notably with parallelization. With parallelization G-CNV has been 6.7x/6.4x/23.4x/2.8x faster to remove the adapter sequences from the SRR001220/SRR001205/SRR005720/SRR921889 dataset. Moreover, G-CNV parallelized on 12 CPU cores resulted to be 18.2x/11x/–/9.4x faster than FASTX-Toolkit and 11.8x/7.3x/58.3x/6.3x NGS QC Toolkit used exploiting the parallelization to remove the adapters from the SRR001220/SRR001205/SRR005720/SRR921889 dataset. Obviously, also for this task the performance of G-CNV improves when compared with NGS QC Toolkit used without parallelization. In this case, G-CNV resulted be 48x/40x/173x/38x faster than NGS QC Toolkit to analyze the SRR001220/SRR001205/SRR005720/SRR921889 dataset. As for the memory consumption, FASTX-Toolkit provides better performance than the other tools. However, G-CNV outperforms NGS QC Toolkit. As FASTX-Toolkit does not support paired-end libraries, it has not been used to analyze the *SRR005720* dataset.

Finally, when used to remove adapters from the *HCS1*, G-CNV required ~50 min and 250 MB of memory, whereas it required ~3 h 20 min and 920 MB for *HCS2*.

3.4. REMOVING DUPLICATED READ SEQUENCES

To assess the performance of G-CNV in the task of removing duplicated sequences, we compared its performance with those of Fulcrum. G-CNV implements a very similar algorithm to that implemented in Fulcrum. In particular, similarly to our tool, Fulcrum clusters together the reads with a similar prefix and looks for duplicates in the same cluster.

Table 7 | Performance evaluation to remove adapter sequences.

Tool	Dataset	Time	Memory
G-CNV[a]	SRR001220	1 min 14 s	17 MB
	SRR001205	2 min 46 s	21 MB
	SRR005720	8 min 12 s	26 MB
	SRR921889	17 min 11 s	20 MB
G-CNV[b]	SRR001220	11 s	0.4 GB
	SRR001205	26 s	0.46 GB
	SRR005720	21 s	0.33 GB
	SRR921889	6 min 10 s	0.84 GB
FASTX-Toolkit	SRR001220	3 min 21 s	516 KB
	SRR001205	4 min 47 s	516 KB
	SRR005720	–	–
	SRR921889	57 min 40 s	516 KB
NGS QC Toolkit[a]	SRR001220	8 min 52 s	217 MB
	SRR001205	17 min 30 s	189 MB
	SRR005720	1 h 48 min	269 MB
	SRR921889	3 h 55 min	226 MB
NGS QC Toolkit[b]	SRR001220	2 min 10 s	1.6 GB
	SRR001205	3 min 10 s	1.3 GB
	SRR005720	20 min 24 s	1.13 GB
	SRR921889	39 min	1.6 GB

The first and the second column of the table report the tool and the analyzed library, respectively. Column third reports the computing time required to analyze the different libraries. The fourth column the peak of memory required to perform the analysis.

Table 8 reports the main parameters that have been used for the experiments. Experiments on the synthetic *S2* library have been performed clustering reads according to a prefix length of 25 bp and looking for identical sequences (i.e., 0 mismatches) and nearly identical sequences with up to 1 mismatch. Results reported in **Table 9** show that both tools have been able to identify the synthetic duplicate sequences. It should be pointed out that *S2* has been built avoiding to generate mismatches among the duplicated sequences in their first 25 bp. As for tests on real-life data, we performed experiments on the larger *SRR921889* dataset. Experiments have been aimed at assessing the performance of G-CNV to remove duplicated sequences according to different constraints. In particular, we performed the experiments on both G-CNV and Fulcrum to cluster sequences according to a prefix size of 10 and 25 bp and to look for duplicated sequences with up to 1 and up to 3 mismatches. Experimental results are reported in **Table 10**. For each experiment were reported the percentage of removed sequences, the computing time and the peak of memory required for the analysis. Results show that both tools remove a similar percentage of duplicated sequences. However, as for the computing time, G-CNV outperforms Fulcrum in all experiments. It should be pointed out that Fulcrum automatically parallelize the computation on all available CPU cores. Therefore, the computing times reported in the table have been obtained running Fulcrum parallelized on 12 CPU cores. Results show that the computing time required by G-CNV depends on both the number of allowed mismatches and the prefix size. The number of sequences that will be classified as duplicated increases with the number of allowed mismatches. Therefore, increasing this value may involve a lower number of sequences comparison. Moreover, the size of a cluster depends on the prefix length. Typically, the size of the clusters increases as the prefix length decreases involving more sequences comparison.

For the sake of completeness, G-CNV performed the clustering step in ~2 s for both length of the prefixes, whereas Fulcrum required 13 min to cluster the reads according to a prefix of 10 bp and 56 min to cluster the reads according to prefix length of 25 bp. However, it should be pointed out that G-CNV can not be used to cluster reads with a prefix longer than 27 bp. Moreover, the clustering phase implemented by G-CNV requires that all prefixes

Table 8 | Tools settings used to remove duplicated sequences.

Tool	
G-CNV	-D ≪*mis*≫ -p ≪*pref*≫
Fulcrum	-b ≪*pref*≫ -s -t s -c ≪*mis*≫

We performed different experiments with different values for both the prefix length and the allowed mismatches. Specific values for the prefixes ≪pref≫ and the allowed mismatches ≪mis≫ are reported in the tables of the results.

Table 9 | Performance evaluation to remove duplicated sequences from the synthetic S2 library.

Tool	Dataset	Mismatches	Percentage removed
G-CNV	S2	0	0
	S2	1	30.1
Fulcrum	S2	0	0
	S2	1	30.6

The first column reports the tool. The second column reports the dataset. Columns third and forth report the allowed mismatches and the percentage of removed duplicated sequences.

Table 10 | Performance evaluation to remove duplicated sequences from the real life dataset SRR921889.

Tool	Prefix	Mismatches	Percentage removed	Time	Memory
G-CNV	10	1	11.2	2 h	17.3 GB
	10	3	11.5	1 h 50 min	17.3 GB
	25	1	11.9	16 min	17.3 GB
	25	3	12.1	8 min	17.3 GB
Fulcrum	10	1	11.3	4 h 01 min	1.6 GB
	10	3	11.4	3 h 23 min	1.6 GB
	25	1	11.6	1 h 24 min	1.6 GB
	25	3	11.9	1 h 33 min	1.6 GB

The first column reports the tool. The second column reports the length of the prefixes used for clustering. Column third reports the allowed mismatches. The fourth column reports percentage of removed sequences. Columns fifth and sixth report the computing time and the memory consumption, respectively.

will be loaded into the memory of the GPU device. This implies a constraint on the size of the analyzed library, which depends on the memory of the GPU. As for the memory consumption, G-CNV undoubtedly requires more memory than Fulcrum. Also in this case, the high memory consumption is due to the need of maximize the occupancy of the grid of the GPU.

Finally, we performed different experiments on the *HCS1* dataset. Experiments have been performed to cluster the reads according to a prefix length of 15 and 27 bp and to look for duplicated with up to 1 and to 3 mismatches. Results are reported in **Table 11**.

Table 11 | Performance evaluation to remove duplicated sequences from the synthetic HCS1 dataset.

Mismatches	Prefix	Time	Memory
1	15	12 h 7 min	8.8 GB
	27	5 h 33 min	6.6 GB
3	15	3 h 20 min	8.7 GB
	27	1 h 30 min	5.7 GB

The first column reports the allowed mismatches. The second column reports the length of the prefix used to cluster the reads. Columns third and fourth report the computing time and memory consumption, respectively.

3.5. GENERATING THE RD SIGNAL

As there are no other specialized tools to generate the RD signal, we cannot assess and compare the performance of G-CNV with other tools. However, we used the *FastQC* tool[7] to assess the reliability of G-CNV in the task of calculating the GC-content that is used to normalize the RD signal. FastQC is a tool that provides some quality control checks on HTS data. In particular, it is able to calculate the distribution of the per-sequence GC-content of the analyzed read sequences.

As G-CNV calculates the GC-content of each observed window in the genome sequences, we generated a synthetic library using as reads the subsequences observed with a window of 100 bp along the MT chromosome of the human genome (build 37.3). Then, we used FastQC to analyze the GC-content of these sequences and compared the results with those generated by G-CNV. Both tools provided the same distribution of the GC-content. It should be pointed out that it was not possible to compare the results with those of FASTX-Toolkit and NGS QC Toolkit as both determine only the per-base GC-content. We did not compare the time required by G-CNV with that required by *FastQC* as it automatically performs several quality checks.

Moreover, to assess the performance of G-CNV to generate a RD signal, we simulated an alignment SAM file on the human genome (build 37.3). The alignment has been simulated by assuming a sequencing experiment on the genome with coverage 30x. We did not simulate sequencing and alignment errors. The SAM file was generated by assuming an ideal aligner able to map the reads uniquely and without errors. In fact, these errors do not affect the computing time to generate the RD signal; they affect the detection of CNVs. However, as for this experiment, we have been mainly interested to assess the computing time of G-CNV in the task of generating the RD signal. G-CNV generated the RD signal with an observing window of length 100 in less than 1 h 56 min and required 10.4 GB of memory. As for the memory used by G-CNV, it depends on the number of alignments in the analyzed genome sequence. G-CNV generates the RD signal analyzing separately the genome sequences. To maximize the parallelization, as many as possible alignments on the analyzed genome sequence are loaded into the GPU.

3.6. HARDWARE AND SOFTWARE CONFIGURATION

Experiments described hereinafter have been carried out on a 12 cores Intel Xeon CPU E5-2667 2.90 GHz with 128 GB of RAM and an NVIDIA Kepler architecture-based Tesla k20c card with 0.71 GHz clock rate and equipped with 4.8 GB of global memory.

[7] http://www.bioinformatics.babraham.ac.uk/projects/fastqc/

4. DISCUSSION

Different RD-based methods and tools have been proposed in the literature to identify CNVs. Typically, these tools do not support most of the preparatory operations for RD analysis. Therefore, a specific analysis pipeline must be built with different third-party tools. G-CNV allows to build the analysis pipeline required to process short-read libraries for RD analysis according to different constraints. However, in our opinion, the added value of G-CNV is the fact that almost all operations are performed on GPUs. In fact, these are data-intensive operations that may require an enormous computing power. GPUs are increasingly used to deal with computational intensive problems. The low cost for accessing the technology and their very high computing power is facilitating the GPUs success. Experimental results show that G-CNV is able to efficiently run the supported operations. However, it should be pointed out that the current release of G-CNV still has some limitations and/or constraints. In particular, as for removing duplicates, there are two main limitations of our algorithm. As for the former, the current release of G-CNV supports removal of duplicates only for single-end reads. As for the latter, there exists a constraint on the clustering phase. Sorting requires that all prefixes will be loaded into the memory of the GPU device. This implies a constraint on the size of the analyzed library, which depends on the memory of the GPU. With a GPU card equipped with 4.8 GB of global memory, libraries of up to 220 M reads can be analyzed. A solution to overcome this constraint is to parallelize the sorting on multiple GPU devices. We are currently working to adapt CUDA-Quicksort to run on multiple GPUs. Although CUDA-Quicksort resulted be the fastest GPU-based implementation of the quicksort algorithm, the Thrust Radix Sort is currently the fastest GPU-based sorting algorithm. However, as for clustering, we adapted and used our CUDA-Quicksort as it has been designed to be easily modified to scale on multiple GPUs. Moreover, we deem that the overall performance of G-CNV can be improved by implementing the trimming of the adapters on GPUs.

AUTHOR CONTRIBUTIONS

Conceived the tool: AM. Conceived and designed the experiments: AM, LM. Performed the experiments: AM, MG, EM, GA.

Analyzed the data: AM, AO, EM, GA, MM, MG, LM. Wrote the manuscript: AM. Revised the manuscript: AM, GA, LM. Wrote the program: AM. Generated the synthetic data: AM. Coordinated the project: LM.

ACKNOWLEDGMENTS

We thank Dario Deledda for his advice and comments on the manuscript. In addition, we thank the reviewers for their very useful and constructive comments and suggestions. *Funding*: The work has been supported by the Italian Ministry of Education and Research through the Flagship InterOmics (PB05) and HIRMA (RBAP11YS7K) projects, and the European MIMOmics (305280) project.

REFERENCES

Abel, H. J., Duncavage, E. J., Becker, N., Armstrong, J. R., Magrini, V. J., and Pfeifer, J. D. (2010). Slope: a quick and accurate method for locating non-snp structural variation from targeted next-generation sequence data. *Bioinformatics* 26, 2684–2688. doi:10.1093/bioinformatics/btq528

Abyzov, A., and Gerstein, M. (2011). Age: defining breakpoints of genomic structural variants at single-nucleotide resolution, through optimal alignments with gap excision. *Bioinformatics* 27, 595–603. doi:10.1093/bioinformatics/btq713

Abyzov, A., Urban, A. E., Snyder, M., and Gerstein, M. (2011). Cnvnator: an approach to discover, genotype, and characterize typical and atypical CNVs from family and population genome sequencing. *Genome Res.* 21, 974–984. doi:10.1101/gr.114876.110

Alkan, C., Coe, B. P., and Eichler, E. E. (2011). Genome structural variation discovery and genotyping. *Nat. Rev. Genet.* 12, 363–376. doi:10.1038/nrg2958

Bao, S., Jiang, R., Kwan, W., Wang, B., Ma, X., and Song, Y.-Q. (2011). Evaluation of next-generation sequencing software in mapping and assembly. *J. Hum. Genet.* 56, 406–414. doi:10.1038/jhg.2011.43

Bochukova, E. G., Huang, N., Keogh, J., Henning, E., Purmann, C., Blaszczyk, K., et al. (2009). Large, rare chromosomal deletions associated with severe early-onset obesity. *Nature* 463, 666–670. doi:10.1038/nature08689

Burriesci, M. S., Lehnert, E. M., and Pringle, J. R. (2012). Fulcrum: condensing redundant reads from high-throughput sequencing studies. *Bioinformatics* 28, 1324–1327. doi:10.1093/bioinformatics/bts123

Carter, N. P. (2007). Methods and strategies for analyzing copy number variation using DNA microarrays. *Nat. Genet.* 39, S16–S21. doi:10.1038/ng2028

Cederman, D., and Tsigas, P. (2008). "A practical quicksort algorithm for graphics processors," in *Algorithms-ESA 2008* (Berlin: Springer), 246–258.

Chen, K., Wallis, J. W., McLellan, M. D., Larson, D. E., Kalicki, J. M., Pohl, C. S., et al. (2009). Breakdancer: an algorithm for high-resolution mapping of genomic structural variation. *Nat. Methods* 6, 677–681. doi:10.1038/nmeth.1363

Chiang, D. Y., Getz, G., Jaffe, D. B., O'Kelly, M. J., Zhao, X., Carter, S. L., et al. (2008). High-resolution mapping of copy-number alterations with massively parallel sequencing. *Nat. Methods* 6, 99–103. doi:10.1038/nmeth.1276

David, M., Dzamba, M., Lister, D., Ilie, L., and Brudno, M. (2011). Shrimp2: sensitive yet practical short read mapping. *Bioinformatics* 27, 1011–1012. doi:10.1093/bioinformatics/btr046

Dohm, J. C., Lottaz, C., Borodina, T., and Himmelbauer, H. (2008). Substantial biases in ultra-short read data sets from high-throughput DNA sequencing. *Nucleic Acids Res.* 36, e105–e105. doi:10.1093/nar/gkn425

Feuk, L., Andrew, R. C., and Stephen, W. S. (2006a). Structural variation in the human genome. *Nat. Rev. Genet.* 7, 85–97. doi:10.1038/nrg1767

Feuk, L., Marshall, C. R., Wintle, R. F., and Scherer, S. W. (2006b). Structural variants: changing the landscape of chromosomes and design of disease studies. *Hum. Mol. Genet.* 15(Suppl. 1), R57–R66. doi:10.1093/hmg/ddl057

Fu, L., Niu, B., Zhu, Z., Wu, S., and Li, W. (2012). Cd-hit: accelerated for clustering the next-generation sequencing data. *Bioinformatics* 28, 3150–3152. doi:10.1093/bioinformatics/bts565

Gomez-Alvarez, V., Teal, T. K., and Schmidt, T. M. (2009). Systematic artifacts in metagenomes from complex microbial communities. *ISME J.* 3, 1314–1317. doi:10.1038/ismej.2009.72

Harismendy, O., Ng, P. C., Strausberg, R. L., Wang, X., Stockwell, T. B., Beeson, K. Y., et al. (2009). Evaluation of next generation sequencing platforms for population targeted sequencing studies. *Genome Biol.* 10, R32. doi:10.1186/gb-2009-10-3-r32

Hillier, L. W., Marth, G. T., Quinlan, A. R., Dooling, D., Fewell, G., Barnett, D., et al. (2008). Whole-genome sequencing and variant discovery in *C. elegans*. *Nat. Methods* 5, 183–188. doi:10.1038/nmeth.1179

Hormozdiari, F., Hajirasouliha, I., Dao, P., Hach, F., Yorukoglu, D., Alkan, C., et al. (2010). Next-generation variationhunter: combinatorial algorithms for transposon insertion discovery. *Bioinformatics* 26, i350–i357. doi:10.1093/bioinformatics/btq216

Hormozdiari, F., Hajirasouliha, I., McPherson, A., Eichler, E. E., and Sahinalp, S. C. (2011). Simultaneous structural variation discovery among multiple paired-end sequenced genomes. *Genome Res.* 21, 2203–2212. doi:10.1101/gr.120501.111

Hurles, M. E., Dermitzakis, E. T., and Tyler-Smith, C. (2008). The functional impact of structural variation in humans. *Trends Genet.* 24, 238–245. doi:10.1016/j.tig.2008.03.001

Iqbal, Z., Caccamo, M., Turner, I., Flicek, P., and McVean, G. (2012). De novo assembly and genotyping of variants using colored De Bruijn graphs. *Nat. Genet.* 44, 226–232. doi:10.1038/ng.1028

Ivakhno, S., Royce, T., Cox, A. J., Evers, D. J., Cheetham, R. K., and Tavaré, S. (2010). CNAseg—a novel framework for identification of copy number changes in cancer from second-generation sequencing data. *Bioinformatics* 26, 3051–3058. doi:10.1093/bioinformatics/btq587

Kircher, M., and Kelso, J. (2010). High-throughput DNA sequencing – concepts and limitations. *Bioessays* 32, 524–536. doi:10.1002/bies.200900181

Klus, P., Lam, S., Lyberg, D., Cheung, M. S., Pullan, G., McFarlane, I., et al. (2012). Barracuda-a fast short read sequence aligner using graphics processing units. *BMC Res. Notes* 5:27. doi:10.1186/1756-0500-5-27

Korbel, J. O., Abyzov, A., Mu, X. J., Carriero, N., Cayting, P., Zhang, Z., et al. (2009). PEMer: a computational framework with simulation-based error models for inferring genomic structural variants from massive paired-end sequencing data. *Genome Biol.* 10, R23. doi:10.1186/gb-2009-10-2-r23

Langmead, B., and Salzberg, S. L. (2012). Fast gapped-read alignment with bowtie 2. *Nat. Methods* 9, 357–359. doi:10.1038/nmeth.1923

Langmead, B., Trapnell, C., Pop, M., and Salzberg, S. L. (2009). Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* 10, R25. doi:10.1186/gb-2009-10-3-r25

Li, H., and Durbin, R. (2009). Fast and accurate short read alignment with burrows–wheeler transform. *Bioinformatics* 25, 1754–1760. doi:10.1093/bioinformatics/btp324

Li, H., Handsaker, B., Wysoker, A., Fennell, T., Ruan, J., Homer, N., et al. (2009a). The sequence alignment/map format and samtools. *Bioinformatics* 25, 2078–2079. doi:10.1093/bioinformatics/btp352

Li, R., Yu, C., Li, Y., Lam, T.-W., Yiu, S.-M., Kristiansen, K., et al. (2009b). Soap2: an improved ultrafast tool for short read alignment. *Bioinformatics* 25, 1966–1967. doi:10.1093/bioinformatics/btp336

Li, H., Ruan, J., and Durbin, R. (2008). Mapping short DNA sequencing reads and calling variants using mapping quality scores. *Genome Res.* 18, 1851–1858. doi:10.1101/gr.078212.108

Li, W., and Godzik, A. (2006). Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. *Bioinformatics* 22, 1658–1659. doi:10.1093/bioinformatics/btl158

Liu, C.-M., Wong, T., Wu, E., Luo, R., Yiu, S.-M., Li, Y., et al. (2012a). Soap3: ultra-fast GPU-based parallel alignment tool for short reads. *Bioinformatics* 28, 878–879. doi:10.1093/bioinformatics/bts061

Liu, Y., Schmidt, B., and Maskell, D. L. (2012b). CUSHAW: a CUDA compatible short read aligner to large genomes based on the burrows–wheeler transform. *Bioinformatics* 28, 1830–1837. doi:10.1093/bioinformatics/bts276

Liu, Y., Schmidt, B., and Maskell, D. L. (2010). Cudasw++ 2.0: enhanced smith-waterman protein database search on CUDA-enabled GPUs based on SIMT and virtualized SIMD abstractions. *BMC Res. Notes* 3:93. doi:10.1186/1756-0500-3-93

Luo, R., Wong, T., Zhu, J., Liu, C.-M., Zhu, X., Wu, E., et al. (2013). Soap3-dp: fast, accurate and sensitive GPU-based short read aligner. *PLoS ONE* 8:e65632. doi:10.1371/journal.pone.0065632

Magi, A., Tattini, L., Pippucci, T., Torricelli, F., and Benelli, M. (2012). Read count approach for DNA copy number variants detection. *Bioinformatics* 28, 470–478. doi:10.1093/bioinformatics/btr707

Manavski, S. A., and Valle, G. (2008). CUDA compatible GPU cards as efficient hardware accelerators for smith-waterman sequence alignment. *BMC Bioinformatics* 9(Suppl. 2):S10. doi:10.1186/1471-2105-9-S2-S10

Manconi, A., Orro, A., Manca, E., Armano, G., and Milanesi, L. (2014a). GPU-bsm: a GPU-based tool to map bisulfite-treated reads. *PLoS ONE* 9:e97277. doi:10.1371/journal.pone.0097277

Manconi, A., Orro, A., Manca, E., Armano, G., and Milanesi, L. (2014b). A tool for mapping single nucleotide polymorphisms using graphics processing units. *BMC Bioinformatics* 15:1–13. doi:10.1186/1471-2105-15-S1-S10

Marco-Sola, S., Sammeth, M., Guigó, R., and Ribeca, P. (2012). The gem mapper: fast, accurate and versatile alignment by filtration. *Nat. Methods* 9, 1185–1188. doi:10.1038/nmeth.2221

Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet J.* 17, 10. doi:10.14806/ej.17.1.200

Medvedev, P., Stanciu, M., and Brudno, M. (2009). Computational methods for discovering structural variation with next-generation sequencing. *Nat. Methods* 6, S13–S20. doi:10.1038/nmeth.1374

Merikangas, A. K., Corvin, A. P., and Gallagher, L. (2009). Copy-number variants in neurodevelopmental disorders: promises and challenges. *Trends Genet.* 25, 536–544. doi:10.1016/j.tig.2009.10.006

Miller, C. A., Hampton, O., Coarfa, C., and Milosavljevic, A. (2011). Readdepth: a parallel r package for detecting copy number alterations from short sequencing reads. *PLoS ONE* 6:e16327. doi:10.1371/journal.pone.0016327

Mills, R. E., Walter, K., Stewart, C., Handsaker, R. E., Chen, K., Alkan, C., et al. (2011). Mapping copy number variation by population-scale genome sequencing. *Nature* 470, 59–65. doi:10.1038/nature09708

Mu, J. C., Jiang, H., Kiani, A., Mohiyuddin, M., Asadi, N. B., and Wong, W. H. (2012). Fast and accurate read alignment for resequencing. *Bioinformatics* 28, 2366–2373. doi:10.1093/bioinformatics/bts450

Munshi, A. (2009). *The Opencl Specification*, Vol. 1. Khronos OpenCL Working Group, l1–l15.

Nijkamp, J. F., van den Broek, M. A., Geertman, J.-M. A., Reinders, M. J., Daran, J.-M. G., and de Ridder, D. (2012). De novo detection of copy number variation by co-assembly. *Bioinformatics* 28, 3195–3202. doi:10.1093/bioinformatics/bts601

NVIDIA Corporation. (2007). *Compute Unified Device Architecture Programming Guide.*

Patel, R. K., and Jain, M. (2012). Ngs qc toolkit: a toolkit for quality control of next generation sequencing data. *PLoS ONE* 7:e30619. doi:10.1371/journal.pone.0030619

Perry, G. H., Tchinda, J., McGrath, S. D., Zhang, J., Picker, S. R., Cáceres, A. M., et al. (2006). Hotspots for copy number variation in chimpanzees and humans. *Proc. Natl. Acad. Sci. U.S.A.* 103, 8006–8011. doi:10.1073/pnas.0602318103

Pireddu, L., Leo, S., and Zanetti, G. (2011). Seal: a distributed short read mapping and duplicate removal tool. *Bioinformatics* 27, 2159–2160. doi:10.1093/bioinformatics/btr325

Ruffalo, M., LaFramboise, T., and Koyutürk, M. (2011). Comparative analysis of algorithms for next-generation sequencing read alignment. *Bioinformatics* 27, 2790–2796. doi:10.1093/bioinformatics/btr477

Rumble, S. M., Lacroute, P., Dalca, A. V., Fiume, M., Sidow, A., and Brudno, M. (2009). Shrimp: accurate mapping of short color-space reads. *PLoS Comput. Biol.* 5:e1000386. doi:10.1371/journal.pcbi.1000386

Sachidanandam, R., Weissman, D., Schmidt, S. C., Kakol, J. M., Stein, L. D., Marth, G., et al. (2001). A map of human genome sequence variation containing 1.42 million single nucleotide polymorphisms. *Nature* 409, 928–933. doi:10.1038/35057149

Schatz, M. C. (2009). Cloudburst: highly sensitive read mapping with mapreduce. *Bioinformatics* 25, 1363–1369. doi:10.1093/bioinformatics/btp236

Shi, H., Schmidt, B., Liu, W., and Müller-Wittig, W. (2010). Quality-score guided error correction for short-read sequencing data using CUDA. *Procedia Comput Sci* 1, 1129–1138. doi:10.1016/j.procs.2010.04.125

Sindi, S., Helman, E., Bashir, A., and Raphael, B. J. (2009). A geometric approach for classification and comparison of structural variants. *Bioinformatics* 25, i222–i230. doi:10.1093/bioinformatics/btp208

Smith, A. D., Chung, W.-Y., Hodges, E., Kendall, J., Hannon, G., Hicks, J., et al. (2009). Updates to the rmap short-read mapping software. *Bioinformatics* 25, 2841–2842. doi:10.1093/bioinformatics/btp533

Smith, A. D., Xuan, Z., and Zhang, M. Q. (2008). Using quality scores and longer reads improves accuracy of solexa read mapping. *BMC Bioinformatics* 9:128. doi:10.1186/1471-2105-9-128

Stefansson, H., Rujescu, D., Cichon, S., Pietiläinen, O. P., Ingason, A., Steinberg, S., et al. (2008). Large recurrent microdeletions associated with schizophrenia. *Nature* 455, 232–236. doi:10.1038/nature07229

Xi, R., Luquette, J., Hadjipanayis, A., Kim, T.-M., and Park, P. J. (2010). Bic-seq: a fast algorithm for detection of copy number alterations based on high-throughput sequencing data. *Genome Biol.* 11(Suppl. 1), O10. doi:10.1186/1465-6906-11-S1-O10

Xie, C., and Tammi, M. T. (2009). Cnv-seq, a new method to detect copy number variation using high-throughput sequencing. *BMC Bioinformatics* 10:80. doi:10.1186/1471-2105-10-80

Xu, H., Luo, X., Qian, J., Pang, X., Song, J., Qian, G., et al. (2012). Fastuniq: a fast de novo duplicates removal tool for paired short reads. *PLoS ONE* 7:e52249. doi:10.1371/journal.pone.0052249

Yang, X., Liu, D., Liu, F., Wu, J., Zou, J., Xiao, X., et al. (2013). Htqc: a fast quality control toolkit for illumina sequencing data. *BMC Bioinformatics* 14:33. doi:10.1186/1471-2105-14-33

Ye, K., Schulz, M. H., Long, Q., Apweiler, R., and Ning, Z. (2009). Pindel: a pattern growth approach to detect break points of large deletions and medium sized insertions from paired-end short reads. *Bioinformatics* 25, 2865–2871. doi:10.1093/bioinformatics/btp394

Yoon, S., Xuan, Z., Makarov, V., Ye, K., and Sebat, J. (2009). Sensitive and accurate detection of copy number variants using read depth of coverage. *Genome Res.* 19, 1586–1592. doi:10.1101/gr.092981.109

Yung, L. S., Yang, C., Wan, X., and Yu, W. (2011). Gboost: a GPU-based tool for detecting gene–gene interactions in genome–wide case control studies. *Bioinformatics* 27, 1309–1310. doi:10.1093/bioinformatics/btr114

Zhang, Z. D., Du, J., Lam, H., Abyzov, A., Urban, A. E., Snyder, M., et al. (2011). Identification of genomic indels and structural variations using split reads. *BMC Genomics* 12:375. doi:10.1186/1471-2164-12-375

Zhao, K., and Chu, X. (2014). G-blastn: accelerating nucleotide alignment by graphics processors. *Bioinformatics* 30, 1384–1391. doi:10.1093/bioinformatics/btu047

Zhao, M., Wang, Q., Wang, Q., Jia, P., and Zhao, Z. (2013). Computational tools for copy number variation (cnv) detection using next-generation sequencing data: features and perspectives. *BMC Bioinformatics* 14(Suppl. 11):S1. doi:10.1186/1471-2105-14-S11-S1

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 10 December 2014; accepted: 19 February 2015; published online: 10 March 2015.

Citation: Manconi A, Manca E, Moscatelli M, Gnocchi M, Orro A, Armano G and Milanesi L (2015) G-CNV: a GPU-based tool for preparing data to detect CNVs with read-depth methods. Front. Bioeng. Biotechnol. ***3****:28. doi: 10.3389/fbioe.2015.00028*

This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology.

frontiers in Bioengineering and Biotechnology

MINI REVIEW
published: 17 March 2015
doi: 10.3389/fbioe.2015.00031

Statistical approaches to detecting and analyzing tandem repeats in genomic sequences

Maria Anisimova[1]*, *Jūlija Pečerska*[2,3] and *Elke Schaper*[3,4]

[1] Institute of Applied Simulation, School of Life Sciences and Facility Management, Zürich University of Applied Sciences (ZHAW), Wädenswil, Switzerland, [2] Department of Biosystems Science and Engineering, ETH Zürich, Basel, Switzerland, [3] Department of Computer Science, ETH Zürich, Zürich, Switzerland, [4] Vital-IT Competency Center, Swiss Institute for Bioinformatics, Lausanne, Switzerland

Tandem repeats (TRs) are frequently observed in genomes across all domains of life. Evidence suggests that some TRs are crucial for proteins with fundamental biological functions and can be associated with virulence, resistance, and infectious/neurodegenerative diseases. Genome-scale systematic studies of TRs have the potential to unveil core mechanisms governing TR evolution and TR roles in shaping genomes. However, TR-related studies are often non-trivial due to heterogeneous and sometimes fast evolving TR regions. In this review, we discuss these intricacies and their consequences. We present our recent contributions to computational and statistical approaches for TR significance testing, sequence profile-based TR annotation, TR-aware sequence alignment, phylogenetic analyses of TR unit number and order, and TR benchmarks. Importantly, all these methods explicitly rely on the evolutionary definition of a tandem repeat as a sequence of adjacent repeat units stemming from a common ancestor. The discussed work has a focus on protein TRs, yet is generally applicable to nucleic acid TRs, sharing similar features.

Keywords: tandem repeats, molecular evolution, protein domain, tandem repeat annotation, sequence profile model

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Ali Masoudi-Nejad, University of Tehran, Iran
Alberto Jesus Martin, Fundación Ciencia & Vida, Chile

***Correspondence:**
Maria Anisimova, Institute of Applied Simulation, School of Life Sciences and Facility Management, Zürich University of Applied Sciences, 8820 Wädenswil, Switzerland
anis@zhaw.ch

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *10 December 2014*
Accepted: *26 February 2015*
Published: *17 March 2015*

Citation:
Anisimova M, Pečerska J and Schaper E (2015) Statistical approaches to detecting and analyzing tandem repeats in genomic sequences. Front. Bioeng. Biotechnol. 3:31. doi: 10.3389/fbioe.2015.00031

Tandem Repeats in Genomic Sequences

A tandem repeat (TR) in genomic sequence is a subsequent recurrence of a single sequence motif. TRs are described by the length of the minimal repeating motif (unit), the number of units, and the similarity among its units. The similarity of initially identical TR units fades with time through point mutations and indels, masking their shared ancestry. Diverged TR units, even when unrecognizable by eye, can maintain structural similarity over long evolutionary times [e.g., **Figure 1** in Kajava (2012)]. While the mechanisms shaping TRs are poorly understood, they can evolve by duplication/loss of TR units, recombination, and gene conversion (Pearson et al., 2005; Richard et al., 2008). TRs can mutate by replication slippage (Levinson and Gutman, 1987; Ellegren, 2000), whereby the mispairing of a slipping-strand during the DNA synthesis causes a loss or gain of units as loops of TR units form hairpin structures (Mirkin, 2006).

Tandem repeats are frequent in coding and non-coding DNA in species throughout the kingdoms of life. Genomic TRs are a rich source for genetic variability, providing a wide range of possible genotypes at a given locus (Nithianantharajah and Hannan, 2007) and apt opportunity for selection, not only on long evolutionary scales but also during somatic cellular processes. Particularly staggering

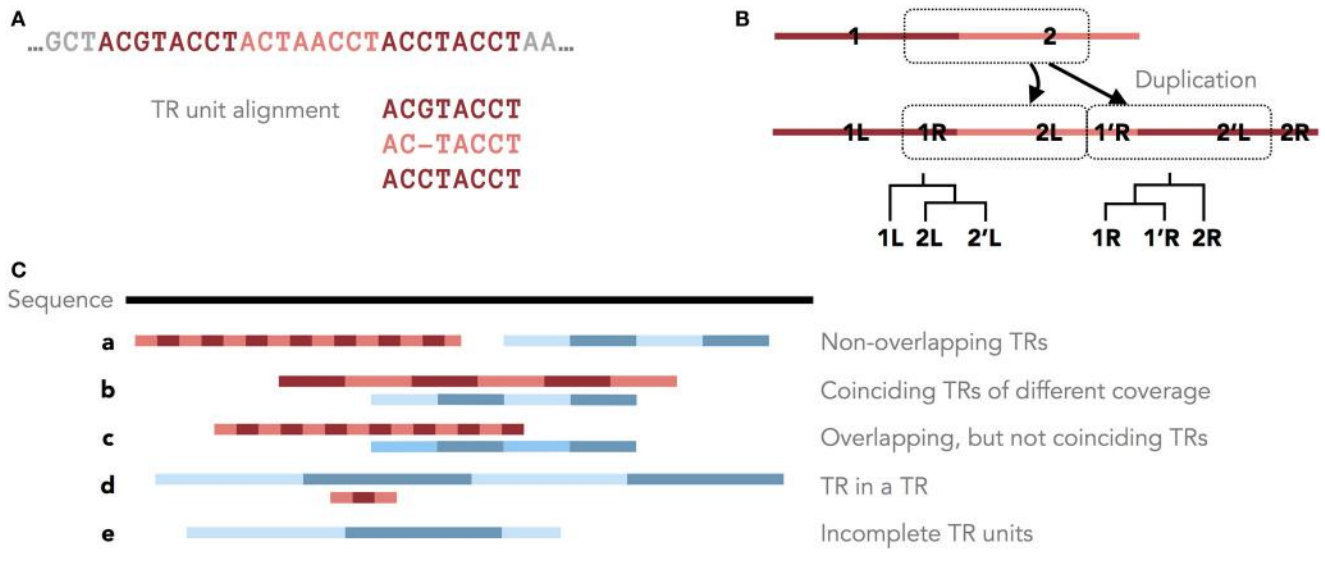


FIGURE 1 | Tandem repeats in genomic sequences. (A) An example TR with three units and the corresponding MSA of its units. **(B)** Different parts of a TR motif (R = right and L = left) have different histories after a single duplication with shifted TR units. Shown are these duplication histories as phylogenies of the right and left parts of the TR motif. **(C)** Five scenarios of overlapping and non-overlapping TR annotations.

variation in TR unit lengths and numbers is characteristic to genomic TRs, such as ribosomal DNA arrays crucial for the translation machinery, and satellite DNA comprising the main component of functional centromeres (Richard et al., 2008). In protein-coding genes, mutations in TRs are likely to alter the structure/function of the protein product. Even in non-coding TRs, mutations can have serious fitness consequences by affecting gene regulation, transcription, and translation (Usdin, 2008). Crucially, TRs have attracted attention because of their medical relevance: many human proteins with TRs have been linked to monogenic disorders, typically affecting the nervous system (Siwach and Ganesh, 2008; Hannan, 2010). There is a high incidence of TRs in virulence factors of pathogenic agents, toxins, and allergens (Jorda et al., 2010).

The Methodological Challenges for Tandem Repeats Detection

Systematic analyses of genomic TRs will help to better understand the biological processes governing and governed by TRs and their functional relevance. Such studies rely on the large-scale TRs detection (TRD). Numerous methods for TRD have been developed. Yet, since they are based on different algorithmic paradigms and heuristics, there is a large discrepancy between TR annotations produced by different algorithms for the same sequence (Leclercq et al., 2007; Merkel and Gemmell, 2008; Mudunuri et al., 2010; Schaper et al., 2012). For example, with four TRD methods applied to the human proteome, the majority of TRs were annotated by a single detector, only 9.8% were annotated by two and a meager 1.1% by at least three (Schaper et al., 2012).

The TR heterogeneity contributes to the large variability among TRD methods. The TRD is relatively simple for identical units. If the TR motif is unknown, this task is computationally expensive for long sequences, requiring an exhaustive search with no information on TR unit length, number, or position in the sequence (search space in $O(N^3)$ for sequence length N). Substitutions and indels in the TR region cause major challenges to TRD: with decreasing unit similarity, TR regions become hard to discern. Indels introduce length variability between individual TR units, increasing the TR search space to $O(2^N N^3)$.

Furthermore, the original TR unit boundaries can be shifted due to new unit duplications (**Figure 1A**, Benson and Dong, 1999; Rivals, 2004). Clear boundaries are preserved only in some cases, for example, when protein TRs are confined by the exon structure. Therefore, unambiguously dividing a TR region into units of similar lengths may not accurately reflect the TR duplication history. Occasionally, the TR history is described by different phylogenies for different parts of the TR motif (**Figure 1B**). Thus, defining the consensus TR motif is problematic, and TRD methods typically differ in the predicted unit lengths and boundaries.

Ultimately, TRD methods differ by TR definitions. One TR definition borrows from string matching in computer science, whereby TRs are defined by a repetitive regular expression, allowing for a fixed proportion of dissimilar characters among TR units. This viewpoint enables straightforward exhaustive TRD algorithms, but lacks biological interpretation. Alternatively, from a structure perspective, protein TRs may be defined by structural repetitions, which allows TR detection for structurally conserved TR units, even with low sequence similarity [see, e.g., the structural TR database Repeats DB, Di Domenico et al. (2014)]. Yet, defining structural repeats is in itself problematic. Finally, the evolutionary definition states that a TR stems from ancestral unit duplications. This viewpoint has a direct biological interpretation reflecting the TR generating mechanism. Most TRD methods,

however, lack an explicit TR definition, which obscures the search objective and TRs are typically detected by empirical properties of unit similarity. This further impedes the ability to evaluate the statistical properties of a method. Improving TRD requires a rigorous statistical framework based on a clear TR definition described as a biologically meaningful mathematical model. Then, a genuine TR can be distinguished from a non-TR sequence by comparison with a model describing random sequences. Relying on a mathematical TR model means that the TRD method's behavior can be predicted for different scenarios, including the evaluation of false predictions.

One possibility is to define TR units as related by a common ancestral unit under a Markov substitution model and a standard phylogeny model reflecting the unit duplication history (Schaper et al., 2012). The evolutionary distance t from currently observed TR units to their common ancestor can be estimated by maximum likelihood. For any tentative TR region with predicted units, this conveniently allows for statistical hypothesis testing using a likelihood ratio test (LRT; Schaper et al., 2012). The null hypothesis "$t = \infty$" represents that the estimated evolutionary distance is so large that TR units have no common origin and could have appeared by chance. Rejecting the null suggests that the given units are related (with finite t) and therefore are assumed to be TRs by definition. Such approach provides a statistical framework to validate predicted TRs, filtering out potential false positive predictions.

The variability in TR annotations produced by different TRD methods warns against relying on one specific algorithm. Different methods not only achieve optimal power for different combinations of TR divergence and unit length, but also vary in their accuracy across the TR space. Therefore, to obtain the most complete and accurate set of TR annotations for a given sequence, we suggest that multiple TR detectors should be used [maximizing the number of true positives, e.g., as in Pellegrini et al. (2012)] followed by validation with an LRT (controlling false positives at a fixed significance level).

Annotating TRs with Sequence Profile Models

Many common protein domains found in tandem are listed in sequence profile databases, such as Pfam (Punta et al., 2011), PROSITE (Sigrist et al., 2010), SMART (Letunic et al., 2012), Repbase (Jurka et al., 2005), and Dfam (Travis et al., 2013). For example, of all *de novo* annotated TRs with unit length $\geq$15, we found that only few had not been described in Pfam (2.1% in *Arabidopsis thaliana* and 11.5% in human). Thus, TR annotation can profit strongly from the existing databases.

Profile-based annotation typically relies on sequence profile hidden Markov models (HMMs). Circular connections in an HMM allow the annotation of full TR units (Bucher et al., 1996; Schaper et al., 2014), as implemented in pftools (Sigrist et al., 2013) and in our Python TR Annotation Library TRAL (http://elkeschaper.github.io/tral/). General profile HMM annotation can be used to detect TR units [e.g., HMMER; Eddy (2011)], but a subsequent analysis is required to annotate the whole TR region, potentially including diverged TR units that, without considering the whole TR region, lack statistical significance.

Importantly, sequence profile HMMs can be used to refine *de novo* annotations. *De novo* detected TR units provide seed motifs that can be converted to sequence profile models (e.g., with *hmmbuild* from the HMMER package). These models can then be used to re-annotate sequences. The advantage is twofold: first, the quality of annotation is homogenized among TRs from different TRD methods; second, annotations on homolog sequences become comparable.

An Example Pipeline for Meta-Prediction of Genomic TRs

A plausible TR annotation pipeline in three steps could be (1) identify a putative TR unit seed motif; (2) detect all tandem occurrences of this motif in the sequence, forming a putative TR; and (3) for each putative TR validate its statistical significance and filter out redundant predictions. We describe this TR annotation workflow in **Figure 2**; all functionalities are implemented in TRAL (http://elkeschaper.github.io/tral/).

Tandem repeat seed motifs are obtained from sequence profile databases and multiple *de novo* TRD algorithms. Circular profile HMMs are built from these TR seeds and consequently used to annotate TR regions in a sequence. All annotated TRs must be statistically validated, for example, using an LRT. A multiple testing correction may be required dependent on the application [e.g., Saville (1990)]. TRs that fail the test are assumed to be false positive predictions and are discarded. Combining several methods leads to redundant predictions, which is not limited to the rare case where two TR predictions fully coincide. Due to differences in predicted unit boundaries or unit numbers, it is often difficult to decide whether two overlapping TR annotations describe the same TR or, rather, nested or neighboring TRs (**Figure 1C**). To filter redundant predictions, several *ad hoc* criteria may be used. For example, overlapping TRs may be seen as redundant (the

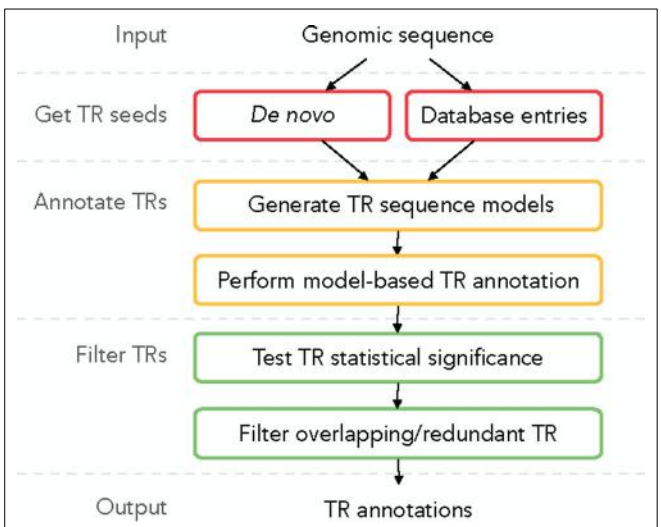


FIGURE 2 | Overview of a generic TR annotation workflow.

required degree of overlap demands another *ad hoc* decision; **Figure 1C**, b–d). Using the evolutionary TR definition, we propose one possible criterion based on the representation of a TR region as a multiple sequence alignment (MSA) of its TR units. Characters grouped in one MSA column are assumed to derive from a common ancestral character. Given this, one of two TR annotations may be seen as redundant if the characters in two annotations are grouped into columns similarly by the two corresponding MSAs (**Figure 1C**, b). Model-based test statistics may be defined to recognize redundant TRs, deciding if two independent TR models provide a better description of these TRs compared to one common model.

We used *de novo* and profile-based methods to annotate TRs in the entire UniprotKB/Swiss-Prot (UniProt Consortium, 2014, v2013-08) with the proposed pipeline (**Figure 2**). A considerable proportion of these sequences contain TRs: 46% in Eukaryotes, 29% in Archaea, and 27% in Bacteria. These TRs, across all kingdoms of life, are dominated by microsatellite (typically < 10 bp) or short minisatellite TRs (~10–100 bp) with few units.

Probabilistic MSA with TRs

When aligning homologous sequences, often we are not aware of the presence of TRs. Yet, due to uneven TR unit gain/loss among the homologs, TR-containing MSAs are error-prone, since standard indel penalty schemes do not account for the potential variation in TR region length. Some MSA methods accounting for sequence repeats produce local alignments (Raphael, 2004; Phuong et al., 2006; Treangen et al., 2009). However, a global alignment is required for evolutionary inferences, such as for the estimation of TR unit history. Sammeth and Heringa (2006) proposed a global MSA method with fixed TR unit boundaries. Yet, unit boundaries may be distorted by indels, slippage, and recombination. Modeling TRs explicitly in the MSA graph representation allows TR units to start at any position, adequately penalizing indels corresponding to unit gains/losses, and to reconstruct the evolutionary history of TR unit events. The implementation ProGraphMSA + TR (Szalkowski and Anisimova, 2013) uses a probabilistic phylogeny-aware approach similar to PRANK (Löytynoja and Goldman, 2005), achieving not only improved alignment quality, but also *a posteriori* estimation of rates of evolutionary events, such as TR unit indels. For example, ProGraphMSA + TR was applied to leucine-rich repeats (LRRs) in a gene family of type III effectors determining the pathogenicity in agriculturally important bacteria *Ralstonia solanacearum*. The estimates of TR indel frequencies in different clades of a gene phylogeny suggested that TR indel rate variation contributes to the diversification of this protein family [Figure 9 of Szalkowski and Anisimova (2013)]. Variation in LRR unit numbers might contribute to adaptive processes in this gene and to pathogenesis on different plant hosts.

