k-mer based analysis for cancer transcriptomics at nucleotide resolution

Haoliang Xue

université