Phylogenetic Approach to Study the Evolution of TRs

Tandem repeat unit phylogenies reconstructed from homologous TRs carry much evolutionary signal, even with short units (Schaper and Anisimova, 2014; Schaper et al., 2014). These phylogenies inform about unit duplication histories and TR unit gain/loss rates, allowing to study selection on TRs and their functional relevance. Clustering patterns in TR unit phylogenies describe the unit conservation between species. If the TR unit number and order in orthologs regions from different species are perfectly conserved throughout the evolution, then the phylogeny of all TR units consists of clades formed by orthologous unit copies, each reflecting the phylogeny of the whole region (or species) [Figure 1B of Schaper et al. (2014)]. In contrast, if TR regions are fully separated, then a speciation event is followed by a series of TR unit gain/loss events, and the TR unit phylogeny consists of species-specific monophyletic clades of TR units [Figure 1A of Schaper et al. (2014)].

Tandem repeat unit phylogenies from pairs of orthologs can be used to backtrack the evolution of TRs from a single species (Schaper et al., 2014) or across an entire species tree – using the all-against-all pair-wise approach (Schaper and Anisimova, 2014). In contrast to multispecies TR unit phylogenies, for pairwise TR unit phylogenies, the statistical significance of observing perfect conservation and separation patterns is computed exactly (Schaper et al., 2014). On the other hand, multispecies TR unit phylogenies suffer fewer reconstruction errors due to the additional information on the unit evolution from additional orthologs.

Our large-scale analyses of eukaryotic proteomes revealed an extremely deep conservation of some protein domain TRs (dTRs), many dating to hundreds million years ago and some even to the times of separation between human and yeast (0.6–1.6 billion years ago) or red algae and green plants (~1.6 billion years ago). Conserved dTRs span much of the TR diversity of proteomes. For example, in human 81% of detected distinct dTR types have been conserved at least to the ancestor of mammals at least in one protein (Schaper et al., 2014). The distribution of conserved domain types is highly heterogeneous: 68% of all conserved dTRs are described by only 5% of all TR types detected in human. Yet, many conserved TRs are rare and occur only in a single protein. Similar numbers were observed in plants (Schaper and Anisimova, 2014).

In contrast, very few dTRs have separated between closely related species. In human, dTRs separated within mammals are dominated by zinc finger repeats (~50%), followed by DUF1220 (~8%; Schaper et al., 2014). In *A. thaliana*, dTRs separated within magnoliophytes are dominated by LRRs (~40%), followed by ankyrin (~12%) (Schaper and Anisimova, 2014). For both species, separated dTRs are enriched in *de novo* annotations, i.e., rare and presumably recent dTRs, which are perhaps more prone to unit gains/losses due to relaxed selection or due to higher mutation rates as a result of a high among-unit sequence similarity.

Simulating Genomic TRs

Benchmarking and hypothesis testing in bioinformatics must often rely on simulated data since the truth is rarely known. For example, only the observation of identical TR units qualifies them as a true TR with certainty. Yet this is only the simplest scenario, which is of little practical relevance. With diverged or shifted TR

units, unbiased benchmarks are challenging to construct from real data. Model-based simulation offers a powerful means to benchmarking and hypothesis testing in bioinformatics studies of TRs. Simulations enable comparisons of competing hypotheses and help to reveal methodological weaknesses or detect and estimate important factors. Simulations are not only crucial for benchmarking new TRD methods, but also to study the underlying evolutionary mechanisms of genomic TRs. For example, to describe the biological mechanisms for specific TR types, patterns or parameter estimates observed in these data can be compared with those obtained from sequences simulated with alternative models of TR evolution. Sequences with TRs may be generated not only with the dedicated models of evolution by unit gain/loss with fuzzy units boundaries, e.g., SlippageSim (Szalkowski and Anisimova, 2013), but also with other general sequence simulators that allow gene family evolution by mutations, indels, gene gain/loss, recombination, etc. [e.g., Dalquen et al. (2012)]. For example, sequences with TRs of different divergence and unit lengths were used to benchmark the MSA method ProGraphMSA + TR that accounts for sequence TRs (see above). The evaluation of the power of TRD methods also relies on simulated sequences with TRs (the alternative hypothesis). Yet the high power is irrelevant without the evaluation of false positive TRD rates, which must be done on TR-free data (the null hypothesis). Furthermore, simulated TR-free data helps to validate TR-specific findings: the comparison of patterns found in simulated TR-free sequences with those observed in sequences with TRs serves to disentangle TR-specific findings from those that may occur in genomic sequences in general. A simple approach is to simulate TR-free data by drawing k-mers from a (k - 1)-th order Markov model based on empirical frequencies (Robin et al., 2007). In contrast to drawing single characters from their frequency distribution, simulating k-mers mimics natural local correlations while choosing small k minimizes the chance of hidden TRs within a k-mer.

Conclusion and Perspectives

Tandem repeats are diverse in their size, type, unit similarity, and distribution across genomes. Methods discussed above enable accurate detection of TR orthologs with strongly conserved unit configurations, or on the contrary, with highly changing unit numbers. Due to the heterogeneity of TRs, large-scale studies should be followed up by studies that focus on specific TR types and their effects on molecular processes. Our TR scans of eukaryotic proteomes provide a plentitude of cases to investigate with respect to their functional roles. While many TRs were linked to key functions, phenotypic changes, or disease predisposition, the biological mechanisms generating and preserving TRs in genomes are poorly understood. New studies of genomic TRs only fuel our fascination with these genomic features, calling for further research and for the development of dedicated methods. Further development of rigorous statistical models of TR generating mechanisms will help to improve TRD methods, and to shed some light on biological forces shaping these sequences.

References

Benson, G., and Dong, L. (1999). Reconstructing the duplication history of a tandem repeat. *Proc. Int. Conf. Intell. Syst. Mol. Biol.* 44–53.

Bucher, P., Karplus, K., Moeri, N., and Hofmann, K. (1996). A flexible motif search technique based on generalized profiles. *Comput. Chem.* 20, 3–23. doi:10.1016/S0097-8485(96)80003-9

Dalquen, D. A., Anisimova, M., Gonnet, G. H., and Dessimoz, C. (2012). ALF – a simulation framework for genome evolution. *Mol. Biol. Evol.* 29, 1115–1123. doi:10.1093/molbev/msr268

Di Domenico, T., Potenza, E., Walsh, I., Gonzalo Parra, R., Giollo, M., Minervini, G., et al. (2014). RepeatsDB: a database of tandem repeat protein structures. *Nucleic Acids Res.* 42, D352–D357. doi:10.1093/nar/gkt1175

Eddy, S. R. (2011). Accelerated profile HMM searches. *PLoS Comput. Biol.* 7:e1002195. doi:10.1371/journal.pcbi.1002195

Ellegren, H. (2000). Microsatellite mutations in the germline: implications for evolutionary inference. *Trends Genet.* 16, 551–558. doi:10.1016/S0168-9525(00)02139-9

Hannan, A. J. (2010). Tandem repeat polymorphisms: modulators of disease susceptibility and candidates for "missing heritability. *Trends. Genet.* 26, 59–65. doi:10.1016/j.tig.2009.11.008

Jorda, J., Xue, B., Uversky, V. N., and Kajava, A. V. (2010). Protein tandem repeats – the more perfect, the less structured. *FEBS J.* 277, 2673–2682. doi:10.1111/j.1742-4658.2010.07684.x

Jurka, J., Kapitonov, V. V., Pavlicek, A., Klonowski, P., Kohany, O., and Walichiewicz, J. (2005). Repbase update, a database of eukaryotic repetitive elements. *Cytogenet. Genome Res.* 110, 462–467. doi:10.1159/000084979

Kajava, A. V. (2012). Tandem repeats in proteins: from sequence to structure. *J. Struct. Biol.* 179, 279–288. doi:10.1016/j.jsb.2011.08.009

Leclercq, S., Rivals, E., and Jarne, P. (2007). Detecting microsatellites within genomes: significant variation among algorithms. *BMC Bioinformatics* 8:125. doi:10.1186/1471-2105-8-125

Letunic, I., Doerks, T., and Bork, P. (2012). SMART 7: recent updates to the protein domain annotation resource. *Nucleic Acids Res.* 40, D302–D305. doi:10.1093/nar/gkr931

Levinson, G., and Gutman, G. A. (1987). Slipped-strand mispairing: a major mechanism for DNA sequence evolution. *Mol. Biol. Evol.* 4, 203–221.

Löytynoja, A., and Goldman, N. (2005). An algorithm for progressive multiple alignment of sequences with insertions. *Proc. Natl. Acad. Sci. U.S.A.* 102, 10557–10562. doi:10.1073/pnas.0409137102

Merkel, A., and Gemmell, N. J. (2008). Detecting microsatellites in genome data: variance in definitions and bioinformatic approaches cause systematic bias. *Evol. Bioinform. Online* 4, 1–6.

Mirkin, S. M. (2006). DNA structures, repeat expansions and human hereditary disorders. *Curr. Opin. Struct. Biol.* 16, 351–358. doi:10.1016/j.sbi.2006.05.004

Mudunuri, S. B., Rao, A. A., Pallamsetty, S., and Nagarajaram, H. A. (2010). "Comparative analysis of microsatellite detecting software: a significant variation in results and influence of parameters," in *Proceedings of the International Symposium on Biocomputing 2010, ISB '10* (New York: ACM). doi:10.1145/1722024.1722068

Nithianantharajah, J., and Hannan, A. J. (2007). Dynamic mutations as digital genetic modulators of brain development, function and dysfunction. *Bioessays* 29, 525–535. doi:10.1002/bies.20589

Pearson, C. E., Edamura, K. N., and Cleary, J. D. (2005). Repeat instability: mechanisms of dynamic mutations. *Nat. Rev. Genet.* 6, 729–742. doi:10.1038/nrg1689

Pellegrini, M., Renda, M. E., and Vecchio, A. (2012). Tandem repeats discovery service (TReaDS) applied to finding novel cis-acting factors in repeat expansion diseases. *BMC Bioinformatics* 13(Suppl. 4):S3. doi:10.1186/1471-2105-13-S4-S3

Phuong, T. M., Do, C. B., Edgar, R. C., and Batzoglou, S. (2006). Multiple alignment of protein sequences with repeats and rearrangements. *Nucleic Acids Res.* 34, 5932–5942. doi:10.1093/nar/gkl511

Punta, M., Coggill, P. C., Eberhardt, R. Y., Mistry, J., Tate, J., Boursnell, C., et al. (2011). The Pfam protein families database. *Nucleic Acids Res.* 40, D290–D301. doi:10.1093/nar/gkr1065

Raphael, B. (2004). A novel method for multiple alignment of sequences with repeated and shuffled elements. *Genome Res.* 14, 2336–2346. doi:10.1101/gr.2657504

Richard, G.-F., Kerrest, A., and Dujon, B. (2008). Comparative genomics and molecular dynamics of DNA repeats in eukaryotes. *Microbiol. Mol. Biol. Rev.* 72, 686–727. doi:10.1128/MMBR.00011-08

Rivals, E. (2004). A survey on algorithmic aspects of tandem repeats evolution. *Int. J. Foun. Comp. Sci.* 15, 225–257.

Robin, S., Schbath, S., and Vandewalle, V. (2007). Statistical tests to compare motif count exceptionalities. *BMC Bioinformatics* 8:84. doi:10.1186/1471-2105-8-84

Sammeth, M., and Heringa, J. (2006). Global multiple-sequence alignment with repeats. *Proteins* 64, 263–274. doi:10.1002/prot.20957

Saville, D. J. (1990). Multiple comparison procedures: the practical solution. *Am. Stat.* 44, 174–180. doi:10.1080/00031305.1990.10475712

Schaper, E., and Anisimova, M. (2014). The evolution and function of protein tandem repeats in plants. *New Phytol.* 206, 397–410. doi:10.1111/nph.13184

Schaper, E., Gascuel, O., and Anisimova, M. (2014). Deep conservation of human protein tandem repeats within the eukaryotes. *Mol. Biol. Evol.* 31, 1132–1148. doi:10.1093/molbev/msu062

Schaper, E., Kajava, A. V., Hauser, A., and Anisimova, M. (2012). Repeat or not repeat? – statistical validation of tandem repeat prediction in genomic sequences. *Nucleic Acids Res.* 40, 10005–10017. doi:10.1093/nar/gks726

Sigrist, C. J. A., Cerutti, L., de Castro, E., Langendijk-Genevaux, P. S., Bulliard, V., Bairoch, A., et al. (2010). PROSITE, a protein domain database for functional characterization and annotation. *Nucleic Acids Res.* 38, D161–D166. doi:10.1093/nar/gkp885

Sigrist, C. J. A., de Castro, E., Cerutti, L., Cuche, B. A., Hulo, N., Bridge, A., et al. (2013). New and continuing developments at PROSITE. *Nucleic Acids Res.* 41, D344–D347. doi:10.1093/nar/gks1067

Siwach, P., and Ganesh, S. (2008). Tandem repeats in human disorders: mechanisms and evolution. *Front. Biosci.* 13:4467–4484. doi:10.2741/3017

Szalkowski, A. M., and Anisimova, M. (2013). Graph-based modeling of tandem repeats improves global multiple sequence alignment. *Nucleic Acids Res.* 41, e162. doi:10.1093/nar/gkt628

Travis, J. W., Clements, J., Eddy, S. R., Hubley, R., Jones, T. A., Jurka, J., et al. (2013). Dfam: a database of repetitive DNA based on profile hidden Markov models. *Nucleic Acids Res.* 41, D70–D82. doi:10.1093/nar/gks1265

Treangen, T. J., Abraham, A.-L., Touchon, M., and Rocha, E. P. C. (2009). Genesis, effects and fates of repeats in prokaryotic genomes. *FEMS Microbiol. Rev.* 33, 539–571. doi:10.1111/j.1574-6976.2009.00169.x

UniProt Consortium. (2014). Activities at the universal protein resource (UniProt). *Nucleic Acids Res.* 42, D191–D198. doi:10.1093/nar/gkt1140

Usdin, K. (2008). The biological effects of simple tandem repeats: lessons from the repeat expansion diseases. *Genome Res.* 18, 1011–1019. doi:10.1101/gr.070409.107

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 08 June 2016
doi: 10.3389/fbioe.2016.00035

Accurate Prediction of the Statistics of Repetitions in Random Sequences: A Case Study in Archaea Genomes

Mireille Régnier[1,2]* *and Philippe Chassignet*[2]

[1] Inria, Palaiseau, France, [2] LIX, Ecole Polytechnique, Palaiseau, France

Repetitive patterns in genomic sequences have a great biological significance and also algorithmic implications. Analytic combinatorics allow to derive formula for the expected length of repetitions in a random sequence. Asymptotic results, which generalize previous works on a binary alphabet, are easily computable. Simulations on random sequences show their accuracy. As an application, the sample case of Archaea genomes illustrates how biological sequences may differ from random sequences.

Keywords: *K*-mers, combinatorics, probability

OPEN ACCESS

Edited by:
Marco Pellegrini,
Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Travis Gagie,
University of Helsinki, Finland
Solon P. Pissis,
King's College London, UK

***Correspondence:**
Mireille Régnier
mireille.regnier@inria.fr

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *03 December 2015*
Accepted: *08 April 2016*
Published: *08 June 2016*

Citation:
Régnier M and Chassignet P (2016) Accurate Prediction of the Statistics of Repetitions in Random Sequences: A Case Study in Archaea Genomes. Front. Bioeng. Biotechnol. 4:35. doi: 10.3389/fbioe.2016.00035

1. INTRODUCTION

This paper provides combinatorial tools to distinguish biologically significant events from random repetitions in sequences. This is a key issue in several genomic problems as many repetitive structures can be found in genomes. One may cite microsatellites, retrotransposons, DNA transposons, long terminal repeats (LTR), long interspersed nuclear elements (LINE), ribosomal DNA, and short interspersed nuclear elements (SINE). In Treangen and Salzberg (2012), it is claimed that half of the genome consists of different types of repeats. Knowledge about the length of a maximal repeat is a key issue for assembly, notably the design of algorithms that rely upon de Bruijn graphs. In re-sequencing, it is a common assumption for aligners that any sequenced "read" should map to a single position in a genome: in the ideal case where no sequencing error occurs, this implies that the length of the reads is larger than the length of the maximal repetition. Average lengths of the repeats are given in Gu et al. (2000). Recently, heuristics have been proposed and implemented (Devillers and Schbath, 2012; Rizk et al., 2013; Chikhi and Medvedev, 2014).

A similar problem has been extensively studied: the prediction of the length of maximal common prefixes for words in a random set. Typical parameters are the background probability model, the size V of the alphabet, the length n of the sequence, and so on. Deviation from uniformity was studied for a uniform model as early as 1988 (Flajolet et al., 1988). A complexity index that captures the richness of the language is addressed in Janson et al. (2004). A distribution model, valid for binary alphabets and biased distributions, was introduced in Park et al. (2009), the so-called *trie profile* and extended to Patricia tries in Magner et al. (2014). The authors pointed out different "regimes" of randomness and a phase transition, by means of analytic combinatorics (Sedgewick and Flajolet, 2009). It was observed in Jacquet and Szpankowski (1994) that the average length of maximal common prefixes in a random set of n words is asymptotically equivalent to the average length of maximal repetitions in a random sequence of length n. Sets of words are considered below in the theoretical analysis. A comparison with the distribution of maximal repetitions in random sequences or real Archaea genomic sequences is presented in Section 3.

Our first goal is to extend results of Park et al. (2009) to the case of a general V-alphabet, including the special case $\{A, C, G, T\}$ where V is 4. A second goal is to compare the results consistency with random data and real genomic data in the finite range.

To achieve the first goal, we rely on an alternative, and simpler, probabilistic and combinatorial approach that is interesting *per se*. It avoids generating functions and the Poissonization–dePoissonization cycle that is used in Park et al. (2009) and it extends to non-binary alphabets. In that case, there is no closed formula for the asymptotic behavior. Nevertheless, the Lagrange multipliers allow to derive it as the solution of an equation that can be computed numerically.

Explicit and computable bounds for the profile of a random set of n words are provided. Three domains can be observed. A first domain is defined by a threshold k for the length, called the *completion length*: any prefix with a length smaller than this threshold occurs at least twice. This threshold is extremely stable over the data sets and it is highly predictable. A similar phenomenon was observed for a uniform model in Fagin et al. (1979a) and a biased model (Mahmoud, 1992; Park et al., 2009). For larger lengths, some prefixes occur only once. In a second domain, called the *transition phase*, the number of maximal common prefixes is sublinear in the size n of the sequence: increasing first, then decreasing slowly, and, finally, dropping rapidly. In the third domain, for a length larger than some *extinction length*, almost no common prefix of that length occurs. Despite the fact that these bounds are asymptotic, a good convergence is shown in practice for random texts when a second-order term is known.

Differences between the model and the observation are studied on the special case of Archaea genomes. A dependency to the GC-content, which is a characteristic of each genome, is exhibited. Regimes and transitions are studied on these genomic data and theoretical results are confirmed, with a drift in the values of transition thresholds. Notably, the length of the largest repetitions is much larger than expected. This difference between the model and the observation arises from the occurrences of long repeated regions.

Section 2 is devoted to Main Results, to be proved in Section 4. First, some notations are introduced; then, an algebraic expression for the expectation of the number of maximal common prefixes in a sequence is derived in Theorem 2.1. Second, this expression is split between two sums that are computable in practical ranges. Then, it is shown that a Large Deviation principle applies. It yields first and second order asymptotic terms, and oscillations, that are provided in Theorem 2.2. A comparison between exact, approximate, and asymptotic expressions is presented in Section 3.

2. MAIN RESULTS

It is assumed throughout this study that sequences and words are randomly generated according to a biased Bernoulli model on an alphabet of size V. Let $p_1, \cdots, p_V$ denote the probabilities of the V characters $\chi_1, \cdots, \chi_V$.

Definition 2.1. For any i in $\{1, \cdots, V\}$, one notes

$$\beta_i = \log \frac{1}{p_i}.$$

Additionally,

$$p_{min} = \min\{p_i; 1 \le i \le V\} \quad \text{and} \quad \alpha_{min} = \frac{1}{\log \frac{1}{p_{min}}} = \frac{1}{\max(\beta_i)}; \tag{1}$$

$$p_{max} = \max\{p_i; 1 \le i \le V\} \quad \text{and} \quad \alpha_{max} = \frac{1}{\log \frac{1}{p_{max}}} = \frac{1}{\min(\beta_i)}. \tag{2}$$

The two values $\min(\beta_i)$ and $\max(\beta_i)$ are different when the Bernoulli model is non-uniform.

2.1. Enumeration

Definition 2.2. Given U a set of words and an integer k, $k \ge 2$, a unique k-mer in U is a word $w\chi_i$ of length k such that

1. w is a prefix of at least two words in U;
2. and $w\chi_i$ is a prefix of a single word.

By convention, a unique 1-mer is a character χ_i that is a prefix of a single word.

Definition 2.3. Let U be a set of n words.

For $k \ge 1$, one denotes $B(n, k)$ the number of unique k-mers in U.

One denotes $\mu(n, k-1)$ the expectation of $B(n, k)$ over all sets of n words.

Remark: It follows from Definition 2.2 that quantity $B(n, k)$ is upper bounded by n. Observe that, for each random set U, it is the sum of a large number – V^k – of correlated random variables. Expectation $\mu(n, k)$ is studied below and compared in Section 3 with $B(n, k+1)$.

Profiles of repetitions can be expressed as a combinatorial sum.

Theorem 2.1. Given a length k, the expectation $\mu(n, k)$ satisfies:

$$\mu(n,k) = n \sum_{k_1+\cdots k_V=k} \binom{k}{k_1, \cdots, k_V} \phi(k_1, \cdots, k_V)\psi_n(k_1, \cdots, k_V) \tag{3}$$

where

$$\phi(k_1, \cdots, k_V) = p_1^{k_1} \cdots p_V^{k_V} \tag{4}$$

$$\psi_n(k_1, \cdots, k_V) = \sum_{i=1}^{V} p_i[(1 - \phi(k_1, \cdots, k_V)p_i)^{n-1} - (1 - \phi(k_1, \cdots, k_V))^{n-1}]. \tag{5}$$

Proof. A word $w\chi_i$ is a unique $(k+1)$-mer iff (i) w has length k and is the prefix of at least two words, including $w\chi_i$; (ii) $w\chi_i$ is not repeated.

Event (i) has probability

$$n\phi(k_1, \cdots, k_V)p_i[1 - (1 - \phi(k_1, \cdots, k_V))^{n-1}].$$

Event (ii), which is a sub-event of (i), has probability

$$n\phi(k_1, \cdots, k_V)p_i[1 - (1 - \phi(k_1, \cdots, k_V)p_i)^{n-1}].$$

2.2. A Combinatorial Expression

Definition 2.4. Given a k-mer w, let α denote $\frac{k}{\log n}$ and k_i denote the number of occurrences of character χ_i in w. The *objective function* is

$$\rho(k_1, \cdots, k_V) = \sum_{i=1}^{V} \frac{k_i}{k}\beta_i - \frac{1}{\alpha}. \tag{6}$$

The character distribution $(k_1, \cdots, k_V)$ of a k-mer may be viewed as *barycentric coordinates* for a point $\beta(k_1, \cdots, k_V) = \sum_{i=1}^{V} \frac{k_i}{k}\beta_i$ that lies in $[\min(\beta_i); \max(\beta_i)] = \left[\frac{1}{\alpha_{\max}}; \frac{1}{\alpha_{\min}}\right]$. The order of β points on that interval allows for a classification of k-mers that is a key to this study.

Definition 2.5. A k-mer w is said

- a common k-mer if $\rho(k_1,\ldots, k_V) < 0$;
- a transition k-mer if $\rho(k_1, \cdots, k_V) \geq 0$ and its ancestor is a common k-mer;
- a rare k-mer, otherwise.

Remark: If $\rho(k_1, \cdots, k_V) = 0$, the condition on the ancestor is trivially satisfied.

Definition 2.6. Given a set U of n words and an integer k, let $D_k(n)$ denote the set of character distributions $(k_1, \cdots, k_V)$ for rare and transition k-mers. Let $E_k(n)$ denote the set of character distributions for common k-mers.

The set $D_k(n)$ is the empty set if $k < \alpha_{min} \log n$ and is the set of character distributions $(k_1, \cdots, k_V)$ if $k > \alpha_{max} \log n$. Computation of (3) is split among the two sets $D_k(n)$ and $E_k(n)$. Computations show that the main contribution arises from transition k-mers. A probabilistic interpretation will be discussed in 2.4.

Notation: Let $S(k)$ and $T(k)$ be

$$S(k) = n \sum_{D_k(n)} \binom{k}{k_1 \cdots k_V} \phi(k_1, \cdots, k_V)\psi_n(k_1, \cdots, k_V); \quad (7)$$

$$T(k) = n \sum_{E_k(n)} \binom{k}{k_1 \cdots k_V} \phi(k_1, \cdots, k_V)\psi_n(k_1, \cdots, k_V). \quad (8)$$

So $\mu(n, k)$ rewrites

$$\mu(n, k) = S(k) + T(k). \quad (9)$$

These sums $S(k)$ and $T(k)$ can be efficiently computed for moderate k, up to a few hundred, approximately. In practice, $\alpha_{max} \log n$ is below this threshold for the sizes of actual genomes and for their ordinary GC content value. The simulations in Section 3 show that this estimation is rather tight. Behavior and asymptotic estimates are derived and discussed in the next section.

2.3. Asymptotic Estimates

In this section, asymptotic estimates for (3) are derived. First, some characteristic functions are introduced. Then, it is observed that a Large Deviation Principle applies for the combinatorial sums to be computed and asymptotics for the dominating term follow. Amortized terms are also computed. It is shown in Section 3 that this second-order term cannot be neglected in the finite range.

2.3.1. Notations

For general alphabets, asymptotic behavior is a function of the solution of an equation and depends on domains whose bounds are defined below.

Definition 2.7. Let $(p_i)_{1 \leq i \leq V}$ be a Bernoulli probability distribution. Let σ_2 denote $\sum_{i=1}^{V} p_i^2$.

The *fundamental ratio*, noted $\tilde{\alpha}$, is $\left(\sum_i p_i \log \frac{1}{p_i}\right)^{-1}$.

The *transition ratio*, noted $\bar{\alpha}$, is $\sigma_2\left(\sum_i p_i^2 \log \frac{1}{p_i}\right)^{-1}$.

The *extinction ratio*, noted α_{ext}, is $\frac{2}{\log \frac{1}{\sigma_2}}$.

Definition 2.8. Let α be a real value in $[\alpha_{min}, \alpha_{max}]$. Let τ_α be the unique real root of the equation

$$\frac{1}{\alpha} = \frac{\sum_{i=1}^{V} \beta_i e^{-\beta_i \tau}}{\sum_{i=1}^{V} e^{-\beta_i \tau}} \quad (10)$$

Let ψ be the function defined in $[\alpha_{min}, \alpha_{ext}]$ as

$$\alpha_{\min} \leq \alpha \leq \bar{\alpha} : \psi(\alpha) = \tau_\alpha + \alpha \log\left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right);$$

$$\bar{\alpha} \leq \alpha : \psi(\alpha) = 2 - \alpha \log \frac{1}{\sigma_2}.$$

Proposition 2.1. The following property holds

$$\alpha_{\min} \leq \tilde{\alpha} \leq \bar{\alpha} \leq \alpha_{\max} \leq \alpha_{\text{ext}}.$$

Function ψ increases on $[\alpha_{min}, \tilde{\alpha}]$ and decreases on $[\tilde{\alpha}, \infty]$. It satisfies

$$\psi(\alpha_{\min}) = \psi(\alpha_{\text{ext}}) = 0 \text{ and } \psi(\tilde{\alpha}) = 1. \quad (11)$$

Remark: Uniqueness of τ_α is shown in Section 4.2. As $\tau_{\bar{\alpha}} = 2$, ψ is continuous at $\alpha = \bar{\alpha}$, with $\psi(\bar{\alpha}) = 2 - \bar{\alpha} \log \frac{1}{\sigma_2}$.

2.3.2. Asymptotic Results

Theorem 2.2. Given a length $\alpha \log n$, when n tends to ∞ the ratio $\frac{\log \mu(n, \alpha \log n)}{\log n}$ satisfies:

$$0 \leq \alpha \leq \alpha_{min} \text{ or } \alpha_{ext} \leq \alpha : \frac{\log \mu(n, \alpha \log n)}{\log n} \leq 0; \quad (12)$$

$$\alpha_{min} \leq \alpha \leq \alpha_{ext} : \frac{\log \mu(n, \alpha \log n)}{\log n} \sim \psi(\alpha). \quad (13)$$

Moreover, let ξ be the function defined in $[\alpha_{min}, \alpha_{ext}]$ as $\xi(\alpha) = \frac{\mu(n, \alpha \log n)}{\log n} - \psi(\alpha)$. It satisfies

$$\alpha_{min} \leq \alpha \leq \bar{\alpha} : \xi(\alpha) \sim -\frac{V-1}{2} \frac{\log(\alpha \log n)}{\log n}; \quad (14)$$

$$\bar{\alpha} \leq \alpha \leq \alpha_{ext} : \xi(\alpha) \sim \frac{\log(1 - \sigma_2)}{\log n}. \quad (15)$$

Proof. The key to the proof when α ranges in $[\alpha_{min}, \alpha_{max}]$ is that $\psi_n(k_1, \cdots k_V)$ is maximal when $\rho(k_1, \cdots k_V)$ is close to 0. Sum $T(k)$ satisfies a Large Deviation Principle.

$$\frac{\log T(\tilde{k})}{k} \sim \max\left\{-\sum_{i=1}^{V} \frac{k_i}{k} \log \frac{k_i}{k}; \rho(k_1, \cdots, k_V) = 0\right\}. \quad (16)$$

The maximization problem rewrites as

$$\max\left\{\sum_{i=1}^{V} \theta_i \log \frac{1}{\theta_i}; \sum_{i=1}^{V} \theta_i = 1; \sum_{i=1}^{V} \beta_i \theta_i = \frac{1}{\alpha}; 0 \leq \theta_i \leq 1\right\} \quad (17)$$

The maximum value is $\tau_\alpha + \alpha \log \left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right)$ that is reached for the V-tuple $\left(\theta_i = \frac{e^{-\beta_i \tau_\alpha}}{\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}}\right)_{1 \le i \le V}$.

$S(k)$ satisfies again a Large Deviation Principle when $\alpha < \bar{\alpha}$, which yields the asymptotic result in this range. For larger α, $S(k)$ is approximately $(1-\sigma_2) n^{1-\alpha \log \frac{1}{\sigma_2}}$ that dominates $T(k)$.

Details for the proof, including the short and long lengths, are provided in Section 4.

Remark: The discussion will depend of the ratio $\alpha = \frac{k}{\log n}$. Possible values for α range over a *discrete* set as they are constrained to be the ratio of an integer by the log of an integer. An interesting property is that, for any real α, the set $T = \{n \in N; \alpha \log n \in N\}$ is either empty or infinite. Indeed, when T is non-empty, it contains all values $n(\alpha)^p$ where $n(\alpha)$ denotes the minimum value of T. It is beyond the scope of this paper to establish the number of other possible solutions.

2.3.3. Domains

Different domains arise from this Theorem, which were observed in Park et al. (2009). Equalities $\psi(\alpha_{min}) = 0$ and $\psi(\bar{\alpha}) = 2 - \bar{\alpha} log \frac{1}{\sigma_2}$ show that there is a continuity between domains.

When α lies inside the domain $[\alpha_{min}, \alpha_{ext}]$, the ratio $\frac{\log \mu(n, \alpha \log n)}{\log n}$ is positive and parameters $\mu(n, \alpha \log n)$ are *sublinear* in the size n of the text: some k-mers – mostly transition k-mers – are unique k-mers. Observe that the maximum value for $\psi(\alpha)$ is 1. When the Bernoulli model is uniform, this central domain is empty.

When the length is smaller than the *completion length* $\alpha_{min} \log n$ or greater than the *extinction length* $\alpha_{ext} \log n$, the ratio $\frac{\log \mu(n, \alpha \log n)}{\log n}$ is negative.

2.3.4. Oscillations

Parameters $(k_1, \cdots, k_V)$ in the combinatorial sums are integers. As the optimum values $(k\theta_i)_{1 \le i \le V}$ may not be integers, the practical maximum is a close point on the lattice $(k_1, \cdots, k_V)$. The difference introduces a multiplicative factor that ranges in $\left[-\log \frac{p_{max}}{p_{min}}, \log \frac{p_{max}}{p_{min}}\right]$. This leads to a small *oscillation* of $\log \mu(n, k)$. For large n, this contribution to $\frac{\log \mu(n,k)}{\log n}$ becomes negligible. As mentioned above, the set of lengths n that are *admissible* for a given α is very sparse. Nevertheless, an approximate value may be used: for instance, for an integer k', $\frac{1}{k'} \log \left\lceil n(\alpha)^{\frac{k'}{k}} \right\rceil$ is very close to α. This oscillation phenomenon was first observed in Nicodème (2005).

2.3.5. Binary Alphabets

Results for binary alphabets in Park et al. (2009) steadily follow from Theorem 2.2. A rewriting of ψ leads to alternative expression (18). This *explicit* expression points out the dependency to the distances to α_{min} and α_{max}, and the behavior around these points.

Corollary 2.1. Assume that the alphabet is binary. Then

$$\psi(\alpha) = \frac{\alpha}{\log \frac{p_{max}}{p_{min}}} \log[s_\alpha{}^{\frac{1}{\alpha} - \frac{1}{\alpha_{min}}} + s_\alpha{}^{\frac{1}{\alpha} - \frac{1}{\alpha_{max}}}] \qquad (18)$$

where

$$s_\alpha = \frac{\alpha_{min}}{\alpha_{max}} \cdot \frac{\alpha - \alpha_{min}}{\alpha_{max} - \alpha}. \qquad (19)$$

A similar result holds for DNA sequences when the alphabet is 4-ary and the probability distribution satisfies $p_A = p_T$ and $p_C = p_G$. Such a distribution is defined by its GC-content $p_G + p_C$.

2.4. A Probabilistic Interpretation

The main contribution to $\mu(n, k)$ arises from k-mers with an objective function close to 0, i.e., transition k-mers. Such k-mers exist in the *transition phase* $[\alpha_{min} \log n, \alpha_{max} \log n]$ where they coexist with rare or common k-mers. Observe that this phase is *shrinked* when the Bernoulli model is uniform, as $p_{min} = p_{max}$ and $\alpha_{min} = \alpha_{max}$. Therefore, most unique k-mers are concentrated on the two lengths $\lfloor \alpha_{min} \log n \rfloor$ and $\lceil \alpha_{min} \log n \rceil$, as observed initially in Fagin et al. (1979b).

Let k be some integer in the transition phase. First, the relative contribution of $S(k)$ and $T(k)$ to $\mu(n, k)$ varies with the length k. For lengths close to $\alpha_{min} \log n$, most words are common and $T(k)$ dominates $S(k)$. When k increases, the proportion of common words decreases and the relative contribution of $T(k)$ decreases.

Second, the dominating term in $\mu(n, k)$ arises from transition k-mers. Let w be a word of length k, the character distribution in w be $(k_1, \cdots, k_V)$ and χ_i be some character. The number of words that admit w or $w\chi_i$ as a prefix fluctuates around the expectations $n\phi(k_1, \cdots, k_V)$ and $n\phi(k_1, \cdots, k_V)p_i$, respectively. On the one hand, when word $w\chi_i$ is a rare word, $n\phi(k_1, \cdots, k_V)$ is less than 1. The smallest $n\phi(k_1, \cdots, k_V)$ is, the less likely the actual number of occurrences of w is greater than 2 and the smallest the contribution of $w\chi_i$ to $S(k)$, and $\mu(n, k)$, is. On the other hand, let $w\chi_i$ be a common $k+1$-mer; w is a common k-mer and then $n\phi(k_1, \cdots, k_V)$ is greater than 1. The largest $n\phi(k_1, \cdots, k_V)$ is, the more likely the word $w\chi_i$ is repeated and the smallest the contribution to $T(k)$, and $\mu(n, k)$, is.

For a short length, i.e., k smaller than the completion length k_{min}, all words are common. In a given sequence, most k-mers are repeated at least twice and there is (almost) no unique k-mers.

For a large length k, i.e., k greater than k_{max}, all words are rare. Nevertheless the number of unique k-mers remains sublinear in n in the range $[\alpha_{max} \log n, \alpha_{ext} \log n]$: the sum of small contributions arising from a large number of possible words is significant.

A folk theorem (Szpankowski, 2001; Jacquet and Szpankowski, 2015) claims that the objective function is concentrated around $\frac{1}{\tilde{\alpha}} - \frac{1}{\alpha}$. Consequently, when $\alpha = \tilde{\alpha}$, most k-mers are transition k-mers and the exponent, the ψ function, is maximal.

3. EXPERIMENTS AND ANALYSIS

Simulations are presented for random and real data. For each simulation, a suffix tree (Ukkonen, 1995) is built, where each leaf represents a unique k-mer. For random cases, the Ukkonen's insertion step is iterated until a tree with exactly n leaves is build. This requires $n + k_{ins}$ insertions of symbols, where $k_{ins} > 0$ is relatively small (there is a value of a few dozen in practice for considered n). One can observe that the event of having n leaves after $n + k - 1$ insertions corresponds to the fact that the trailing k-mer is unique in the sequence of length $n + k - 1$.

Even if a statistical bias exists, with respect to the case of a set of N random words analyzed in previous sections, this bias for respective values on k and n is below the numeric precision used for tables below.

Then, one simulation that is related to the case of a set of n random words, requires the generation of the order of N random symbols from a small alphabet, following a Bernoulli scheme. For this range of n, and even in the case of a hundred consecutive simulations, this corresponds to a regular use of a common random number generator (Knuth, 1998).

A first set of simulation deals with the case of random sequences over a binary alphabet, since the results can be compared with previous work. A second set addresses the case of random sequences over a quaternary alphabet $\{A, C, G, T\}$ with a constrained distribution such that probabilities $p_A \approx p_T$ and $p_C \approx p_G$ as it is the case for DNA sequences (where the sum $p_C + p_G$ is also known as the GC-content). Results on such random sequences are then compared with the sample biological sequence of an Archaea (*Haloferax volcanii*).

An implementation with a suffix array (Manber and Myers, 1993) allows for a compact representation and an efficient counting (Beller et al., 2013).

3.1. Random data

A hundred binary sequences were randomly generated. The number of leaves in each tree was fixed to $n = 5000000$ and the Bernoulli parameter was $p_{max} = 0.7000$. Therefore, $p_{min} = 0.3000$, $\tilde{p} = 0.5429$, and $\log n = 15.4249$. The thresholds for α and the corresponding lengths $\alpha \log n$ are:

$\alpha_{min} = 0.8306$	$\tilde{\alpha} = 1.6370$	$\bar{\alpha} = 2.0484$	$\alpha_{max} = 2.8035$	$\alpha_{ext} = 3.6714$
$k_{min} = 12.81$	$\tilde{k} = 25.25$	$\bar{k} = 31.60$	$k_{max} = 43.24$	$k_{ext} = 56.63$

3.1.1. Statistical Behavior on Random Sets

Throughout experiments, every sample profile for a given sequence fluctuates very little around the expectation.

Table 1 provides experimental results averaged over a hundred binary sequences. Short length with no observed unique k-mer is removed. Column 2 gives the mean of $B(k+1)$, i.e., the mean number of observed leaves at depth $k+1$, over the set of a hundred simulations. Columns 3 to 5 give the computed values for $S(k)$, $T(k)$, and $\mu(k)$, using the expressions, equations (7–9).

The actual number of leaves $B(n, k+1)$ is very close to the average value $\mu(n, k)$, and simulations show that this is the general case when (only) a hundred simulations are performed: $\mu(n, k)$ is a very good prediction.

Observed lengths of extinction also show very little variations. In array below, each column gives n_k, the number of sequences out of the one hundred sample set for which the longest repetition had length k.

Distribution of the extinction level for 100 random binary sequences. p_{max} is 0.7.

k	51	52	53	54	55	56	57	58	59	60	61	62	63	64
n_k	10	16	13	19	14	14	6	1	1	2	1	1	0	2

In the binary case, the predicted extinction length is between 56 and 57. It is noticeable that, in most cases, the observed depth is slightly smaller than this value. In **Table 1**, value 0.04 for $\mu(n, 61)$ means that one expects a total of four leaves at depth 60 over one hundred sequences. In that run, exists a total amount of 8.

3.1.2. Quality of Estimates

1. *Tightness of the asymptotic estimates.* Asymptotic estimates (13) given in Column 7 significantly *overestimate* the observed values in Column 6 that is computed directly from Column 2 and n. A first conclusion is that first-order asymptotics provide a *poor prediction*: next term is $O\left(\frac{1}{\log n}\right)$ that goes slowly to 0.
2. *Tightness of the second-order asymptotics.* Second term for the asymptotic $\xi(\alpha)$ ensures a much better approximation in Column 8.
3. *Growth of asymptotic estimates.* Observed values increase with length until $k = \tilde{k}$ and then decrease. This is consistent with the variation of asymptotic values $\psi(\alpha)$.

3.1.3. Dependency to Probability Bias

Thresholds were computed for a given sequence length n and various probabilities. The more p_{max} departs from 0.5, the value for the uniform model, the largest the extinction length is. The completion length, k_{min}, slightly decreases, while the extinction length significantly increases. Nevertheless, this effect is limited when the largest probability p_{max} remains in the range [0.5;0.7].

Dependency of thresholds to p_{max} for binary alphabets, $n = 5{,}000{,}000$.

p_{max}	k_{min}	$\tilde{k}$	$\bar{k}$	k_{max}	k_{ext}
0.50	22.25	22.25	22.25	22.25	44.51
0.55	19.32	22.42	22.74	25.80	45.16
0.60	16.83	22.92	24.27	30.20	47.18
0.65	14.69	23.82	27.06	35.81	50.83
0.70	12.81	25.25	31.60	43.25	56.63
0.75	11.13	27.43	38.80	53.62	65.64
0.80	9.58	30.83	50.63	69.13	79.99
0.85	8.13	36.49	71.78	94.91	104.80
0.90	6.70	47.45	116.72	146.40	155.45
0.95	5.15	77.70	259.56	300.72	309.05

3.2. Long Repetitions in Archaea Genomes

The experimental data set is the sequence from *Haloferax volcanii DS2 chromosome, complete genome* (Hartman et al., 2010). The alphabet is quaternary. Profile results are shown in **Table 2**.

Sequence length is $n = 2847757$. The observed symbol frequencies are $p_A = 0.1655$; $p_C = 0.3334$; $p_G = 0.3330$; $p_T = 0.1681$. Therefore, observed *GC-content* is 0.6664. Parameters for an approximate degenerated quaternary model are $p_A = p_T = p_{min} = 0.1668$; $p_C = p_G = p_{max} = 0.3332$; $\tilde{p} = 0.2645$; and $\log n = 14.8620$. The thresholds for the domain are

$\alpha_{min} = 0.5584$	$\tilde{\alpha} = 0.7520$	$\bar{\alpha} = 0.8079$	$\alpha_{max} = 0.9099$	$\alpha_{ext} = 1.5609$
$k_{min} = 8.30$	$\tilde{k} = 11.18$	$\bar{k} = 12.01$	$k_{max} = 13.52$	$k_{ext} = 23.20$

Statistics on one hundred random sequences with same parameters are shown in **Table 3**. GC-content is 0.6664. Extinction level is provided in **Table 4**. Observe first a good match between the observed values, the predicted values for $\mu(n, k)$, and the asymptotic values for random data. As shown for binary alphabets,

TABLE 1 | Mean profile for 100 random binary sequences.

	Observed	Predicted			Observed	Asymptotic		
k	$B(k+1)$	$S(k)$	$T(k)$	$\mu(n,k)$	$\frac{\log B(k+1)}{\log n}$	$\psi(\alpha)$	$\psi(\alpha)+\xi(\alpha)$	
11	0.29	0	0.3	0.3	−0.0803			
12	7.91	0	8.3	8.3	0.1341			
								k_{min}
13	87.87	0.1	86.9	87.1	0.2902	0.0843	*0.0012*	
14	552.88	1.2	550.3	551.5	0.4094	0.3340	*0.2485*	
15	2456.77	86.6	2366.4	2453.0	0.5061	0.4962	*0.4085*	
16	8269.20	209.4	8069.1	8278.5	0.5848	0.6181	*0.5282*	
17	22516.20	406.1	22097.7	22503.8	0.6497	0.7136	*0.6218*	
18	51085.15	4823.8	46267.2	51091.0	0.7028	0.7897	*0.6960*	
19	99387.01	6636.1	92717.6	99353.7	0.7460	0.8504	*0.7549*	
20	169303.03	37415.5	131882.6	169298.1	0.7805	0.8984	*0.8013*	
21	256358.10	42003.9	214454.4	256458.3	0.8074	0.9357	*0.8370*	
22	349801.23	137615.9	212264.2	349880.1	0.8276	0.9635	*0.8634*	
23	434625.83	134807.6	299824.7	434632.4	0.8416	0.9830	*0.8814*	
24	495572.93	122283.1	373279.8	495562.8	0.8501	0.9949	*0.8919*	
25	522788.19	255284.4	267476.3	522760.7	0.8536	0.9998	*0.8955*	
								$\tilde{k}$
26	513374.76	211204.2	302252.5	513456.7	0.8524	0.9982	*0.8926*	
27	472126.51	315154.7	157087.0	472241.6	0.8470	0.9906	*0.8838*	
28	408946.76	242583.4	166360.3	408943.7	0.8377	0.9772	*0.8692*	
29	335080.05	273441.0	61579.7	335020.7	0.8248	0.9582	*0.8491*	
30	260999.29	198163.4	62712.5	260875.9	0.8086	0.9339	*0.8236*	
31	194100.36	137502.0	56463.1	193965.1	0.7894	0.9043	*0.7930*	
								$\bar{k}$
32	138437.13	122218.3	16090.9	138309.2	0.7675	0.8699	*0.8136*	
33	95017.33	80937.1	14067.8	95004.9	0.7431	0.8346	*0.7783*	
34	63082.67	60397.1	2744.6	63141.7	0.7165	0.7993	*0.7430*	
35	40742.97	38411.9	2368.9	40780.8	0.6882	0.7639	*0.7077*	
36	25679.21	23888.2	1817.4	25705.6	0.6582	0.7286	*0.6724*	
37	15860.59	15622.9	255.8	15878.7	0.6270	0.6933	*0.6371*	
38	9645.84	9455.0	194.2	9649.2	0.5948	0.6580	*0.6018*	
39	5791.32	5772.7	15.9	5788.6	0.5617	0.6227	*0.5664*	
40	3433.87	3426.4	12.1	3438.5	0.5278	0.5874	*0.5311*	
41	2032.57	2027.2	0.4	2027.6	0.4938	0.5520	*0.4958*	
42	1188.84	1189.0	0.3	1189.3	0.4590	0.5167	*0.4605*	
43	692.28	694.8	0.2	695.0	0.4240	0.4814	*0.4252*	
								k_{max}
44	402.75	405.1	0	405.1	0.3889	0.4461	*0.3899*	
45	233.35	235.7	0	235.7	0.3535	0.4108	*0.3545*	
46	135.42	137.0	0	137.0	0.3182	0.3755	*0.3192*	
47	78.39	79.6	0	79.6	0.2828	0.3401	*0.2839*	
48	44.69	46.2	0	46.2	0.2463	0.3048	*0.2486*	
49	25.35	26.8	0	26.8	0.2096	0.2695	*0.2133*	
50	14.57	15.6	0	15.6	0.1737	0.2342	*0.1780*	
51	8.44	9.0	0	9.0	0.1383	0.1989	*0.1426*	
52	4.76	5.2	0	5.2	0.1012	0.1636	*0.1073*	
53	2.76	3.0	0	3.0	0.0658	0.1282	*0.0720*	
54	1.74	1.8	0	1.8	0.0359	0.0929	*0.0367*	
55	1.02	1.0	0	1.0	0.0013	0.0576	*0.0014*	
56	0.64	0.6	0	0.6	−0.0289	0.0223	*−0.0339*	
								k_{ext}
57	0.32	0.3	0	0.3	−0.0739	−0.0130		
58	0.18	0.2	0	0.2	−0.1112	−0.0483		
59	0.16	0.1	0	0.1	−0.1188	−0.0836		
60	0.12	0.07	0	0.07	−0.1375	−0.1190		
61	0.08	0.04	0	0.04	−0.1637	−0.1543		
62	0.06	0.02	0	0.02	−0.1824	−0.1896		
63	0.04	0.01	0	0.01	−0.2087	−0.2249		
64	0.04	0.008	0	0.008	−0.2087	−0.2602		

(p_{max}; p_{min}) = (0.7; 0.3).

TABLE 2 | Profile for the sequence from *Haloferax volcanii* DS2 chromosome, complete genome.

	Observed	Predicted			
k	$B(k+1)$	$S(k)$	$T(k)$	$\mu(n, k)$	
6	4	0	0.05	0.05	
7	1975	0	4e+02	4e+02	
8	41349	0	2e+04	2e+04	
					k_{min}
9	178523	781.2	213568.8	214350.1	
10	382032	66858.4	617279.6	684137.9	
11	542386	171711.2	742379.1	914090.3	
					$\tilde{k}$
12	570499	407976.5	215942.2	623918.7	
					$\bar{k}$
13	459330	259860.7	6512.5	266373.2	
					k_{max}
14	305002	87488.6	0	87488.6	
15	169317	25704.4	0	25704.4	
16	86379	7264.7	0	7264.7	
17	40391	2028.2	0	2028.2	
18	17432	564.1	0	564.1	
19	7866	156.7	0	156.7	
20	3830	43.5	0	43.5	
21	1957	12.1	0	12.1	
22	1229	3.4	0	3.4	
23	910	0.9	0	0.9	
					k_{ext}
24	733	0.3	0	0.3	
25	617	0.07	0	0.07	
26	561	0.02	0	0.02	
27	492	0.006	0	0.006	
28	446	0.002	0	0.002	
29	436	0.0005	0	0.0005	
30	397	0.0001	0	0.0001	
31	374	1e−05	0	1e−05	
32	359	2e−06	0	2e−06	
33	322	2e−08	0	2e−08	
...	*truncated*	...	*truncated*	...	

the observed extinction level for random sequences departs very little from the predicted k_{ext} level.

Numerous differences with random data can be observed on real genomes.

Interestingly, the behavior for short lengths and in the transition phase is similar to the random behavior. Observation and prediction have the same order of magnitude. In particular, the number of unique k-mers is maximum for length $\tilde{k}$ where observation and prediction coincide. For a real genome and a length k smaller than k_{min}, observed $B(n, k+1)$ is larger than predicted $\mu(n, k)$. This indicates, at a level $k+1$ where completion is expected, more leaves in the real trie, more missing words at level $k+2$. Simultaneously, less internal nodes occur at level $k+1$ because the total sum is constant and equal to V^{k+1}.

The effect of (non-random) repetitions is more sensible in the decreasing domain. First, the number of unique k-mers decreases much more slowly than expected for lengths larger than k_{max}. A significant gap can be observed around extinction level k_{ext}. The decrease rate, which was around 0.02–0.04 drops to 0.007 and then becomes even lower. Finally, the extinction level is much larger than the predicted value 23: the largest repetition is 1395 bp long.

To evaluate the contribution of long repetitions, one may erase the longest ones. When a word w is repeated, any proper suffix of w is also repeated. Consequently, once the longest repeated word is erased, one unique k-mer (only) disappears for each length larger than the length of the second largest subsequence (here, 935). The profile remains far from the random profile. This observation is still true if the 10 longest subsequences are erased.

4. COMBINATORIAL AND ANALYTIC DERIVATION

4.1. Lagrange Multipliers

Lagrange multipliers method allows to maximize an expression under constraints. To compute (17), one sets

$$F = \sum_{i=1}^{V} \theta_i \log \theta_i; \tag{20}$$

$$G = \sum_{i=1}^{V} \theta_i; \tag{21}$$

$$H = \sum_{i=1}^{V} \theta_i \beta_i. \tag{22}$$

Two constraints are given:

$$G = 1 \text{ and } H = \frac{1}{\alpha}.$$

An intermediary function $\phi_\alpha(\tau_1, \cdots \tau_V)$ is defined

$$\phi_\alpha = F + \lambda_\alpha G + \tau_\alpha H \tag{23}$$

In order to maximize ϕ under these two constraints, ϕ function is derived with respect to each random variable τ_i. This yields V equations

$$1 + \log \theta_i + \lambda_\alpha + \tau_\alpha \beta_i = 0. \tag{24}$$

Two indices i_{min} and i_{max} are chosen that satisfy $\beta_{i_{min}} \neq \beta_{i_{max}}$. For instance

$$\beta_{i_{min}} = \min(\beta_i)_{1 \leq i \leq V} = \log \frac{1}{p_{max}};$$
$$\beta_{i_{max}} = \max(\beta_i)_{1 \leq i \leq V} = \log \frac{1}{p_{min}}.$$

Solving equation (24) with indices i_{min} and i_{max} yields

$$\tau_\alpha = \frac{\log \theta_{i_{min}} - \log \theta_{i_{max}}}{\beta_{i_{max}} - \beta_{i_{min}}} = \log \frac{\theta_{i_{min}}}{\theta_{i_{max}}}^{\frac{1}{\beta_{i_{max}} - \beta_{i_{min}}}};$$
$$1 + \lambda_\alpha = \frac{\beta_{i_{min}} \log \theta_{i_{max}} - \beta_{i_{max}} \log \theta_{i_{min}}}{\beta_{i_{max}} - \beta_{i_{min}}}.$$

Remaining equations rewrite:

$$\log \theta_i = \log \theta_{i_{min}} + \tau_\alpha (\beta_{i_{min}} - \beta_i). \tag{25}$$

Using the constraint $\sum_{i=1}^{V} \theta_i = 1$ that yields

$$\theta_{i_{min}} e^{\beta_{i_{min}} \tau_\alpha} \sum_{i=1}^{V} e^{-\beta_i \tau_\alpha} = 1,$$

TABLE 3 | Mean profile for 100 random degenerated quaternary sequences.

	Observed	Predicted			Observed	asymptotic		
k	$B(k+1)$	$S(k)$	$T(k)$	$\mu(n,k)$	$\frac{\log B(k+1)}{\log n}$	$\psi(\alpha)$	$\psi(\alpha)+\xi(\alpha)$	
6	0.03	0	0.0	0.0	−0.2359			
7	363.29	0	363.9	363.9	0.3967			
8	21236.17	0	21252.2	21252.2	0.6704			k_{min}
9	214371.12	781.6	213574.7	214356.3	0.8260	0.7242	*0.5024*	
10	684344.68	66877.4	617315.1	684192.5	0.9041	0.9280	*0.6956*	
11	914013.67	171742.8	742383.0	914125.8	0.9235	0.9985	*0.7564*	$\tilde{k}$
12	623870.12	407973.4	215914.6	623888.0	0.8978	0.9655	*0.7147*	$\bar{k}$
13	266366.73	259826.1	6510.8	266336.9	0.8406	0.8792	*0.8574*	k_{max}
14	87424.58	87471.6	0	87471.6	0.7656	0.7930	*0.7711*	
15	25704.95	25698.5	0	25698.5	0.6832	0.7068	*0.6849*	
16	7253.72	7262.9	0	7262.9	0.5981	0.6206	*0.5987*	
17	2025.99	2027.6	0	2027.6	0.5123	0.5344	*0.5125*	
18	565.97	563.9	0	563.9	0.4265	0.4482	*0.4263*	
19	155.90	156.7	0	156.7	0.3397	0.3620	*0.3401*	
20	43.52	43.5	0	43.5	0.2539	0.2758	*0.2539*	
21	12.28	12.1	0	12.1	0.1688	0.1895	*0.1677*	
22	3.06	3.4	0	3.4	0.0753	0.1033	*0.0814*	
23	0.80	0.9	0	0.9	−0.0150	0.0171	*−0.0048*	k_{ext}
24	0.28	0.3	0	0.3	−0.0857	−0.0691	*−0.0910*	
25	0.14	0.1	0	0.1	−0.1323	−0.1553	*−0.1772*	

GC-content is 0.6664.

TABLE 4 | Distribution of the extinction level for 100 random degenerated quaternary sequences.

k	21	22	23	24	25
n_k	26	42	18	7	7

GC-content is 0.6664.

and an expression for $\theta_{i_{min}}$ follows. Therefore Equation 25 rewrites:

$$\theta_i = \frac{e^{-\beta_i \tau_\alpha}}{\sum_{i=1}^{V} \beta_i e^{-\beta_i \tau_\alpha}}. \tag{26}$$

Finally, Equation $\sum_{i=1}^{V} \theta_i \beta_i = \frac{1}{\alpha}$ yields equation (10).

$$\frac{1}{\alpha} = \frac{\sum_{i=1}^{V} \beta_i e^{-\beta_i \tau_\alpha}}{\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}}.$$

For this V-tuple

$$\sum_{i=1}^{V} \theta_i \log \theta_i = -\left(\sum_{i=1}^{V} \theta_i \beta_i\right) \tau_\alpha - \left(\sum_{i=1}^{V} \theta_i\right) \log\left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right)$$
$$= -\frac{\tau_\alpha}{\alpha} - \log\left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right).$$

4.2. Approximation Orders

Derivating the RHS of (10) yields $\frac{\sum_{i \neq j} (\beta_i + \beta_j)^2 e^{-(\beta_i+\beta_j)\tau}}{(\sum_i e^{-\beta_i \tau})^2}$ that is positive. Therefore, for any α, the solution to (10) is unique. Moreover, τ_α increases with α. Let

$$\psi_1(\alpha) = \tau_\alpha + \alpha \log\left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right); \tag{27}$$

$$\psi_2(\alpha) = 2 - \alpha \log \frac{1}{\sigma_2}. \tag{28}$$

Notably, the solutions τ_α of (10) associated with the four increasing values of α: $(\alpha_{min}, \tilde{\alpha}, \bar{\alpha}, \alpha_{max})$ are $(-\infty, 1+2, +\infty)$. Computing ψ for these values yields (11) and Equality $\psi_1(\tilde{\alpha}) = \psi_2(\tilde{\alpha})$.

Derivating both expressions yields

$$\frac{\partial \psi_1}{\partial \alpha}(\alpha) = \log\left(\sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right); \tag{29}$$

$$\frac{\partial \psi_1}{\partial \alpha}(\alpha) - \frac{\partial \psi_2}{\partial \alpha}(\alpha) = \log\left(\frac{1}{\sigma_2} \sum_{i=1}^{V} e^{-\beta_i \tau_\alpha}\right). \tag{30}$$

Both derivatives are monotone functions of τ_α. In equation (30), derivative is 0 when $\alpha = \bar{\alpha}$. Therefore, ψ is the maximum of the two values ψ_1 and ψ_2 over the interval $[\alpha_{min}, \alpha_{max}]$. The former equation is 0 if $\alpha = \tilde{\alpha}$. Therefore, ψ is maximum when $\alpha = \tilde{\alpha}$.

4.3. Approximations

4.3.1. Short Lengths

Assume that $k \leq \alpha_{min} \log n$. Each term $\phi(k_1, \cdots, k_V)$ is lower bounded by $p_{min}^k = n^{\alpha log p_{min}} = n^{-\frac{\alpha}{\alpha_{min}}}$. Each term $\psi_n(k_1, \cdots, k_V)$ is trivially bounded by $e^{-n^{1-\frac{\alpha}{\alpha_{min}}}}$ that is upper bounded by 1 and $n\psi_n(k_1, \cdots, k_V)$ tends to 0 when n goes to ∞. As $\sum \binom{k}{k_1 \cdots k_V} \phi(k_1, \cdots, k_V) = 1$, the ratio $\frac{\log \mu(n,k)}{\log n}$ is negative.

4.3.2. Moderate and Large Lengths

For a length k in the transition domain $[\alpha_{min} \log n, \alpha_{max} \log n]$, the objective function may be either positive or negative. When $k > \alpha_{max} \log n$, set $E_k(n)$ is empty and $\mu(n, k)$ reduces to $S(k)$.

The maximum M among the terms $e^{k\left(-\sum_i \frac{k_i}{k} \log \frac{k_i}{k} - \frac{1}{k} \log n\phi(k_1,\cdots,k_V)\right)}$ in $T(k)$ is reached when $\rho(k_1, \cdots, k_V)$ is 0. Due to the exponential decrease of $e^{-n\phi(k_1,\cdots,k_V)}$ when $n\phi(k_1, \cdots, k_V) \geq 1$, $\frac{T(k)}{k}$ is upper bounded. Computation of $\log M$ is done with Lagrange multipliers, as explained above.

Computation of $S(k)$ relies on the local development of $\psi_n(k_1, \cdots, k_V)$, that is $n(1-\sigma_2)\phi\ (k_1, \cdots, k_V)$. $S(k)$ rewrites ${\sigma_2}^k \tilde{S}(k) + (S(k) - {\sigma_2}^k \tilde{S}(k))$ where $\tilde{S}(k) = \sum_{\rho(k_1,\cdots,k_V)\leq 0} \binom{k}{k_i} \left(\frac{p_1^2}{\sigma_2}\right)^{k_1} \cdots \left(\frac{p_V^2}{\sigma_2}\right)^{k_V}$. This sum satisfies a Large Deviation Principle when $\rho(k_1, \cdots, k_V) + \frac{1}{\alpha} \geq \frac{1}{\tilde{\alpha}}$, or $\alpha < \tilde{\alpha}$. In this range, $\frac{\tilde{S}(k)}{k} \sim \max\left\{-\sum_{i=1}^V \frac{k_i}{k} \log \frac{k_i}{k}\right\}$, which was shown to be $\psi(\alpha)$.

When $\alpha > \tilde{\alpha}$, sum $\tilde{S}(k)$ rewrites $1 - \bar{S}(k)$ where

$$\bar{S}(k) = \sum_{\rho(k_1,\cdots,k_V)+\frac{1}{\alpha}<\frac{1}{\tilde{\alpha}}} \binom{k}{k_i} \left(\frac{p_1^2}{\sigma_2}\right)^{k_1} \cdots \left(\frac{p_V^2}{\sigma_2}\right)^{j_V}.$$

This sum satisfies a Large Deviation Principle and

$$\frac{\bar{S}(k)}{k} \sim \max\left\{-\sum_{i=1}^V \frac{k_i}{k} \log \frac{k_i}{k} + \sum_{i=1}^V \frac{k_i}{k} \log \frac{p_i^2}{\sigma_2}\right\}.$$

As $\sum_{i=1}^V \frac{k_i}{k} \log \frac{p_i^2}{\sigma_2} = -\frac{2}{\alpha} + \log \frac{1}{\sigma_2}$, this maximum is

$$-\frac{1}{\alpha}[2 - \alpha \log \frac{1}{\sigma_2} - \psi(\alpha)]$$

that is negative.

4.4. Binary Case

Barycentric coordinates of α are unique. Indeed, equation (10) reduces to a linear equation on the variable $e^{-(\beta_2-\beta_1)\tau}$

$$\frac{1}{\alpha} = \frac{\beta_1 + \beta_2 e^{-(\beta_2-\beta_1)\tau}}{1 + e^{-(\beta_2-\beta_1)\tau}}$$

where $\beta_2 - \beta_1 = \beta_{min} - \beta_{max} = \log \frac{p_{max}}{p_{min}}$. Therefore, $e^{-(\beta_2-\beta_1)\tau} = \frac{1-\alpha\beta_1}{\alpha\beta_2-1}$. Finally

$$\tau_\alpha = \frac{1}{\log \frac{p_{max}}{p_{min}}} \log \frac{\alpha\beta_2 - 1}{1 - \alpha\beta_1} = \frac{1}{\log \frac{p_{max}}{p_{min}}} \log \frac{\frac{1}{\alpha_{min}} - \frac{1}{\alpha}}{\frac{1}{\alpha} - \frac{1}{\alpha_{max}}}.$$

Function ψ rewrites, in the binary case:

$$\psi_\alpha = \tau_\alpha = \alpha \log e^{-\frac{1}{\alpha}\tau_\alpha} \left(e^{-\left(\beta_1 - \frac{1}{\alpha}\right)\tau_\alpha} + e^{-\left(\beta_2 - \frac{1}{\alpha}\right)\tau_\alpha}\right).$$

Observing that $e^{-(\beta_2-\beta_2)\tau_\alpha} = s_\alpha$ and changing variable τ_α into $(\beta_2 - \beta_1)$ yields $e^{-\left(\beta_1 - \frac{1}{\alpha}\right)\tau_\alpha} = {s_\alpha}^{-\left(\frac{1}{\alpha_{min}} - \frac{1}{\alpha}\right)}$ and $e^{-\left(\beta_2 - \frac{1}{\alpha}\right)\tau_\alpha} = {s_\alpha}^{-\left(\frac{1}{\alpha_{max}} - \frac{1}{\alpha}\right)}$.

5. CONCLUSION

This paper describes the behavior of the number of unique or repeated k-mers in a random sequence, on a general alphabet. Derivation relies on a combination of analytic combinatorics and on Lagrange multipliers. It simplifies an approach provided for binary alphabets and allows to address larger alphabets, including the quaternary alphabets, such as DNA alphabet. Precise asymptotic estimates are provided and a probabilistic interpretation is given. They are validated on random simulated data and shown to be valid in the finite range. Therefore, they provide a valuable tool to estimate a suitable read length for assembly purposes and tune parameters for assembly algorithms. Real genomes significantly depart from the random behavior for long repetitions. The general shape of the trie profile is observed, with a maximum of the number of unique k-mers at the expected length. However, for real genomes, a number of very short k-mers are missing and, on the contrary, one observes a number of very long repetitions. Besides these events, the behaviors are rather similar.

In the future, it is worth extending the method to generalized Patricia tries, Markov models and approximate repetitions.

AUTHOR CONTRIBUTIONS

Both authors contributed equally.

FUNDING

Inria-Cnrs-Poncelet grant Carnage.

REFERENCES

Beller, T., Gog, S., Ohlebusch, E., and Schnattinger, T. (2013). Computing the longest common prefix array based on the burrows–wheeler transform. *J. Discrete Algorithms* 18, 22–31. doi:10.1016/j.jda.2012.07.007

Chikhi, R., and Medvedev, P. (2014). Informed and automated k-mer size selection for genome assembly. *Bioinformatics* 30, 31–37. doi:10.1093/bioinformatics/btt310

Devillers, H., and Schbath, S. (2012). Separating significant matches from spurious matches in dna sequences. *J. Comput. Biol.* 19, 1–12. doi:10.1089/cmb.2011.0070

Fagin, R., Nievergelt, J., Pippenger, N., and Strong, H. R. (1979a). Extendible hashingâ – a fast access method for dynamic files. *ACM Trans. Database Syst.* 4, 315–344. doi:10.1145/320083.320092

Fagin, R., Nievergelt, J., Pippenger, N., and Strong, R. (1979b). Extendible hashing: a fast access method for dynamic files. *ACM Trans. Database Syst.* 4, 315–344. doi:10.1145/320083.320092

Flajolet, P., Kirschenhofer, P., and Tichy, R. F. (1988). Deviations from uniformity in random strings. *Probab. Theory Relat. Fields* 80, 139–150. doi:10.1007/BF00348756

Gu, Z., Wang, H., Nekrutenko, A., and Li, W. H. (2000). Densities, length proportions, and other distributional features of repetitive sequences in the human genome estimated from 430 megabases of genomic sequence. *Gene* 259, 81–88. doi:10.1016/S0378-1119(00)00434-0

Hartman, A. L., Norais, C., Badger, J. H., Delmas, S., Haldenby, S., Madupu, R., et al. (2010). The complete genome sequence of haloferax volcanii ds2, a model archaeon. *PLoS One* 5:e9605. doi:10.1371/journal.pone.0009605

Jacquet, P., and Szpankowski, W. (1994). Autocorrelation on words and its applications: analysis of suffix trees by string-ruler approach. *J. Comb. Theory A* 66, 237–269. doi:10.1016/0097-3165(94)90065-5

Jacquet, P., and Szpankowski, W. (2015). *Analytic Pattern Matching: From DNA to Twitter*. Reading, MA: Cambridge University Press.

Janson, S., Lonardi, S., and Szpankowski, W. (2004). "On the average sequence complexity," in *Combinatorial Pattern Matching*, eds S. C. Sahinalp, S. Muthukrishnan, and U. Dogrusoz (Berlin Heidelberg: Springer), 74–88.

Knuth, D. (1998). *The Art of Computer Programming, Volume Two, Seminumerical Algorithms*. Reading, MA.

Magner, A., Knessl, C., and Szpankowski, W. (2014). "Expected external profile of patricia tries," in *Proceedings of the Meeting on Analytic Algorithmics and Combinatorics* (Society for Industrial and Applied Mathematics), 16–24.

Mahmoud, H. (1992). *Evolution of Random Search Trees*. New York: John Wiley & Sons.

Manber, U., and Myers, G. (1993). Suffix arrays: a new method for on-line string searches. *SIAM J. Comput.* 22, 935–948. doi:10.1137/0222058

Nicodème, P. (2005). "Average profiles, from tries to suffix-trees," in *2005 International Conference on Analysis of Algorithms*, *Volume AD of DMTCS Proceedings*, ed. C. Martìnez (Barcelona, Spain: Discrete Mathematics and Theoretical Computer Science), 257–266.

Park, G., Hwang, H.-K., Nicodeme, P., and Szpankowski, W. (2009). Profile of trie. *SIAM J. Comput.* 38, 1821–1880. doi:10.1137/070685531

Rizk, G., Lavenier, D., and Chikhi, R. (2013). Dsk: k-mer counting with very low memory usage. *Bioinformatics* 29, 652–653. doi:10.1093/bioinformatics/btt020

Sedgewick, R., and Flajolet, P. (2009). *Analytic Combinatorics*. Reading, MA: Cambridge University Press.

Szpankowski, W. (2001). *Average Case Analysis of Algorithms on Sequences*. New York: John Wiley and Sons.

Treangen, T. J., and Salzberg, S. L. (2012). Repetitive dna and next-generation sequencing: computational challenges and solutions. *Nat. Rev. Genet.* 13, 36–46. doi:10.1038/nrg3117

Ukkonen, E. (1995). On-line construction of suffix trees. *Algorithmica* 14, 249–260. doi:10.1007/BF01206331

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 09 June 2015
doi: 10.3389/fbioe.2015.00071

Transposable element insertions in long intergenic non-coding RNA genes

Sivakumar Kannan[1†], Diana Chernikova[2†], Igor B. Rogozin[1†], Eugenia Poliakov[3], David Managadze[1], Eugene V. Koonin[1] and Luciano Milanesi[4]*

[1] National Center for Biotechnology Information, National Library of Medicine, National Institutes of Health, Bethesda, MD, USA, [2] Department of Genetics, Institute for Quantitative Biomedical Sciences, Geisel School of Medicine, Dartmouth College, Hanover, NH, USA, [3] Laboratory of Retinal Cell and Molecular Biology, National Eye Institute, National Institutes of Health, Bethesda, MD, USA, [4] Institute for Biomedical Technologies, National Research Council, Segrate, Italy

Transposable elements (TEs) are abundant in mammalian genomes and appear to have contributed to the evolution of their hosts by providing novel regulatory or coding sequences. We analyzed different regions of long intergenic non-coding RNA (lincRNA) genes in human and mouse genomes to systematically assess the potential contribution of TEs to the evolution of the structure and regulation of expression of lincRNA genes. Introns of lincRNA genes contain the highest percentage of TE-derived sequences (TES), followed by exons and then promoter regions although the density of TEs is not significantly different between exons and promoters. Higher frequencies of ancient TEs in promoters and exons compared to introns implies that many lincRNA genes emerged before the split of primates and rodents. The content of TES in lincRNA genes is substantially higher than that in protein-coding genes, especially in exons and promoter regions. A significant positive correlation was detected between the content of TEs and evolutionary rate of lincRNAs indicating that inserted TEs are preferentially fixed in fast-evolving lincRNA genes. These results are consistent with the repeat insertion domains of LncRNAs hypothesis under which TEs have substantially contributed to the origin, evolution, and, in particular, fast functional diversification, of lincRNA genes.

Keywords: mobile elements, molecular domestication, exaptation, junk DNA, long non-coding RNA, repetitive elements

OPEN ACCESS

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Roderic Guigo, Center for Genomic Regulation, Spain
Justin Blumenstiel, University of Kansas, USA

***Correspondence:**
Luciano Milanesi, Institute for Biomedical Technologies, National Research Council, Via Fratelli Cervi 93, 20090 Segrate, Italy
luciano.milanesi@itb.cnr.it

†Sivakumar Kannan, Diana Chernikova and Igor B. Rogozin have contributed equally to this work.

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *10 December 2014*
Accepted: *06 May 2015*
Published: *09 June 2015*

Citation:
Kannan S, Chernikova D, Rogozin IB, Poliakov E, Managadze D, Koonin EV and Milanesi L (2015) Transposable element insertions in long intergenic non-coding RNA genes. Front. Bioeng. Biotechnol. 3:71. doi: 10.3389/fbioe.2015.00071

Introduction

Traditionally, genomes have been perceived mostly as repositories of protein-coding genes. Although this might be largely true in the case of viruses, prokaryotes, and unicellular eukaryotes, numerous recent studies on the genomes of multicellular eukaryotes, particularly animals, have revealed a vast non-coding RNome, i.e., numerous genes encoding various classes of non-coding RNAs (ncRNAs) (Carninci et al., 2005; Mattick and Makunin, 2006; Ponting et al., 2009; Derrien et al., 2012; Amaral et al., 2013). Strikingly, the total number of genes for ncRNAs that are expressed from a mammalian genome seems to exceed the number of protein-coding genes several fold (Mattick and Makunin, 2006; Amaral et al., 2013). The classification of ncRNAs and validation of their functionality remain matters of intensive investigation and debate (Van Bakel and Hughes, 2009; Ponting and Belgard, 2010; Graur et al., 2013). Among many distinct classes of ncRNAs, the long non-coding RNA (lncRNA) is probably the most enigmatic group. The definition of a lncRNA is based solely on the transcript size: lncRNAs are defined as ncRNAs longer than 200 nt (Mattick and Makunin, 2006; Ponting et al., 2009).

Many lncRNAs are spliced, 5′capped, and polyadenylated (Okazaki et al., 2002; Carninci et al., 2005; Kapranov et al., 2007; Ponjavic et al., 2007). Based on the localization in the genome, lncRNAs can be divided into two distinct classes: (i) transcripts that overlap protein-coding genes, many of which are likely to be involved in sense–antisense regulation (Chen et al., 2005; Ponting and Belgard, 2010; Rinn and Chang, 2012) and (ii) long intergenic non-coding (linc)RNAs that are transcribed from genome regions separating protein-coding genes (Ponjavic et al., 2007; Mercer et al., 2008; Ponting et al., 2009).

The current knowledge on the functions of long intergenic non-coding RNAs (lincRNAs) is scarce because very few of the lincRNAs have been experimentally characterized. Nevertheless, the functional range of this class of ncRNA is believed to be broad on the basis of indirect evidence (Bertone et al., 2004; Ponjavic et al., 2007; Mercer et al., 2008; Ponting and Belgard, 2010; Ulitsky et al., 2011; Glazko et al., 2012; Ng et al., 2013). It has been proposed that lincRNAs could be involved in the regulation of many cellular processes (Mattick and Makunin, 2006; Loewer et al., 2010; Wang et al., 2011; Rinn and Chang, 2012). For example, they can affect transcription locally on the gene level (Martens et al., 2004; Martianov et al., 2007; Osato et al., 2007; Hirota et al., 2008) as well as target transcription regulators and thus affect transcription of many genes (Feng et al., 2006; Goodrich and Kugel, 2006). They can also target RNA polymerase II in human and mouse (Espinoza et al., 2007; Mariner et al., 2008) and thus affect the expression of an even broader range of genes. Furthermore, lincRNAs participate in the regulation of splicing (Munroe and Lazar, 1991; Beltran et al., 2008) and translation (Wang et al., 2005; Centonze et al., 2007). Well-characterized examples of lincRNAs involved in epigenetic processes are *Xist* (Brockdorff et al., 1992; Elisaphenko et al., 2008), *Kcnq1ot1* (Umlauf et al., 2004; Pandey et al., 2008), and *Air* (Nagano et al., 2008).

It is well established that, compared to protein-coding sequences and structural RNAs, lincRNAs are weakly conserved in evolution. Many early studies, therefore, branded the lincRNAs "transcriptional dark matter" and considered them to be generally non-functional (Van Bakel and Hughes, 2009; Robinson, 2010). However, low level or lack of detectable conservation does not necessarily imply that these molecules have no function (Pang et al., 2006). A case in point is the best-characterized, functionally important lincRNA gene, *Xist*, which is weakly conserved although it does contain evolutionary constrained regions (Elisaphenko et al., 2008). In general, lincRNAs show reduced substitution and insertion–deletion rates, which has been attributed to purifying selection (Ponjavic et al., 2007; Managadze et al., 2011). Taking into account that some lincRNA genes originated from protein-coding genes [for example, *Xist* (Duret et al., 2006; Elisaphenko et al., 2008)], it appears likely that many properties of lincRNAs would generally resemble those of protein-coding genes, despite the typically lower level of constraint. In particular, protein-coding genes that are highly expressed in many tissues typically evolve slower than genes with lower expression level and breadth (Duret and Mouchiroud, 2000; Krylov et al., 2003; Drummond and Wilke, 2008), and a similar dependence has been observed for lincRNA genes (Managadze et al., 2011). Taken together, these findings imply that an unknown but substantial fraction of lincRNAs are functional molecules rather than transcriptional noise and have evolutionary properties similar to those of protein-coding genes. However, the number of functionally characterized lincRNAs remains scarce (Amaral et al., 2013).

The origin of lincRNA genes generally remains enigmatic. However, analysis of the well-characterized *Xist* lincRNA has revealed fragmentary homology to a protein-coding gene *Lnx3* suggesting that the *Xist* genes emerged in early eutherians via integration of transposable elements (TEs) into the *Lnx3* gene, which gave rise to simple tandem repeats (Duret et al., 2006; Elisaphenko et al., 2008). The *Xist* gene promoter region and 4 of its 10 exons retain homology to exons of the *Lnx3* gene. The remaining six *Xist* exons including those containing simple tandem repeats show similarity to different TEs (Elisaphenko et al., 2008). Integration of TEs into the *Xist* gene apparently had been occurring throughout the course of evolution of this gene and most likely continues in contemporary eutherian species. Additionally, it has been shown that the combination of remnants of protein-coding sequences and TEs is not unique to the *Xist* gene but is also found in neighboring genes that encode non-coding nuclear RNAs (Elisaphenko et al., 2008; Kolesnikov and Elisafenko, 2010).

The discovery of the pivotal contribution of TEs to the evolution of the *Xist* gene prompts the question on a possible general role of TEs in the evolution of lincRNAs. Diverse TEs are widespread and abundant in the genomes of most eukaryotes (Smit, 1996; Brosius, 1999; Kidwell and Lisch, 2001; Deininger and Batzer, 2002). Different classes of TEs include mobile retrovirus-like elements, or retrotransposons, which transpose within the genome via RNA intermediates, and DNA transposons, which can relocate directly. Retrotransposons including long interspersed repetitive elements (LINEs), short interspersed repetitive elements (SINEs), and long terminal repeat (LTR) retrotransposons are widely represented in mammals (Smit, 1996; Deininger and Batzer, 2002). The LINEs are transcribed by RNA polymerase II and contain open reading frames (ORFs) (Temin, 1985). A complete and transpositionally active L1 element (the most common variety of LINEs) is ~7 kb long and contains a 5′-untranslated region (UTR) with an internal promoter, two ORFs (ORF1 and ORF2) and a 3′-UTR terminated by a polyadenylate-rich tail (Smit, 1996; Deininger and Batzer, 2002). The ORF1 encodes a putative RNA-binding protein ~40 kDa in size (Martin, 2006) whereas ORF2 encodes a protein with endonuclease and reverse transcriptase (RT) activities that generates cDNAs from RNA transcripts of the element (Loeb et al., 1986). The mobility of the LINE elements had been demonstrated in mouse and human genomes (Kazazian et al., 1988; Boccaccio et al., 1990). The SINEs are characterized by the presence of a split intragenic RNA polymerase III promoter and a 3′A-rich region often followed by an oligo(A) tail (Smit, 1996; Rogozin et al., 2000; Kapitonov and Jurka, 2003). The SINEs do not contain long ORFs and do not encode enzymes for transposition. Instead, transposition of SINEs apparently requires RT encoded by other TEs, in particular, LINEs (Smit, 1996; Deininger and Batzer, 2002). The LTR retrotransposons have LTRs that range from ~100 bp to over 5000 bp in size (Smit, 1996; Deininger and Batzer, 2002). The LTR retrotransposons are similar to retroviruses in organization, with transcriptional regulatory sequences located in the flanking LTRs, a RT priming site that is typically located immediately downstream

of an first LTR, and several ORFs encoding proteins involved in retrotransposition, in particular, RT and integrase (Smit, 1996; Deininger and Batzer, 2002).

The TEs are the primary contributors to the bulk of the genomic DNA in many eukaryotes, in particular mammals, and have the potential to contribute to the evolution of the hosts by providing novel regulatory or coding sequences (Makalowski, 2000). Different classes of regulatory regions in the human genome have been surveyed for the presence of TE-derived sequences (TES) to systematically assess the potential contribution of TEs to the regulation of human genes, and almost 25% of the analyzed promoter regions have been found to contain TES (Jordan et al., 2003; Feschotte, 2008; Bourque, 2009). In addition, numerous examples where experimentally characterized *cis*-regulatory elements are derived from TE sequences have been identified (Jordan et al., 2003; Bourque et al., 2008; Faulkner et al., 2009). Thus, thousands of human (and other mammalian) genes appear to be regulated, at least in part, by sequences derived from TEs (Jordan et al., 2003; Feschotte, 2008; Bourque, 2009). The TES are likely to have substantially contributed to evolutionary change in both gene specific and global patterns of mammalian protein-coding gene regulation (Makalowski, 2000; Jordan et al., 2003).

In light of the regulatory and structural effects that some TEs exert on host protein-coding and lncRNA genes (Makalowski, 2000; Jordan et al., 2003; Elisaphenko et al., 2008; Mattick et al., 2010; Wang et al., 2011; Kapusta et al., 2013; Johnson and Guigo, 2014), we sought to examine the contribution and conservation of TES to regulatory regions, exons and introns of human and mouse lincRNA genes. We found that introns of lincRNA genes contain the highest fraction of TES, followed by exons. The promoters of the lincRNAs contain the lowest fraction of TES but the largest fraction of ancient TES that are conserved between primates and rodents. The content of TES in lincRNA genes is substantially greater than in protein-coding genes, particularly in exons and promoter regions. These results are compatible with the view that TEs are major contributors to the origin and evolution of lincRNAs. We further sought to assess the potential utility of TES as an "evolutionary variable" by analyzing the correlations between the TES content, lincRNA expression, and sequence conservation.

Materials and Methods

Human and mouse lincRNA genes, the corresponding genomic alignments and expression data were taken from our previous work (Managadze et al., 2011) where the procedures of data processing are described in full details. Briefly, complete mouse and human probe sets were downloaded from the NRED database (Dinger et al., 2009) in the tab delimited and browser extensible data (BED, containing genomic coordinates) formats. The probe sets from platform GNF Atlas 2 (Mouse and Human), with the target classification "Non-coding Only," were used for further analysis. This protocol yielded 917 human and 5444 mouse probe sets. Only the probe sets that mapped to intergenic regions of the human and mouse genomes (i.e., between two adjacent protein-coding genes) were used for further analysis. The resulting list of lincRNAs was further filtered: sequences shorter than 200 nt were removed. This procedure yielded the final set of NCBI GenBank Accession IDs of 2390 mouse and 589 human lincRNAs and their corresponding microarray expression probe sets. The genomic coordinates and sequences of exons and introns of lincRNA and protein-coding genes were downloaded from the UCSC Table Browser (Karolchik et al., 2004), specifically, from "all_mrna" tables of mouse mm8 and human hg18 assemblies. Multiple alignments of these regions were fetched from Galaxy (Goecks et al., 2010). For the detection of TES, lincRNA and protein-coding genes were analyzed using RepeatMasker version open-3.1.3[1] with the following parameters: -w -s -no_is -cutoff 255 -frag 20000 -gff -species mouse/human. A TE insert was considered ancient if the pairwise alignment between human and mouse orthologous TE sequences was longer than 100 bp and contained <5% insertions/deletions (stringent definition) or 25% insertions/deletions (relaxed definition). Microarray data for normal (non-cancerous) tissues (73 human and 61 mouse tissues) were used to analyze the lincRNA expression. Log2-normalized median values of expression for each probe set across the tissues were calculated (Managadze et al., 2011). As an alternative method of measuring expression levels, the mouse RNA-seq data for eight tissues (the ENCODE project; modENCODE Consortium) were downloaded from the UCSC genome browser Web site[2] and pooled together. The RPKM value was calculated for each mouse lincRNA (Managadze et al., 2011). Pairwise evolutionary distances for human–macaque and mouse–rat lincRNA alignments were calculated using the DNADIST program from the PHYLIP package (Felsenstein, 1996), with the Kimura nucleotide substitution model. The lists of lincRNA genes and expression data are available at ftp://ftp.ncbi.nlm.nih.gov/pub/managdav/paper_suppl/TEs_lincRNA/.

One of the problems in the analysis of lincRNAs is that there is little overlap between lincRNA sets produced in different studies (Ulitsky et al., 2011; Chew et al., 2013; Managadze et al., 2013; Schuler et al., 2014). We used human and mouse datasets because these curated lincRNA sets have known evolutionary and gene expression properties (Managadze et al., 2011, 2013). Another reason for this choice is that we sought to analyze lincRNA datasets as different as possible from those used in previous studies (Kapusta et al., 2013; Johnson and Guigo, 2014) and to check how small sample size of human lincRNA set influences results. As shown below, the sample size does not perceptibly affect the conclusions of this study.

Results

Transposable Elements in Human and Mouse lincRNA Genes

Transposable element-derived sequences (TES for short) comprise at least half of the mammalian genomes, and in particular, are found in most lincRNAs. We identified TES in 69% of the human lincRNAs and 51% of the mouse lincRNAs. These values are somewhat lower than the previously reported 83% of TES in human lincRNAs (Kelley and Rinn, 2012) but nevertheless clearly show the importance of TE for lincRNA evolution. The

[1] http://www.repeatmasker.org

[2] http://hgdownload.cse.ucsc.edu/goldenPath/mm9/encodeDCC/wgEncodeLicrRnaSeq/

distribution of TES in 5′ flanking regions (putative core promoter regions), lincRNA exons, and introns is shown in the **Figure 1**. The lowest fraction of TES was found in the predicted core promoter regions (100 bp upstream regions), and the highest fraction of TES was observed in introns, whereas exonic sequences showed intermediate densities of TES (**Figure 1A**). This distribution of TES is compatible with the previously described general tendency of TES avoidance in functionally important regions of protein-coding genes (Jordan et al., 2003). In particular, similar to the protein-coding genes, the TES density in extended promoter regions has been found to be significantly greater than that in core promoter regions (Jordan et al., 2003). Notably, the fractions of TES in introns of lincRNAs and protein-coding genes are nearly identical, suggestive of comparable (weak) functional constraints. By contrast, in the exons and the core promoter regions of lincRNA genes, the fractions of TES are substantially and statistically significantly ($P < 10^{-5}$ according to the Fisher exact test) higher than in the respective regions of protein-coding genes (compare **Figures 1A,B**). These findings are consistent with the results of a previous study that employed different datasets of lincRNA genes (Kapusta et al., 2013), indicating that the distribution of TES in lincRNA genes is a robust feature. A more detailed analysis of the distribution of TES across lincRNAs is shown in Figure S1 in Supplementary Material. The most prominent feature of this distribution is the high fraction of lincRNAs with a low TES content: in 66% of human lincRNAs and 78% of mouse lincRNAs, the fraction of TES is <20% (Figure S1 in Supplementary Material).

The avoidance of TES in lincRNAs is consistent with purifying selection, which is an important feature of lincRNA evolution (Ponjavic et al., 2007; Managadze et al., 2011). The significant positive correlation between the evolutionary rate and the content of TES was observed for both human and mouse lincRNA sets (for human and mouse, respectively, the Pearson correlation coefficient are 0.183 and 0.337, $P < 10^{-5}$ for both comparisons) (**Figure 2**). We also tested the correlation between the expression level and the content of TES (Figure S2 in Supplementary Material). In many independent previous studies, it has been shown that protein-coding genes that are highly expressed in many tissues typically evolve slower than genes with lower expression level and breadth (Duret and Mouchiroud, 2000; Krylov et al., 2003; Drummond and Wilke, 2008), and a similar dependence has been observed for lincRNA genes (Managadze et al., 2011). Consistent with these observations, here we found a significant negative correlation between the content of TES and the expression level of lincRNAs (Figure S2 in Supplementary Material; for mouse RNA-seq data, Pearson correlation coefficient is −0.158, $P < 10^{-5}$; for mouse microarray data, Pearson correlation coefficient is −0.07, $P < 10^{-5}$; for human microarray data, Pearson correlation coefficient is −0.253, $P < 10^{-5}$).

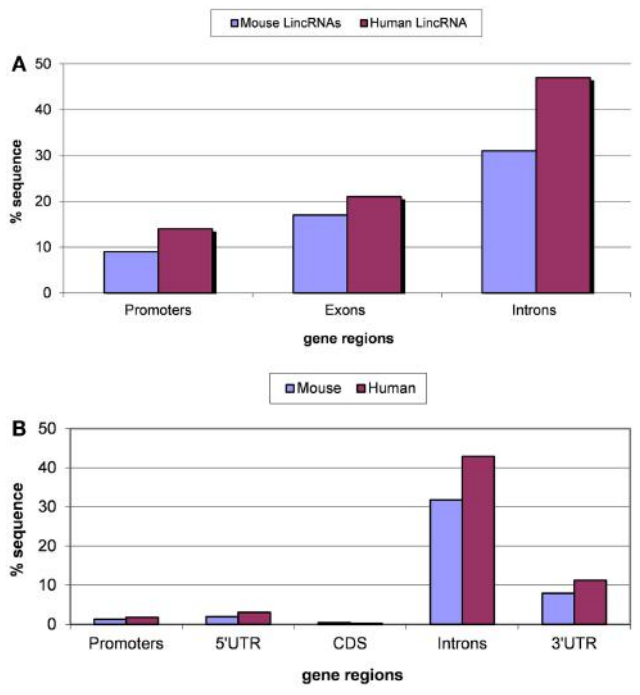


FIGURE 1 | Fractions (proportions) of lincRNA gene regions (concatenated promoters, exons, and introns) (A) and protein-coding gene regions (B) occupied by TE-derived sequences. The differences for pairwise comparisons "promoters vs. introns" and "exons vs. introns" are statistically significant for both classes of genes ($P < 10^{-5}$ according to the Fisher exact test; the raw counts of nucleotides in TES vs. the raw counts of nucleotides in TE-free regions was used as the input for 2 × 2 contingency tables).

Different Classes of Transposable Elements in lincRNA Genes

Analysis of different classes of TEs indicates that the fractions of each class are similar for introns of lincRNA and protein-coding

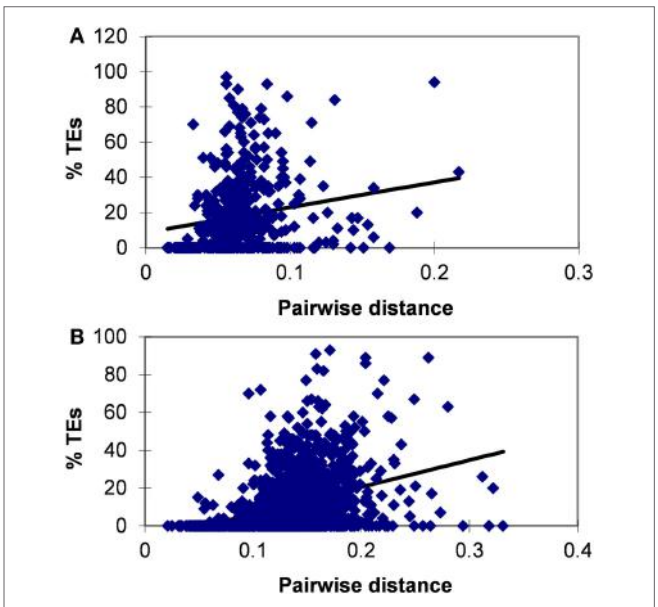


FIGURE 2 | Correlation between the fraction (proportion) of TEs in concatenated exons and evolutionary rate for human (A) and mouse (B) lincRNAs. Pearson correlation coefficient is 0.183 for human and 0.337 for mouse ($P < 10^{-5}$ for both comparisons).

genes and whole genomes (**Figures 3** and **4**). In each case, the fraction of LINEs is substantially greater than those of SINEs and LTR elements (**Figures 3A** and **4A**). However, there is a significant suppression of LINEs in exonic and promoter regions, in both human and mouse (**Figures 3A** and **4A**). This effect cannot be explained by fluctuations of the base composition in different gene regions because there are no significant compositional differences between exons, introns, and promoter regions for human and mouse lincRNA genes (results not shown). The same trend was observed for different lincRNA sets (Kapusta et al., 2013) suggesting that re-distribution of TEs is a general property of mammalian lincRNA genes. Furthermore, similar tendency is observed in promoter sequences of protein-coding genes (**Figures 3B** and **4B**), the overall lower abundance of TEs notwithstanding. Conceivably, when the smaller SINEs are inserted into functionally important parts of genes, they typically exert a milder deleterious effect than the larger LINEs and LTR elements and accordingly, are more often fixed in the course of evolution.

Higher Frequency of Ancient Transposable Element-Derived Sequences in Promoters and Exons Compared to Introns

Evolutionary conservation of TEs is likely to reflect molecular domestication of the respective elements (Jordan et al., 2003; Feschotte, 2008; Jurka, 2008; Bourque, 2009; Sinzelle et al., 2009). We analyzed the fraction of ancient mobile elements in different regions of lincRNA genes (**Figure 5**). A significantly higher abundance of ancient TEs ($P < 10^{-5}$ according to the Fisher exact test) was detected in exons and especially in promoter regions compared to introns (**Figure 5**). This finding is consistent with the hypothesis that TEs, in some cases, may perform novel functions in the host organisms (Makalowski, 2000; Jordan et al., 2003). The excess of ancient TEs was more pronounced in human compared to mouse lincRNA genes (**Figure 5**), possibly reflecting differences in evolutionary processes in rodents and primates although a bias caused by technical problems with the detection of 5′-ends of human lincRNA sequences cannot be ruled out (Kutter et al., 2012). We searched the putative promoter regions of lincRNA genes for the presence of TATA boxes and found a substantially elevated frequency of TATA-like sequences in the region −25 to −35 (Figure S3 in Supplementary Material). Given that a similar distribution is observed in many well-annotated human protein-coding genes (Yang et al., 2007), these observations suggest acceptable accuracy of 5′-end identification in lincRNA genes. The fractions of TATA-containing promoters are similar for protein-coding genes (10–25%) (Yang et al., 2007; Anish et al., 2009) and the analyzed sets of lincRNA genes (19–30%; Table S1 in Supplementary Material). The higher frequency of ancient TEs in promoters and exons compared to introns (**Figure 5**; Table S2 in Supplementary Material) suggests that many lincRNA genes emerged before the split of primates and rodents, and that TEs contributed to the origin of these ancient lincRNAs. This finding is consistent with recent observations that 60–70% of the lincRNAs genes are conserved between human and mouse (Kutter et al., 2012; Managadze et al., 2013), and with the observed

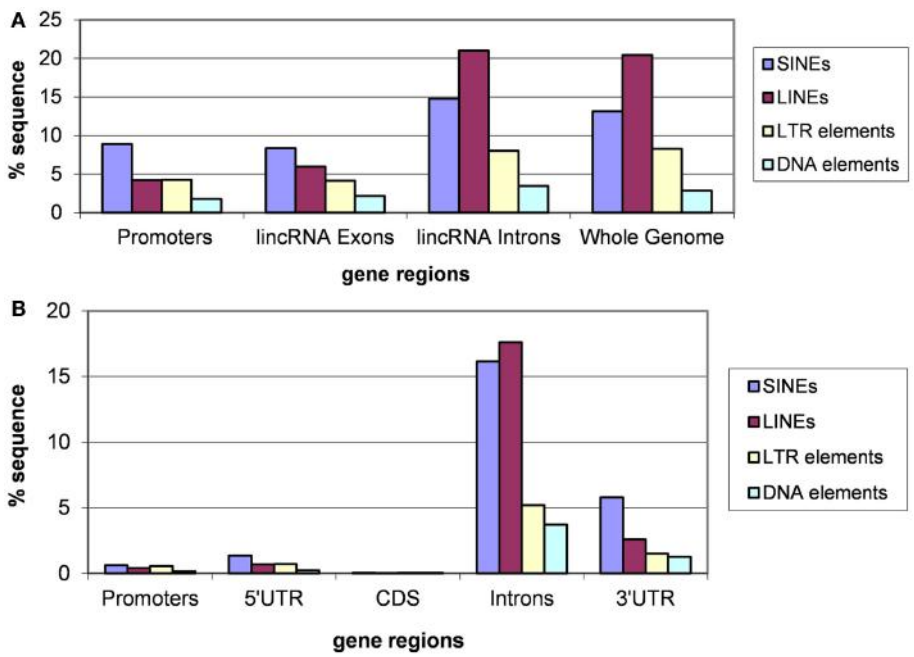


FIGURE 3 | Fractions (proportions) of human lincRNA gene regions (concatenated promoters, exons, and introns) and the whole genome sequence (A) and protein-coding gene regions (B) occupied by sequences derived from different types of TEs. Differences for pairwise comparisons "SINEs vs. LINEs" and "LTRs vs. LINEs" are statistically significant for both classes of genes ($P < 10^{-5}$ according to the Fisher exact test; the raw counts of nucleotides in TES vs. the raw counts of nucleotides in TE-free regions was used as the input for 2 × 2 contingency tables).

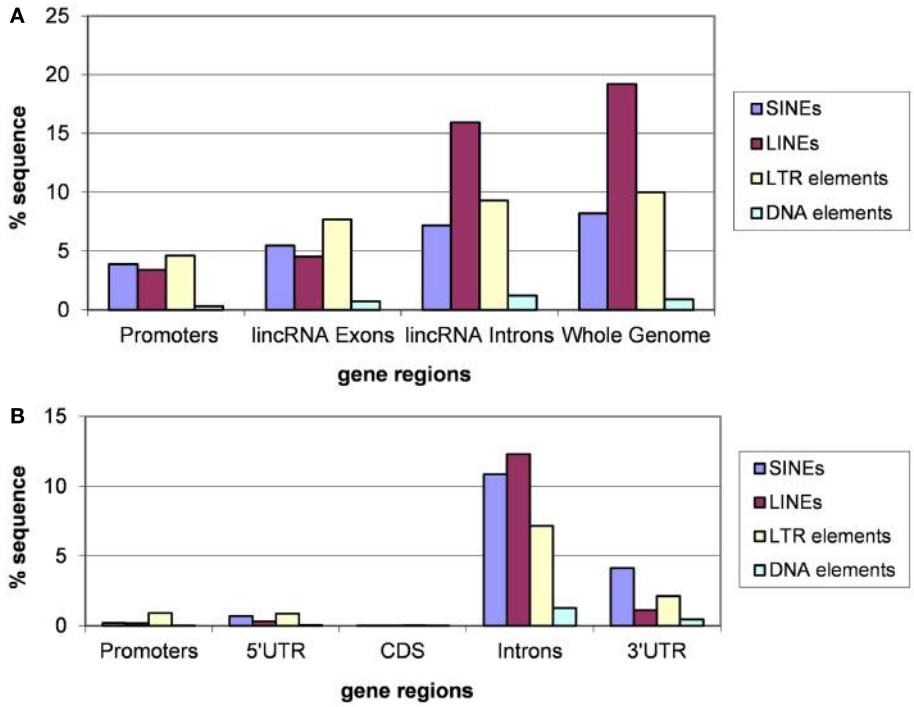


FIGURE 4 | Fractions (proportions) of mouse lincRNA gene regions (concatenated promoters, exons, and introns) and the whole genome sequence [the data for the whole genome sequence were from Waterston et al. (2002)] (A) and protein-coding gene regions (B) occupied by sequences derived from different types of TEs. Differences for pairwise comparisons "SINEs vs. LINEs" and "LTRs vs. LINEs" are statistically significant for both classes of genes ($P < 10^{-5}$ according to the Fisher exact test; the raw counts of nucleotides in TES vs. the raw counts of nucleotides in TE-free regions was used as the input for 2 × 2 contingency tables).

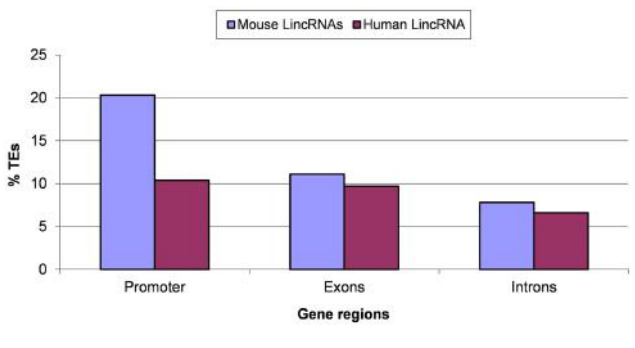


FIGURE 5 | Ancient transposable elements (TEs) in putative promoter regions, exons, and introns of lincRNA genes. A TE was considered ancient if the alignment between human–mouse orthologous TE sequences was longer than 100 bp and contained <5% insertions/deletions (the stringent threshold). "% TEs" stands for the fraction (proportion) of ancient TEs. Results for the relaxed threshold (the alignment between human–mouse orthologous TE sequences was longer than 100 bp and contained no more than 25% insertions/deletions) are shown in the Table S2 in Supplementary Material. Differences between pairwise comparisons "promoters vs. introns" and "exons vs. introns" are statistically significant ($P < 10^{-5}$ according to the Fisher exact test; the raw counts of ancient TES vs. the raw counts of lineage-specific TES was used as the input for 2 × 2 contingency tables).

higher conservation of lincRNA promoter regions compared to exons (Elisaphenko et al., 2008; Kapusta et al., 2013; Johnson and Guigo, 2014).

Discussion

The staggering evolutionary success of TEs in eukaryotes is often attributed to their ability to out replicate the host genomes in which they reside, as opposed to any selective advantage that they might provide to their hosts. Indeed, it has been shown that TEs can spread within and among genomes even in the face of a selective cost to the host (Hickey, 1982). Hence, the selfish DNA concept of TEs focuses on the parasitic nature of these elements and emphasizes the deleterious effects of transposition as well as the negligible evolutionary benefit that TEs provide to their hosts (Orgel et al., 1980; Gould and Vrba, 1982). However, the sheer abundance of TEs in the genome, as well as the variety of mutation effects induced by their mobility, suggest that they might, in some cases, be exapted (Gould and Vrba, 1982) or domesticated (Miller et al., 1999), to serve the evolutionary interests of the host (Makalowski, 2000; Jordan et al., 2003). Indeed, multiple lines of evidence indicate that the presence of TEs can result in host adaptation by shaping and reshaping the genome in many different ways (Smit, 1996; Makalowski, 2000; Rogozin et al., 2000; Kidwell and Lisch, 2001; Deininger and Batzer, 2002; Jordan et al., 2003).

The TEs comprise at least half of the mammalian genomes, and in particular, are found in most lincRNAs [this study and Kelley and Rinn (2012)]. Here, we demonstrate that TEs substantially contribute to the evolution of lincRNAs and their promoter regions. Although the densities of TES in these regions are much lower than those in introns, ostensibly, due to the purifying selection that

affects functional regions, the contributions of TEs to the evolution of these regions is substantially greater than in the respective regions of protein-coding genes. The higher density of TES in the exons of lincRNAs compared to protein-coding exons appears to reflect the much lower level of functional constraint characteristic of the former (Ponjavic et al., 2007; Managadze et al., 2011). The promoters of lincRNA appear to similarly enjoy greater plasticity and flexibility compared to the promoters of protein-coding genes.

Thus, TE insertion is an important factor that affects lincRNA evolution and biological function. An analysis of TEs in human lincRNAs revealed that the TES composition in lincRNA genes significantly differs from genomic averages: LINEs and SINEs are depleted whereas LTR retrotransposons are enriched (Kelley and Rinn, 2012). The TES occur in biased positions and orientations at lincRNA transcription start sites suggesting a functional role in lincRNA transcriptional regulation (Kelley and Rinn, 2012). In many cases, lincRNAs devoid of TES are expressed at higher levels than lincRNAs containing TES in all tested tissues and cell lines (Kelley and Rinn, 2012). Thus, it has been suggested that TES divide lincRNAs into classes and have contributed to lincRNA evolution and function by conferring tissue-specific expression from extant transcriptional regulatory signals (Kelley and Rinn, 2012). Here, we add another facet to these observations by showing that the promoter regions of lincRNAs are specifically enriched for ancient TES. This finding indicates that not only have many lincRNA genes evolved before the radiation of primates and rodents but also that at least some features of their regulation were already established at that time through TE insertion.

The possibility that some lincRNA genes encode short peptides that are translated, perhaps in a tissue-specific manner, is the subject of an ongoing debate (Brosius and Tiedge, 2004; Mattick and Makunin, 2006; Dinger et al., 2008; Makalowska et al., 2010; Carvunis et al., 2012; Chew et al., 2013). It is extremely hard to rule out such a role for a fraction of purported lincRNAs. A recent peptidomics study demonstrated that most annotated lincRNAs do not generate stable protein (peptide) products (Banfai et al., 2012). Furthermore, ribosomal profiling of lincRNAs suggests that ribosomal engagement with lincRNAs is likely to be regulatory (Chew et al., 2013). The presence of ORFs in the analyzed lincRNA data sets had been analyzed before using different approaches (Managadze et al., 2011, 2013). Importantly, removal of ORF-containing lincRNAs did not affect the conclusions of both studies (Managadze et al., 2011, 2013). The much higher abundance of TES in lincRNA compared to 5′UTR and protein-coding regions of mRNAs is consistent with the low frequency of protein-coding regions in the analyzed data sets.

It has been proposed that lncRNAs are organized into combinations of discrete functional domains, but the nature of these domains and their identification remain elusive (Guttman and Rinn, 2012). Insertion of TEs and exaptation of TES could represent an important route of evolution of the domain structure of lncRNAs. More specifically, Johnson and Guigo (2014) have proposed that exonic TES comprise functional domains of lncRNAs that they dubbed repeat insertion domains of LncRNAs (RIDLs). A growing number of RIDLs have been experimentally identified whereby lncRNA TES function as RNA-, DNA-, and protein-binding domains/motifs (Elisaphenko et al., 2008; Kelley and Rinn, 2012; Grote and Herrmann, 2013; Holdt et al., 2013; Johnson and Guigo, 2014). These examples are likely to reflect a more general phenomenon of exaptation and/or domestication during lncRNA evolution whereby TES are employed as DNA-, RNA-, and protein-binding domain/motifs (Johnson and Guigo, 2014). The RIDL hypothesis has the potential to explain how functional evolution can keep pace with the fast evolution observed in many lncRNA genes (Johnson and Guigo, 2014). The findings on the distribution of TES across different regions of lincRNA genes, the higher occurrence of TES in lincRNA promoters and exons compared to introns, and significant correlations between the content of TES and evolutionary rate presented here appear to be compatible with the RIDL hypothesis. More specifically, even if a substantial fraction of TES are not fixed in lincRNA exons and promoter regions, those TES that are fixed tend to persist in the genome longer than intronic TES. Moreover, given the near ubiquity of recognizable TES in lincRNA genes, TE mapping can be a useful approach for characterization of lincRNAs and possibly even prediction of their functions. The correlations between the content of TES and various features of lincRNA genes described here could be useful for the characterization of lincRNA functions.

Conclusion

The results of the present analysis, along with several previous studies, indicate that TEs have contributed to the evolution of many if not most mammalian lincRNAs. Whereas the density of TES in the introns of lincRNA genes is about the same as in introns of protein-coding genes exons, and promoters of lincRNAs are markedly enriched in TES compared to the counterparts in protein-coding genes. This high prevalence of TES reflects the relatively weak evolutionary constraints on lincRNA genes and itself appears to contribute to the plasticity and functional diversification of lincRNAs. Furthermore, the distribution of TE types in the functional regions of lincRNA genes significantly differs from that in introns (or whole genomes), conceivably, because the smaller SINEs that encode no proteins are more suitable for exaptation than the larger, protein-coding LINEs. The prodigious exaptation of TE could account, at least in part, for the functionality of many lincRNAs despite their rapid evolution.

Acknowledgments

We thank Joshua Cherry, Jean Thierri-Mieg, Kira Makarova, and Yuri Wolf for useful discussions. *Funding*: This research was supported in part by the Intramural Research Program of the National Library of Medicine at the National Institutes of Health and the National Institutes of Health Intramural Research Program of the National Eye Institute (US Department of Health and Human Services), Italian Ministry of Education, University and Research (MIUR) Flagship "InterOmics" (PB05), HIRMA (RBAP11YS7K) and by the "MIMOMICS" European projects.

Supplementary Material

The Supplementary Material for this article can be found online at http://www.frontiersin.org/article/10.3389/fbioe.2015.00071/abstract

References

Amaral, P. P., Dinger, M. E., and Mattick, J. S. (2013). Non-coding RNAs in homeostasis, disease and stress responses: an evolutionary perspective. *Brief. Funct. Genomics* **12**, 254–278. doi:10.1093/bfgp/elt016

Anish, R., Hossain, M. B., Jacobson, R. H., and Takada, S. (2009). Characterization of transcription from TATA-less promoters: identification of a new core promoter element XCPE2 and analysis of factor requirements. *PLoS ONE* **4**:e5103. doi:10.1371/journal.pone.0005103

Banfai, B., Jia, H., Khatun, J., Wood, E., Risk, B., Gundling, W. E. Jr., et al. (2012). Long noncoding RNAs are rarely translated in two human cell lines. *Genome Res.* **22**, 1646–1657. doi:10.1101/gr.134767.111

Beltran, M., Puig, I., Pena, C., Garcia, J. M., Alvarez, A. B., Pena, R., et al. (2008). A natural antisense transcript regulates Zeb2/Sip1 gene expression during Snail1-induced epithelial-mesenchymal transition. *Genes Dev.* **22**, 756–769. doi:10.1101/gad.455708

Bertone, P., Stolc, V., Royce, T. E., Rozowsky, J. S., Urban, A. E., Zhu, X., et al. (2004). Global identification of human transcribed sequences with genome tiling arrays. *Science* **306**, 2242–2246. doi:10.1126/science.1103388

Boccaccio, C., Deschatrette, J., and Meunier-Rotival, M. (1990). Empty and occupied insertion site of the truncated LINE-1 repeat located in the mouse serum albumin-encoding gene. *Gene* **88**, 181–186. doi:10.1016/0378-1119(90)90030-U

Bourque, G. (2009). Transposable elements in gene regulation and in the evolution of vertebrate genomes. *Curr. Opin. Genet. Dev.* **19**, 607–612. doi:10.1016/j.gde.2009.10.013

Bourque, G., Leong, B., Vega, V. B., Chen, X., Lee, Y. L., Srinivasan, K. G., et al. (2008). Evolution of the mammalian transcription factor binding repertoire via transposable elements. *Genome Res.* **18**, 1752–1762. doi:10.1101/gr.080663.108

Brockdorff, N., Ashworth, A., Kay, G. F., McCabe, V. M., Norris, D. P., Cooper, P. J., et al. (1992). The product of the mouse Xist gene is a 15 kb inactive X-specific transcript containing no conserved ORF and located in the nucleus. *Cell* **71**, 515–526. doi:10.1016/0092-8674(92)90519-I

Brosius, J. (1999). Genomes were forged by massive bombardments with retroelements and retrosequences. *Genetica* **107**, 209–238. doi:10.1023/A:1004018519722

Brosius, J., and Tiedge, H. (2004). RNomenclature. *RNA Biol.* **1**, 81–83. doi:10.4161/rna.1.2.1228

Carninci, P., Kasukawa, T., Katayama, S., Gough, J., Frith, M. C., Maeda, N., et al. (2005). The transcriptional landscape of the mammalian genome. *Science* **309**, 1559–1563. doi:10.1126/science.1112014

Carvunis, A. R., Rolland, T., Wapinski, I., Calderwood, M. A., Yildirim, M. A., Simonis, N., et al. (2012). Proto-genes and de novo gene birth. *Nature* **487**, 370–374. doi:10.1038/nature11184

Centonze, D., Rossi, S., Napoli, I., Mercaldo, V., Lacoux, C., Ferrari, F., et al. (2007). The brain cytoplasmic RNA BC1 regulates dopamine D2 receptor-mediated transmission in the striatum. *J. Neurosci.* **27**, 8885–8892. doi:10.1523/JNEUROSCI.0548-07.2007

Chen, J., Sun, M., Hurst, L. D., Carmichael, G. G., and Rowley, J. D. (2005). Human antisense genes have unusually short introns: evidence for selection for rapid transcription. *Trends Genet.* **21**, 203–207. doi:10.1016/j.tig.2005.02.003

Chew, G. L., Pauli, A., Rinn, J. L., Regev, A., Schier, A. F., and Valen, E. (2013). Ribosome profiling reveals resemblance between long non-coding RNAs and 5' leaders of coding RNAs. *Development* **140**, 2828–2834. doi:10.1242/dev.098343

Deininger, P. L., and Batzer, M. A. (2002). Mammalian retroelements. *Genome Res.* **12**, 1455–1465. doi:10.1101/gr.282402

Derrien, T., Johnson, R., Bussotti, G., Tanzer, A., Djebali, S., Tilgner, H., et al. (2012). The GENCODE v7 catalog of human long noncoding RNAs: analysis of their gene structure, evolution, and expression. *Genome Res.* **22**, 1775–1789. doi:10.1101/gr.132159.111

Dinger, M. E., Pang, K. C., Mercer, T. R., Crowe, M. L., Grimmond, S. M., and Mattick, J. S. (2009). NRED: a database of long noncoding RNA expression. *Nucleic Acids Res.* **37**, D122–D126. doi:10.1093/nar/gkn617

Dinger, M. E., Pang, K. C., Mercer, T. R., and Mattick, J. S. (2008). Differentiating protein-coding and noncoding RNA: challenges and ambiguities. *PLoS Comput. Biol.* **4**:e1000176. doi:10.1371/journal.pcbi.1000176

Drummond, D. A., and Wilke, C. O. (2008). Mistranslation-induced protein misfolding as a dominant constraint on coding-sequence evolution. *Cell* **134**, 341–352. doi:10.1016/j.cell.2008.05.042

Duret, L., Chureau, C., Samain, S., Weissenbach, J., and Avner, P. (2006). The Xist RNA gene evolved in eutherians by pseudogenization of a protein-coding gene. *Science* **312**, 1653–1655. doi:10.1126/science.1126316

Duret, L., and Mouchiroud, D. (2000). Determinants of substitution rates in mammalian genes: expression pattern affects selection intensity but not mutation rate. *Mol. Biol. Evol.* **17**, 68–74. doi:10.1093/oxfordjournals.molbev.a026239

Elisaphenko, E. A., Kolesnikov, N. N., Shevchenko, A. I., Rogozin, I. B., Nesterova, T. B., Brockdorff, N., et al. (2008). A dual origin of the Xist gene from a protein-coding gene and a set of transposable elements. *PLoS ONE* **3**:e2521. doi:10.1371/journal.pone.0002521

Espinoza, C. A., Goodrich, J. A., and Kugel, J. F. (2007). Characterization of the structure, function, and mechanism of B2 RNA, an ncRNA repressor of RNA polymerase II transcription. *RNA* **13**, 583–596. doi:10.1261/rna.310307

Faulkner, G. J., Kimura, Y., Daub, C. O., Wani, S., Plessy, C., Irvine, K. M., et al. (2009). The regulated retrotransposon transcriptome of mammalian cells. *Nat. Genet.* **41**, 563–571. doi:10.1038/ng.368

Felsenstein, J. (1996). Inferring phylogenies from protein sequences by parsimony, distance, and likelihood methods. *Meth. Enzymol.* **266**, 418–427. doi:10.1016/S0076-6879(96)66026-1

Feng, J., Bi, C., Clark, B. S., Mady, R., Shah, P., and Kohtz, J. D. (2006). The Evf-2 noncoding RNA is transcribed from the Dlx-5/6 ultraconserved region and functions as a Dlx-2 transcriptional coactivator. *Genes Dev.* **20**, 1470–1484. doi:10.1101/gad.1416106

Feschotte, C. (2008). Transposable elements and the evolution of regulatory networks. *Nat. Rev. Genet.* **9**, 397–405. doi:10.1038/nrg2337

Glazko, G. V., Zybailov, B. L., and Rogozin, I. B. (2012). Computational prediction of polycomb-associated long non-coding RNAs. *PLoS ONE* **7**:e44878. doi:10.1371/journal.pone.0044878

Goecks, J., Nekrutenko, A., and Taylor, J. (2010). Galaxy: a comprehensive approach for supporting accessible, reproducible, and transparent computational research in the life sciences. *Genome Biol.* **11**, R86. doi:10.1186/gb-2010-11-8-r86

Goodrich, J. A., and Kugel, J. F. (2006). Non-coding-RNA regulators of RNA polymerase II transcription. *Nat. Rev. Mol. Cell Biol.* **7**, 612–616. doi:10.1038/nrm1946

Gould, S. J., and Vrba, S. (1982). Exaptation – a missing term in the science of form. *Paleobiology* **8**, 4–14.

Graur, D., Zheng, Y., Price, N., Azevedo, R. B., Zufall, R. A., and Elhaik, E. (2013). On the immortality of television sets: "function" in the human genome according to the evolution-free gospel of ENCODE. *Genome Biol. Evol.* **5**, 578–590. doi:10.1093/gbe/evt028

Grote, P., and Herrmann, B. G. (2013). The long non-coding RNA Fendrr links epigenetic control mechanisms to gene regulatory networks in mammalian embryogenesis. *RNA Biol.* **10**, 1579–1585. doi:10.4161/rna.26165

Guttman, M., and Rinn, J. L. (2012). Modular regulatory principles of large non-coding RNAs. *Nature* **482**, 339–346. doi:10.1038/nature10887

Hickey, D. A. (1982). Selfish DNA: a sexual*ly-transmitted* nuclear parasite. *Genetics* **101**, 519–531.

Hirota, K., Miyoshi, T., Kugou, K., Hoffman, C. S., Shibata, T., and Ohta, K. (2008). Stepwise chromatin remodelling by a cascade of transcription initiation of non-coding RNAs. *Nature* **456**, 130–134. doi:10.1038/nature07348

Holdt, L. M., Hoffmann, S., Sass, K., Langenberger, D., Scholz, M., Krohn, K., et al. (2013). Alu elements in ANRIL non-coding RNA at chromosome 9p21 modulate atherogenic cell functions through trans-regulation of gene networks. *PLoS Genet.* **9**:e1003588. doi:10.1371/journal.pgen.1003588

Johnson, R., and Guigo, R. (2014). The RIDL hypothesis: transposable elements as functional domains of long noncoding RNAs. *RNA* **20**, 959–976. doi:10.1261/rna.044560.114

Jordan, I. K., Rogozin, I. B., Glazko, G. V., and Koonin, E. V. (2003). Origin of a substantial fraction of human regulatory sequences from transposable elements. *Trends Genet.* **19**, 68–72. doi:10.1016/S0168-9525(02)00006-9

Jurka, J. (2008). Conserved eukaryotic transposable elements and the evolution of gene regulation. *Cell. Mol. Life Sci.* **65**, 201–204. doi:10.1007/s00018-007-7369-3

Kapitonov, V. V., and Jurka, J. (2003). A novel class of SINE elements derived from 5S rRNA. *Mol. Biol. Evol.* **20**, 694–702. doi:10.1093/molbev/msg075

Kapranov, P., Cheng, J., Dike, S., Nix, D. A., Duttagupta, R., Willingham, A. T., et al. (2007). RNA maps reveal new RNA classes and a possible function for pervasive transcription. *Science* **316**, 1484–1488. doi:10.1126/science.1138341

Kapusta, A., Kronenberg, Z., Lynch, V. J., Zhuo, X., Ramsay, L., Bourque, G., et al. (2013). Transposable elements are major contributors to the origin, diversification, and regulation of vertebrate long noncoding RNAs. *PLoS Genet.* **9**:e1003470. doi:10.1371/journal.pgen.1003470

Karolchik, D., Hinrichs, A. S., Furey, T. S., Roskin, K. M., Sugnet, C. W., Haussler, D., et al. (2004). The UCSC table browser data retrieval tool. *Nucleic Acids Res.* **32**, D493–D496. doi:10.1093/nar/gkh103

Kazazian, H. H. Jr., Wong, C., Youssoufian, H., Scott, A. F., Phillips, D. G., and Antonarakis, S. E. (1988). Haemophilia A resulting from de novo insertion of L1 sequences represents a novel mechanism for mutation in man. *Nature* **332**, 164–166. doi:10.1038/332164a0

Kelley, D., and Rinn, J. (2012). Transposable elements reveal a stem cell-specific class of long noncoding RNAs. *Genome Biol.* **13**, R107. doi:10.1186/gb-2012-13-11-r107

Kidwell, M. G., and Lisch, D. R. (2001). Perspective: transposable elements, parasitic DNA, and genome evolution. *Evolution* **55**, 1–24. doi:10.1554/0014-3820(2001)055[0001:PTEPDA]2.0.CO;2

Kolesnikov, N. N., and Elisafenko, E. A. (2010). Comparative organization and the origin of noncoding regulatory RNA genes from X-chromosome inactivation center of human and mouse. *Genetika* **46**, 1386–1391. doi:10.1134/S1022795410100200

Krylov, D. M., Wolf, Y. I., Rogozin, I. B., and Koonin, E. V. (2003). Gene loss, protein sequence divergence, gene dispensability, expression level, and interactivity are correlated in eukaryotic evolution. *Genome Res.* **13**, 2229–2235. doi:10.1101/gr.1589103

Kutter, C., Watt, S., Stefflova, K., Wilson, M. D., Goncalves, A., Ponting, C. P., et al. (2012). Rapid turnover of long noncoding RNAs and the evolution of gene expression. *PLoS Genet.* **8**:e1002841. doi:10.1371/journal.pgen.1002841

Loeb, D. D., Padgett, R. W., Hardies, S. C., Shehee, W. R., Comer, M. B., Edgell, M. H., et al. (1986). The sequence of a large L1Md element reveals a tandemly repeated 5' end and several features found in retrotransposons. *Mol. Cell. Biol.* **6**, 168–182.

Loewer, S., Cabili, M. N., Guttman, M., Loh, Y. H., Thomas, K., Park, I. H., et al. (2010). Large intergenic non-coding RNA-RoR modulates reprogramming of human induced pluripotent stem cells. *Nat. Genet.* **42**, 1113–1117. doi:10.1038/ng.710

Makalowska, I., Rogozin, I. B., and Makalowski, W. (2010). Genome evolution. *Adv. Bioinformatics* **2010**, 643701. doi:10.1155/2010/643701

Makalowski, W. (2000). Genomic scrap yard: how genomes utilize all that junk. *Gene* **259**, 61–67. doi:10.1016/S0378-1119(00)00436-4

Managadze, D., Lobkovsky, A. E., Wolf, Y. I., Shabalina, S. A., Rogozin, I. B., and Koonin, E. V. (2013). The vast, conserved mammalian lincRNome. *PLoS Comput. Biol.* **9**:e1002917. doi:10.1371/journal.pcbi.1002917

Managadze, D., Rogozin, I. B., Chernikova, D., Shabalina, S. A., and Koonin, E. V. (2011). Negative correlation between expression level and evolutionary rate of long intergenic noncoding RNAs. *Genome Biol. Evol.* **3**, 1390–1404. doi:10.1093/gbe/evr116

Mariner, P. D., Walters, R. D., Espinoza, C. A., Drullinger, L. F., Wagner, S. D., Kugel, J. F., et al. (2008). Human Alu RNA is a modular transacting repressor of mRNA transcription during heat shock. *Mol. Cell* **29**, 499–509. doi:10.1016/j.molcel.2007.12.013

Martens, J. A., Laprade, L., and Winston, F. (2004). Intergenic transcription is required to repress the *Saccharomyces cerevisiae* SER3 gene. *Nature* **429**, 571–574. doi:10.1038/nature02538

Martianov, I., Ramadass, A., Serra Barros, A., Chow, N., and Akoulitchev, A. (2007). Repression of the human dihydrofolate reductase gene by a non-coding interfering transcript. *Nature* **445**, 666–670. doi:10.1038/nature05519

Martin, S. L. (2006). The ORF1 protein encoded by LINE-1: structure and function during L1 retrotransposition. *J. Biomed. Biotechnol.* **2006**, 45621. doi:10.1155/JBB/2006/45621

Mattick, J. S., and Makunin, I. V. (2006). Non-coding RNA. *Hum. Mol. Genet.* **15**, R17–R29. doi:10.1093/hmg/ddl046

Mattick, J. S., Taft, R. J., and Faulkner, G. J. (2010). A global view of genomic information – moving beyond the gene and the master regulator. *Trends Genet.* **26**, 21–28. doi:10.1016/j.tig.2009.11.002

Mercer, T. R., Dinger, M. E., Sunkin, S. M., Mehler, M. F., and Mattick, J. S. (2008). Specific expression of long noncoding RNAs in the mouse brain. *Proc. Natl. Acad. Sci. U.S.A.* **105**, 716–721. doi:10.1073/pnas.0706729105

Miller, W. J., McDonald, J. F., Nouaud, D., and Anxolabehere, D. (1999). Molecular domestication – more than a sporadic episode in evolution. *Genetica* **107**, 197–207. doi:10.1023/A:1004070603792

Munroe, S. H., and Lazar, M. A. (1991). Inhibition of c-erbA mRNA splicing by a naturally occurring antisense RNA. *J. Biol. Chem.* **266**, 22083–22086.

Nagano, T., Mitchell, J. A., Sanz, L. A., Pauler, F. M., Ferguson-Smith, A. C., Feil, R., et al. (2008). The air noncoding RNA epigenetically silences transcription by targeting G9a to chromatin. *Science* **322**, 1717–1720. doi:10.1126/science.1163802

Ng, S. Y., Lin, L., Soh, B. S., and Stanton, L. W. (2013). Long noncoding RNAs in development and disease of the central nervous system. *Trends Genet.* **29**, 461–468. doi:10.1016/j.tig.2013.03.002

Okazaki, Y., Furuno, M., Kasukawa, T., Adachi, J., Bono, H., Kondo, S., et al. (2002). Analysis of the mouse transcriptome based on functional annotation of 60,770 full-length cDNAs. *Nature* **420**, 563–573. doi:10.1038/nature01266

Orgel, L. E., Crick, F. H., and Sapienza, C. (1980). Selfish DNA. *Nature* **288**, 645–646. doi:10.1038/288645a0

Osato, N., Suzuki, Y., Ikeo, K., and Gojobori, T. (2007). Transcriptional interferences in cis natural antisense transcripts of humans and mice. *Genetics* **176**, 1299–1306. doi:10.1534/genetics.106.069484

Pandey, R. R., Mondal, T., Mohammad, F., Enroth, S., Redrup, L., Komorowski, J., et al. (2008). Kcnq1ot1 antisense noncoding RNA mediates lineage-specific transcriptional silencing through chromatin-level regulation. *Mol. Cell* **32**, 232–246. doi:10.1016/j.molcel.2008.08.022

Pang, K. C., Frith, M. C., and Mattick, J. S. (2006). Rapid evolution of noncoding RNAs: lack of conservation does not mean lack of function. *Trends Genet.* **22**, 1–5. doi:10.1016/j.tig.2005.10.003

Ponjavic, J., Ponting, C. P., and Lunter, G. (2007). Functionality or transcriptional noise? Evidence for selection within long noncoding RNAs. *Genome Res.* **17**, 556–565. doi:10.1101/gr.6036807

Ponting, C. P., and Belgard, T. G. (2010). Transcribed dark matter: meaning or myth? *Hum. Mol. Genet.* **19**, R162–R168. doi:10.1093/hmg/ddq362

Ponting, C. P., Oliver, P. L., and Reik, W. (2009). Evolution and functions of long noncoding RNAs. *Cell* **136**, 629–641. doi:10.1016/j.cell.2009.02.006

Rinn, J. L., and Chang, H. Y. (2012). Genome regulation by long noncoding RNAs. *Annu. Rev. Biochem.* **81**, 145–166. doi:10.1146/annurev-biochem-051410-092902

Robinson, R. (2010). Dark matter transcripts: sound and fury, signifying nothing? *PLoS Biol.* **8**:e1000370. doi:10.1371/journal.pbio.1000370

Rogozin, I. B., Mayorov, V. I., Lavrentieva, M. V., Milanesi, L., and Adkison, L. R. (2000). Prediction and phylogenetic analysis of mammalian short interspersed elements (SINEs). *Brief. Bioinformatics* **1**, 260–274. doi:10.1093/bib/1.3.260

Schuler, A., Ghanbarian, A. T., and Hurst, L. D. (2014). Purifying selection on splice-related motifs, not expression level nor RNA folding, explains nearly all constraint on human lincRNAs. *Mol. Biol. Evol.* **31**, 3164–3183. doi:10.1093/molbev/msu249

Sinzelle, L., Izsvak, Z., and Ivics, Z. (2009). Molecular domestication of transposable elements: from detrimental parasites to useful host genes. *Cell. Mol. Life Sci.* **66**, 1073–1093. doi:10.1007/s00018-009-8376-3

Smit, A. F. (1996). The origin of interspersed repeats in the human genome. *Curr. Opin. Genet. Dev.* **6**, 743–748. doi:10.1016/S0959-437X(96)80030-X

Temin, H. M. (1985). Reverse transcription in the eukaryotic genome: retroviruses, pararetroviruses, retrotransposons, and retrotranscripts. *Mol. Biol. Evol.* **2**, 455–468.

Ulitsky, I., Shkumatava, A., Jan, C. H., Sive, H., and Bartel, D. P. (2011). Conserved function of lincRNAs in vertebrate embryonic development despite rapid sequence evolution. *Cell* **147**, 1537–1550. doi:10.1016/j.cell.2011.11.055

Umlauf, D., Goto, Y., Cao, R., Cerqueira, F., Wagschal, A., Zhang, Y., et al. (2004). Imprinting along the Kcnq1 domain on mouse chromosome 7 involves repressive histone methylation and recruitment of polycomb group complexes. *Nat. Genet.* **36**, 1296–1300. doi:10.1038/ng1467

Van Bakel, H., and Hughes, T. R. (2009). Establishing legitimacy and function in the new transcriptome. *Brief. Funct. Genomic. Proteomic.* **8**, 424–436. doi:10.1093/bfgp/elp037

Wang, H., Iacoangeli, A., Lin, D., Williams, K., Denman, R. B., Hellen, C. U., et al. (2005). Dendritic BC1 RNA in translational control mechanisms. *J. Cell Biol.* **171**, 811–821. doi:10.1083/jcb.200506006

Wang, J., Gong, C., and Maquat, L. E. (2011). Control of myogenesis by rodent SINE-containing lncRNAs. *Genes Dev.* **27**, 793–804. doi:10.1101/gad.212639.112

Waterston, R. H., Lindblad-Toh, K., Birney, E., Rogers, J., Abril, J. F., Agarwal, P., et al. (2002). Initial sequencing and comparative analysis of the mouse genome. *Nature* **420**, 520–562. doi:10.1038/nature01262

Yang, C., Bolotin, E., Jiang, T., Sladek, F. M., and Martinez, E. (2007). Prevalence of the initiator over the TATA box in human and yeast genes and identification of DNA motifs enriched in human TATA-less core promoters. *Gene* **389**, 52–65. doi:10.1016/j.gene.2006.09.029

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in Bioengineering and Biotechnology

ORIGINAL RESEARCH
published: 08 May 2015
doi: 10.3389/fbioe.2015.00058

SPECTRA: an integrated knowledge base for comparing tissue and tumor-specific PPI networks in human

Giovanni Micale[1], *Alfredo Ferro*[2], *Alfredo Pulvirenti*[2*†] and *Rosalba Giugno*[2*†]

[1] Department of Computer Science, University of Pisa, Pisa, Italy, [2] Department of Clinical and Molecular Biomedicine, University of Catania, Catania, Italy

OPEN ACCESS

Edited by:
Marco Pellegrini,
Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Arsen Arakelyan,
Institute of Molecular Biology, Armenia
Mohammed El-Kebir,
Brown University, USA

***Correspondence:**
Alfredo Pulvirenti and Rosalba Giugno,
Department of Clinical and Molecular Biomedicine, University of Catania, Via Andrea Doria 6, Catania 95037, Italy
apulvirenti@dmi.unict.it; giugno@dmi.unict.it

[†]Alfredo Pulvirenti and Rosalba Giugno have contributed equally to this work.

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: 31 October 2014
Accepted: 17 April 2015
Published: 08 May 2015

Citation:
Micale G, Ferro A, Pulvirenti A and Giugno R (2015) SPECTRA: an integrated knowledge base for comparing tissue and tumor-specific PPI networks in human. Front. Bioeng. Biotechnol. 3:58. doi: 10.3389/fbioe.2015.00058

Protein–protein interaction (PPI) networks available in public repositories usually represent relationships between proteins within the cell. They ignore the specific set of tissues or tumors where the interactions take place. Indeed, proteins can form tissue-selective complexes, while they remain inactive in other tissues. For these reasons, a great attention has been recently paid to tissue-specific PPI networks, in which nodes are proteins of the global PPI network whose corresponding genes are preferentially expressed in specific tissues. In this paper, we present SPECTRA, a knowledge base to build and compare tissue or tumor-specific PPI networks. SPECTRA integrates gene expression and protein interaction data from the most authoritative online repositories. We also provide tools for visualizing and comparing such networks, in order to identify the expression and interaction changes of proteins across tissues, or between the normal and pathological states of the same tissue. SPECTRA is available as a web server at http://alpha.dmi.unict.it/spectra.

Keywords: tissue, tumor, database, proteins, interactions

1. Introduction

In the last 10 years, there has been a rapid growth of available protein–protein interaction (PPI) data. They represent all known physical interactions between proteins within a cell. The collection of PPI data yields a network depicting a global overview of the relationships between proteins.

Nowadays, PPIs of species and associated data are stored in many databases, which usually are weekly or monthly updated. Primary sources of PPI data include BioGRID (Stark et al., 2006), DIP (Xenarios et al., 2000), HPRD (Peri et al., 2004), IntAct teorchard2013mintact, and MINT (Licata et al., 2012).

DIP (Xenarios et al., 2000) was the first database, which combined information from multiple observations and experimental techniques into networks of interacting proteins for different species. HPRD (Peri et al., 2004) contains manually curated proteomic information regarding human proteins, which are annotated and linked to OMIM database (Hamosh et al., 2005). BioGRID (Stark et al., 2006) collects protein–protein and genetic interactions for all major model organisms trying to remove redundancy and create a single mapping of interactions. The IntAct database (Orchard et al., 2013) provides tools for both textual and graphical representations of protein interactions. Interacting proteins can be annotated with GO terms for functional analysis. MINT (Licata et al., 2012), which is based on the IntAct database

infrastructure, collects experimentally verified PPIs by extracting experimental evidences from the scientific literature.

Some databases integrate PPIs data of human and other organisms from primary sources, by removing redundancies and assigning a unique reliability score. These include STRING (Franceschini et al., 2013), IRefIndex (Razick et al., 2008), ConsensusPathDB (Kamburov et al., 2013), and HitPredict (Patil et al., 2011).

STRING (Franceschini et al., 2013) combines physical interaction data and curated pathways of different organisms with predicted interactions from text mining, genomic features and interactions transferred from model organisms based on orthology. IRefIndex (Razick et al., 2008) is a set of tools to index and retrieve proteins and interactions from major public databases. Indexes are built according to protein sequences and taxonomy identifiers and mapping scores evaluate the quality of the mapping. ConsensusPathDB (Kamburov et al., 2013) integrates human protein–protein interactions, biochemical pathways, gene regulatory, and drug–target interactions into a global network, containing genes, proteins, and metabolites, which can be visualized, analyzed, and annotated. HitPredict (Patil et al., 2011) combines PPI data from IntAct (Orchard et al., 2013), BIOGRID (Stark et al., 2006), and HPRD (Peri et al., 2004), by assigning a confidence score based on sequence, structure, and functional annotations of the interacting proteins. The reliability score is calculated using the Bayesian networks.

The analysis of PPI networks has provided novel biological insights on the function of many previously uncharacterized proteins in *Human* through module identification (Bader and Hogue, 2003; Adamcsek et al., 2006; Mete et al., 2008; Rhrissorrakrai and Gunsalus, 2011), network querying (Ferro et al., 2007; Banks et al., 2008; Bruckner et al., 2010), and network alignment (Flannick et al., 2006; Kalaev et al., 2009; Liao et al., 2009; Sahraeian and Yoon, 2013; Micale et al., 2014a) algorithms. Furthermore, the annotation of PPI networks with external data (i.e., diseases, expression data, phenotypes) has helped to classify genes according to the expression profiles (Dao et al., 2011), predict new gene–disease associations (Huang et al., 2012; Zhao et al., 2012), and discover new drugs (Huang et al., 2012; Alaimo et al., 2013; Csermerly et al., 2013).

These tasks have been accomplished thanks to the availability of authoritative repositories of gene expression data in normal/cancer tissues and at different diseases stages (Uhlen et al., 2010; Barrett et al., 2013; Rustici et al., 2013). For example, ArrayExpress (Rustici et al., 2013) and GEO (Barrett et al., 2013) include gene expression data from microarray and high-throughput sequencing experiments, which can be easily queried or downloaded. Users can also submit data directly by using the standard MIAME format. More recently, new projects have started with the aim of cataloging tissue or tumor sequencing data. The Cancer Genome Atlas (TCGA)[1] collects complete high-throughput genome data (clinical information, expressions data, methylations, mutations) for specific cancer tissues, with the purpose of helping the diagnosis and the treatment of cancers. The Human Protein Atlas (Uhlen et al., 2010) is a database with histological images showing the spatial distribution of proteins in normal and cancer tissues. Protein Atlas contains also transcription expression levels, protein expression profiles, and subcellular localization data.

All above PPI networks data are constructed by ignoring the role of proteins in human tissues. On the other hand, human diseases often occur in specific tissues (Lage et al., 2008). Some genes can be predominantly expressed in one or few tissues and can control the formation of protein complexes (Emig and Albrecht, 2011). Furthermore, genes can use alternative splicing as a powerful mechanism to enlarge the number of their interactors and perform distinct functions in different tissues (Emig and Albrecht, 2011). Therefore, the integration of PPI networks with tissue-specific gene expression data can help to highlight the role of some genes in specific disease or tumors. The result of such integration gives the so called Tissue-Specific PPI (TS-PPI) network (Bossi and Lehner, 2009), which is a subgraph of a PPI network where the genes corresponding to both interacting proteins are expressed in one or more selected tissues.

Some studies focus on the analysis of global and local properties of TS-PPI networks. In Bossi and Lehner (2009), authors prove that most housekeeping proteins form highly tissue-specific protein interactions, suggesting a key role of those proteins in tissue-specific biological processes. Emig and Albrecht (2011) show that the number of tissue-specific proteins is very low and the receptor-activated signaling processes and the transcriptional regulation are two key factors for tissue specificity. In Souiai et al. (2011), a gradient model is used to describe the structure of TS-PPI networks, containing interactions of regulatory and developmental functions at the core of the TS-PPI network and physiological functions at the periphery.

Several recent works highlight the advantages of using TS-PPI networks. In Lopes et al. (2011), a set of proteins related to the response of viral infection in a TS-PPI network lead to a more reliable functional enrichment. Magger et al. (2012) use TS-PPI networks to improve the prioritization of candidate disease-causing genes with respect to a generic PPI network. In Chen and Wang (2012), authors identify functional modules in TS-PPI networks using CFinder (Adamcsek et al., 2006) and show that they exhibit more biological meaning than modules in a PPI network. Xiao et al. (2014) propose a new method for the identification of multi-tissue gene co-expression networks associated with specific functional processes relevant for phenotype variation and disease in humans. Barshir et al. (2014) show that genes causing hereditary diseases tend to have higher transcript levels and more interacting partners in the TS-PPI network of disease tissues than in the TS-PPI network of unaffected tissues.

To the best of our knowledge, few tools are available for querying and analyzing TS-PPI networks (Barshir et al., 2013; Nersisyan et al., 2014). CyKeggParser (Nersisyan et al., 2014) is a Cytoscape app for generating and analyzing tissue-specific KEGG pathways. Pathways can be checked for inconsistencies and modified based on gene expression data from normal and cancer tissues. TissueNet (Barshir et al., 2013) is a dataset of TS-PPIs in humans, which integrates a collection of four PPInetworks (BioGRID, DIP, IntAct, and MINT) with three expression datasets (GEO, Human Protein Atlas, and Illumina Body Map 2.0). The database provides

[1]http://cancergenome.nih.gov

a web interface for retrieving tissue-specific interactions of a query protein. However, it handles only 16 normal tissues and does not provide any tool for the analyses of TS-PPI networks.

In this paper, we propose SPECTRA (SPECific Tissue/Tumor Related PPI networks Analyzer), a framework to build and analyze TS-PPI networks. SPECTRA integrates tissue and tumor-specific gene expression data from the most authoritative online repositories such as Protein Atlas, ArrayExpress, GEO, and TCGA. Expression data are then integrated with high-quality protein–protein interactions, taken from HPRD, BioGRID, MIPS, IntAct, and the work of Havugimana et al. (2012). We provide a web interface for constructing, visualizing, and comparing TS-PPI networks, with the aim of identifying differential interaction/expression patterns in TS-PPI networks (i.e., distinct tissues, or normal and pathological states of the same tissue). The TS-PPI networks together with the results of differential analysis can be easily visualized by using Cytoscape facilities (Shannon et al., 2003) and downloaded as text files for further investigations. SPECTRA is free for all users and available at http://alpha.dmi.unict.it/spectra.

2. Materials and Methods

SPECTRA combines protein–protein interactions in human with gene expressions, by integrating 13 authoritative resources. The final integrated SPECTRA database contains 16,435 protein coding genes and 175,841 gene interactions (GIs), 1,350,637 tissue-specific gene expression data entries covering 107 normal tissues, and 2,171,808 tumor-specific expression data entries covering 160 different tumors.

2.1. Interaction Datasets

Human protein interaction data were taken from BioGRID[2], DIP[3], a recent work by Havugimana et al. (2012), HPRD[4], IntAct[5], and MINT[6].

Table 1 describes the features of the PPI networks integrated in SPECTRA. Networks taken from the work of Havugimana et al. (2012), IntAct and MINT are weighted with edge weights ranging in [0,1], while the other PPI networks are unweighted. Proteins of the considered PPI networks, including splicing isoforms, were first mapped to the corresponding gene. Next, a global GI network was built, by collecting all interactions reported in at least one dataset. We assigned to each edge a pair consisting of the average value of weights across the datasets that report that interaction and the percentage of datasets giving the interaction (dataset coverage). Average edge weights range from 0.131 to 1.

Figure 1 depicts a Venn diagram of common gene interactions between PPI datasets. Interaction databases generally show low overlap, with only 25 interactions shared by all datasets and only 7,783 interactions in common between MINT, BioGRID, IntAct, and HPRD, which are the biggest ones. The final integrated network has 16,435 nodes, 175,841 edges and 17 connected components, with a high average diameter (9) and low clustering coefficient (0.289). The average degree is 21.398 and the degree distribution follows a power law (**Figure 2**).

[2]http://thebiogrid.org
[3]http://dip.doe-mbi.ucla.edu/dip
[4]http://www.hprd.org
[5]http://www.ebi.ac.uk/intact
[6]http://mint.bio.uniroma2.it/mint

TABLE 1 | Features of PPI networks integrated in SPECTRA.

Network	Nodes	Edges	Type
BioGRID	15,290	135,677	Unweighted
DIP	2,338	3,427	Unweighted
Havugimana et al. (2012)	3,003	13,989	Weighted
HPRD	9,506	37,054	Unweighted
IntAct	11,637	63,030	Weighted
MINT	6,551	18,478	Weighted

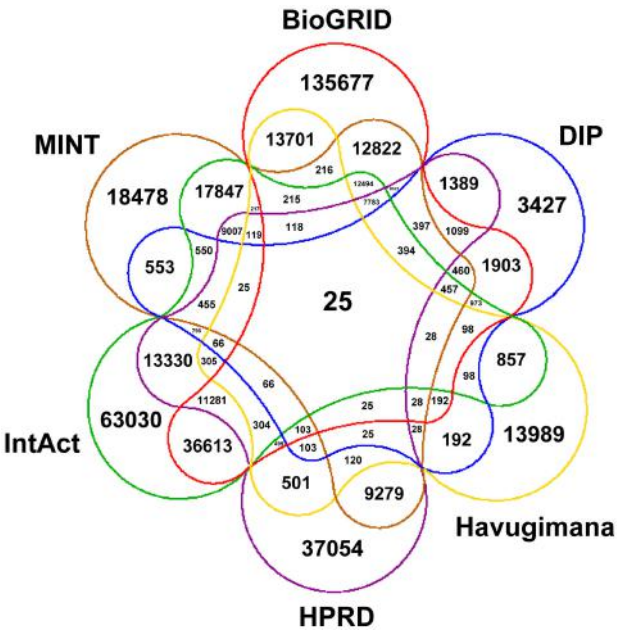


FIGURE 1 | Venn diagram showing the number of common interactions across PPI datasets in SPECTRA.

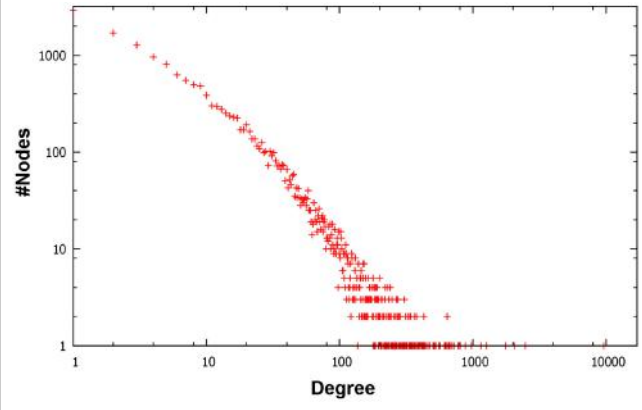


FIGURE 2 | Log–log plot of degree distribution of the final integrated PPI network in SPECTRA.

TABLE 2 | Features of expression datasets integrated in SPECTRA.

Dataset	Platform	Tissues	Tumors
E-MTAB-62 (Lukk et al., 2010)	GPL96	46	110
GDS181 (Su et al., 2002)	GPL91	29	6
GDS596 (Su et al., 2004)	GPL96	57	5
GDS1096 (Ge et al., 2005)	GPL96	36	0
GDS3113 (Dezso et al., 2008)	GPL2986	32	0
ProteinAtlas	GPL11154	28	33
TCGA	Agilent G4502A-07-3	0	27

2.2. Expression Datasets

Gene expression data for various tissues and tumors were downloaded from ArrayExpress[7], GEO[8], ProteinAtlas[9], and TCGA (see text footnote 1). **Table 2** lists the gene expression datasets integrated within SPECTRA, the platform used to detect the expressions and the number of covered tissues and tumors.

Figure 3 depicts a Venn diagram of common tissues and tumors across expression datasets. While tissue names are generally shared, tumor names are much differentiated, resulting in a poor overlap between datasets. In particular, TCGA contains data for very specific tumors and partially overlap only with E-MTAB-62 dataset, which is the richest one. Note that the numbers reported in **Figure 3** only refer to specific tumors and not to tumor classes. So, for instance, "breast carcinoma" and "breast adenocarcinoma" are considered distinct tumors, even though they belong to the same class of tumors, "breast cancer."

As regards the integration of expression data, we followed the work of Guo et al. (2013), where authors show that there is a positive correlation between normalized Affymetrix and RNA-Seq data. We performed RMA normalization (McCall et al., 2010) for datasets based on Affymetrix platforms (GDS181, GDS596, and GDS1096), using the corresponding R Bioconductor package. For GDS3113, ProteinAtlas, and TCGA, we first computed the $\log_2$ of the number of fragments and then we normalized values using the quantile normalization method of Bolstad et al. (2003). For GDS181, GDS596, and GDS1096 datasets, normalized values were computed from raw data. Then, probes, which were present in a particular microarray dataset, were mapped to the corresponding genes. Finally, the expression of a gene for a specific tissue was computed as the average expression value of probes mapping to that gene in the tissue. For GDS3113, ProteinAtlas, and TCGA, instead, we directly normalized gene expression values for different tissues. We did not normalize EMTAB-62 data, since its source values were already normalized with RMA.

Finally, we assigned to each pair gene–tissue a unique positive expression score, given by the average normalized expression value of the gene in that tissue, according to the different datasets. Expression scores in SPECTRA range from 3.566 to 17.366 for tissues and from 0.01 to 17.343 for tumors.

[7]http://www.ebi.ac.uk/arrayexpress
[8]http://www.ncbi.nlm.nih.gov/geo
[9]http://www.proteinatlas.org

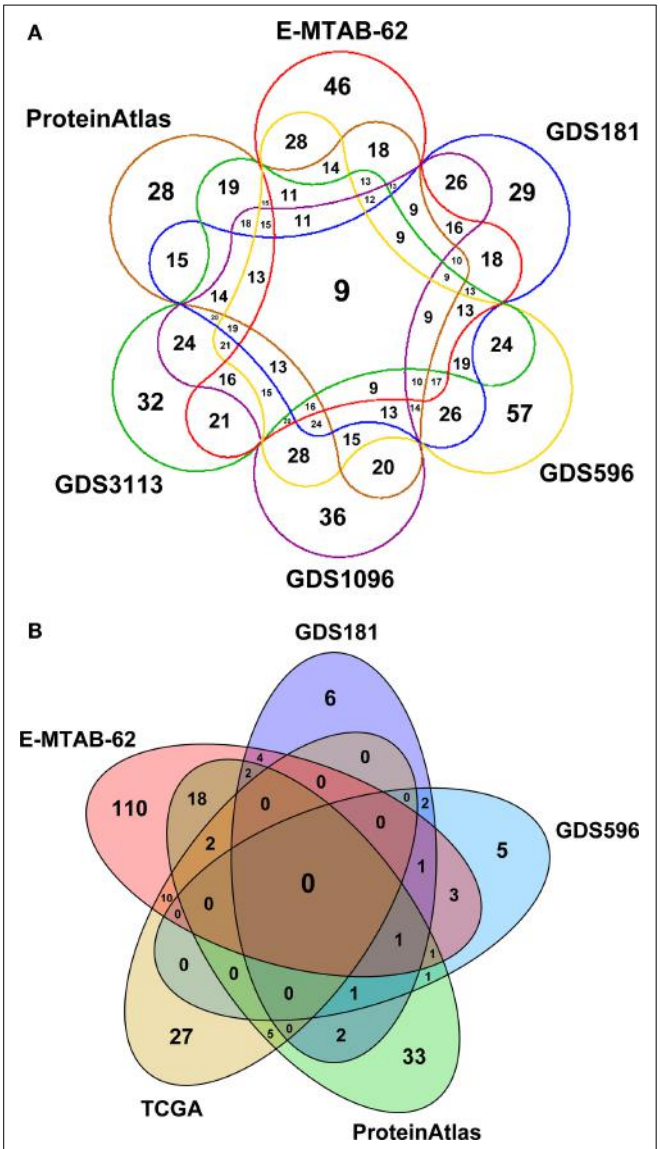


FIGURE 3 | Venn diagrams depicting: (A) the number of common tissues and (B) the number of common cancer types across expression datasets in SPECTRA. The zero overlap among the cancer datasets is due to the fact that cancers are considered according to their specific names and not to their class (breast cancer, prostate cancer, etc.).

2.3. Data Schema

SPECTRA database is structured as a MySQL relational database with six tables: *Genes*, *Tissues*, *Tumors*, *Interactions*, *Expr_normal*, and *Expr_tumor*.

The *Genes* table contains the list of all expressed and interacting genes. Each entry is identified by the gene symbol and contains associated data, including a description string, aliases, and cross references to Entrez Gene (if available).

The *Tissues* and *Tumors* tables have the same structure. Tissues and tumors are associated to different classes, depending on the organism part they refer to. Each entry is identified by a unique number and contains a description and the corresponding class.

SPECTRA contains 26 distinct classes of tissues and 32 distinct classes of tumors.

The *Interactions* table lists all the PPIs integrated in SPECTRA. Interactions are identified by a couple of gene symbols and the edge weight for each integrated dataset (when available) is stored, together with the average interaction weight across dataset reporting that interaction and the dataset coverage.

Expr_normal and *Expr_tumor* contain all the gene expressions in normal and cancer tissues. The unique identifier of *Expr_normal* is a couple gene–tissue, while entries in *Expr_tumor* are uniquely identified by the couple gene–tumor. In both tables, the normalized expression value for each integrated dataset (where available) and the average expression score are included as associated data.

2.4. An Algorithm for Differential Local Alignment of TS-PPI Networks

TS-PPI networks are compared in SPECTRA for identifying patterns of differential gene expressions between multiple TS-PPI networks.

Our goal is to find conserved sub-regions in the TS-PPI networks, which maximize the difference of expression values of aligned genes. The problem is related to that of finding maximal-scoring connected subgraphs, which is NP-hard, even in a common simpler setting where the aligning TS-PPI networks have the same set of nodes and edges (e.g., TS-PPI networks built starting from different expression data and the same interaction datasets) (Ideker et al., 2002).

In the case of two TS-PPI networks with the same set of nodes and edges (representing for instance case and control expression data), heuristic (Ideker et al., 2002; Sohler et al., 2004; Cabusora et al., 2005; Rajagopalan and Agarwal, 2005; Guo et al., 2007) and exact (Dittrich et al., 2008) solutions have been proposed. However, as far as we are concerned, no solutions are known for the multiple case. Here, we propose an approximate solution to the multiple differential alignment problem based on a modified version of the GASOLINE algorithm (Micale et al., 2014a). For simplicity, we consider TS-PPI networks with no multiple edges between two nodes.

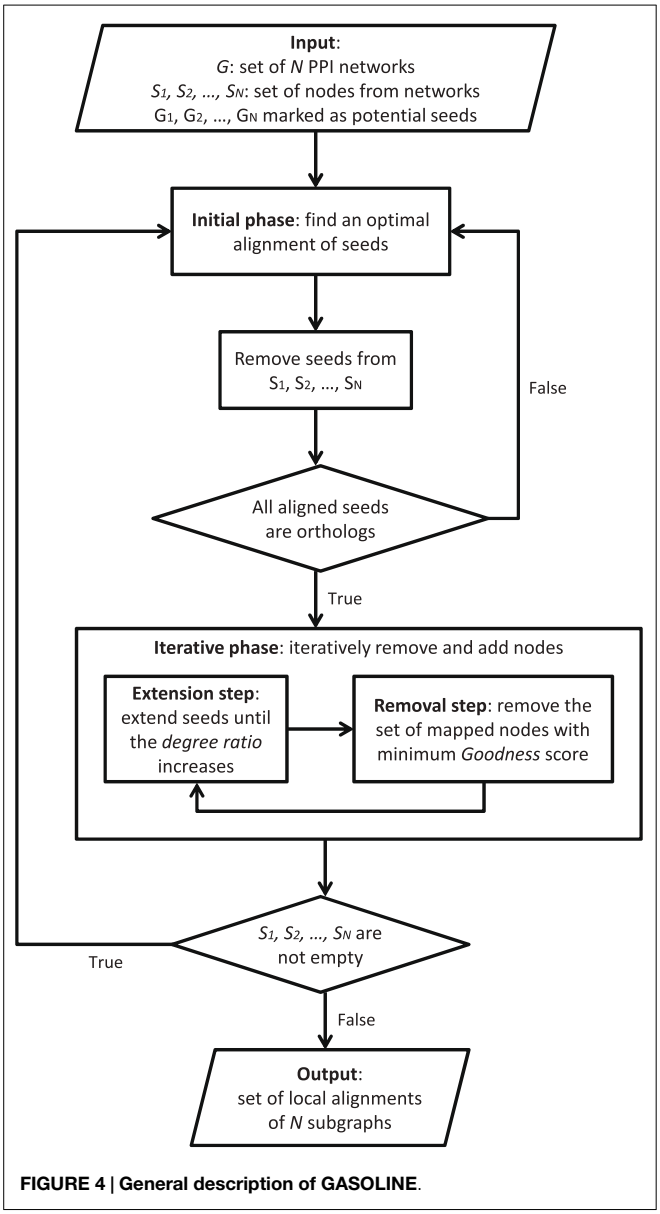


FIGURE 4 | General description of GASOLINE.

2.4.1. The GASOLINE Algorithm

GASOLINE (Micale et al., 2014a) is a greedy and stochastic algorithm for multiple local alignment of protein–protein interaction networks. Flowchart in **Figure 4** provides a general description of GASOLINE.

Given N weighted PPI networks of different species, where edge weights are probabilities of interaction between proteins, local alignment aims at finding a set of connected subnetworks, one from each network, that are conserved in their sequence and interaction pattern. Such subnetworks could represent evolutionary conserved complexes or pathways across different organisms.

Such a problem is related to subgraph isomorphism, which is known to be NP-complete (Cook, 1971). GASOLINE proposes an approximate solution through a stochastic-greedy strategy consisting of two phases.

In the first step, called bootstrap phase, we look for orthologous proteins across the networks and build a set of seeds. The set of seeds initially consists of proteins, one from each network, and includes all the starting nodes of the suboptimal local network alignment.

The second step, called iterative phase, repeatedly adds (extension step) or removes (removal step) nodes in the network alignment, trying to maximize the final alignment score. Each extension step adds, in each network, a single node to the corresponding seed. During the extension step, the seeds grow up producing a set of subgraphs, one from each network. The extension process is regulated by a properly defined degree ratio measuring the average density of the aligned subgraphs with respect to their neighbors in the networks. The extension is performed until the degree ratio increases.

Each removal step replaces from the current alignment the set of proteins (one from each network) with minimum topology similarity score.

The bootstrap phase and each extension step are performed through Gibbs sampling (Geman and Geman, 1984). In both cases, the Gibbs sampling builds a chain, where each state represents a combination (i.e., alignment) of single proteins, one from each network. First, a random initial state is selected. Then, the sampling method iteratively performs a transition from a state to another, by replacing a randomly chosen protein of the current alignment with a protein of the same network, according to a properly defined transition probability distribution.

Due to its non-deterministic nature, different iterations of GASOLINE may produce different local alignments. The above steps are iterated to produce a set of local networks alignments, which are then ranked according to an Index of Structural Conservation (ISC) score. ISC score measures the percentage of conserved interactions in the final alignment. The higher is ISC, the better is the alignment.

GASOLINE implements preprocessing and post-processing steps. During preprocessing, the search space for potential seeds is reduced. This is obtained by marking only proteins having orthologs in all aligning networks and with a significant interaction degree in each network. All marked nodes in each network $G_i (1 \leq i \leq N)$ are added to a set called S_i. These sets will be used in the initial phase and will be updated at each iteration. Finally, a post-processing filters the final set of local alignments returned by GASOLINE by removing highly overlapping complexes.

GASOLINE does not allow many-to-many mapping between aligned nodes. However, experimental results show that the algorithm can produce more reliable results than methods implementing many-to-many mapping. Moreover, GASOLINE is clearly faster than the state-of-art algorithms (Micale et al., 2014a).

2.4.2. The Adapted GASOLINE

We implemented a customized version of GASOLINE to compare two or more Tissue-Specific PPI (TS-PPI) networks for local differential alignment problem. GASOLINE algorithm was extended to deal with gene expressions as weights to the nodes.

Let A and B two genes and $Expr(A)$ and $Expr(B)$ their expression values, with $Expr(A) \geq Expr(B)$. In order to evaluate the expression difference between A and B, we compute the *log fold change*, defined as follows:

$$\text{LogFold}(A, B) = \log_2 \left(\frac{Expr(A)}{Expr(B)} \right) \quad (1)$$

Given N TS-PPI networks and a set of aligned genes $G = \{G_1, G_2, \ldots, G_N\}$, one for each TS-PPI network, MaxLogFold is the maximum value of LogFold function among all pairs of genes in G:

$$\text{MaxLogFold}(G) = \max\{\text{LogFold}(G_i, G_j) \ \forall \ 1 \leq i < j \leq n\} \quad (2)$$

We applied the following changes to original GASOLINE algorithm:

- We included the LogFold function in the Gibbs sampling procedure of bootstrap and iterative phases, by multiplying it by the topology and homology scores in the computation of node similarities;
- The number of iterations of Gibbs sampling both in the bootstrap and in the extension phase is governed by a new parameter, is α, which is a probability threshold related to N, the number of networks, according to the following formula:

$$k = \max \left\{ k' : \left(\frac{N-1}{N} \right)^{k'} > \alpha \right\} \quad (3)$$

  where $P = \left(\frac{N-1}{N}\right)^{k'}$ is the probability that a gene is never selected in k' consecutive iterations of Gibbs sampling. The idea is to stop Gibbs sampling when an alignment does not change for k consecutive iterations. The lower is α, the higher is k, so the more precise and slower will be the sampling procedure:
- We introduced a new threshold, *MaxLogFoldThreshold*, for the value of MaxLogFol function, and we used it to tune the extension process in place of the degree ratio: in particular, we extend the current alignment until the average value of *MaxLogFold* between the sets of aligned nodes is above such a threshold;
- In the remove phase, the set of aligning nodes with minimum value of MaxLogFold is deleted from the current local alignment;
- Given a local alignment $A = \{A_1, A_2, \ldots, A_w\}$, where w is the size of the alignment and $A_1, \ldots, A_w$ are the set of aligned genes, an average value of MaxLogFold(A_i) is computed together with the *ISC* score to evaluate the quality of the alignment.

3. Results

SPECTRA is a framework for retrieving and analyzing protein–protein interaction data specific for a given set of normal or cancer tissues. The underlying graph model in SPECTRA is the Tissues-Specific PPI network (or TS-PPI network), in which the genes of corresponding interacting proteins are both expressed in one or more tissues. The architecture of SPECTRA is composed by (i) the *searching tool*, which allows to build TS-PPIs; (ii) the *comparison tool* to look for shared differential expressions patterns between genes of two or more TS-PPI networks. Results can be graphically visualized by using Cytoscape.js or downloaded as text files.

3.1. SPECTRA Search Tool: Building TS-PPI Networks in SPECTRA

SPECTRA builds TS-PPI networks starting from a user-defined set of genes, tissues, expression data, and interaction data. **Figure 5** depicts the search interface of SPECTRA.

In the "Gene data" section (**Figure 5A**), the user can look for all genes expressed in a set of tissues or restrict the search to a specific list of genes. Genes can be provided with their official names or Aliases (e.g., Ensembl Gene, Entrez Gene, Affy).

In the "Expression data" section (**Figure 5B**), the user limits the search to a set of tissues/tumors and to a set of expression datasets or uploads a text file with custom expression data. Note that the two options are mutually exclusive, that is, all the settings concerning datasets and tissue/tumors will be ignored if the user

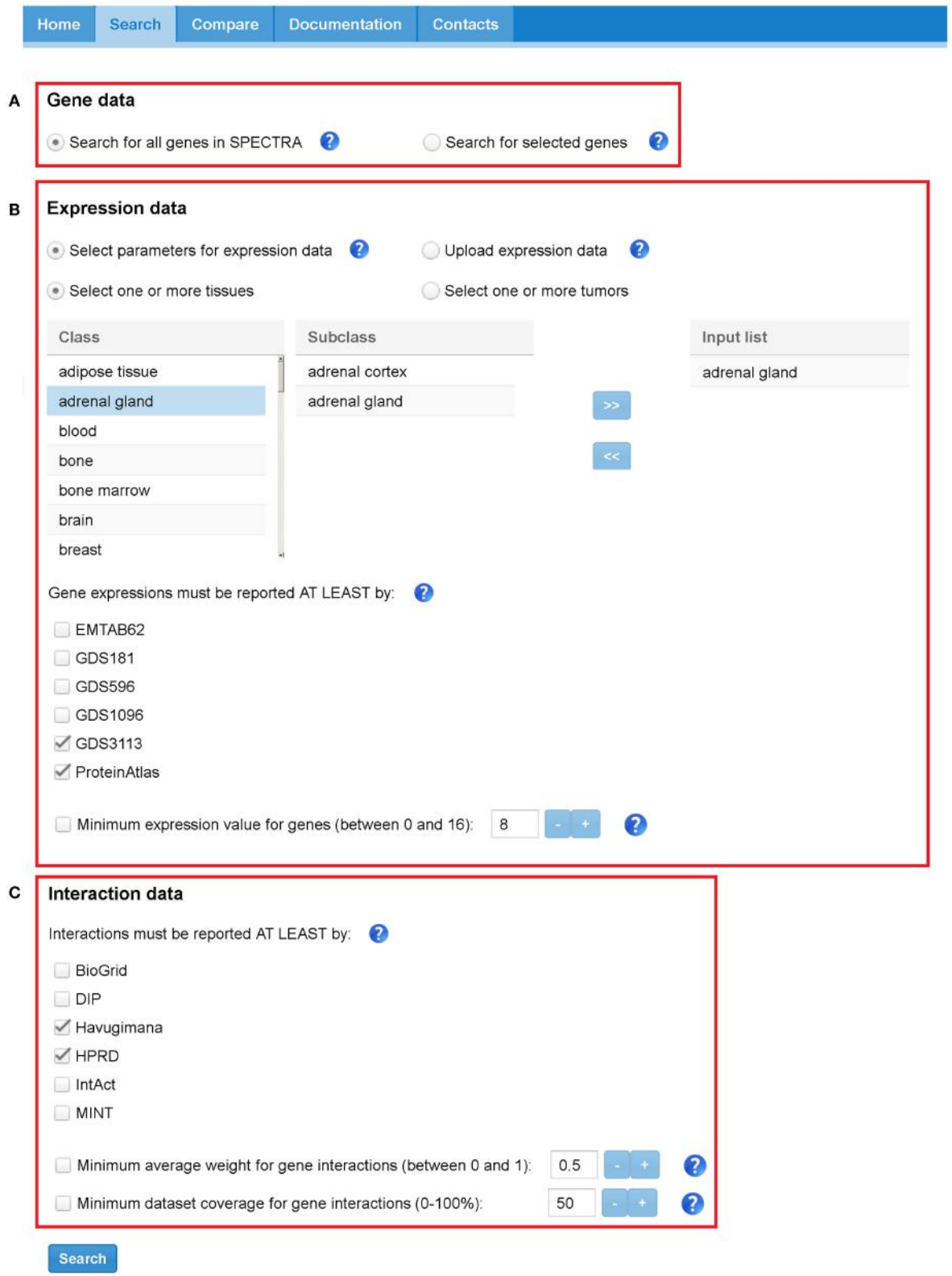


FIGURE 5 | SPECTRA search tabbed panel. Red boxes highlight the three sections: **(A)** "Gene data," **(B)** "Expression data," and **(C)** "Interaction data." In this case, the parameters have been set to indicate that we want to retrieve all the interactions that are present at least in Havugimana and HPRD, involving genes that are expressed in adrenal gland" tissue according at least to GDS3113 and ProteinAtlas. In this example, we neither restrict our search to a predefined set of genes nor provide a threshold for interaction weights, dataset coverage, and expression scores.

provides a custom text file. Available tissues and tumors in SPECTRA are listed in a table and can be easily included in the input query list with a double click in each entry. When no data are provided, all the tissues and tumors in SPECTRA are considered. Tissues and tumors are also mutually exclusive, meaning that a TS-PPI network built-in SPECTRA cannot contains interactions defined on both normal and tumor tissues. However, two TS-PPI networks defined upon a specific set of tissues and tumors, respectively, can be always compared for differential analysis with the adapted GASOLINE. The user can also select one or more datasets from which the expression have to be reported. When the expression is in other datasets it will be also given. When no dataset is selected, all expression data in SPECTRA are considered.

Finally, a further filter on genes can be applied by indicating a threshold for the minimum normalized value of gene expressions to be considered.

The "Interaction data" section (**Figure 5C**) contains the parameters for filtering interaction data. As above, the user can select one or more datasets where protein interactions have to be reported. If no interaction dataset is selected, all PPIs in SPECTRA are considered. A threshold can be provided to select interaction weights above a given value and a minimum dataset coverage.

When all input parameters have been specified, the user clicks on the "Search" button. At the end of the process, all the TS-PPIs found are listed in a result table (**Figure 6**). For each TS-PPI, we show the interacting genes, the tissues where they are expressed, the expression values of genes in tissues, the average interaction weights and dataset coverages of corresponding proteins. Results are ordered by dataset coverage and average interaction weight. Expression values and interaction weights are depicted with colored progress bar, where colors range from cyan (low values) to red (high values).

By selecting a specific TS-PPI in the result table, additional data about the interaction and the interacting genes are shown (**Figures 6** and **7**). A list of datasets reporting the interaction and the corresponding interaction weight is reported on the right of the result table (**Figure 6**). Below the result table, two panels with details about the interacting genes are shown (**Figure 7**). For each gene, description and aliases are provided, together with the lists of tissues and tumors where the gene is expressed, according to the different expression datasets, ordered by expression score.

3.2. SPECTRA Comparison Part: Compare TS-PPI Subnetworks

TS-PPI networks can be compared in SPECTRA for identifying patterns of differential gene expressions between multiple TS-PPI networks. The goal is to find conserved sub-regions in the TS-PPI networks, which maximize the difference of expression values of aligned genes.

Figure 8 shows the "Compare" tabbed panel in SPECTRA. Before running the adapted GASOLINE, the user has to upload at least two TS-PPI networks. For each network, the number of nodes and edges are reported. Networks can also be renamed by double clicking on the corresponding cell. Note that uploaded TS-PPI networks with multi-edges between nodes will be always treated as simple networks, where multi-edges are replaced by a single edge with weight equals to the average weight of multi-edges and label given by the concatenation of the multi-edge labels.

Once the networks have been uploaded, the user can click on "Run GASOLINE" button to set the input parameters for the adapted GASOLINE (**Figure 8**).

We briefly describe their meaning (default values are reported in brackets):

- "Sigma": the minimum degree of candidate nodes for the initial alignment of seeds (1);
- "Alpha": a value between 0 and 1, which regulates the number of iterations of Gibbs sampling in the bootstrap and extend phases (default 0.05);
- "Overlap threshold": a maximum average overlap threshold between local alignments, which is used to remove highly

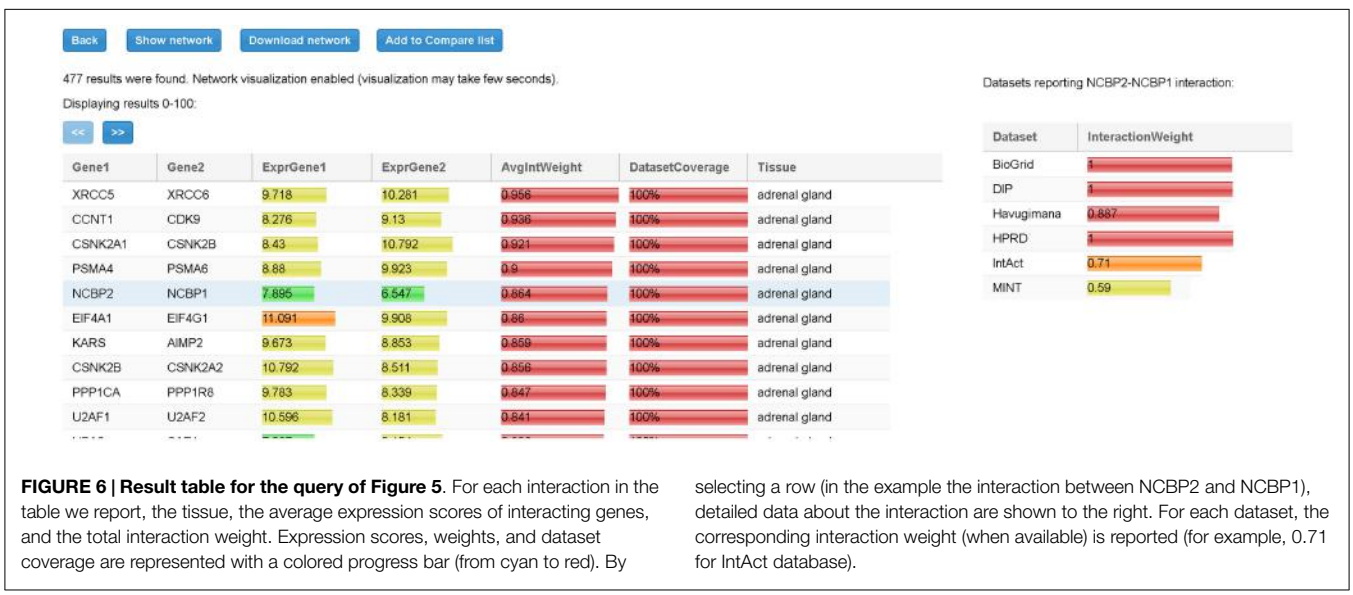


FIGURE 6 | Result table for the query of Figure 5. For each interaction in the table we report, the tissue, the average expression scores of interacting genes, and the total interaction weight. Expression scores, weights, and dataset coverage are represented with a colored progress bar (from cyan to red). By selecting a row (in the example the interaction between NCBP2 and NCBP1), detailed data about the interaction are shown to the right. For each dataset, the corresponding interaction weight (when available) is reported (for example, 0.71 for IntAct database).

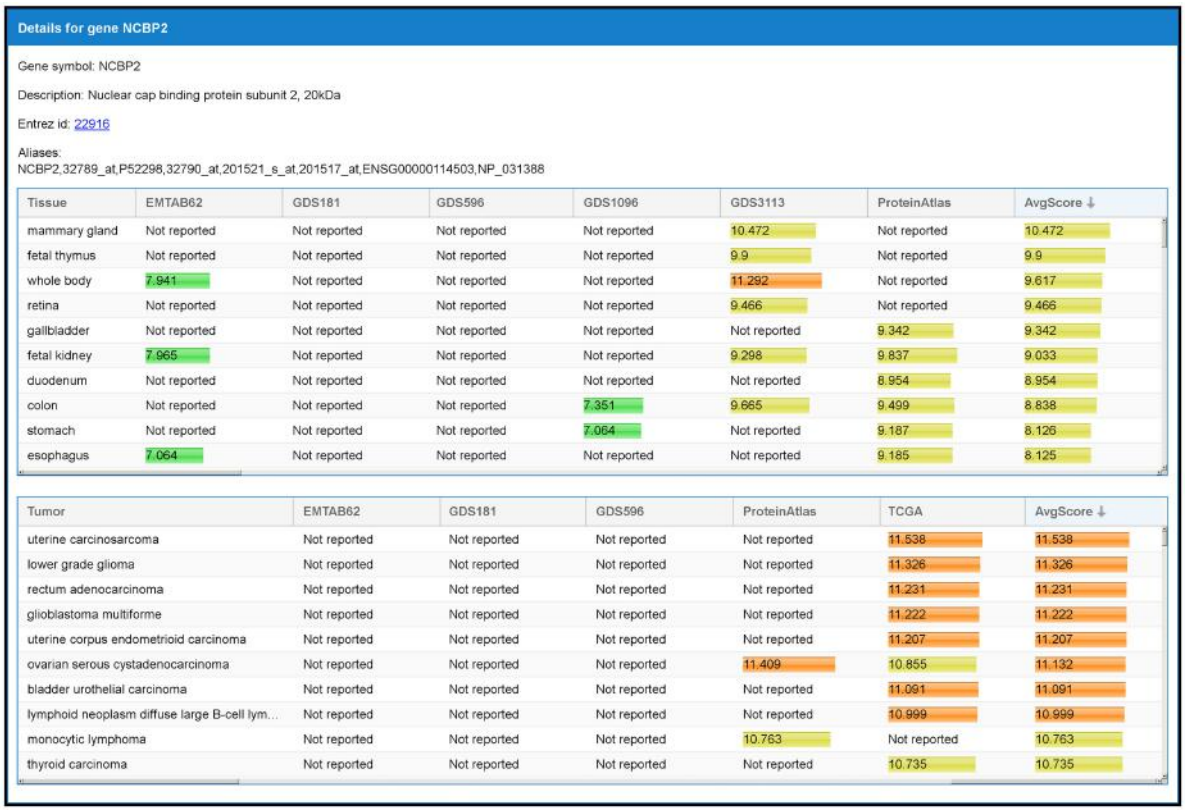


FIGURE 7 | The panel with detailed information of a gene. When an interaction is selected from the result table (**Figure 6**), two panels with additional data, one for each interacting gene, are shown. This example refers to the detailed panel for gene NCBP2, which appears when the row table of **Figure 6** is selected. In the detailed panel, the gene symbol, the description, the corresponding ID in Entrez Gene database (when available), and aliases (including references in other databases) are reported. Finally, two tables with the set of tissues and tumors where the gene is expressed are shown. These are shown in decreasing order with respect to the average expression scores.

overlapping alignments. It takes values between 0 and 1 (default 0.5, which means 50%);

- "Refine iterations": the number of iterations of the iterative phase, i.e., extend steps followed by a removal step (default 10);
- "Minimum alignment size": the minimum size of a local alignment. Local alignments with size lower this minimum size are not reported in final list (default 3);
- "Minimum gene expression log fold change threshold": value for *MaxLogFoldThreshold*, which controls the extension process (default 0.6).

According to the experiments reported in Micale et al. (2014a,b), we assigned to each parameter default values, which guarantee a good tradeoff between speed and accuracy of GASOLINE.

"Alpha" and "Refine iterations" parameters are strictly related to the stochastic nature of the algorithm. Lower values for "Alpha" and higher values for "Iter Refine" can be assigned to improve accuracy; however, the suggested default values are enough to yield good alignment results. Higher values of "Sigma" can be used to restrict the search to alignments starting from central genes in the input networks and to speedup the algorithm. Lower values of "Overlap threshold" and higher values of "Minimum alignment size" allow to prune the final set of local alignments.

MaxLogFoldThreshold is the most critical parameter for GASOLINE. By increasing this threshold, the number and the size of final local alignments can be highly decreased and the algorithm could become much faster. Notice that there is no constant ideal value for *MaxLogFoldThreshold*, because it is highly dependent on the properties of input expression data. For log-transformed gene expression data, like the one which are present in SPECTRA database, low values of *MaxLogFoldThreshold* (0.2–1) are recommended.

Before running the adapted GASOLINE by clicking on "Run GASOLINE" button, the user has to indicate an homology scoring scheme between proteins of different aligning TS-PPI networks (**Figure 8**). The default naive solution is to use gene names for computing similarities: if two nodes have the same label, then they are considered homologs. Otherwise, user can upload an homology score file.

When the adapted GASOLINE ends, it gives as output a list of local alignments (if any, see **Figure 9**). For each alignment, the size, the average value of *MaxLogFold*, and the *ISC* score are reported.

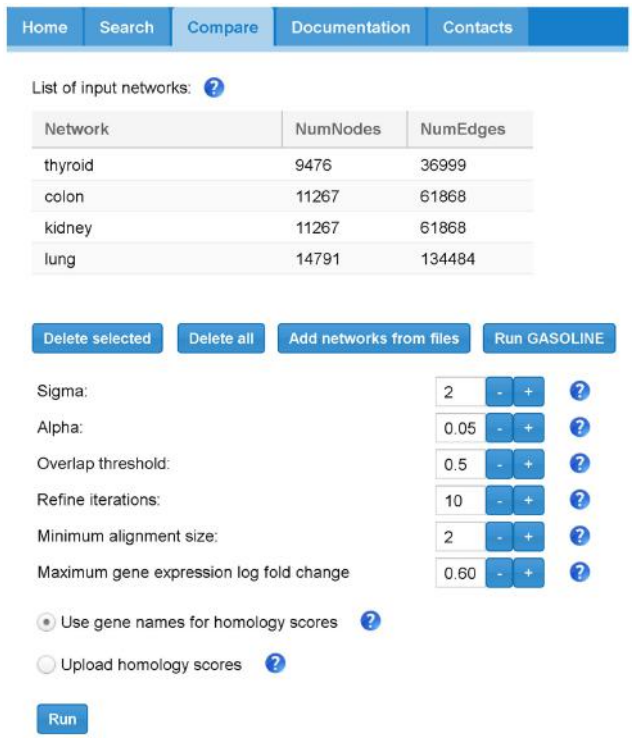


FIGURE 8 | The SPECTRA compare panel. In this example, we first loaded 4 different TS-PPI networks from files using the "Add networks" button. Then by clicking on "Run Gasoline" the form for the selection of the adapted GASOLINE input parameters appears.

By selecting an alignment, its details are reported on the right (**Figure 9**). Alignment data include the set of nodes and edges attributes. The final mapping of aligned nodes is represented as a matrix in which columns contain nodes of the same network and rows represent the mapped genes.

3.3. Alternative Input for SPECTRA

User can upload text files in SPECTRA for building and comparing network. Expression data can be provided as text files in the "Expression data" section (**Figure 5B**) by selecting the Üpload expression data" option. Expression data files should have a matrix format with a row header representing tissues, a column header representing genes, and matrix elements indicating the gene expression value in a tissue.

There are two ways to provide input TS-PPI networks for comparison. User can either upload a text file or create the TS-PPI network with the SPECTRA searching tool and pass it to the comparison page. In the first case, network files are uploaded by clicking on "Add networks from files" in the "Compare" tabbed panel (**Figure 8**).

TS-PPI network files for comparison follows the same format of the result table in SPECTRA (**Figure 6**), except for the dataset coverage, with fields separated by tab characters. In the second case, one or more TS-PPI networks for specific tissues are passed to the comparison tool, by clicking on the "Add to compare list" button. The network is then added as input to the comparison list (**Figure 8**). By default, networks are added with the name of the corresponding tissue, optionally followed by a progressive number whenever two or more TS-PPI networks for the same tissue are already present in the table. Anyway, networks can be later renamed by the user from the comparison table, before running GASOLINE.

In the homology file, needed to run the adapted GASOLINE algorithm, each row contains a pair of nodes of different TS-PPI networks, followed by a positive score value.

3.4. SPECTRA Output

TS-PPI networks (or subnetworks of them) are downloadable from the result panel, by clicking on "Download network" button (**Figure 6**). The user can filter the set of tissues upon which the TS-PPI network is defined. TS-PPI networks will be saved into different text files, one for each selected tissue or tumor. The file format is the same of the result table (**Figure 6**), with fields separated by tab characters.

The set of differential alignments returned by the adapted GASOLINE can be saved as .zip archive. The archive will contain a text file for each alignment. Each file contains the same alignment information reported in **Figure 9**.

Results can also be visualized by using Cytoscape.js[10], a JavaScript library for the analysis and visualization of networks. In the 2D visualization, TS-PPI networks can be navigated and zoomed. A TS-PPI network can be visualized from the result panel (**Figure 6**). **Figure 10** shows two different examples of visualizations of TS-PPI networks within SPECTRA, with one (**Figure 10A**) or more (**Figure 10B**) tissues. Nodes and edges are differently colored according to the tissues of the TS-PPI network. Nodes are represented as pies with multiple colored slices. The diameter of the pie is proportional to the total expression score of the gene (considering all tissues of the TS-PPI network) and the size of each pie slice is proportional to the expression score of the gene in the corresponding tissue. Edge line widths are proportional to the interaction weights.

The alignments can be visualized in 2D (**Figure 11**), by selecting them from the list of local alignments and clicking on the "Show alignment network" button (**Figure 9**). Aligned nodes are colored according to the network they belong to and their sizes are proportional to the genes expressions. Edges are divided into two categories: intra-edges and inter-edges. Intra-edges connect nodes of the same subnetwork and are represented with solid lines with variable width, depending on the interaction weights. Inter-edges connect aligned nodes of different networks and are drawn with dashed black lines. In both cases, we used the Constraint-Based Layout (COLA) for network visualization.

3.5. Case Study

In this section, we show a practical usage of SPECTRA through a case study. We compared a set of four TS-PPI networks, built from genes expression data in normal and well differentiated, moderately differentiated, and poorly differentiated breast cancer tissues. The aim is to identify subnetworks of differentially expressed genes across the normal breast and the three different grades of breast tumors.

[10]http://js.cytoscape.org

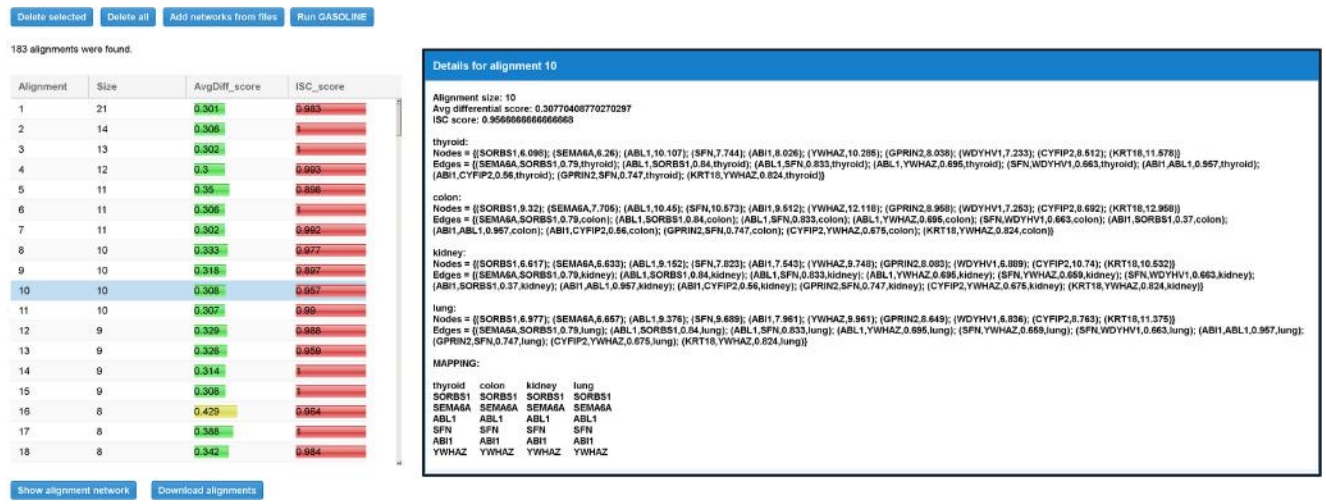


FIGURE 9 | Result table for the differential local alignment of the four TS-PPI networks of Figure 8 with the Adaptive GASOLINE. The table reports, for each alignment, the size (i.e., the number of aligned nodes), the average expression difference between aligned nodes, and the ISC (Index of Structural Conservation) score. When the user selects a row in the table, a panel with alignment details is shown to the right. Details include the list of aligned subnetworks (defined by the set of nodes and edges) and the mapping between aligned nodes. Nodes of aligned networks are represented by the corresponding ids, followed by their weights, while edges are represented by the ids of interacting proteins, followed by the interaction weights and the corresponding tissues. Alignment mapping is represented as a matrix where rows contain aligned proteins and columns represent nodes of the same subnetwork.

3.5.1. Data Preprocessing

We downloaded four breast cancer expression datasets for which information about the stage of breast tumors were available: GSE2361 (Ge et al., 2005), GSE2990 (Sotiriou et al., 2006), GSE4922 (Ivshina et al., 2006), and GSE7390 (Desmedt et al., 2007). We normalized data using RMA in R Bioconductor package (McCall et al., 2010).

The four expression datasets were then combined using COMBAT (Johnson and Li, 2007) into the R InSilicoDbMerging package. Finally, we grouped samples of the integrated dataset into four categories according to the grade of breast tumor (0 for normal tissue, 1 for well-differentiated tumor cells, 2 for moderately differentiated cells, and 3 for poorly differentiated cells). For each category, we computed the average expression value of each gene among samples. Results are stored into four different files (one per category).

3.5.2. Uploading Data in SPECTRA and Building Breast TS-PPI Networks

We loaded the expression files in the "Expression data" panel in SPECTRA (**Figure 5B**) and we selected BioGRID and IntAct as PPI datasets in the "Interaction data" panel (**Figure 5C**). SPECTRA builds four TS-PPI networks, each of them has 7,472 nodes and 29,765 edges. We added each network to the comparison list of GASOLINE (**Figure 8**), by clicking on *Add to compare list* from the Result panel (**Figure 6**).

3.5.3. Results of GASOLINE on TS-PPI Networks

Networks have been aligned by clicking on *Run GASOLINE* with the following parameters:

- Sigma = 1;
- Alpha = 0.05;
- Overlap threshold = 0.5;
- Refine iterations = 10;
- Minimum complex size = 2;
- Maximum gene expression log fold change threshold = 0.3;
- Use gene names for homology score.

GASOLINE took 27 s to complete the task and returned 20 local alignments. In **Figure 11**, the two biggest alignments are shown using the SPECTRA visualization tool. Both alignments contain genes that are known to be involved in breast cancer at different stages.

More precisely, the major group of aligned nodes in **Figure 11A** is formed by the chemokine proteins (CXCL10, CXCL9, CXCL11, CCL5) and the chemokine receptors CXCR3 and CCR1, which are all highly overexpressed across the different grades of breast tumor. Chemokines can be responsible for leukocyte migration during processes of tissue development and formation, or can attract immune cells to a site of inflammation. Chemokines and chemokine receptors are known to have an important role on cancer metastasis, by facilitating tumor dissemination (Muller et al., 2000; Karnoub and Weinberg, 2007). DPP4 gene has a lower expression variation but ensures the communication between CCL5, CCR1, and the other chemokine proteins. This result agrees with the key role of DPP4 in signal transduction and tumor progression (Pro and Dang, 2004).

The alignment of **Figure 11B** is characterized by the Human Leukocyte Antigen (HLA) system (HLA-DRB1, HLA-DMB, HLA-DMA, HLA-DRA). The HLA system is composed by proteins on cell surface that are responsible for regulation of the immune system. HLA genes exhibit very high differential expression between normal and tumor cells and their overexpression in breast cancers is confirmed by several papers (Bartek et al., 1987; Kaneko et al., 2011; Da Silva et al., 2013).

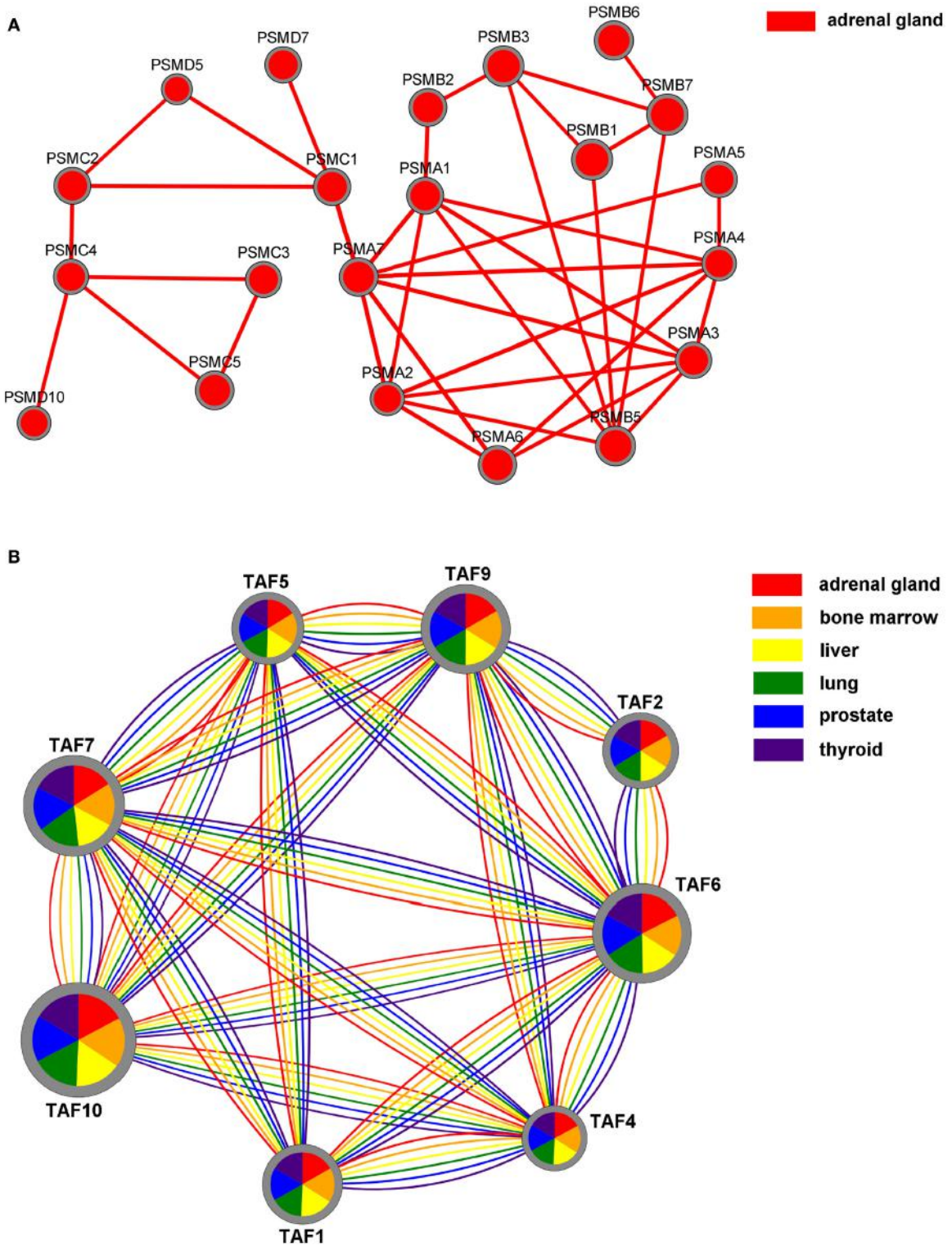


FIGURE 10 | The Network visualization in SPECTRA. **(A)** A TS-PPI network for a single tissue; **(B)** A TS-PPI network for multiple tissues. In this case, nodes are represented as pies with slice sizes proportional to the expression of corresponding gene in a tissue. Nodes and edges are colored according to the corresponding tissue and node dimensions are proportional to the total gene expression score.

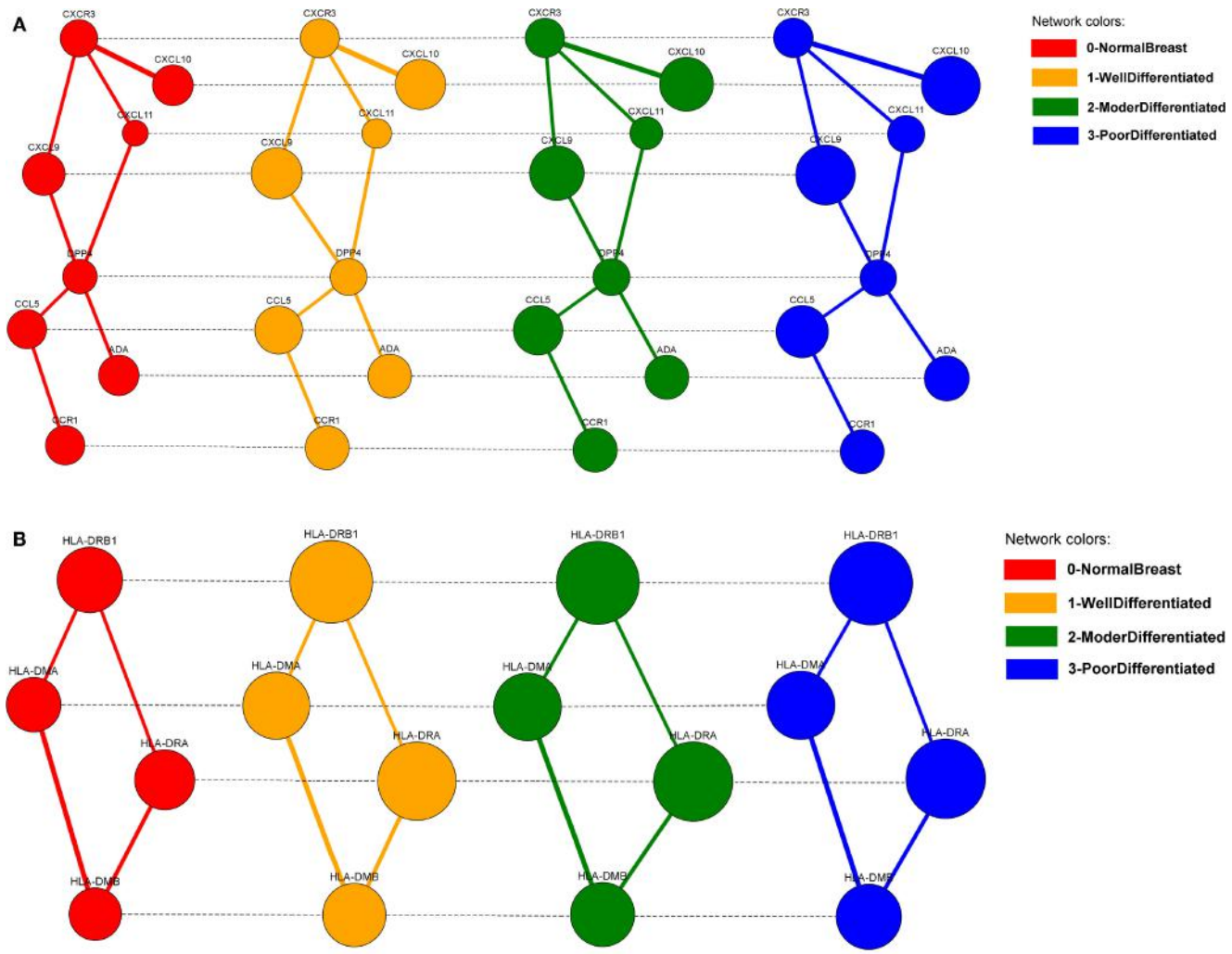


FIGURE 11 | The two biggest local differential alignments found by the Adaptive GASOLINE for the TS-PPI networks of normal breast cells (grade 0), well-differentiated cells (grade 1), moderately differentiated cells (grade 2), and poorly differentiated cells (grade 3). **(A)** A complex of chemokine proteins. **(B)** The Human Leukocyte Antigen (HLA) system. Nodes and edges are colored according to the corresponding network. Edge widths are proportional to the strength of interaction. Node dimensions are proportional to the gene expressions. Solid lines (intra-edges) connect the nodes of the same network, while dashed lines (inter-edges) connect the aligned nodes.

The above case study highlights the capability of SPECTRA in helping researchers in producing novel biologically sound hypothesis and insight in the study of tissue-specific diseases.

4. Discussion

SPECTRA is a knowledge base to build and compare tissue or tumor-specific PPI networks. It overcomes the current PPI network analysis limitations mainly due (i) to the spreading of data in several databases with low overlap; (ii) to be unaware of the role of proteins in human tissues and diseases. SPECTRA integrates 13 databases of both protein–protein interactions and expressions data. Moreover, it provides an algorithm to compare built-in or custom tissue and tumor-specific PPI networks and identify subnetworks of differentially expressed genes. Finally, the results can easily browsed trough a lightweight web application equipped with a 2D visualization network tool based on Cytoscape.js. Experiments performed on four TS-PPI networks built from gene expression data consisting of normal and breast cancer tissues show that the comparison algorithm can produce biologically significant results. SPECTRA database will go under update twice a year, with a semi-automatic curation of data downloaded from the online repositories. Future developments of SPECTRA aim to provide further network mining algorithms devoted to the analysis of expression data and the validation and annotation with ontologies of results.

Acknowledgments

Publication of this article has been funded by PON 2007-2013 grant, SIGMA – PON01 00683 – CUP B61H11000380005.

References

Adamcsek, B., Palla, G., Farkas, I., Derenyi, I., and Vicsek, T. (2006). Cfinder: locating cliques and overlapping modules in biological networks. *Bioinformatics* 22, 1021–1023. doi:10.1093/bioinformatics/btl039

Alaimo, S., Pulvirenti, A., Giugno, R., and Ferro, A. (2013). Drug-target interaction prediction through domain-tuned network-based inference. *Bioinformatics* 29, 2004–2008. doi:10.1093/bioinformatics/btt307

Bader, G., and Hogue, C. (2003). An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics* 4:2. doi:10.1186/1471-2105-4-2

Banks, E., Nabieva, E., Peterson, R., and Singh, M. (2008). Netgrep: fast network schema searches in interactomes. *Genome Biol.* 9, R138. doi:10.1186/gb-2008-9-9-r138

Barrett, T., Wilhite, S., Ledoux, P., Evangelista, C., Kim, I. F., Tomashevsky, M., et al. (2013). Ncbi geo: archive for functional genomics data sets-update. *Nucleic Acids Res.* 41, D991–D995. doi:10.1093/nar/gks1193

Barshir, R., Basha, O., Eluk, A., Smoly, I., Lan, A., and Yeger-Lotem, E. (2013). The tissuenet database of human tissue protein-protein interactions. *Nucleic Acids Res.* 41, D841–D844. doi:10.1093/nar/gks1198

Barshir, R., Shwartz, O., Smoly, I., and Yeger-Lotem, E. (2014). Comparative analysis of human tissue interactomes reveals factors leading to tissue-specific manifestation of hereditary diseases. *PLoS Comput. Biol.* 10:e1003632. doi:10.1371/journal.pcbi.1003632

Bartek, J., Petrek, M., Vojtesek, B., Bartkova, J., Kovarik, J., and Rejthar, A. (1987). Hla-dr antigens on differentiating human mammary gland epithelium and breast tumours. *Br. J. Cancer* 56, 727–733. doi:10.1038/bjc.1987.278

Bolstad, B., Irizarry, R., Astrand, M., and Speed, T. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics* 19, 185–193. doi:10.1093/bioinformatics/19.2.185

Bossi, A., and Lehner, B. (2009). Tissue specificity and the human protein interaction network. *Mol. Syst. Biol.* 5, 260. doi:10.1038/msb.2009.17

Bruckner, S., Huffner, F., Karp, R., Shamir, R., and Sharan, R. (2010). Topology-free querying of protein interaction networks. *J. Comput. Biol.* 17, 237–252. doi:10.1089/cmb.2009.0170

Cabusora, L., Sutton, E., Fulmer, A., and Forst, C. (2005). Differential network expression during drug and stress response. *Bioinformatics* 21, 2898–2905. doi:10.1093/bioinformatics/bti440

Hamosh, A., Scott, A., Amberger, J., Bocchini, C., and McKusick, V. (2005). Online mendelian inheritance in man (OMIM), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Res.* 33, D514–D517. doi:10.1093/nar/gki033

Chen, G., and Wang, J. (2012). Identifying functional modules in tissue specific protein interaction network. *IEEE Int. Conf. Bioinform. Biomed. Workshops* 2012, 581–586. doi:10.1109/BIBMW.2012.6470204

Cook, S. (1971). "The complexity of theorem-proving procedures," in *Proc. 3rd ACM Symposium on Theory of Computing* (New York: ACM), 151–158. doi:10.1145/800157.805047

Csermerly, P., Korcsmaros, T., Kiss, H., London, G., and Nussinov, R. (2013). Structure and dynamics of molecular networks: a novel paradigm of drug discovery: a comprehensive review. *Pharmacol. Ther.* 138, 333–408. doi:10.1016/j.pharmthera.2013.01.016

Da Silva, G., Silva, T., Duarte, R., Neto, N., Carrara, H., Donadi, E. A., et al. (2013). Expression of the classical and nonclassical hla molecules in breast cancer. *Int. J. Breast Cancer* 2013, 250435. doi:10.1155/2013/250435

Dao, P., Wang, K., Collins, C., Ester, M., Lapuk, A., and Sahinalp, S. (2011). Optimally discriminative subnetwork markers predict response to chemotherapy. *Bioinformatics* 27, i205–i213. doi:10.1093/bioinformatics/btr245

Desmedt, C., Piette, F., Loi, S., Wang, Y., Lallemand, F., Haibe-Kains, B., et al. (2007). Strong time dependence of the 76-gene prognostic signature for node-negative breast cancer patients in the transbig multicenter independent validation series. *Clin. Cancer Res.* 13, 3207–3214. doi:10.1158/1078-0432.CCR-06-2765

Dezso, Z., Nikolski, Y., Sviridov, E., Shi, W., Serebriyskaya, T., Dosymbekov, D., et al. (2008). A comprehensive functional analysis of tissue specificity of human gene expression. *BMC Biol.* 6:49. doi:10.1186/1741-7007-6-49

Dittrich, M., Klau, G., Rosenwald, A., Dandekar, T., and Muller, T. (2008). Identifying functional modules in protein-protein interaction networks: an integrated exact approach. *Bioinformatics* 24, i223–i231. doi:10.1093/bioinformatics/btn161

Emig, D., and Albrecht, M. (2011). Tissue-specific proteins and functional implications. *J. Proteome Res.* 10, 1893–1903. doi:10.1021/pr101132h

Ferro, A., Giugno, R., Pigola, G., Pulvirenti, A., Skripin, D., Bader, G., et al. (2007). Netmatch: a cytoscape plugin for searching biological networks. *Bioinformatics* 23, 910–912. doi:10.1093/bioinformatics/btm032

Flannick, J., Novak, A., Srinivasan, B., McAdams, H., and Batzoglou, S. (2006). Graemlin: general and robust alignment of multiple large interaction networks. *Genome Res.* 16, 1169–1181. doi:10.1101/gr.5235706

Franceschini, A., Szklarczyk, D., Frankild, S., Kuhn, M., Simonovic, M., Roth, A., et al. (2013). String v9.1: protein-protein interaction networks, with increased coverage and integration. *Nucleic Acids Res.* 41, D808–D815. doi:10.1093/nar/gks1094

Ge, X., Yamamoto, S., Tsutsumi, S., Midorikawa, Y., Ihara, S., Wang, S. M., et al. (2005). Interpreting expression profiles of cancers by genome-wide survey of breadth of expression in normal tissues. *Genomics* 86, 127–141. doi:10.1016/j.ygeno.2005.04.008

Geman, S., and Geman, D. (1984). Stochastic relaxation, gibbs distributions, and the Bayesian restoration of images. *IEEE Trans. Pattern Anal. Mach. Intell.* 6, 721–741. doi:10.1109/TPAMI.1984.4767596

Guo, Y., Sheng, Q., Li, J., Ye, F., Samuels, D., and Shyr, Y. (2013). Large scale comparison of gene expression levels by microarrays and RNAseq using tcga data. *PLoS ONE* 8:e71462. doi:10.1371/journal.pone.0071462

Guo, Z., Li, Y., Gong, X., Yao, C., Ma, W., Wang, D., et al. (2007). Edge-based scoring and searching method for identifying condition-responsive protein-protein interaction sub-network. *Bioinformatics* 23, 2121–2128. doi:10.1093/bioinformatics/btm294

Havugimana, P., Hart, G., Nepusz, T., Yang, H., Turinsky, A., and Zhihua, A. (2012). A census of human soluble protein complexes. *Cell* 150, 1068–1081. doi:10.1016/j.cell.2012.08.011

Huang, H., Wu, X., Pandey, R., Li, J., Zhao, G., Ibrahim, S., et al. (2012). C2maps: a network pharmacology database with comprehensive disease-gene-drug connectivity relationships. *BMC Genomics* 13:S17. doi:10.1186/1471-2164-13-S6-S17

Ideker, T., Ozier, O., Schwikowski, B., and Siegel, A. (2002). Discovering regulatory and signalling circuits in molecular interaction networks. *Bioinformatics* 18, S233–S240. doi:10.1093/bioinformatics/18.suppl_1.S233

Ivshina, A., Joshy, G., Senko, O., Mow, B., Putti, T. C., Smeds, J., et al. (2006). Genetic reclassification of histologic grade delineates new clinical subtypes of breast cancer. *Cancer Res.* 66, 10292–10301. doi:10.1158/0008-5472.CAN-05-4414

Johnson, W., and Li, C. (2007). Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics* 8, 118–127. doi:10.1093/biostatistics/kxj037

Kalaev, M., Bafna, V., and Sharan, R. (2009). Fast and accurate alignment of multiple protein networks. *J. Comput. Biol.* 16, 989–999. doi:10.1089/cmb.2009.0136

Kamburov, A., Stelzl, U., Lehrach, H., and Herwig, R. (2013). The consensuspathdb interaction database: 2013 update. *Nucleic Acids Res.* 41, D793–D800. doi:10.1093/nar/gks1055

Kaneko, K., Ishigami, S., Kijma, Y., Funasako, Y., Hirata, M., Okumura, H., et al. (2011). Clinical implication of hla class i expression in breast cancer. *BMC Cancer* 11:454. doi:10.1186/1471-2407-11-454

Karnoub, A., and Weinberg, R. (2007). Chemokine networks and breast cancer metastasis. *Breast Dis.* 26, 75–85.

Lage, K., Hansen, N., Karlberg, E., Eklund, A., Roque, F., Donahoe, P. K., et al. (2008). A large-scale analysis of tissue-specific pathology and gene expression of human disease genes and complexes. *Proc. Natl. Acad. Sci. U.S.A.* 105, 20870–20875. doi:10.1073/pnas.0810772105

Liao, C., Lu, K., Baym, M., Singh, R., and Berger, B. (2009). Isorankn: spectral methods for global alignment of multiple protein networks. *Bioinformatics* 25, 253–258. doi:10.1093/bioinformatics/btp203

Licata, L., Briganti, L., Peluso, D., Perfetto, L., Iannuccelli, M., Galeota, E., et al. (2012). Mint, the molecular interaction database: 2012 update. *Nucleic Acids Res.* 40, D857–D861. doi:10.1093/nar/gkr930

Lopes, T., Schaefer, M., Shoemaker, J., Matsuoka, Y., Fontaine, J. F., Neumann, G., et al. (2011). Tissue-specific subnetworks and characteristics of publicly available human protein interaction databases. *Bioinformatics* 27, 2414–2421. doi:10.1093/bioinformatics/btr414

Lukk, M., Kapushesky, M., Nikkila, J., Parkinson, H., Goncalves, A., Huber, W., et al. (2010). A global map of human gene expression. *Nat. Biotechnol.* 28, 322–324. doi:10.1038/nbt0410-322

Magger, O., Waldman, Y., Ruppin, E., and Sharan, R. (2012). Enhancing the prioritization of disease-causing genes through tissue specific protein interaction networks. *PLoS Comput. Biol.* 8:e1002690. doi:10.1371/journal.pcbi.1002690

McCall, M., Bolstad, B., and Irizarry, R. (2010). Frozen robust multiarray analysis (frma). *Biostatistics* 11, 242–253. doi:10.1093/biostatistics/kxp059

Mete, M., Tang, F., Xu, X., and Nurcan, Y. (2008). A structural approach for finding functional modules from large biological networks. *BMC Bioinformatics* 9:S19. doi:10.1186/1471-2105-9-S9-S19

Micale, G., Pulvirenti, A., Giugno, R., and Ferro, R. (2014a). Gasoline: a greedy and stochastic algorithm for optimal local multiple alignment of interaction networks. *PLoS ONE* 9:e98750. doi:10.1371/journal.pone.0098750

Micale, G., Pulvirenti, A., Giugno, R., and Ferro, R. (2014b). Proteins comparison through probabilistic optimal structure local alignment. *Front. Genet.* 5:302. doi:10.3389/fgene.2014.00302

Muller, A., Homey, B., Soto, H., Ge, N., Catron, D., Buchanan, M. E., et al. (2000). Involvement of chemokine receptors in breast cancer metastasis. *Nature* 410, 50–56. doi:10.1038/35065016

Nersisyan, L., Samsonyan, R., and Arakelyan, A. (2014). Cykeggparser: tailoring kegg pathways to fit into systems biology analysis workflows. *F1000Res.* 3, 145. doi:10.12688/f1000research.4410.2

Orchard, S., Ammari, M., Aranda, B., Brueza, L., Briganti, L., Broackes-Carter, F., et al. (2013). The mintact project-intact as a common curation platform for 11 molecular interaction databases. *Nucleic Acids Res.* 42, D358–D363. doi:10.1093/nar/gkt1115

Patil, A., Nakai, K., and Nakamura, H. (2011). Hitpredict: a database of quality assessed protein-protein interactions in nine species. *Nucleic Acids Res.* 39, D744–D749. doi:10.1093/nar/gkq897

Peri, S., Navarro, J., Kristiansen, T., Amanchy, R., Surendranath, V., Muthusamy, B., et al. (2004). Human protein reference database as a discovery resource for proteomics. *Nucleic Acids Res.* 32, D497–D501. doi:10.1093/nar/gkh070

Pro, B., and Dang, N. (2004). Cd26/dipeptidyl peptidase iv and its role in cancer. *Histol. Histopathol.* 19, 1345–1351.

Rajagopalan, D., and Agarwal, P. (2005). Inferring pathways from gene lists using a literature-derived network of biological relationships. *Bioinformatics* 21, 788–793. doi:10.1093/bioinformatics/bti069

Razick, S., Magklaras, G., and Donaldson, I. (2008). irefindex: a consolidated protein interaction database with provenance. *BMC Bioinformatics* 9:405. doi:10.1186/1471-2105-9-405

Rhrissorrakrai, K., and Gunsalus, K. (2011). Mine: module identification in networks. *BMC Bioinformatics* 12:192. doi:10.1186/1471-2105-12-192

Rustici, G., Kolesnikov, N., Brandizi, M., Burdett, T., Dylag, M., Emam, I., et al. (2013). Arrayexpress update-trends in database growth and links to data analysis tools. *Nucleic Acids Res.* 41, D987–D990. doi:10.1093/nar/gks1174

Sahraeian, S. M. E., and Yoon, B. (2013). Smetana: accurate and scalable algorithm for probabilistic alignment of large-scale biological networks. *PLoS ONE* 8:e67995. doi:10.1371/journal.pone.0067995

Shannon, P., Markiel, A., Ozier, O., Baliga, N., Wang, J., Ramage, D., et al. (2003). Cytoscape: a software environment for integrated models of biomolecular interaction networks. *Genome Res.* 13, 2498–2504. doi:10.1101/gr.1239303

Sohler, F., Hanisch, D., and Zimmer, R. (2004). New methods for joint analysis of biological networks and expression data. *Bioinformatics* 20, 1517–1521. doi:10.1093/bioinformatics/bth112

Sotiriou, C., Wirapati, P., Loi, S., Harris, A., Fox, S., Smeds, J., et al. (2006). Gene expression profiling in breast cancer: understanding the molecular basis of histologic grade to improve prognosis. *J. Natl. Cancer Inst.* 98, 262–272. doi:10.1093/jnci/djj052

Souiai, O., Becker, E., Prieto, C., Benkahla, A., De Las Rivas, J., and Brun, C. (2011). Functional integrative levels in the human interactome recapitulate organ organization. *PLoS ONE* 6:e22051. doi:10.1371/journal.pone.0022051

Stark, C., Breitkreutz, B., Reguly, T., Boucher, L., Breitkreutz, A., and Tyers, M. (2006). Biogrid: a general repository for interaction datasets. *Nucleic Acids Res.* 34, D535–D539. doi:10.1093/nar/gkj109

Su, A., Cooke, M., Ching, K., Hakak, Y., Walker, J., Wiltshire, T., et al. (2002). Large-scale analysis of the human and mouse transcriptomes. *Proc. Natl. Acad. Sci. U.S.A.* 99, 4465–4470. doi:10.1073/pnas.012025199

Su, A., Wiltshire, T., Batalov, S., Lapp, H., Ching, K., Block, D., et al. (2004). A gene atlas of the mouse and human protein-encoding transcriptomes. *Proc. Natl. Acad. Sci. U.S.A.* 101, 6062–6067. doi:10.1073/pnas.0400782101

Uhlen, M., Oksvold, P., Fagerberg, L., Lundberg, E., Jonasson, K., Forsberg, M., et al. (2010). Towards a knowledge-based human protein atlas. *Nat. Biotechnol.* 28, 1248–1250. doi:10.1038/nbt1210-1248

Xenarios, I., Salwinski, L., Duan, X., Higney, P., Kim, S., and Eisenberg, D. (2000). Dip: the database of interacting proteins. *Nucleic Acids Res.* 28, 289–291. doi:10.1093/nar/28.1.289

Xiao, X., Moreno-Moral, A., Rotival, M., Bottolo, L., and Petretto, E. (2014). Multi-tissue analysis of co-expression networks by higher-order generalized singular value decomposition identifies functionally coherent transcriptional modules. *PLoS Genet.* 10:e1004006. doi:10.1371/journal.pgen.1004006

Zhao, J., Lee, S., Huss, M., and Holme, P. (2012). The network organization of cancer-associated protein complexes in human tissues. *Sci. Rep.* 3, 1583. doi:10.1038/srep01583

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in
BIOENGINEERING AND BIOTECHNOLOGY

MINI REVIEW ARTICLE
published: 09 February 2015
doi: 10.3389/fbioe.2015.00012

Searching and indexing genomic databases via kernelization

Travis Gagie* and Simon J. Puglisi

Helsinki Institute for Information Technology (HIIT) and Department of Computer Science, University of Helsinki, Helsinki, Finland

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Thierry Lecroq, University of Rouen, France
Sebastian Wandelt, Humboldt University of Berlin, Germany

****Correspondence:***
Travis Gagie, Department of Computer Science, University of Helsinki, P.O. Box 68 (Gustaf Hällströmin katu 2b), Helsinki FI-00014, Finland
e-mail: travis.gagie@cs.helsinki.fi

The rapid advance of DNA sequencing technologies has yielded databases of thousands of genomes. To search and index these databases effectively, it is important that we take advantage of the similarity between those genomes. Several authors have recently suggested searching or indexing only one reference genome and the parts of the other genomes where they differ. In this paper, we survey the 20-year history of this idea and discuss its relation to kernelization in parameterized complexity.

Keywords: approximate pattern matching, data compression, genomic databases, indexing, kernelization, random-access reading, string algorithms

1. INTRODUCTION

The Human Genome Project took 13 years and three billion dollars to sequence a human genome, but the latest next-generation sequencing methods take only a few days and a few thousand dollars. With these methods, initiatives such as the 1000 Genomes Project and the 100,000 Genomes Project are now feasible. Advances in sequencing have far outstripped advances in computer processors and random-access memory, however, so it is increasingly challenging to make use of the data available. For example, while modern aligners can easily hold in memory the index for approximate pattern matching on a single human genome, they cannot handle thousands of human genomes. Schneeberger et al. (2009) proposed that we index the common parts of the genomes only once for them all, but we index the parts near variation sites for each genome. Ferrada et al. (2014b) suggested indexing the parts of all the genomes near boundaries between phrases in the LZ77 parse of the database. This is more general and may give better compression but requires the LZ77 parse, which is difficult to compute when the database does not fit in memory. Wandelt et al. (2013) proposed using a modified parse in which phrases must occur in a reference genome, which is easier to compute. (When papers have appeared in journals we cite those versions, although their chronological order may differ from that of previous versions.) Danek et al. (2014) recently showed that with this general approach we can store an index for approximate pattern matching on the database from the 1000 Genomes Project, in the memory of a commodity personal computer. This has so far not been possible with competing approaches, as surveyed by Vyverman et al. (2012).

When we are not given an upper bound on the pattern length, we can use one of the competing indexes that does not require such a bound or we can scan, with an online pattern-matching algorithm, the reference genome, and the parts of the other genomes near phrase boundaries. Wandelt and Leser (2012) and Rahn et al. (2014) proposed the latter idea specifically for approximate pattern matching in genomic databases, but the general approach has a 20-year history in the field of compressed pattern matching. In this paper, we survey that history and relate it to current research: in Section 2 we discuss some relevant data compression schemes and how they have been augmented to support fast random-access reading; in Section 3 we discuss how they have been used to speed up pattern-matching; in Section 4 we discuss how they have been used in compressed indexing. While writing this survey, we realized that scanning or indexing only parts of the database and then mapping the solution for those parts onto a solution for the whole database, is like kernelization in parameterized complexity (We note that kernels in parameterized complexity bear no relation to operating system kernels nor to kernels in machine learning.). We emphasize this perspective because we feel that computing a pattern-matching kernel is an interesting problem in itself, regardless of how we process it later, and deserving of further study. Of course, the nature and even the existence of the kernel depend on the problem we are trying to solve.

2. COMPRESSION WITH RANDOM-ACCESS READING

In general, the best compression of highly repetitive datasets is achieved with the LZ77 algorithm by Ziv and Lempel (1977). Suppose $S[1, \ldots, n]$ is a string with $S[n] = \$$, which is an end-of-file symbol that does not occur elsewhere in S. LZ77 works by parsing S into phrases such that, for each phrase $S[i, \ldots, j]$, $S[i, \ldots, j-1]$ occurs in $S[1, \ldots, j-2]$ but $S[i, \ldots, j]$ does not occur in $S[1, \ldots, j-1]$; that phrase is stored as a triple consisting of a pointer to $S[i, \ldots, j]$'s first occurrence in S (which is called the phrase's source), $j - i$, and $S[j]$. The LZ77 encoding of S takes $\mathcal{O}(z \log n)$ bits, where z is the number of phrases in the parse. For example, in the following verses vertical lines indicate phrase boundaries:

9|9-|b|o|t|t|l|e|s|-o|f|-be|e|r|-on|-t|h|e-|w|a|ll|-9|9-bottles-of-beer-
I|f-o|n|e-o|f-t|h|o|se|-bottles-s|hou|ld|-h|ap|pe|n-to|-f|all-
98|-bottles-of-beer-on-the-wall-

98|-bottles-of-beer-on-the-wall-98-bottles-of-beer-
I|f-one-of-those-bottles-should-happen-to-fall-
97|-bottles-of-beer-on-the-wall . . .

(We have displayed the verses with linebreaks to increase readability, but we have not considered them while computing the parse.) Although these verses may be annoyingly similar by the standards of natural language, they are far less similar than human genomes. Indeed, most repetitive biological datasets are much too similar (as well as much too large) for us to use them as informative examples.

One drawback of LZ77 compression is that reading a character in a compressed string can be very slow. Rytter (2003) and Charikar et al. (2005) showed how we can turn that parse into a balanced straight-line program (SLP) for S with $\mathcal{O}(z \log n)$ rules. An SLP for S is a context-free grammar in Chomsky normal form that generates S and only S; it is balanced if the height of each subtree in the parse tree is logarithmic in that subtree's size. It follows from Rytter's and Charikar et al.'s results that we can store S in $\mathcal{O}(z \log^2 n)$ bits and support random-access reading of any substring of S with length l in $\mathcal{O}(\log n + l)$ time. Verbin and Yu (2013) showed that this is nearly optimal in the worst case. Bille et al. (2011) showed how, given even an unbalanced SLP for S with r rules, we can store S in $\mathcal{O}(r \log n)$ bits and support random-access reading in $\mathcal{O}(\log n + l)$ time. Rytter's, Charikar et al.'s, and Bille et al.'s constructions are not practical, but there are practical grammar-based compressors, such as those by Larsson and Moffat (1999) and Maruyama and Tabei (2014). As far as we know, block graphs by Gagie et al. (2011) and Gagie et al. (2014c) are the most practical grammar-like representations for random-access reading. The LZ78 algorithm by Ziv and Lempel (1978) does not compress repetitive datasets as well as LZ77, but the LZ78 encoding of S can easily be augmented to support random-access reading in $\mathcal{O}(\log \log n + l)$ time. LZ78 also works by parsing S into phrases but then each phrase must extend a previous phrase plus one character. Because of this property, the LZ78 encoding of S has $\Omega(\sqrt{n})$ phrases, even when $S = a^n$.

In the example above, the first verse contains many phrase boundaries but the second verse contains only three. Kuruppu et al. (2010) proposed that, given a set of similar strings (or one string that can easily be divided into similar substrings), we store the first string in plain text as a reference and compress the others with a version of LZ77 that restricts phrases' sources to occur in the reference. They called this scheme Relative Lempel–Ziv (RLZ) and showed it compresses genomic databases very well in practice (although it too uses $\Omega(\sqrt{n})$ phrases, even when $S = a^n$) and there are several implementations of this approach, such as those by Deorowicz and Grabowski (2011), Kuruppu et al. (2012), and Ferrada et al. (2014a). Even when there is no obvious reference, Kuruppu et al. (2011) showed we can often build one by sampling the dataset: intuitively, if a substring is common then it is likely to appear in our sample, and if it is not then we lose little by not compressing it well; this can be formalized using results about SLPs.

3. SEARCHING

Farach and Thorup (1998) observed that the first occurrence of any pattern $P[1, \ldots, m]$ in S must cross or end at a phrase boundary in the LZ77 parse. Kärkkäinen and Ukkonen (1996) showed how, if we already know the locations of P's occurrences in S that cross or end at phrase boundaries, then we can deduce the locations of all its other occurrences from the structure of the parse. By the same arguments, LZ78 also has these properties and (Karpinski et al., 1997) simultaneously proved similar results for SLPs. Bille et al. (2009) observed that any substring of S within edit distance k of P (i.e., any of P's approximate matches) has length at most $m + k$, and any such substring that does not cross or end at an LZ78 phrase boundary must be an exact copy of an earlier one that does. They gave an algorithm for approximate pattern matching in LZ78 strings that works by extracting the $m + k$ and $m + k - 1$ characters before and after each LZ78 phrase boundary, respectively, using a technique similar to those discussed in Section 2; scanning the resulting substrings with any online algorithm for approximate pattern matching in uncompressed strings; and then deducing the locations of the other approximate matches from the structure of the parse.

Bille et al. (2011) extended this approach to show how we can find all P's approximate matches in S from an SLP for S. Recently, Gagie et al. (2014b) extended it further to show how we can preprocess the LZ77 parse of S in $\mathcal{O}(z \log n)$ time such that later, given P and k, we can find all P's occ approximate matches in $\mathcal{O}(z \min(mk, m + k^4) + occ)$ time. Their algorithm works by extracting the $m + k$ and $m + k - 1$ characters before and after each LZ77 phrase boundary, respectively, and then continuing as with the algorithm by Bille et al. (2009). The set of substrings we extract is like a kernel in parameterized complexity: the total length of the substrings can be much smaller than n, but a solution on them can quickly be mapped to a solution on all of S. For our example from Section 2 with $m = 4$ and $k = 1$, the kernel is:

99-bottles-of-beer-on-the-wall-99-bo
eer-If-one-of-those-bottles-should-happen-to-fall-98-bot
ll-98-bot
eer-If-on
ll-97-bot . . .

If we want a kernel consisting of only a single string, we can concatenate the substrings with $k + 1$ copies of \$ between each consecutive pair. Notice that if we are careful, we can avoid scanning the fourth substring "eer-If-on," since it occurs in the second substring.

We do not wish to leave the impression that kernelization is the only approach used in compressed pattern matching, nor even that the papers mentioned above are the only ones that use it. We have focused on those papers because we feel they are the most relevant to the practical bioinformatics papers by Wandelt and Leser (2012) and Rahn et al. (2014) mentioned in Section 1. Those authors were apparently unaware of the field of compressed pattern matching and re-invented kernelization specifically for approximate pattern matching in genomic databases, with kernels based on RLZ instead of LZ77, LZ78, or SLPs. This may be because the earlier researchers using kernelization for pattern matching did not perform convincing experiments on large enough datasets,

publicize their ideas in interdisciplinary forums or implement their ideas in tools usable by other scientists.

4. INDEXING

Kärkkäinen and Ukkonen (1996) gave the first LZ-based index, which supported exact pattern matching and stored S separately and uncompressed. They used Patricia trees and range reporting to find a set of candidate matches crossing or ending at LZ77 phrase boundaries; verified them by checking S; and then used more range reporting to find the other matches. We can obtain various time-space tradeoffs by compressing S and use the methods discussed in Section 1 to extract the characters needed to verify candidate matches. Claude and Navarro (2012) modified Kärkkäinen and Ukkonen's index to use a grammar-compressed encoding of S, and Kreft and Navarro (2013) modified it to use the encoding of S produced by a version of LZ77 they called LZ-End, which supports fast random-access reads starting at phrase boundaries. Arroyuelo et al. (2012) and Do et al. (2014) gave indexes based on LZ78 and RLZ, respectively, and Maruyama et al. (2013) and Takabatake et al. (2014) gave indexes based on the edit-sensitive parsing by Cormode and Muthukrishnan (2007). Gagie et al. (2014a) recently gave a version of Kärkkäinen and Ukkonen's index that uses a total of $\mathcal{O}(z \log^2 n)$ bits and returns the locations of all P's occ occurrences in S in $\mathcal{O}(m \log m + occ \log \log n)$ time. These indexes require no assumptions about the pattern.

Kärkkäinen and Sutinen (1998) gave an index based on a version of LZ77 that allows phrases to overlap by $q-1$ characters, where q is a parameter. If P has length exactly q, then their index returns the locations of all P's occurrences in S in optimal $\mathcal{O}(m + occ)$ time. If we are given an upper bound M on the pattern length at construction time, then even with Kärkkäinen and Ukkonen's original version, we need keep only a kernel of the text and can use $\mathcal{O}(z \log n + zM \log \sigma)$ bits in total, where σ is the size of the alphabet. We suspect this escaped investigation for so long because it seemed too obvious and inelegant to be theoretically interesting, and the need to index massive, highly repetitive datasets in practice has become pressing only since the development of next-generation sequencing methods.

The use of kernelization for indexing was eventually investigated by Schneeberger et al. (2009), although they did not present kernelization as a separate process because their work was application-driven. As noted in Section 1, they proposed that, given a database of genomes from the same species, we index the common parts of the genomes only once for them all, but we index the parts near variation sites for each genome. Wandelt et al. (2013) and Danek et al. (2014) gave similar results, essentially using a kernel based on the RLZ parse. Like Schneeberger et al., these authors indexed the kernels using specific methods based on q-grams or seeds. Danek et al.'s index for the database for the 1000 Genomes Project is the first one to fit in the memory of a commodity personal computer. Ferrada et al. (2014b) emphasized kernelization (albeit not under that name) in terms of the LZ77 parse, which is more general and may give better compression, and pointed out that we can use any index for approximate pattern matching to store the kernel. One point they did not comment on, and which we hope to have clarified in this paper, is that we can consider kernels based on LZ77, LZ78, RLZ, other compression schemes, or possibly other algorithms entirely. These kernels may be easier to compute when the database does not fit in memory, or have other useful properties that make them preferable in some situations. One interesting problem is how we can best maintain a dynamic kernel for an expanding database. This could allow us to align reads against a genomic database and then add the newly assembled genome, which could be useful when dealing with mutating cancer genomes or changing strains of a disease during an outbreak.

ACKNOWLEDGMENTS

Many thanks to Fabio Cunial, Paweł Gawrychowski, Szymon Grabowski, Juha Kärkkäinen, Veli Mäkinen, Gonzalo Navarro, Esa Pitkänen, Yasuo Tabei, and Niko Välimäki, for helpful discussions, and to the anonymous reviewers for their comments. *Funding:* The authors are funded by Academy of Finland grants 268324, 258308 and 250345 (CoECGR).

REFERENCES

Arroyuelo, D., Navarro, G., and Sadakane, K. (2012). Stronger Lempel-Ziv based compressed text indexing. *Algorithmica* 62, 54–101. doi:10.1007/s00453-010-9443-8

Bille, P., Fagerberg, R., and Gørtz, I. L. (2009). Improved approximate string matching and regular expression matching on Ziv-Lempel compressed texts. *ACM Trans. Algorithms* 6:3. doi:10.1145/1644015.1644018

Bille, P., Landau, G. M., Raman, R., Sadakane, K., Satti, S. R., and Weimann, O. (2011). "Random access to grammar-compressed strings," in *Proceedings of the 22nd Symposium on Discrete Algorithms (SODA)* (Philadephia: Society for Industrial and Applied Mathematics (SIAM)), 373–389.

Charikar, M., Lehman, E., Liu, D., Panigrahy, R., Prabhakaran, M., Sahai, A., et al. (2005). The smallest grammar problem. *IEEE Trans. Inf. Theory* 51, 2554–2576. doi:10.1109/TIT.2005.850116

Claude, F., and Navarro, G. (2012). "Improved grammar-based compressed indexes," in *Proceedings of the 19th Symposium on String Processing and Information Retrieval (SPIRE)* (Berlin: Springer-Verlag), 180–192.

Cormode, G., and Muthukrishnan, S. (2007). The string edit distance matching problem with moves. *ACM Trans. Algorithms* 3:2. doi:10.1145/1186810.1186812

Danek, D. A., Deorowicz, S., and Grabowski, S. (2014). Indexes of large genome collections on a PC. *PLoS ONE* 9:e109384. doi:10.1371/journal.pone.0109384

Deorowicz, S., and Grabowski, S. (2011). Robust relative compression of genomes with random access. *Bioinformatics* 27, 2979–2986. doi:10.1093/bioinformatics/btr505

Do, H. H., Jansson, J., Sadakane, K., and Sung, W. K. (2014). Fast relative Lempel-Ziv self-index for similar sequences. *Theor. Comp. Sci.* 532, 14–30. doi:10.1016/j.tcs.2013.07.024

Farach, M., and Thorup, M. (1998). String matching in Lempel-Ziv compressed strings. *Algorithmica* 20, 388–404. doi:10.1007/PL00009202

Ferrada, H., Gagie, T., Gog, S., and Puglisi, S. J. (2014a). "Relative Lempel-Ziv with constant-time random access," in *Proceedings of the 21st Symposium on String Processing and Information Retrieval (SPIRE)* (Berlin: Springer-Verlag), 13–17.

Ferrada, H., Gagie, T., Hirvola, T., and Puglisi, S. J. (2014b). Hybrid indexes for repetitive datasets. *Philos. Trans. R. Soc. A* 327, 2016. doi:10.1098/rsta.2013.0137

Gagie, T., Gawrychowski, P., Kärkkäinen, J., Nekrich, Y., and Puglisi, S. J. (2014a). "LZ77-based self-indexing with faster pattern matching," in *Proceedings of the 11th Latin American Symposium on Theoretical Informatics (LATIN)* (Berlin: Springer-Verlag), 731–742.

Gagie, T., Gawrychowski, P., and Puglisi, S. J. (2014b). Faster approximate pattern matching in compressed repetitive texts. *J. Discrete Algorithms.* doi:10.1016/j.jda.2014.10.003

Gagie, T., Hoobin, C., and Puglisi, S. J. (2014c). "Block graphs in practice," in *Proceedings of the 2nd International Conference on Algorithms for Big Data (ICABD)* (Aachen: CEUR-WS.org), 30–36.

Gagie, T., Gawrychowski, P., and Puglisi, S. J. (2011). "Faster approximate pattern matching in compressed repetitive texts," in *Proceedings of the 22nd International*

Symposium on Algorithms and Computation (ISAAC) (Berlin: Springer-Verlag), 653–662.

Kärkkäinen, J., and Sutinen, E. (1998). Lempel-Ziv index for *q*-grams. *Algorithmica* 21, 137–154. doi:10.1007/PL00009205

Kärkkäinen, J., and Ukkonen, E. (1996). "Lempel-Ziv parsing and sublinear-size index structures for string matching," in *Proceedings of the 3rd South American Workshop on String Processing (WSP)* (Ottawa: Carleton University Press), 141–155.

Karpinski, M., Rytter, W., and Shinohara, A. (1997). An efficient pattern-matching algorithm for strings with short descriptions. *Nordic J. Comput.* 4, 172–186.

Kreft, S., and Navarro, G. (2013). On compressing and indexing repetitive sequences. *Theor. Comp. Sci.* 483, 115–133. doi:10.1016/j.tcs.2012.02.006

Kuruppu, S., Beresford-Smith, B., Conway, T. C., and Zobel, J. (2012). Iterative dictionary construction for compression of large DNA data sets. *IEEE/ACM Trans. Comput. Biol. Bioinform.* 9, 137–149. doi:10.1109/TCBB.2011.82

Kuruppu, S., Puglisi, S. J., and Zobel, J. (2010). "Relative Lempel-Ziv compression of genomes for large-scale storage and retrieval," in *Proceedings of the 17th Symposium on String Processing and Information Retrieval (SPIRE)* (Berlin: Springer-Verlag), 201–206.

Kuruppu, S., Puglisi, S. J., and Zobel, J. (2011). "Reference sequence construction for relative compression of genomes," in *Proceedings of the 18th Symposium on String Processing and Information Retrieval (SPIRE)* (Berlin: Springer-Verlag), 420–425.

Larsson, N. J., and Moffat, A. (1999). "Offline dictionary-based compression," in *Proceedings of the Data Compression Conference (DCC)* (Hoboken, NJ: IEEE Press), 296–305.

Maruyama, S., Nakahara, M., Kishiue, N., and Sakamoto, H. (2013). ESP-index: a compressed index based on edit-sensitive parsing. *J. Discrete Algorithms* 18, 100–112. doi:10.1016/j.jda.2012.07.009

Maruyama, S., and Tabei, Y. (2014). "Fully online grammar compression in constant space," in *Proceedings of the Data Compression Conference (DCC)* (Hoboken, NJ: IEEE Press), 173–182.

Rahn, R., Weese, D., and Reinert, K. (2014). Journaled string tree – a scalable data structure for analyzing thousands of similar genomes on your laptop. *Bioinformatics* 30, 3499–3505. doi:10.1093/bioinformatics/btu438

Rytter, W. (2003). Application of Lempel-Ziv factorization to the approximation of grammar-based compression. *Theor. Comp. Sci.* 302, 211–222. doi:10.1016/S0304-3975(02)00777-6

Schneeberger, K., Hagmann, J., Ossowski, S., Warthmann, N., Gesing, S., Kohlbacher, O. D., et al. (2009). Simultaneous alignment of short reads against multiple genomes. *Genome Biol.* 10, R98. doi:10.1186/gb-2009-10-9-r98

Takabatake, Y., Tabei, Y., and Sakamoto, H. (2014). "Improved ESP-index: a practical self-index for highly repetitive texts," in *Proceedings of the 13th Symposium on Experimental Algorithms (SEA)* (Berlin: Springer-Verlag), 338–350.

Verbin, E., and Yu, W. (2013). "Data structure lower bounds on random access to grammar-compressed strings," in *Proceedings of the 24th Symposium on Combinatorial Pattern Matching (CPM)* (Berlin: Springer-Verlag), 247–258.

Vyverman, M., Baets, B. D., Fack, V., and Dawyndt, P. (2012). Prospects and limitations of full-text index structures in genome analysis. *Nucleic Acids Res.* 40, 6993–7015. doi:10.1093/nar/gks408

Wandelt, S., and Leser, U. (2012). "String searching in referentially compressed genomes," in *Proceedings of the Conference on Knowledge Discovery and Information Retrieval (KDIR)* (SciTePress), 95–102.

Wandelt, S., Starlinger, J., Bux, M., and Leser, U. (2013). "RCSI: scalable similarity search in thousand(s) of genomes," in *Proceedings of the VLDB Endowment*, Vol. 6 (San Jose, CA: VLDB Endowment), 1534–1545. doi:10.14778/2536258.2536265

Ziv, J., and Lempel, A. (1977). A universal algorithm for sequential data compression. *IEEE Trans. Inf. Theory* 23, 337–343. doi:10.1109/83.663496

Ziv, J., and Lempel, A. (1978). Compression of individual sequences via variable-rate coding. *IEEE Trans. Inf. Theory* 24, 530–536. doi:10.1109/TIT.1978.1055911

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 11 December 2014; accepted: 22 January 2015; published online: 09 February 2015.

*Citation: Gagie T and Puglisi SJ (2015) Searching and indexing genomic databases via kernelization. Front. Bioeng. Biotechnol. **3**:12. doi: 10.3389/fbioe.2015.00012*

This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology.

frontiers in Bioengineering and Biotechnology

REVIEW
published: 24 September 2015
doi: 10.3389/fbioe.2015.00143

Tandem repeats in proteins: prediction algorithms and biological role

*Marco Pellegrini**

Laboratory for Integrative Systems Medicine (LISM), Istituto di Informatica e Telematica, and Istituto di Fisiologia Clinica, Consiglio Nazionale delle Ricerche, Pisa, Italy

Tandem repetitions in protein sequence and structure is a fascinating subject of research which has been a focus of study since the late 1990s. In this survey, we give an overview on the multi-faceted aspects of research on protein tandem repeats (PTR for short), including prediction algorithms, databases, early classification efforts, mechanisms of PTR formation and evolution, and synthetic PTR design. We also touch on the rather open issue of the relationship between PTR and flexibility (or disorder) in proteins. Detection of PTR either from protein sequence or structure data is challenging due to inherent high (biological) signal-to-noise ratio that is a key feature of this problem. As early *in silico* analytic tools have been key enablers for starting this field of study, we expect that current and future algorithmic and statistical breakthroughs will have a high impact on the investigations of the biological role of PTR.

Keywords: proteins, tandem repeats, biological significance, protein TR detection algorithms, protein TR properties

OPEN ACCESS

Edited by:
John Hancock,
The Genome Analysis Centre, UK

Reviewed by:
Silvio C. E. Tosatto,
University of Padua, Italy
Michelle M. Simon,
Medical Research Council, UK

***Correspondence:**
Marco Pellegrini,
Istituto di Informatica e Telematica,
Consiglio Nazionale delle Ricerche,
Via Moruzzi 1, Pisa 56124, Italy
marco.pellegrini@iit.cnr.it

Specialty section:
This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology

Received: *12 June 2015*
Accepted: *07 September 2015*
Published: *24 September 2015*

Citation:
Pellegrini M (2015) Tandem repeats in proteins: prediction algorithms and biological role. Front. Bioeng. Biotechnol. 3:143. doi: 10.3389/fbioe.2015.00143

Introduction

A seminal paper (Andrade et al., 2001) reports the observation that repetitive subsequences that appear in tandem repetitions (TR) within the protein primary sequence often form integrated assemblies when these residues are mapped to their corresponding three-dimensional folded conformation. These TR confer multiple binding opportunities and may play a structural role by giving rigidity to a protein, and by exposing functional domains. Moreover, Andrade et al. (2001) remark that *tandem repeated structures* should not be assimilated to the traditional notions of *domains* and *motifs* that may appear singly or in multiple interspersed copies in each protein (while they can be repeated across families of protein), since they constitute a rather distinct class. They also remark that repeats in protein sequences are usually hard to detect because on average the repeating unit is relatively short, and moreover there can be considerable sequence divergence among units of the same TR. We will refer throughout this article to these repetitive sub-sequences as *Protein Tandem Repeats* (PTR or Protein-TR, for short).

A study by Marcotte et al. (1998) indicates that internal subsequence repetitions in protein primary structure are quite widespread. They have been detected in about 14% of all the then known proteins, with eukaryotic proteins being three times more as likely to have internal repeats than prokaryotic ones. More recent measurements in (Pellegrini et al., 2012) give a count of about 25% of the proteins in the Uniprot database (Apweiler et al., 2004) holding a PTR of length at least 20 aa.

A recent survey of some algorithmic aspects of PTR detection in protein sequences is in Luo and Nijveen (2014). In this survey, we will touch lightly on the multi-faceted aspects of PTR

research, including prediction algorithms, databases, early classification efforts, mechanisms of PTR formation and evolution, and synthetic PTR design. We also touch on the rather open issue of the relationship between PTR and flexibility (or disorder) in proteins.

Protein-TR Detection Algorithms Based on Sequence

Structural and functional properties of Protein-TR are often preserved also in presence of high divergence among the subsequences corresponding to the PTR units, both at the level of DNA coding sequence and at the level of AA sequence. This property makes automatic PTR detection a challenging task, and a variety of approaches have been implemented since the late 1990s. More recently, a tendency to integrating basic sequence data with evolutionary or biochemical annotations has emerged. **Table 1** reports the list of sequence-based algorithms.

Interestingly, early algorithms by Marcotte et al. (1998), Pellegrini et al. (1999), and Andrade et al. (2000) were instrumental to the first PTR classification efforts, while more recent tools have been aimed at providing web-server-based utilities, or at populating databases.

REP in Andrade et al. (2000) is one of the first PTR detection algorithms which uses a homology-based method to identify statistically significant protein repeats.

Other early methods developed for finding TRs in proteins are based on detecting sub-optimal alignments in the self-alignment matrix generated by the Smith-Waterman algorithm (or similar methods). Some methods developed along this line are `Internal Repeat Finder` (Marcotte et al., 1998; Pellegrini et al., 1999), `prospero` (Mott, 1999), RADAR (Heger and Holm, 2000), REPRO (Heringa and Argos, 1993; George and Heringa, 2000), and TRUST (Szklarczyk and Heringa, 2004). These methods often detect both tandem and interspersed repeats.

XSTREAM (Newman and Cooper, 2007) uses a seed expansion approach, while Jorda and Kajava (2009) proposed T-REKS, which uses a clustering approach based on k-means.

The systems HHrep (Soding et al., 2006) and HHRepID (Biegert and Soding, 2008) are instead based on building and matching Hidden Markov Models for the repeating substrings to be sought (not necessarily tandem).

Some approaches based on neural networks aim at detecting particular repetitive structures. For example, Palidwor et al. (2009) developed a classification technique for detecting alpharods repeats, a specific important repetitive structure [see also Rubinson and Eichman (2012)].

For the class of protein solenoid repeats, REPETITA, by Marsella et al. (2009), uses several AA biochemical properties (including polarity, secondary structure, molecular volume, electric charge, and codon diversity) and a discrete fourier transform approach to detect self-similarities.

Pellegrini et al. (2012) propose the notion of *fuzzy TR* (FTR) for proteins, which is based on using a normalized BLOSUM-weighted edit distance between AA sub-strings and in assuming that in a FTR, even if the constitutive unit elements may be pairwise at high divergence, there exists an "origin" string, not necessarily still part of the protein in exam, that is at a relatively small divergence from any of its unit elements. Here, the notion of high/low divergence is relative to the divergence between random AA strings under the chosen weighted edit distance. An exhaustive search of FTRs in long proteins is computationally demanding, since the bare definition leads to an NP-hard problem. Thus, an efficient heuristic is used in *PTRStalker* to guess the candidate "origin" strings.

Gruber et al. (2005) propose REPPER a meta searching approach that combines the output of different algorithms. A web-based meta-search server that allows to run and compare easily several tools on the same input is also described in Schaper et al. (2015).

Shapper et al. (Schaper et al., 2012; Anisimova et al., 2015) propose a statistical method based on phylogenetic fingerprints and ML-estimation that, in conjunction with one or more standard predictors, is able to filter out predicted TR that are more likely to be false-positive.

As screening large portions of protein sequence DB looking for TR patterns is time consuming, Richard and Kajava (2014)

TABLE 1 | Synthetic table of resources for PTR studies: sequence-based algorithms.

Name	Type	Year	Reference	Notes
INTREP	Alg	1999	Pellegrini et al. (1999)	http://nihserver.mbi.ucla.edu/Repeats/
prospero	Alg	1999	Mott (1999)	http://www.well.ox.ac.uk/rmott/ARIADNE/prospero.shtml
REP	Alg	2000	Andrade et al. (2000)	http://www.bork.embl.de/~andrade/papers/rep/search.html
RADAR	Alg	2000	Heger and Holm (2000)	https://github.com/AndreasHeger/radar/
REPRO	Alg	2000	George and Heringa (2000)	http://www.ibi.vu.nl/programs/reprowww/
TRUST	Alg	2004	Szklarczyk and Heringa (2004)	http://www.ibi.vu.nl/programs/trustwww/
REPPER	Alg	2005	Gruber et al. (2005)	http://toolkit.tuebingen.mpg.de/repper/
HHrep	Alg	2006	Soding et al. (2006)	http://toolkit.tuebingen.mpg.de/hhrep
TRED	Alg	2006	Sokol et al. (2007)	Available upon request
XSTREAM	Alg	2007	Newman and Cooper (2007)	http://jimcooperlab.mcdb.ucsb.edu/xstream/
HHRepID	Alg	2008	Biegert and Soding (2008)	http://toolkit.tuebingen.mpg.de/hhrepid/
ARD2	Alg	2009	Palidwor et al. (2009)	http://cbdm.mdc-berlin.de/~ard2/
T-REKS	Alg	2009	Jorda and Kajava (2009)	http://bioinfo.montp.cnrs.fr/
REPETITA	Alg	2009	Marsella et al. (2009)	http://protein.bio.unipd.it/repetita/
PTRStalker	Alg	2012	Pellegrini et al. (2012)	http://bioalgo.iit.cnr.it/
TRDistiller	Alg	2014	Richard and Kajava (2014)	Available upon request

propose a pre-screening tool (`TRDistiller`) whose purpose is to quickly filter out proteins that almost surely do not contain a TR, while retaining for further analysis the proteins carrying a TR with high probability.

As the list of possible tools to choose from becomes longer, there is an emerging need for guidance on which tool is most suitable for a given task. Unfortunately, at the best of my knowledge, no such comprehensive comparative study has been attempted yet. More limited comparative tests can be found in Pellegrini et al. (2012) where five methods (RADAR, TRUST, T-REKS, XSTREAM, and PTRStalker) are compared in their ability to detect very long PTRs ($\geq$ 4000 AA), with XSTREAM and PTRStalker emerging as the best choice for this task. A second test is aimed at detecting dimeric proteins by five tools (RADAR, TRUST, HHRep, HHRepID, and PTRStalker), with PTRStalker, TRUST, and HHRepID being able to successfully uncover such dimeric structures in some of the tested proteins. In Jorda and Kajava (2009), four methods (T-REKS, XSTREAM, `Internal Repeat Finder`, and TRED) are compared by the number of sequences they could identify as holding a PTR longer than 14 AA in the SWISSPROT database, with T-REKS giving the highest number (almost doubling the closest competitor). In Marsella et al. (2009), three methods (REPETITA, TRUST, and RADAR) are compared to assess their ability in guessing the correct periodicity in solenoid repeats, with REPETITA having an edge over the other two methods.

Protein-TR Detection Algorithms Based on Structure

Functional features are more readily linked to the structural features of a protein rather than to their primary sequence, thus available structural data should also be used to detect protein 3d symmetries and repetitive 3d motifs (Goodsell and Olson, 2000). However, only for a fraction of the known protein sequences, the corresponding 3D conformation could be determined, therefore the range of applicability of structure-based methods is limited w.r.t. the range of the sequence-based methods.

In this case, the algorithmic challenge lies in the multidimensional nature of the data, and on the fact that the space of rigid transformations (rotations, translations) as well as the inherent flexibility of proteins must be taken into account when attempting to match 3d substructures in order to detect the PTR periodicity. **Table 2** reports the list of structure-based algorithms.

In Murray et al. (2002), both the sequence and the structure signals are integrated within a continuous wavelet transform approach to detect repeating motifs. In particular, the sequence is represented by values of the Kyte–Doolittle hydrophobicity scale, while structure is characterized via the relative accessible surface area. This approach has been shown to be successful on most of the well known types of repetitive motifs.

DAVROS (Murray et al., 2004) is a PTR prediction system that builds upon a structural alignment program (SAP) that evaluates internal structural symmetries via a protein self-similarity matrix and employs a Fourier Transform approach to identify strong signals over the noisy background.

Swelfe (Abraham et al., 2008) finds internal repeats by combining three abstraction levels. Swelfe quickly identifies statistically significant internal repeats in DNA sequence, in the amino acid sequence and in the 3D structures using dynamic programing. The associated web server also shows the relationships between repeating feature at each level and facilitates visualization of the results.

SymD (Kim et al., 2010) is an algorithm that aims at detecting internal spatial symmetries of proteins. It uses the alignment method in Kim et al. (2009) on pairs of structure formed by the target protein and its shifted versions built by all circular permutations of its residues. Although not all PTR give rise to symmetric 3D structures, many do, therefore this approach often indicates the presence of a PTR. Other methods based on this symmetry detection approach are RQA (Chen et al., 2009), OPAAS (Shih and Hwang, 2004), and Gplus (Guerler et al., 2009).

ProSTRIP (Sabarinathan et al., 2010) uses dynamic programing to find similar structural repeats in a protein structure encoded by the protein backbone dihedral angles.

RAPHAEL (Walsh et al., 2012b) is a more recent method for the detection of solenoids in protein structures. It aims at mimicking the periodicity and distance patterns detection criteria a human curator is likely to exploit when assessing the presence of a solenoid visually. In particular, the candidate protein is subject to a random rotation and translation, and subsequently for each of the three C-alpha coordinates a projection is performed. This operation produces a profile curve, in which the distance between consecutive local maxima is a candidate periodicity value. By averaging over multiple random rotations and translations, a robust

TABLE 2 | Synthetic table of resources for PTR studies: structure-based algorithms.

Name	Type	Year	Reference	Notes
DAVROS	Alg	2004	Murray et al. (2004)	http://www.ebi.ac.uk/~murray/davros/
OPAAS	Alg	2004	Shih and Hwang (2004)	http://www.ibms.sinica.edu.tw/
Swelfe	Alg	2008	Abraham et al. (2008)	http://wwwabi.snv.jussieu.fr/public/Swelfe/
RQA	Alg	2009	Chen et al. (2009)	
Gplus	Alg	2009	Guerler et al. (2009)	http://agknapp.chemie.fu-berlin.de/gplus/
SymD	Alg	2010	Kim et al. (2010)	http://symd.nci.nih.gov/
ProSTRIP	Alg	2010	Sabarinathan et al. (2010)	http://cluster.physics.iisc.ernet.in/prostrip/
RAPHAEL	Alg	2012	Walsh et al. (2012b)	http://protein.bio.unipd.it/raphael/
Frustratometer	Alg	2013	Parra et al. (2013)	http://www.proteinphysiologylab.tk/
ConSole	Alg	2014	Hrabe and Godzik (2014)	http://console.sanfordburnham.org/
PRIGSA	Alg	2014	Chakrabarty and Parekh (2014)	http://bioinf.iiit.ac.in/PRIGSA/

period estimation is attained. Additional simple rules allow to further detect non-periodic residues interspersed in the solenoid periodic structure.

Parra et al. (2013) use the structural alignment tool TopMatch (Sippl, 2008) to search exhaustively the space of possible sub-structures that tile a large fraction of a given structure, and thus can represent a *bona fide* structural repetitive element of the input protein.

PRIGSA (Chakrabarty and Parekh, 2014) represents distance information among residues in an adjacency matrix, and it is based on the observation that similar sub-structures can be recognized as unique profiles of the principal eigenspectra of this matrix.

ConSole (Hrabe and Godzik, 2014) aims at detecting solenoid domains having as input structural information, by searching repetitive patterns in a *contact matrix*, which, for every pair of residues *i,j* in a protein, encodes a value 1 if the two residues have at least a pair of heavy atoms at Euclidean distance below a threshold t (set at $t = 4.5$ Å). *Ad hoc* rules are further applied in order to handle insertions in the solenoid repetitive patterns.

As in the case of sequence-based methods, very few comparative studies among the proposed structure-based tools have been done. In Kim et al. (2010), six methods (DAVROS, OPAAS, Swelfe, RQA, Gplus, and SymD) are compared in their ability to identify characteristic symmetries in fold families from CATH, SCOP, and ASTRAL databases, with SymD having an overall better performance. In Sabarinathan et al. (2010), two methods (ProSTRIP and Swelfe) are compared over well known families of repeat proteins, for the task of detecting periodicity and exact repeat positions. On well known PTR proteins, both methods detect approximatively the correct period, however, ProSTRIP detects more repeating units. On the harder class of multidomain proteins ProSTRIP is also better at guessing the correct periodicity. In Walsh et al. (2012b) five methods (both sequence and structure based) are compared (namely Swelfe, RAPHAEL, REPETITA, TRUST, and RADAR) in their ability to guess the PTR periodicity, with RAPHAEL giving better predictions, when we allow for a slackness of 5 AA in the predicted value. For exact predictions, RAPHAEL, REPETITA, and TRUST are about equivalent.

Databases for Protein-TR

Information about PTR can be retrieved as annotations in general purpose integrated protein databases. However, such annotations often cover only the well studied PTR, therefore in recent years a number of special purpose repositories have been assembled with the objective of making large scale PTR analysis easier. We list here in **Table 3** only DBs that are available on-line at the present time, as many older published articles refer to DBs no longer available.

RepSeq (Depledge et al., 2007) is a specialized DB for PTR in lower eukaryotic pathogens.

PRDB is a PTR database that supports queries on protein tandem repeats found in sequence data bases. Currently, it holds about 1.25M PTR extracted from the Swissprot, PDB, and NR databases in early 2010 using the T-REKS detection tool (Jorda et al., 2012). This database has been instrumental for uncovering original biological correlations in Jorda et al. (2010).

TABLE 3 | Synthetic table of resources for PTR studies: databases.

Name	Type	Year	Reference	Notes
RepSeq	DB	2007	Depledge et al. (2007)	http://www.repseq.org/
PRDB	DB	2012	Jorda et al. (2012)	http://bioinfo.montp.cnrs.fr/
ProRepeat	DB	2012	Luo et al. (2012)	http://prorepeat.bioinformatics.nl/
PTRStalkerDB	DB	2012	Pellegrini et al. (2012)	http://bioalgo.iit.cnr.it/
RepeatsDB	DB	2013	Di Domenico et al. (2013)	http://repeatsdb.bio.unipd.it/

PTRStalkerDB lists the PTR found with the PTRStalker method on the SwissProt database release 57.15 of March 2, 2010 that contains 515,203 sequence entries.

ProRepeat (Luo et al., 2012) is a curated and integrated data base and analysis platform for research on the biological features of amino acid tandem repeats. ProRepeat collects PTR of protein sequences listed in the UniProt knowledge base from different species; moreover, it includes 85 completely sequenced eukaryotic proteomes from the RefSeq collection. The latest datasets used in ProRepeat are UniProtKB Release May 2011 and RefSeq Release 40.

RepeatsDB (Di Domenico et al., 2013) is a database of annotated tandem repeat protein *structures* that uses both a state of the art detection method (RAPHAEL) and manual curation to survey the protein structures listed in PDB. The latest version 2.0.0 (beta) released in 2015 holds 10,039 PTR structures (including manually classified and predicted PTR). Automated updates every 3 months are planned.

Although progress in the area of databases for PTR has come about in the past few years, there is also much scope for improvement, in particular, as the amount of proteomic data increases rapidly, it is important to maintain the PTR databases aligned with the latest releases of the reference protein sequence and structure. Also, given the variety of algorithms and approaches to PTR prediction, DB that uses one single algorithm as source of data could suffer for the specific algorithm's biases, and more robust prediction could be obtained instead by using multiple detecting algorithms.

Classification of Protein-TR

Kajava (2012) reports an extensive survey of bioinformatic tools to support various analysis of TR in proteins, including tools for identification of TR in proteins, databases reporting PTR (either exclusively, or as an annotation in a larger protein DB), classification of repetitive 3D structures, and tools for structural prediction targeting proteins with PTR (as opposed to globular ones).

Early surveys by Marcotte et al. (1998), Andrade et al. (2001), and Kajava (2001) are very much concerned with the task of identifying specific classes of proteins highly characterized by their PTR content with the aim of finding corresponding structural and functional regularities. Andrade et al. (2001) propose a taxonomy of six main classes (β-propellers, β-trefoils,

TPR-like, Ankyrin-like, Armadillo/HEAT-like and Leucine-Rich). Instead Kajava (2001) uses a classification based on the repeating unit length (1–2 residues = class I crystalline aggregates, 3–4 residues = class II fibrous proteins, 5–40 residues = class III solenoid-like proteins, and class IV beads-on-string proteins with repeats longer than 30 residues folded into globular domains). Later in Kajava (2012), a refinement of this classification by splitting class III into two sub-classes of *solenoid* and *non-solenoid* structures has been proposed. The database RepeatsDB (Di Domenico et al., 2013) uses the classification proposed by Kajava (2012).

Mechanisms of Protein-TR Expansion During Evolution

Björklund et al. (2006) and Moore et al. (2008) analyze the internal sequence similarity in proteins of several species and note that the domain repeats are often expanded through simultaneous duplications of several domains in one event, while the duplication of one domain at a time is a less common event. Moreover, many of the repeats appear to have been duplicated in the middle of the repeat region. This behavior is in contrast to the evolution of other proteins that mainly happens through additions of single domains at either terminus of the protein. No common mechanism for the expansion of all repeats could be detected in this study, for example, duplication patterns show no dependence on the size of the domains. Repeat expansion in some families can possibly be explained by shuffling of exons but exon shuffling does not appear to be a general formation mechanism.

Some domain families show distinct specific duplication patterns, for example, nebulin domains have mainly been expanded with groups of seven domains at a time, while duplications of other domain families involve varying numbers of domains for each event. A more detailed analysis of nebulin domains evolution is in Björklund et al. (2010).

By mapping the Protein TR back onto their coding DNA sequences, Street et al. (2006) study the conservation of intron/exon patterns across several species and show evidence that subdivide the repeat protein genes into two classes. The first class has random-length exons that are likely produced by accumulating introns though random insertion within the array of repetitive units. The second class is composed exclusively of exons corresponding to the multiple of the repeating unit, and thus is likely to be formed by local duplications of intron/exon modules.

Protein-TR Evolutionary Conservation

In Schaper et al. (2014), it is described a proteome-wide analysis of the evolution of TR in human proteins, using a database of 61 eukaryotes. The main finding is that the vast majority of human PTR are ancient, with TR unit number and order preserved intact since remote speciation events. Moreover, no human PTR shows evidence of a recent duplication or deletion event. Thus, presumably, most PTRs fold into stable and conserved structures, indispensable for their function. Similar findings for plants are shown in Schaper and Anisimova (2015). The analysis of PTR in *Drosophila melanogaster* reported in Ponting et al. (2001) led to the identification of novel PTR in the products of disease-related human genes homologous to those in *Drosophila melanogaster*.

Protein-TR in Protein Design

Different structures which arise from tandem arrays of a repeated structural motif have generated significant interest with respect to protein engineering and synthetic protein design (Forrer et al., 2003, 2004; Main et al., 2003, 2005; Javadi and Itzhaki, 2013). Several results are reported in these articles about re-engineering of PTR binding specificities, with attention to protein folding kinetics and protein stability.

Sawyer et al. (2013) present a "module-based" design approach in which modules composed of tandem repeats are aligned to identify repeat-specific features that will be important to include in future repeat protein design templates.

Parmeggiani et al. (2015) describe a general database-driven approach for reliable generation of synthetic stable modular repeat proteins. Concomitant to the distillation of general design principles for PTR engineering, research activities have been also concentrated toward specific classes of Protein-TR which have shown a more promising potential for applications (Stumpp et al., 2015). A notable example is that of *Designed Ankyrin Repeat Proteins (DARPins)* (Binz2003) that have been extensively studied [see a recent survey by Plückthun (2015) and references therein], since they provide a biochemically stable scaffold for designing protein variants able to recognize targets with affinity and specificity that are equal or possibly superior to that of antibodies. Similar promising studies focus also on *armadillo repeat proteins* (Reichen et al., 2014) and *leucine-rich-repeat proteins* (Park et al., 2015).

Order, Disorder, and Protein-TR

While our view of protein functions is often linked to the presence of a well defined 3-dimensional protein conformations, it has been recognized (Tompa, 2002) that many important protein functions are also linked to proteins (or regions within a protein) that lack a folded structure, but display a highly flexible random-coil-like conformation under physiological conditions [named intrinsically unstructured proteins (IUP) or intrinsically unstructured regions (IUR)].

The concept of order and disorder in protein segments (Dunker et al., 2001; Tompa, 2002) has been often investigated in correlation with the presence or absence of PTR at the sequence level. For example in Tompa (2002), 21 IUP are examined, and further 21 cases are cited in Dunker et al. (2001). It is noticed that IUR often correspond to regions of low compositional complexity (low sequence entropy) and sometimes to repetitive sub-sequences in fibrillar proteins. Tompa and Fersht (2009) discuss in detail the cases of PTR in PEVK regions of human Titin, in prion proteins and in the CTD domain of RNA polymerase. These findings on specific instances are, however, hard to generalize.

A general property observed by Jorda et al. (2010) is that higher level of repeat perfection correlates positively with the disordered state of protein sub-chains.

The emergence of IUP/IUR prediction tools, such as IUpred (Dosztányi et al., 2005), ESpritz (Walsh et al., 2012a), and DISOPRED (Jones and Cozzetto, 2015), to name a few, and

comprehensive databases of IUP/IUR, such as DisProt (Sickmeier et al., 2007) and MobiDB 2.0 (Potenza et al., 2015), can be quite useful for finding generalizable connections between PTR and ordered/disordered states of protein regions.

Correlation of Protein-TR with Other Protein Properties

In Turutina et al. (2006), the sequences of the *Swiss-Prot* protein families are analyzed in order to detect family-specific latent periodicity fingerprints induced by PTR, using the method in Korotkov et al. (2003), and 94 such protein families are reported as well-characterized by such fingerprints.

A complete analysis of PDB sequences using RADAR is reported in Rajathei and Selvaraj (2013), where a good correlation among PTR, structural similarity, and functionally involved residues is highlighted.

In Mularoni et al. (2007) and Mularoni et al. (2010), the function and evolution of a particular class of PTR formed by repetitions of a *single AA* are investigated (homo-TR). These two studies concentrated on human and mouse homo-TR of length four. The protein stabilizing properties of homo-TR are also reported in Katti et al. (2000). A more general statistical analysis of homo-PTR in human proteins is in Jorda and Kajava (2010).

Conclusion

The present survey on Protein-TR touches several aspects of this research fields, including detection algorithms (Sections "Protein-TR Detection Algorithms Based on Sequence" and "Protein-TR Detection Algorithms Based on Structure"), databases (Section "Databases for Protein-TR"), classification (Section "Classification of Protein-TR"), the relationship between PTR and biologically relevant concepts (Sections "Mechanisms of Protein-TR Expansion During Evolution," "Protein-TR Evolutionary Conservation," "Order, Disorder, and Protein-TR," and "Correlation of Protein-TR with Other Protein Properties"), and it highlights also recent progress in the design of synthetic PTR (Section "Protein-TR in Protein Design").

Although there has been steady progress in the last 15 years in devising new prediction tools, both sequence and structure based, very little comparative or integrative work has been done. Most of the proteome-wise studies use only one tool to define and detect PTR and draw conclusions on PTR distributions and statistics. Though this approach was completely justified in the pioneering times (late 1990s and early 2000s), it is necessary now to refine these methodologies and make full use of the wealth of algorithms and approaches devised in the last decade. A more robust assessment of the distribution and annotations of PTR over the entire proteome could be attained by applying and merging the outcomes of multiple tools. In this context, the manually curated databases of PTRs can provide the necessary validation benchmarks.

From the point of view of the design of prediction tools, one open challenge is to devise sequence-based tools that are able to come close to the performance of structure-based tools. Thus providing higher quality PTR predictions for a larger pool of sequenced proteins.

Funding

The present work is partially supported by the Flagship project InterOmics (PB. P05), funded by the Italian Ministry for Instruction University and Research (MIUR) and CNR organizations, and by the joint IIT-IFC Laboratory of Integrative Systems Medicine (LISM).

References

Abraham, A.-L., Rocha, E. P. C., and Pothier, J. (2008). Swelfe: a detector of internal repeats in sequences and structures. *Bioinformatics* 24, 1536–1537. doi:10.1093/bioinformatics/btn234

Andrade, M. A., Perez-Iratxeta, C., and Ponting, C. P. (2001). Protein repeats: structures, functions, and evolution. *J. Struct. Biol.* 134, 117–131. doi:10.1006/jsbi.2001.4392

Andrade, M. A., Ponting, C. P., Gibson, T. J., and Bork, P. (2000). Homology-based method for identification of protein repeats using statistical significance estimates. *J. Mol. Biol.* 298, 521–537. doi:10.1006/jmbi.2000.3684

Anisimova, M., Pečerska, J., and Schaper, E. (2015). Statistical approaches to detecting and analyzing tandem repeats in genomic sequences. *Front. Bioeng. Biotechnol.* 3:31. doi:10.3389/fbioe.2015.00031

Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., et al. (2004). Uniprot: the universal protein knowledgebase. *Nucleic Acids Res.* 32(Suppl. 1), D115–D119. doi:10.1093/nar/gkh131

Biegert, A., and Soding, J. (2008). De novo identification of highly diverged protein repeats by probabilistic consistency. *Bioinformatics* 24, 807–814. doi:10.1093/bioinformatics/btn039

Björklund, A. K., Light, S., Sagit, R., and Elofsson, A. (2010). Nebulin: a study of protein repeat evolution. *J. Mol. Biol.* 402, 38–51. doi:10.1016/j.jmb.2010.07.011

Björklund, ÅK., Ekman, D., and Elofsson, A. (2006). Expansion of protein domain repeats. *PLoS Comput. Biol.* 2:e114. doi:10.1371/journal.pcbi.0020114

Chakrabarty, B., and Parekh, N. (2014). Prigsa: protein repeat identification by graph spectral analysis. *J. Bioinform. Comput. Biol.* 12, 1442009. doi:10.1142/S0219720014420098

Chen, H., Huang, Y., and Xiao, Y. (2009). A simple method of identifying symmetric substructures of proteins. *Comput. Biol. Chem.* 33, 100–107. doi:10.1016/j.compbiolchem.2008.07.026

Depledge, D. P., Lower, R. P., and Smith, D. F. (2007). Repseq – a database of amino acid repeats present in lower eukaryotic pathogens. *BMC Bioinformatics* 8:122. doi:10.1186/1471-2105-8-122

Di Domenico, T., Potenza, E., Walsh, I., Gonzalo Parra, R., Giollo, M., Minervini, G., et al. (2013). Repeatsdb: a database of tandem repeat protein structures. *Nucleic Acids Res.* 42, D352–D357. doi:10.1093/nar/gkt1175

Dosztányi, Z., Csizmok, V., Tompa, P., and Simon, I. (2005). Iupred: web server for the prediction of intrinsically unstructured regions of proteins based on estimated energy content. *Bioinformatics* 21, 3433–3434. doi:10.1093/bioinformatics/bti541

Dunker, A., Lawson, J., Brown, C. J., Williams, R. M., Romero, P., Oh, J. S., et al. (2001). Intrinsically disordered protein. *J. Mol. Graph. Model.* 19, 26–59. doi:10.1016/S1093-3263(00)00138-8

Forrer, P., Binz, H. K., Stumpp, M. T., and Plückthun, A. (2004). Consensus design of repeat proteins. *Chembiochem* 5, 183–189. doi:10.1002/cbic.200300762

Forrer, P., Stumpp, M. T., Binz, H., and Plückthun, A. (2003). A novel strategy to design binding molecules harnessing the modular nature of repeat proteins. *FEBS Lett.* 539, 2–6. doi:10.1016/S0014-5793(03)00177-7

George, R., and Heringa, J. (2000). The repro server: finding protein internal sequence repeats through the web. *Trends Biochem. Sci.* 25, 515–517. doi:10.1016/S0968-0004(00)01643-1

Goodsell, D. S., and Olson, A. J. (2000). Structural symmetry and protein function. *Annu. Rev. Biophys. Biomol. Struct.* 29, 105–153. doi:10.1146/annurev.biophys.29.1.105

Gruber, M., Soding, J., and Lupas, A. N. (2005). REPPER-repeats and their periodicities in fibrous proteins. *Nucleic Acids Res.* 33(Suppl._2), W239–W243. doi:10.1093/nar/gki405

Guerler, A., Wang, C., and Knapp, E.-W. (2009). Symmetric structures in the universe of protein folds. *J. Chem. Inf. Model.* 49, 2147–2151. doi:10.1021/ci900185z

Heger, A., and Holm, L. (2000). Rapid automatic detection and alignment of repeats in protein sequences. *Proteins* 41, 224–237. doi:10.1002/1097-0134(20001101)41:2<224::AID-PROT70>3.0.CO;2-Z

Heringa, J., and Argos, P. (1993). A method to recognize distant repeats in protein sequences. *Proteins* 17, 391–411. doi:10.1002/prot.340170407

Hrabe, T., and Godzik, A. (2014). Console: using modularity of contact maps to locate solenoid domains in protein structures. *BMC Bioinformatics* 15:119. doi:10.1186/1471-2105-15-119

Javadi, Y., and Itzhaki, L. S. (2013). Tandem-repeat proteins: regularity plus modularity equals design-ability. *Curr. Opin. Struct. Biol.* 23, 622–631. doi:10.1016/j.sbi.2013.06.011

Jones, D. T., and Cozzetto, D. (2015). Disopred3: precise disordered region predictions with annotated protein-binding activity. *Bioinformatics* 31, 857–863. doi:10.1093/bioinformatics/btu744

Jorda, J., Baudrand, T., and Kajava, A. V. (2012). Prdb: protein repeat database. *Proteomics* 12, 1333–1336. doi:10.1002/pmic.201100534

Jorda, J., and Kajava, A. V. (2009). T-REKS: identification of Tandem REpeats in sequences with a K-meanS based algorithm. *Bioinformatics* 25, 2632–2638. doi:10.1093/bioinformatics/btp482

Jorda, J., and Kajava, A. V. (2010). "Protein homorepeats: sequences, structures, evolution, and functions," in *Advances in Protein Chemistry and Structural Biology*, Vol. 79, ed. A. McPherson (Waltham, MA: Academic Press), 59–88.

Jorda, J., Xue, B., Uversky, V. N., and Kajava, A. V. (2010). Protein tandem repeats: the more perfect, the less structured. *FEBS J.* 277, 2673–2682. doi:10.1111/j.1742-4658.2010.07684.x

Kajava, A. V. (2001). Review: proteins with repeated sequencestructural prediction and modeling. *J. Struct. Biol.* 134, 132–144. doi:10.1006/jsbi.2000.4328

Kajava, A. V. (2012). Tandem repeats in proteins: from sequence to structure. *J. Struct. Biol.* 179, 279–288. doi:10.1016/j.jsb.2011.08.009

Katti, M. V., Sami-Subbu, R., Ranjekar, P. K., and Gupta, V. S. (2000). Amino acid repeat patterns in protein sequences: their diversity and structural-functional implications. *Protein Sci.* 9, 1203–1209. doi:10.1110/ps.9.6.1203

Kim, C., Basner, J., and Lee, B. (2010). Detecting internally symmetric protein structures. *BMC Bioinformatics* 11:303. doi:10.1186/1471-2105-11-303

Kim, C., Tai, C.-H., and Lee, B. (2009). Iterative refinement of structure-based sequence alignments by seed extension. *BMC Bioinformatics* 10:210. doi:10.1186/1471-2105-10-210

Korotkov, E. V., Korotkova, M. A., and Kudryashov, N. A. (2003). Information decomposition method to analyze symbolical sequences. *Phys. Lett. A* 312, 198–210. doi:10.1016/S0375-9601(03)00641-8

Luo, H., Lin, K., David, A., Nijveen, H., and Leunissen, J. A. M. (2012). Prorepeat: an integrated repository for studying amino acid tandem repeats in proteins. *Nucleic Acids Res.* 40, D394–D399. doi:10.1093/nar/gkr1019

Luo, H., and Nijveen, H. (2014). Understanding and identifying amino acid repeats. *Brief. Bioinformatics* 15, 582–591. doi:10.1093/bib/bbt003

Main, E. R., Jackson, S. E., and Regan, L. (2003). The folding and design of repeat proteins: reaching a consensus. *Curr. Opin. Struct. Biol.* 13, 482–489. doi:10.1016/S0959-440X(03)00105-2

Main, E. R., Lowe, A. R., Mochrie, S. G., Jackson, S. E., and Regan, L. (2005). A recurring theme in protein engineering: the design, stability and folding of repeat proteins. *Curr. Opin. Struct. Biol.* 15, 464–471. doi:10.1016/j.sbi.2005.07.003

Marcotte, E. M., Pellegrini, M., Yeates, T. O., and Eisenberg, D. (1998). A census of protein repeats. *J. Mol. Biol.* 293, 151–160. doi:10.1006/jmbi.1999.3136

Marsella, L., Sirocco, F., Trovato, A., Seno, F., and Tosatto, S. C. (2009). Repetita: detection and discrimination of the periodicity of protein solenoid repeats by discrete fourier transform. *Bioinformatics* 25, i289–i295. doi:10.1093/bioinformatics/btp232

Moore, A. D., Bjorklund, A. K., Ekman, D., Bornberg-Bauer, E., and Elofsson, A. (2008). Arrangements in the modular evolution of proteins. *Trends Biochem. Sci.* 33, 444–451. doi:10.1016/j.tibs.2008.05.008

Mott, R. (1999). Local sequence alignments with monotonic gap penalties. *Bioinformatics* 15, 455–462. doi:10.1093/bioinformatics/15.6.455

Mularoni, L., Ledda, A., Toll-Riera, M., and Albà, M. M. (2010). Natural selection drives the accumulation of amino acid tandem repeats in human proteins. *Genome Res.* 20, 745–754. doi:10.1101/gr.101261.109

Mularoni, L., Veitia, R. A., and Albà, M. M. (2007). Highly constrained proteins contain an unexpectedly large number of amino acid tandem repeats. *Genomics* 89, 316–325. doi:10.1016/j.ygeno.2006.11.011

Murray, K. B., Gorse, D., and Thornton, J. M. (2002). Wavelet transforms for the characterization and detection of repeating motifs. *J. Mol. Biol.* 316, 341–363. doi:10.1006/jmbi.2001.5332

Murray, K. B., Taylor, W. R., and Thornton, J. M. (2004). Toward the detection and validation of repeats in protein structure. *Proteins* 57, 365–380. doi:10.1002/prot.20202

Newman, A., and Cooper, J. (2007). Xstream: a practical algorithm for identification and architecture modeling of tandem repeats in protein sequences. *BMC Bioinformatics* 8:382. doi:10.1186/1471-2105-8-382

Palidwor, G. A., Shcherbinin, S., Huska, M. R., Rasko, T., Stelzl, U., Arumughan, A., et al. (2009). Detection of alpha-rod protein repeats using a neural network and application to huntingtin. *PLoS Comput. Biol.* 5:e1000304. doi:10.1371/journal.pcbi.1000304

Park, K., Shen, B. W., Parmeggiani, F., Huang, P.-S., Stoddard, B. L., and Baker, D. (2015). Control of repeat-protein curvature by computational protein design. *Nat. Struct. Mol. Biol.* 22, 167–174. doi:10.1038/nsmb.2938

Parmeggiani, F., Huang, P.-S., Vorobiev, S., Xiao, R., Park, K., Caprari, S., et al. (2015). A general computational approach for repeat protein design. *J. Mol. Biol.* 427, 563–575. doi:10.1016/j.jmb.2014.11.005

Parra, R. G., Espada, R., Sánchez, I. E., Sippl, M. J., and Ferreiro, D. U. (2013). Detecting repetitions and periodicities in proteins by tiling the structural space. *J. Phys. Chem. B* 117, 12887–12897. doi:10.1021/jp402105j

Pellegrini, M., Marcotte, E. M., and Yeates, T. O. (1999). A fast algorithm for genome-wide analysis of proteins with repeated sequences. *Proteins* 35, 440–446. doi:10.1002/(SICI)1097-0134(19990601)35:4<440::AID-PROT7>3.0.CO;2-Y

Pellegrini, M., Renda, M. E., and Vecchio, A. (2012). Ab initio detection of fuzzy amino acid tandem repeats in protein sequences. *BMC Bioinformatics* 13(Suppl. 3):S8. doi:10.1186/1471-2105-13-S3-S8

Plückthun, A. (2015). Designed ankyrin repeat proteins (darpins): binding proteins for research, diagnostics, and therapy. *Annu. Rev. Pharmacol. Toxicol.* 55, 489–511. doi:10.1146/annurev-pharmtox-010611-134654

Ponting, C. P., Mott, R., Bork, P., and Copley, R. R. (2001). Novel protein domains and repeats in drosophila melanogaster: insights into structure, function, and evolution. *Genome Res.* 11, 1996–2008. doi:10.1101/gr.198701

Potenza, E., Domenico, T. D., Walsh, I., and Tosatto, S. C. E. (2015). Mobidb 2.0: an improved database of intrinsically disordered and mobile proteins. *Nucleic Acids Res.* 43, 315–320. doi:10.1093/nar/gku982

Rajathei, D. M., and Selvaraj, S. (2013). Analysis of sequence repeats of proteins in the {PDB}. *Comput. Biol. Chem.* 47, 156–166. doi:10.1016/j.compbiolchem.2013.09.001

Reichen, C., Madhurantakam, C., Plückthun, A., and Mittl, P. R. (2014). Crystal structures of designed armadillo repeat proteins: implications of construct design and crystallization conditions on overall structure. *Protein Sci.* 23, 1572–1583. doi:10.1002/pro.2535

Richard, F. D., and Kajava, A. V. (2014). Trdistiller: a rapid filter for enrichment of sequence datasets with proteins containing tandem repeats. *J. Struct. Biol.* 186, 386–391. doi:10.1016/j.jsb.2014.03.013

Rubinson, E. H., and Eichman, B. F. (2012). Nucleic acid recognition by tandem helical repeats. *Curr. Opin. Struct. Biol.* 22, 101–109. doi:10.1016/j.sbi.2011.11.005

Sabarinathan, R., Basu, R., and Sekar, K. (2010). Prostrip: a method to find similar structural repeats in three-dimensional protein structures. *Comput. Biol. Chem.* 34, 126–130. doi:10.1016/j.compbiolchem.2010.03.006

Sawyer, N., Chen, J., and Regan, L. (2013). All repeats are not equal: a module-based approach to guide repeat protein design. *J. Mol. Biol.* 425, 1826–1838. doi:10.1016/j.jmb.2013.02.013

Schaper, E., and Anisimova, M. (2015). The evolution and function of protein tandem repeats in plants. *New Phytol.* 206, 397–410. doi:10.1111/nph.13184

Schaper, E., Gascuel, O., and Anisimova, M. (2014). Deep conservation of human protein tandem repeats within the eukaryotes. *Mol. Biol. Evol.* 31, 1132–1148. doi:10.1093/molbev/msu062

Schaper, E., Kajava, A. V., Hauser, A., and Anisimova, M. (2012). Repeat or not repeat? Statistical validation of tandem repeat prediction in genomic sequences. *Nucleic Acids Res.* 40, 10005–10017. doi:10.1093/nar/gks726

Schaper, E., Korsunsky, A., Messina, A., Murri, R., Pečerska, J., Stockinger, H., et al. (2015). Tral: tandem repeat annotation library. *Bioinformatics* 31, 3051–3053. doi:10.1093/bioinformatics/btv306

Shih, E. S., and Hwang, M.-J. (2004). Alternative alignments from comparison of protein structures. *Proteins* 56, 519–527. doi:10.1002/prot.20124

Sickmeier, M., Hamilton, J. A., LeGall, T., Vacic, V., Cortese, M. S., Tantos, A., et al. (2007). Disprot: the database of disordered proteins. *Nucleic Acids Res.* 35(Suppl. 1), D786–D793. doi:10.1093/nar/gkl893

Sippl, M. J. (2008). On distance and similarity in fold space. *Bioinformatics* 24, 872–873. doi:10.1093/bioinformatics/btn040

Soding, J., Remmert, M., and Biegert, A. (2006). HHrep: de novo protein repeat detection and the origin of TIM barrels. *Nucleic Acids Res.* 34(Suppl._2), W137–W142. doi:10.1093/nar/gkl130

Sokol, D., Benson, G., and Tojeira, J. (2007). Tandem repeats over the edit distance. *Bioinformatics* 23, e30–e35. doi:10.1093/bioinformatics/btl309

Street, T. O., Rose, G. D., and Barrick, D. (2006). The role of introns in repeat protein gene formation. *J. Mol. Biol.* 360, 258–266. doi:10.1093/bioinformatics/btl309

Stumpp, M. T., Forrer, P., Binz, H. K., and Pluckthun, A. (2015). *Repeat Protein from Collection of Repeat Proteins Comprising Repeat Modules*. US Patent 9,006,389.

Szklarczyk, R., and Heringa, J. (2004). Tracking repeats using significance and transitivity. *Bioinformatics* 20(Suppl. 1), i311–i317. doi:10.1093/bioinformatics/bth911

Tompa, P. (2002). Intrinsically unstructured proteins. *Trends Biochem. Sci.* 27, 527–533. doi:10.1016/S0968-0004(02)02169-2

Tompa, P., and Fersht, A. (2009). *Structure and Function of Intrinsically Disordered Proteins*. Baca Raton, FL: Chapman and Hall/CRC.

Turutina, V. P., Laskin, A. A., Kudryashov, N. A., Skryabin, K. G., and Korotkov, E. V. (2006). Identification of amino acid latent periodicity within 94 protein families. *J. Comput. Biol.* 13, 946–964. doi:10.1089/cmb.2006.13.946

Walsh, I., Martin, A. J. M., Di Domenico, T., and Tosatto, S. C. E. (2012a). Espritz: accurate and fast prediction of protein disorder. *Bioinformatics* 28, 503–509. doi:10.1093/bioinformatics/btr682

Walsh, I., Sirocco, F. G., Minervini, G., Di Domenico, T., Ferrari, C., and Tosatto, S. C. E. (2012b). Raphael: recognition, periodicity and insertion assignment of solenoid protein structures. *Bioinformatics* 28, 3257–3264. doi:10.1093/bioinformatics/bts550

Conflict of Interest Statement: The author declares that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

frontiers in
BIOENGINEERING AND BIOTECHNOLOGY

METHODS ARTICLE
published: 24 February 2015
doi: 10.3389/fbioe.2015.00018

Knowledge in the investigation of A-to-I RNA editing signals

Giovanni Nigita[1†], Salvatore Alaimo[2†], Alfredo Ferro[3], Rosalba Giugno[3*‡] and Alfredo Pulvirenti[3*‡]

[1] Department of Molecular Virology, Immunology and Medical Genetics, Ohio State University, Columbus, OH, USA
[2] Department of Mathematics and Computer Science, University of Catania, Catania, Italy
[3] Department of Clinical and Experimental Medicine, University of Catania, Catania, Italy

Edited by:
Marco Pellegrini, Consiglio Nazionale delle Ricerche, Italy

Reviewed by:
Ernesto Picardi, University of Bari, Italy
Eran Eyal, Sheba Medical Center, Israel

***Correspondence:**
Rosalba Giugno and Alfredo Pulvirenti, Department of Clinical and Experimental Medicine, University of Catania, Via Santa Sofia, Catania 78 | I-95122, Italy
e-mail: giugno@dmi.unict.it; apulvirenti@dmi.unict.it

[†] Giovanni Nigita and Salvatore Alaimo have contributed equally to this work.
[‡] Rosalba Giugno and Alfredo Pulvirenti have contributed equally to this work.

RNA editing is a post-transcriptional alteration of RNA sequences that is able to affect protein structure as well as RNA and protein expression. Adenosine-to-inosine (A-to-I) RNA editing is the most frequent and common post-transcriptional modification in human, where adenosine (A) deamination produces its conversion into inosine (I), which in turn is interpreted by the translation and splicing machineries as guanosine (G). The disruption of the editing machinery has been associated to various human diseases such as cancer or neurodegenerative diseases. This biological phenomenon is catalyzed by members of the adenosine deaminase acting on RNA (ADAR) family of enzymes and occurs on dsRNA structures. Despite the enormous efforts made in the last decade, the real biological function underlying such a phenomenon, as well as ADAR's substrate features still remain unknown. In this work, we summarize the major computational aspects of predicting and understanding RNA editing events. We also investigate the detection of short motif sequences potentially characterizing RNA editing signals and the use of a logistic regression technique to model a predictor of RNA editing events. The latter, named AIRIINER, an algorithmic approach to assessment of A-to-I RNA editing sites in non-repetitive regions, is available as a web app at: http://alpha.dmi.unict.it/airliner/. Results and comparisons with the existing methods encourage our findings on both aspects.

Keywords: A-to-I RNA editing, motif analysis, prediction, ADARs, logistic regression

BACKGROUND

In recent times, there has been a change in the range of research on many types of diseases. In the past decades, the principal aim was to add information about the molecular pathways involved in some disease through the study of DNA mutations. Lately, the focus has indeed moved to the analysis of post-transcriptional modification events, such as RNA editing. The knowledge that the activity of RNA editing is higher in mammalian brain than in other tissues (Paul and Bass, 1998), hints that editing may play a crucial role in the central nervous system (Nishikura, 2006). Therefore, malfunctions of RNA editing machineries could lead to serious consequences (Galeano et al., 2012; Tomaselli et al., 2014).

RNA editing is a type of post-transcriptional modification, taking place in eukaryotes, which alters the sequence of primary RNA transcripts by deleting, inserting, or modifying residues. Despite the discovery of several distinct types of RNA editing over the years, adenosine-to-inosine (A-to-I) RNA editing is now considered the most predominant in mammalians (Nishikura, 2010). Through the deamination process, adenosine (A) is converted into inosine (I), which in turn is interpreted as guanosine (G) by both the splicing and the translation machineries (Rueter et al., 1999). Enzymes members of the adenosine deaminase acting on RNA (ADAR) family catalyze this biological phenomenon only on dsRNA structures (Bass, 2002; Jepson and Reenan, 2008; Nishikura, 2010).

Adenosine-to-inosine RNA sites abundantly occur in intronic regions as well as in 3′-UTRs. RNA editing events can modify RNA molecules in several cellular contexts causing: the creation and/or destruction of splicing sites (Rueter et al., 1999); the modulation of gene expression pathways (Bazak et al., 2014b) during translation (Nishikura, 2010); the gain or loss of miRNA recognition elements (MRE) during mRNA targeting (Nishikura, 2006; Borchert et al., 2009) (i.e., MRE can be created or deleted even with a single post-transcriptional modification). As it has been reported in the last few years, RNA editing sites can be found in non-coding RNA molecules, especially within pri-miRNA (Kawahara et al., 2008; Kawahara, 2012), lncRNA (Mitra et al., 2012), and precursor-tRNA (Su and Randau, 2011), the latter deaminated by adenosine deaminases acting on tRNA (ADAT) enzymes.

It is possible to distinguish two forms of A-to-I RNA editing, *promiscuous* and *specific*. The *promiscuous* A-to-I editing occurs within longer duplexes of hundreds of nucleotides, as in the case of stem–loops that are formed by the pairing of repetitive elements (e.g., Alu elements), as seen above. In those cases, up to 60% of adenosines could be edited (Carmi et al., 2011; Bazak et al., 2014b). The *specific* A-to-I RNA editing occurs in short and/or unstable duplex RNA regions (Wahlstedt and O'Hman, 2011), in which at least 10% of their adenosines selectively could undergo deamination. A-to-I RNA editing events in small non-coding RNAs, such as microRNAs, are perfect examples of *specific* editing (Nishikura, 2010).

One of the main challenges in the study of the RNA editing phenomenon is certainly RNA editing occurrence. The detection of editing sites in RNA molecules in particular cellular conditions is very difficult considering that RNA editing is a dynamic spatial–temporal process. In the last decade, the application of global approaches to the study of A-to-I editing, including in a first phase bioinformatics methods and, lately, high-throughput sequencing technology (HTS) based pipelines, have led to important advances, allowing the discovery of a large amount of editing sites in the human transcriptome. Despite the enormous efforts made in recent years, the real biological function underlying such a phenomenon, as well as ADAR's substrate features still remain unknown.

In this work, we give an overview of the current state of knowledge on the editing phenomenon, as well as provide the main features of editing sites as highlighted today. We also investigate, inspired by previous results, methods for the detection of signals characterizing editing events and the prediction of novel A-to-I editing sites in non-repetitive regions. These techniques are based on the analysis of nucleotide profiles within a distance-radius of the probable editing site. Results on the signal detection show that editing sites may not have strong defined signal patterns.

Finally, by using a logistic regression technique we developed AIRlINER, an algorithmic approach for the prediction of A-to-I RNA editing sites in non-repetitive regions. This method has been compared with *InosinePredict* (Eggington et al., 2011), a similar technique, which analyzes the nucleotides flanking the editing site. *InosinePredict* assumes a multiplicative relationship between the coefficients necessary to compute the percentage of editing. Our results clearly show that AIRlINER improves the quality of predictions with respect to *InosinePredict* and suggest further research directions. AIRlINER is available at the following address: http://alpha.dmi.unict.it/airliner/.

KNOWLEDGE AND FEATURES OF EDITING SITES SIGNALS

At the end of 80s, ADARs, initially identified as associated with an unknown dsRNA-unwinding activity (Bass and Weintraub, 1987; Rebagliati and Melton, 1987), were discovered as RNA editing machineries able to alter adenosine into inosine through deamination, especially in dsRNA structures (Bass and Weintraub, 1988; Wagner et al., 1989). In the next 10 years, three members of the ADAR gene family were identified in humans: two isoforms of ADAR1 (N-terminally truncated ADAR1p110 and a full-length ADAR1p150) (Kim et al., 1994; Patterson and Samuel, 1995), ADAR2 (Lai et al., 1997) (both these members expressed in many tissues), and ADAR3 (Chen et al., 2000) present only in the central nervous system. While for ADAR1 and ADAR2 the enzymatic activity was established, for ADAR3 it remains unknown. Unlike ADAR1 and ADAR2, an interesting feature about ADAR3 is the presence of the R domain, which enables the enzyme to bind to single strand structures. ADAR1 and ADAR2 have two common functional regions, an N-terminal dsRNA-binding domain (dsRBD) and a C-terminal deaminase domain, but only ADAR1 contains two Z-DNA-binding domains, Zα and Zβ. Some editing events are edited only by ADAR1 or ADAR2, showing a significant difference in their RNA-substrate interactions (Wong et al., 2001; Riedmann et al., 2008). For instance, the serotonin B site is deaminated not only by ADAR1, while the serotonin D and the GluR-B Q/R sites are deaminated exclusively by ADAR2 (Burns et al., 1997; Yang et al., 1997), but also ADAR1 and ADAR2 can edited the same target, as in the cases of serotonin A and C editing sites (Burns et al., 1997). Subsequently, the characterization of the neighborhood profiles of both ADAR1 and ADAR2 were established. In particular, ADAR1 has 5′ neighboring base preference consisting of uracil, adenosine, cytosine, and guanosine in order ($U \approx A > G > C$), but not 3′ neighbor preference has been identified (Polson and Bass, 1994). Similarly, ADAR2 has a 5′ neighbor preference, but, differently from ADAR1, ADAR2 has a 3′ neighboring base preference ($U = G > C = A$) forming particular trinucleotide sequences with an adenosine at the second base (U*A*U, A*A*G, U*A*G, A*A*U) (Lehmann and Bass, 2000).

In 2003, Hoopengardner et al. (2003) discovered that highly conserved regions, which in turn form a dsRNA structure, surround many editing sites. Later, by considering these findings, bioinformatics methods mapping ESTs against a reference genome were able to discover tens of thousands of A-to-I RNA editing sites, with more than 90% of them occurring within Alu repeats (Athanasiadis et al., 2004; Kim et al., 2004; Levanon et al., 2004). A significant problem in all the bioinformatics approaches for RNA editing detection, as described above, still remains the limitations posed by sequencing technologies, specifically, the inability to distinguish a guanosine originating from an I-to-G replacement from a guanosine as a product of noise, sequencing errors or SNP. A solution to this issue was proposed by Sakurai et al. (2010) who designed a biochemical method, called inosine chemical erasing (ICE), able to identify inosine sites on RNA molecules by employing inosine-specific cyanoethylation with reverse transcription. This is a reliable and accurate biochemical method to detect inosines in RNA strands.

The recent years have been characterized by the development of several approaches for editing discovery based on deep sequencing. It was recently hypothesized that more than 100 million editing sites could be found in human Alu repeats, located mainly in genic regions (Bazak et al., 2014a). Although these recent methods prove to be more accurate than previous ones, some of them nonetheless present limitations in terms of false positives produced (Kleinman and Majewski, 2012; Lin et al., 2012; Pickrell et al., 2012). In recent years, a considerable number of RNAseq based methods have emerged (Li et al., 2009; Ju et al., 2011; Bahn et al., 2012; Peng et al., 2012; Picardi et al., 2012; Ramaswami et al., 2012, 2013; Bazak et al., 2014a), gradually improved the accuracy in discovering new editing sites, leading, in addition, to the identification of a set of human editing sites orders of magnitude larger than before. Recently, Sakurai et al. (2014) combined the ICE method with HTS (ICE seq) for an unbiased genome-wide screening of novel A-to-I editing sites. ICE seq is able to detect editing sites in both repeat elements and short hairpins, rendering this a currently unique method for genome-wide identification of A-to-I editing events in both tissues and clinical specimens without genomic DNAs.

The application of HTS technology to RNA editing discovery has not only brought improvements in the editing discovery but also helped to increase the knowledge about the features inherent to the phenomenon. In fact, thanks to the analysis of a large RNA-seq data, Bazak et al. (2014b) studied the global characteristics that

affect the editability at the Alu level, uncovering some important features. An important parameter that influences the editing of the Alu is the distance to the nearest complementary inverse sequence. Indeed, the editing, on average, exponentially decays with this distance, with a typical length of about 800 nt. Another aspect is that the editing levels are positively correlated with the number of reversely complementary repeats in the flanking regions of the Alu. Instead, they are negatively correlated with the number of same-strand repeats. Furthermore, the editing level depends on both the lengths of the Alu repeats and their closest reversely oriented sequence, additionally to whether the latter resides in the same intron/exon. Finally, the consensus strand of the Alus is more edited than the reverse strand.

Lately, Pinto et al. (2014) conducted a study with the scope to find mammalian conserved editing sites. Surprisingly, only a very small fraction (0.004%) of human editing sites is conserved in mammals. Noteworthy, by considering the nucleotide frequency, the 10-nt upstream and downstream regions of conserved editing sites are stronger than the ones of all non-Alu human editing sites.

The large number of editing sites discovered by these methodologies has given rise to the need for public databases to record such information in order to further elucidate the biological functions underlying the RNA editing phenomenon. The first centralized repository was DARNED[1] (Kiran and Baranov, 2010), whose last release contains more than 300,000 editing sites (Kiran and Baranov, 2010; Kiran et al., 2013). Later, Ramaswami and Li (2014) built RADAR[2], a rigorously manually curated database of annotated A-to-I editing sites, amounting to about 1.4 million editing events. Unfortunately, both DARNED and RADAR do not offer a grade of confidence for each editing site due to the heterogeneity of the discovering methods applied, making the creation of a standard measure of confidence necessary in the future.

INVESTIGATION OF MOTIFS CHARACTERIZING THE RNA EDITING EVENTS

It is well known that the vast majority of editing events occur in repetitive regions. Recently, Ramaswami et al. (2012) developed a computational framework to identify editing events both Alu and non-Alu regions (repetitive non-Alu and non-repetitive regions) by analyzing the genomic DNA and RNA sequences. Through this method they found that more than 97% of the discovered editing events occur in Alu regions, also speculating that the remaining non-Alu editing sites are related to nearby edited Alu ones. This makes the identification of sequence motifs able to characterize RNA editing a very challenging problem. Therefore, any approach aimed at the search of sequence or structural motifs associated to RNA editing events should take into account the bias introduced by repetitive regions. Consequently, the searching should be done outside of repetitive regions in order to detect signals independent of the background.

Our strategy has been the following. First, we selected a set of non-Alu editing events and then generated edited regions (ERs) based on the distances between non-Alu editing site, as described below. Next, we applied MEME (Bailey et al., 2009) in order to discover motifs within such a set of sequences. MEME analyzes the input data and searches for significant ungapped sequence patterns shared among the sequences.

In order to obtain the ERs, considering the human editing sites listed in the RADAR database (Ramaswami and Li, 2014), we firstly filtered the A-to-I editing sites, which resulted to be SNPs, as compared to dbSNP141 (Solomon et al., 2014). We then computed δ as the weighted average distance between the editing sites. We obtained that on average there are 6,057 nt between two editing events. This value has been considered as a *breakpoint* during the construction of ERs. In particular, starting from a generic editing site x, we searched for the next one y. When y falls within a distance less than or equal to δ, the editing site y is included in the ER and the process continues. Otherwise, if the next site is found at a distance greater than δ, the ER is no longer extended. As a result, a total of 55,952 ERs have been defined. Additionally, we separated ERs containing repetitive elements from those, which do not contain any, obtaining a total of 48,164 repetitive ERs and 7,788 non-repetitive ERs. The fact that ERs possess different lengths could allow us to take into account the possibility that they may contain motifs close to the editing sites in secondary structures.

Figure 1 shows that repetitive ERs are longer than non-repetitive ones, with the largest number of editing sites found in regions containing some repetitive elements, as confirmed in the literature (Wahlstedt and O'Hman, 2011). We built a training set of non-repetitive ERs by selecting those regions with a length of 2,000–6,000 nt, containing at least 10 editing sites. Hence, we obtained a final dataset of 47 ERs, in particular, 29 regions are in positive strand with 479 editing sites and 18 ones are in negative strand with 319 editing sites.

We ran MEME on such dataset by searching both palindromic and non-palindromic motifs with a length ranging from 6 to 50 nt. We bound the number of motifs to 50 palindromic and 50 non-palindromic.

From these 100 motifs we took only those with an E-value <0.05. Next, we filter out motifs that were contained in a set of human ultra-conserved sequences having no known editing site (Bejerano et al., 2004), with respect to DARNED and RADAR databases. Finally, a total of 16 motifs (4 palindromic and 12 non-palindromic) have been discovered.

In order to validate the filtered motifs, we performed a permutation test using 100 samples of 1,000 randomly taken 3′ UTR sequences (hg19) with masked repetitive regions. As shown in **Table 1**, only 13 motifs were significant (p-value <0.01).

FROM NUCLEOTIDE FREQUENCY TO AN APPROACH TO ASSESSMENT OF A-TO-I RNA EDITING SITES

Starting from the idea proposed by Pinto et al. (2014), we used a logistic regression technique to determine a model from which we can compute the probability that an adenosine in a non-repetitive region of the genome is affected by the A-to-I editing phenomenon. Our method, called AIRlINER, determines the editing probability of an adenosine by analyzing its flanking region of 10 nt. Such pattern is then combined with a similar model calculated from un-edited sequences, resulting in the estimation of an unbiased editing probability.

[1] http://darned.ucc.ie/
[2] http://rnaedit.com/

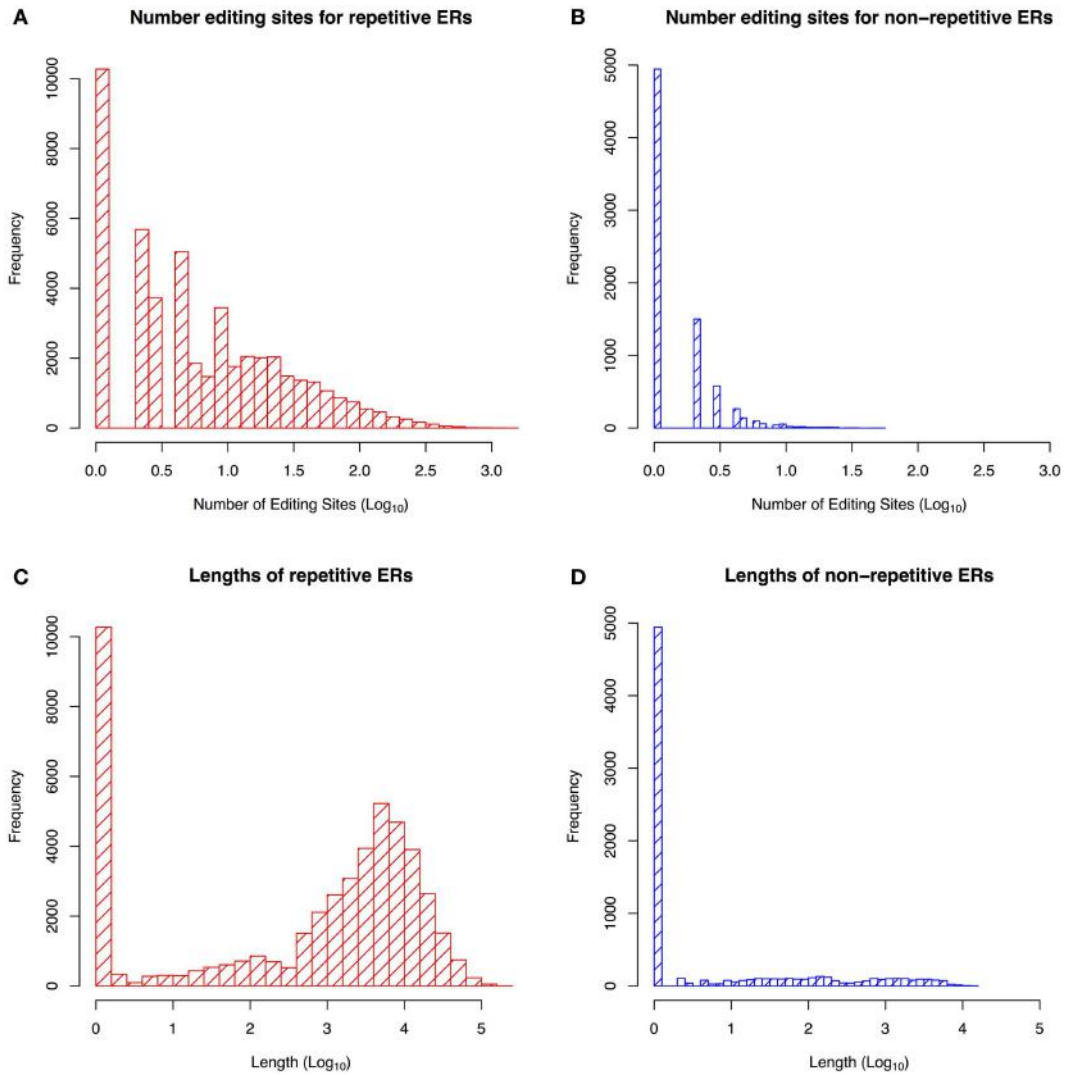


FIGURE 1 | Statistics about the *repetitive* and *non-repetitive* edited regions (ER). Distribution of editing sites frequency in *repetitive* ERs **(A)** and *non-repetitive* ERs **(B)**. Distribution of *repetitive* ERs sequence length **(C)** and *non-repetitive* ERs sequence length **(D)**. The figure shows that the non-repetitive ERs are shorter than repetitive ones and contain fewer editing sites.

In order to train our method, we built a dataset composed of 30,280 sequences of 21 nt centered on an adenosine, from the human genome (hg19). According to their provenance, our dataset can be divided equally into two sets: known editing sites and random sites. For the purpose of retrieving known editing sites in non-repetitive regions, only human sites which do not have any repetitive elements in their flanking regions of 2,000 nt were selected from the RADAR database (Ramaswami and Li, 2014). Random sites were chosen by randomly selecting a number of sequences equal to that of the known editing sites. From such a selection, we excluded known editing sites in both repetitive and non-repetitive regions.

From such a dataset, two probabilities $P(j, i)$ and $P'(j, i)$ can be computed: the first one corresponds to the probability of finding nucleotide j in position i of a region affected by editing, while the second one represents the probability of finding nucleotide j in position i of an un-ER. Starting from these probabilities, we computed the graphs in **Figure 2**, which represent the distributions of the nucleotides for the two types of regions.

Therefore, let s be a nucleotide sequence and $P(s)$ its editing probability, using the previously defined probabilities we are able to train a logistic regression model such as:

$$\log\left(\frac{P(s)}{1-P(s)}\right) = \beta_0 + \sum_{i=1}^{21} \beta_i P\left(s\left[i\right], i\right) - \sum_{i=1}^{21} \beta_i' P'\left(s\left[i\right], i\right),$$

where $s[i]$ is the i-th nucleotide in a sequence. Now we can use this model to estimate the editing probability of any sequence of 21 nt centered on an adenosine, and if such probability is >0.5, we can say that such a sequence may be affected by editing.

To tune and validate our method, we applied a 10-fold cross validation procedure and computed a mean error. To compare our

Table 1 | Filtered motifs in ERs (47 edited regions).

Motif	Sequence (*Best possible match*)	Width	Type	*E*-value
1	CCAGGCTGGAGTGCAGTGGCGCAATCTCA	29	Non-palindromic	1E-126
2	GGATTACAGGCGTGAGCCACCGCGCCTGG	29	Non-palindromic	3,60E-123
3	GAGGTGCTGGGATTATAGGGG	21	Non-palindromic	8,50E-35
4	CCTGACCTCATGAGA	15	Non-palindromic	4,10E-22
5	AGACATGGAACCAACCTAAATGCCCACCA	29	Non-palindromic	9,40E-17
6	AGGAGGCAAAGGAAG	15	Non-palindromic	7,00E-11
7	TGGGATTGCAGGCAT	15	Non-palindromic	1,20E-06
8	TTTCATGGCTGCATAGTATTCTATTGTGT	29	Non-palindromic	1,00E-05
9	TGTAAATTAGTACAGCCTTTATGGAAAAC	29	Non-palindromic	2,90E-12
10	AGTCCCAGCTTCTCGAGAAGCTGGGACT	28	Palindromic	2,7E-97
11	TGCACCCCAGGCTGGGGTGCA	21	Palindromic	8,4E-50
12	CTTGTACTCCCAACATGTTGGGAGTACAAG	30	Palindromic	5,2E-72
13	CTTGAACCTCGGAGGTTCAAG	21	Palindromic	3,9E-28

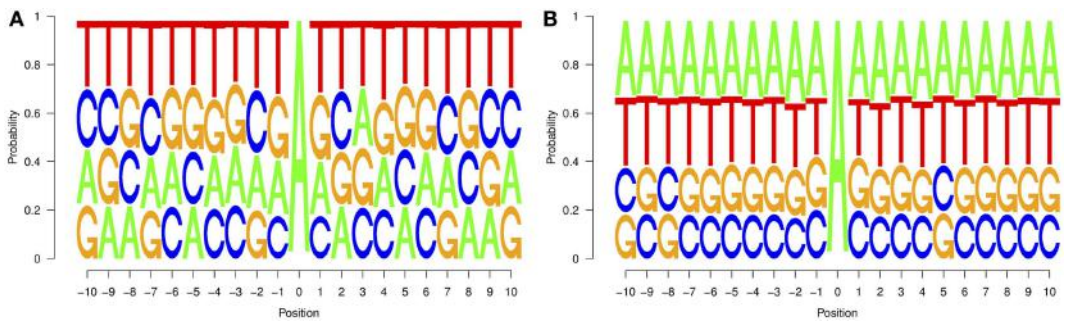


FIGURE 2 | Neighborhood preferences that we computed for experimentally verified editing sites in non-repetitive regions (A) and random sites (B) chosen among those for which no editing event is reported. Neighborhood preferences are coherent with the upstream nucleotide distribution of editing site sequence contexts reported in Eggington et al. (2011).

method with *InosinePredict*, we used a threshold to establish the presence or absence of editing in a specific sequence. Such a threshold was set to 9.6% for *InosinePredict*, as shown in Eggington et al. (2011). For our algorithm, we choose all sites for which an editing probability >0.5 is computed. We also took into account the fact that *InosinePredict* can produce predictions for both hADAR1 and hADAR2. We do not have this information in our dataset, so we chose to select the maximum score produced by *InosinePredict* for editing sites, and the minimum score for random sequences. Consequently, we are able to ensure a fair comparison with our method despite the absence of information on which ADAR affects each editing site.

In **Tables 2** and **3**, we show the confusion matrices computed using the previously described procedure. The two algorithms were applied to the dataset and the values computed for the central adenosines in each sequence were used to determine the presence or absence of editing. Our method significantly reduces the number of false negatives compared to *InosinePredict*, thus resulting in a better editing sites prediction quality. AIRlINER is also able to achieve a substantial reduction of false positives, even if nothing can be stated with certainty about them, as the absence of editing in these sites can also be determined by lack of experimental tests. The best quality in predicting editing sites, however, may reflect the fact that the random sequences classified as non-edited could be with high probability considered as such.

Table 2 | Confusion matrix computed by applying InosinePredict (Eggington et al., 2011) to our dataset.

		Prediction outcome	
		Editing site	**Non-editing site**
Actual value	Editing sites	58.48	41.52
	Random sites	60.18	39.82

Editing percentages for each sites have been divided into two classes (editing/non-editing) using the thresholds defined in Eggington et al. (2011).

Further confirmation of the quality of our methodology is represented by the receiver operating characteristic curves (ROCs), **Figure 3**, computed from the results produced by the two algorithms. The curves demonstrate a significant improvement in performance. Such curves also show that the threshold chosen to distinguish editing sites from non-editing ones does not affect the performance difference between the two algorithms. As a confirmation of this, *InosinePredict* obtains an average area under the ROC curve (AUC) of 0.5072, while AIRlINER reaches 0.7466. In

Table 3 | Confusion matrix computed by applying AIRlINER to our dataset.

		Prediction outcome	
		Editing site	**Non-editing site**
Actual value	Editing sites	71.18	28.82
	Random sites	34.05	65.95

All editing sites for which editing probability is >0.5 were classified as editing while the remaining as non-editing.

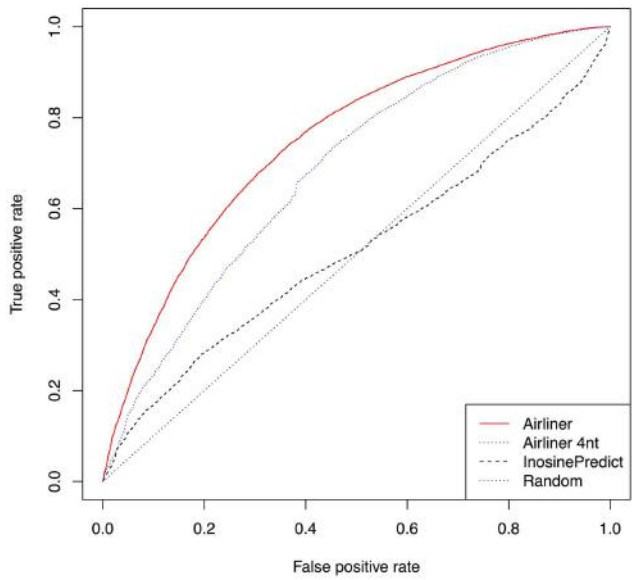


FIGURE 3 | Receiver operating characteristic curve (ROC) computed for the two prediction algorithms. We also provide a ROC curve for a variant of our algorithm (AIRlINER 4 nt), which takes into account only the flanking region of 4 nt around an adenosine. Such a curve is useful to compare the performance with our algorithm using the same flanking region. AIRlINER shows an average area under the ROC curve (AUC) equal to 0.7466, while InosinePredict gets an AUC of 0.5072. AIRlINER 4 nt has an AUC of 0.7464.

Figure 3, we also compare a variant of our method, AIRlINER 4 nt, with *InosinePredict*. Such a variant computes the editing probability of an adenosine by considering its flanking region of 4 nt. This comparison shows that our strategy is superior to *InosinePredict* even when the prediction is calculated from this same region around an adenosine.

Furthermore, we investigated that ADAR acts on each editing site in our training set by building an additional data set from editing sites experimentally identified in (Bahn et al., 2012). Using human cell lines U87MG in which the gene expression of ADAR1 was repressed, the authors were able to identify about 4,000 ADAR1-specific editing sites. Four hundreds of such sites were identified in non-repetitive regions. From the latter, we have built a training set using the same procedure described above and trained our model. In **Figure 4**, we show the results of this experiment by means of ROC curves. Even in this case, the AIRlINER

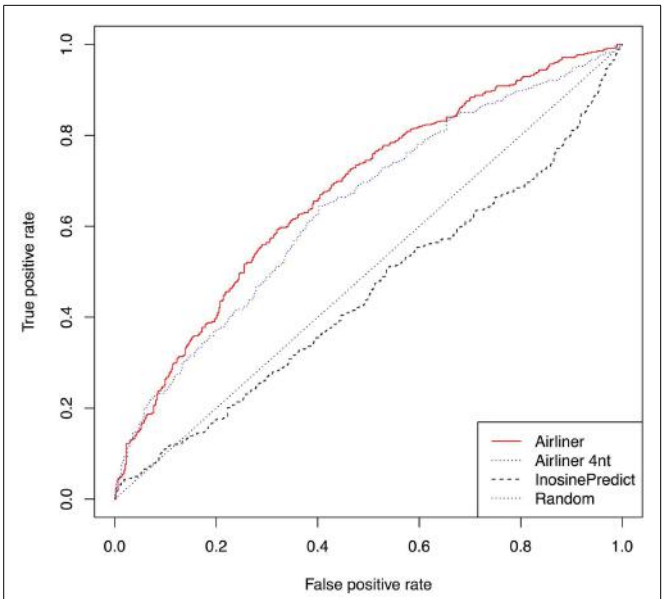


FIGURE 4 | Comparison between AIRlINER and InosinePredict by means of receiver operating characteristic curve (ROC) computed using the data set built from Bahn et al. (2012). Here we also show a ROC curve for a variant of the proposed algorithm (AIRlINER 4 nt), which takes into account only the flanking region of 4 nt around an adenosine. AIRlINER shows an average area under the ROC curve (AUC) equal to 0.6763, while InosinePredict gets an AUC of 0.4498. AIRlINER 4 nt has an AUC of 0.6435.

methodology is significantly better than *InosinePredict*. As further confirmation, we also computed the AUC, which amounts to 0.6763 for AIRlINER, and 0.4498 for *InosinePredict*.

Finally, to verify the quality of the editing sites predicted by our algorithm, we selected from the literature 52 experimentally validated sites by Sanger method and 7 sites validated as non-edited (as shown in Table S1 in Supplementary Material). We then applied the two methodologies and checked how many of them are correctly identified. AIRlINER is able to predict 42 of 52 editing sites and 5 of 7 non-editing sites while *InosinePredict* identifies 26 editing sites and 4 non-editing ones. More details can be found in the Table S1 in Supplementary Material.

AIRlINER is available as a web app at the following URL: http://alpha.dmi.unict.it/airliner/.

CONCLUSION AND FUTURE DIRECTIONS

RNA editing is a post-transcriptional phenomenon that occurs in eukaryotes and contributes to the diversity of transcriptome. A-to-I is the most common form of RNA editing in mammals, altering the sequence of primary RNA transcripts by adenosine deamination. In this last decade, computational methods and RNAseq based approaches to RNA editing discovery have emerged, contributing to the identification of more than a million editing events in human, many of which located close to or within Alu repeats. Despite the enormous efforts made so far, the biological significance of the editing phenomenon remains largely unknown.

In the first part of this work, we summarized some of the most important characteristics discovered for RNA editing. Inspired by

literature, we investigated the presence of motifs in non-repetitive regions characterizing the editing events, finding a small set of candidates. Moreover, we considered the frequency of the 20 nt centered on each RNA editing site to compute the probability that an adenosine in a non-repetitive region of the genome may be affected by the A-to-I editing phenomenon. Our method, available on line, significantly reduces the number of false negatives with respect to existing methods, thus indicating a better editing-site prediction quality.

Future work will concern the use of different motif-detecting algorithms to confirm the consistency of our current findings. Motif detection methods may make use of information from the secondary structure of the editing regions with respect also to the different classes of ADAR. Finally, further investigation is needed to highlight any significant combination of motif patterns.

ACKNOWLEDGMENTS

GN has been supported by Italian Foundation for Cancer Research (NG 15046). We also wish to thank Dario Veneziano for reviewing the English of the final version of the article. AP, RG and AF have been partially supported by a PON 2007–2013 grant, SIGMA – PON01_00683 – CUP B61H11000380005.

SUPPLEMENTARY MATERIAL

The Supplementary Material for this article can be found online at: http://www.frontiersin.org/Journal/10.3389/fbioe.2015.00018/abstract

REFERENCES

Athanasiadis, A., Rich, A., and Maas, S. (2004). Widespread A-to-I RNA editing of Alu-containing mRNAs in the human transcriptome. *PLoS Biol.* 2:e391. doi:10.1371/journal.pbio.0020391

Bahn, J. H., Lee, J.-H., Li, G., Greer, C., Peng, G., and Xiao, X. (2012). Accurate identification of A-to-I RNA editing in human by transcriptome sequencing. *Genome Res.* 22, 142–150. doi:10.1101/gr.124107.111

Bailey, T. L., Boden, M., Buske, F. A., Frith, M., Grant, C. E., Clementi, L., et al. (2009). MEME SUITE: tools for motif discovery and searching. *Nucleic Acids Res.* 37, W202–W208. doi:10.1093/nar/gkp335

Bass, B. L. (2002). RNA editing by adenosine deaminases that act on RNA. *Annu. Rev. Biochem.* 71, 817–846. doi:10.1146/annurev.biochem.71.110601.135501

Bass, B. L., and Weintraub, H. (1987). A developmentally regulated activity that unwinds RNA duplexes. *Cell* 48, 607–613. doi:10.1016/0092-8674(87)90239-X

Bass, B. L., and Weintraub, H. (1988). An unwinding activity that covalently modifies its double-stranded RNA substrate. *Cell* 55, 1089–1098. doi:10.1016/0092-8674(88)90253-X

Bazak, L., Haviv, A., Barak, M., Jacob-Hirsch, J., Deng, P., Zhang, R., et al. (2014a). A-to-I RNA editing occurs at over a hundred million genomic sites, located in a majority of human genes. *Genome Res.* 24, 365–376. doi:10.1101/gr.164749.113

Bazak, L., Levanon, E. Y., and Eisenberg, E. (2014b). Genome-wide analysis of Alu editability. *Nucleic Acids Res.* 42, 6876–6884. doi:10.1093/nar/gku414

Bejerano, G., Pheasant, M., Makunin, I., Stephen, S., Kent, W. J., Mattick, J. S., et al. (2004). Ultraconserved elements in the human genome. *Science* 304, 1321–1325. doi:10.1126/science.1098119

Borchert, G. M., Gilmore, B. L., Spengler, R. M., Xing, Y., Lanier, W., Bhattacharya, D., et al. (2009). Adenosine deamination in human transcripts generates novel microRNA binding sites. *Hum. Mol. Genet.* 18, 4801–4807. doi:10.1093/hmg/ddp443

Burns, C. M., Chu, H., Rueter, S. M., Hutchinson, L. K., Canton, H., Sanders-Bush, E., et al. (1997). Regulation of serotonin-2C receptor G-protein coupling by RNA editing. *Nature* 387, 303–308. doi:10.1038/387303a0

Carmi, S., Borukhov, I., and Levanon, E. Y. (2011). Identification of widespread ultra-edited human RNAs. *PLoS Genet.* 7:e1002317. doi:10.1371/journal.pgen.1002317

Chen, C. X., Cho, D. S., Wang, Q., Lai, F., Carter, K. C., and Nishikura, K. (2000). A third member of the RNA-specific adenosine deaminase gene family, ADAR3, contains both single- and double-stranded RNA binding domains. *RNA* 6, 755–767. doi:10.1017/S1355838200000170

Eggington, J. M., Greene, T., and Bass, B. L. (2011). Predicting sites of ADAR editing in double-stranded RNA. *Nat. Commun.* 2, 319. doi:10.1038/ncomms1324

Galeano, F., Tomaselli, S., Locatelli, F., and Gallo, A. (2012). A-to-I RNA editing: the "ADAR" side of human cancer. *Sem. Cell Dev. Biol.* 23, 244–250. doi:10.1016/j.semcdb.2011.09.003

Hoopengardner, B., Bhalla, T., Staber, C., and Reenan, R. (2003). Nervous system targets of RNA editing identified by comparative genomics. *Science* 301, 832–836. doi:10.1126/science.1086763

Jepson, J. E. C., and Reenan, R. A. (2008). RNA editing in regulating gene expression in the brain. *Biochim. Biophys. Acta* 1779, 459–470. doi:10.1016/j.bbagrm.2007.11.009

Ju, Y. S., Kim, J.-I., Kim, S., Hong, D., Park, H., Shin, J.-Y., et al. (2011). Extensive genomic and transcriptional diversity identified through massively parallel DNA and RNA sequencing of eighteen Korean individuals. *Nat. Genet.* 43, 745–752. doi:10.1038/ng.872

Kawahara, Y. (2012). Quantification of adenosine-to-inosine editing of microRNAs using a conventional method. *Nat. Protoc.* 7, 1426–1437. doi:10.1038/nprot.2012.073

Kawahara, Y., Megraw, M., Kreider, E., Iizasa, H., Valente, L., Hatzigeorgiou, A. G., et al. (2008). Frequency and fate of microRNA editing in human brain. *Nucleic Acids Res.* 36, 5270–5280. doi:10.1093/nar/gkn479

Kim, D. D. Y., Kim, T. T. Y., Walsh, T., Kobayashi, Y., Matise, T. C., Buyske, S., et al. (2004). Widespread RNA editing of embedded Alu elements in the human transcriptome. *Genome Res.* 14, 1719–1725. doi:10.1101/gr.2855504

Kim, U., Wang, Y., Sanford, T., Zeng, Y., and Nishikura, K. (1994). Molecular cloning of cDNA for double-stranded RNA adenosine deaminase, a candidate enzyme for nuclear RNA editing. *Proc. Natl. Acad. Sci. U.S.A.* 91, 11457–11461. doi:10.1073/pnas.91.24.11457

Kiran, A., and Baranov, P. V. (2010). DARNED: a database of RNa editing in humans. *Bioinformatics* 26, 1772–1776. doi:10.1093/bioinformatics/btq285

Kiran, A. M., O'Mahony, J. J., Sanjeev, K., and Baranov, P. V. (2013). Darned in 2013: inclusion of model organisms and linking with Wikipedia. *Nucleic Acids Res.* 41, D258–D261. doi:10.1093/nar/gks961

Kleinman, C. L., and Majewski, J. (2012). Comment on "widespread RNA and DNA sequence differences in the human transcriptome". *Science* 335, 1302–1302. doi:10.1126/science.1209658

Lai, F., Chen, C. X., Carter, K. C., and Nishikura, K. (1997). Editing of glutamate receptor B subunit ion channel RNAs by four alternatively spliced DRADA2 double-stranded RNA adenosine deaminases. *Mol. Cell. Biol.* 17, 2413–2424.

Lehmann, K. A., and Bass, B. L. (2000). Double-stranded RNA adenosine deaminases ADAR1 and ADAR2 have overlapping specificities †. *Biochemistry* 39, 12875–12884. doi:10.1021/bi001383g

Levanon, E. Y., Eisenberg, E., Yelin, R., Nemzer, S., Hallegger, M., Shemesh, R., et al. (2004). Systematic identification of abundant A-to-I editing sites in the human transcriptome. *Nat. Biotechnol.* 22, 1001–1005. doi:10.1038/nbt996

Li, J. B., Levanon, E. Y., Yoon, J.-K., Aach, J., Xie, B., Leproust, E., et al. (2009). Genome-wide identification of human RNA editing sites by parallel DNA capturing and sequencing. *Science* 324, 1210–1213. doi:10.1126/science.1170995

Lin, W., Piskol, R., Tan, M. H., and Li, J. B. (2012). Comment on "widespread RNA and DNA sequence differences in the human transcriptome". *Science* 335, 1302. doi:10.1126/science.1210624

Mitra, S. A., Mitra, A. P., and Triche, T. J. (2012). A central role for long non-coding RNA in cancer. *Front. Genet.* 3:17. doi:10.3389/fgene.2012.00017

Nishikura, K. (2006). Editor meets silencer: crosstalk between RNA editing and RNA interference. *Nat. Rev. Mol. Cell Biol.* 7, 919–931. doi:10.1038/nrm2061

Nishikura, K. (2010). Functions and regulation of RNA editing by ADAR deaminases. *Annu. Rev. Biochem.* 79, 321–349. doi:10.1146/annurev-biochem-060208-105251

Patterson, J. B., and Samuel, C. E. (1995). Expression and regulation by interferon of a double-stranded-RNA-specific adenosine deaminase from human cells: evidence for two forms of the deaminase. *Mol. Cell. Biol.* 15, 5376–5388.

Paul, M. S., and Bass, B. L. (1998). Inosine exists in mRNA at tissue-specific levels and is most abundant in brain mRNA. *EMBO J.* 17, 1120–1127. doi:10.1093/emboj/17.4.1120

Peng, Z., Cheng, Y., Tan, B. C.-M., Kang, L., Tian, Z., Zhu, Y., et al. (2012). Comprehensive analysis of RNA-Seq data reveals extensive RNA editing in a human transcriptome. *Nat. Biotechnol.* 30, 253–260. doi:10.1038/nbt.2122

Picardi, E., Gallo, A., Galeano, F., Tomaselli, S., and Pesole, G. (2012). A novel computational strategy to identify A-to-I RNA editing sites by RNA-Seq data: de novo detection in human spinal cord tissue. *PLoS One* 7:e44184. doi:10.1371/journal.pone.0044184

Pickrell, J. K., Gilad, Y., and Pritchard, J. K. (2012). Comment on "widespread RNA and DNA sequence differences in the human transcriptome". *Science* 335, 1302. doi:10.1126/science.1210484

Pinto, Y., Cohen, H. Y., and Levanon, E. Y. (2014). Mammalian conserved ADAR targets comprise only a small fragment of the human editosome. *Genome Biol.* 15, R5. doi:10.1186/gb-2014-15-1-r5

Polson, A. G., and Bass, B. L. (1994). Preferential selection of adenosines for modification by double-stranded RNA adenosine deaminase. *EMBO J.* 13, 5701–5711.

Ramaswami, G., and Li, J. B. (2014). RADAR: a rigorously annotated database of A-to-I RNA editing. *Nucleic Acids Res.* 42, D109–D113. doi:10.1093/nar/gkt996

Ramaswami, G., Lin, W., Piskol, R., Tan, M. H., Davis, C., and Li, J. B. (2012). Accurate identification of human Alu and non-Alu RNA editing sites. *Nat Meth* 9, 579–581. doi:10.1038/nmeth.1982

Ramaswami, G., Zhang, R., Piskol, R., Keegan, L. P., Deng, P., O'Connell, M. A. A., et al. (2013). Identifying RNA editing sites using RNA sequencing data alone. *Nat. Methods* 10, 128–132. doi:10.1038/nmeth.2330

Rebagliati, M. R., and Melton, D. A. (1987). Antisense RNA injections in fertilized frog eggs reveal an RNA duplex unwinding activity. *Cell* 48, 599–605. doi:10.1016/0092-8674(87)90238-8

Riedmann, E. M., Schopoff, S., Hartner, J. C., and Jantsch, M. F. (2008). Specificity of ADAR-mediated RNA editing in newly identified targets. *RNA* 14, 1110–1118. doi:10.1261/rna.923308

Rueter, S. M., Dawson, T. R., and Emeson, R. B. (1999). Regulation of alternative splicing by RNA editing. *Nature* 399, 75–80. doi:10.1038/19992

Sakurai, M., Ueda, H., Yano, T., Okada, S., Terajima, H., Mitsuyama, T., et al. (2014). A biochemical landscape of A-to-I RNA editing in the human brain transcriptome. *Genome Res.* 24, 522–534. doi:10.1101/gr.162537.113

Sakurai, M., Yano, T., Kawabata, H., Ueda, H., and Suzuki, T. (2010). Inosine cyanoethylation identifies A-to-I RNA editing sites in the human transcriptome. *Nat. Chem. Biol.* 6, 733–740. doi:10.1038/nchembio.434

Solomon, O., Bazak, L., Levanon, E. Y., Amariglio, N., Unger, R., Rechavi, G., et al. (2014). Characterizing of functional human coding RNA editing from evolutionary, structural, and dynamic perspectives. *Proteins* 82, 3117–3131. doi:10.1002/prot.24672

Su, A. A. H., and Randau, L. (2011). A-to-I and C-to-U editing within transfer RNAs. *Biochemistry(Mosc.)* 76, 932–937. doi:10.1134/S0006297911080098

Tomaselli, S., Locatelli, F., and Gallo, A. (2014). The RNA editing enzymes ADARs: mechanism of action and human disease. *Cell Tissue Res.* 356, 527–532. doi:10.1007/s00441-014-1863-3

Wagner, R. W., Smith, J. E., Cooperman, B. S., and Nishikura, K. (1989). A double-stranded RNA unwinding activity introduces structural alterations by means of adenosine to inosine conversions in mammalian cells and *Xenopus* eggs. *Proc. Natl. Acad. Sci. U.S.A.* 86, 2647–2651. doi:10.1073/pnas.86.8.2647

Wahlstedt, H., and O'Hman, M. (2011). Site-selective versus promiscuous A-to-I editing. *Wiley Interdiscip Rev RNA* 2, 761–771. doi:10.1002/wrna.89

Wong, S. K., Sato, S., and Lazinski, D. W. (2001). Substrate recognition by ADAR1 and ADAR2. *RNA* 7, 846–858. doi:10.1017/S135583820101007X

Yang, J. H., Sklar, P., Axel, R., and Maniatis, T. (1997). Purification and characterization of a human RNA adenosine deaminase for glutamate receptor B pre-mRNA editing. *Proc. Natl. Acad. Sci. U.S.A.* 94, 4354–4359. doi:10.1073/pnas.94.9.4354

Conflict of Interest Statement: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Received: 14 November 2014; accepted: 07 February 2015; published online: 24 February 2015.

Citation: Nigita G, Alaimo S, Ferro A, Giugno R and Pulvirenti A (2015) Knowledge in the investigation of A-to-I RNA editing signals. Front. Bioeng. Biotechnol. ***3****:18. doi: 10.3389/fbioe.2015.00018*

This article was submitted to Bioinformatics and Computational Biology, a section of the journal Frontiers in Bioengineering and Biotechnology.

Advantages of publishing in Frontiers

OPEN ACCESS
Articles are free to read, for greatest visibility

COLLABORATIVE PEER-REVIEW
Designed to be rigorous – yet also collaborative, fair and constructive

FAST PUBLICATION
Average 85 days from submission to publication (across all journals)

COPYRIGHT TO AUTHORS
No limit to article distribution and re-use

TRANSPARENT
Editors and reviewers acknowledged by name on published articles

SUPPORT
By our Swiss-based editorial team

IMPACT METRICS
Advanced metrics track your article's impact

GLOBAL SPREAD
5'100'000+ monthly article views and downloads

LOOP RESEARCH NETWORK
Our network increases readership for your article

Frontiers
EPFL Innovation Park, Building I • 1015 Lausanne • Switzerland
Tel +41 21 510 17 00 • Fax +41 21 510 17 01 • info@frontiersin.org
www.frontiersin.org

Find us on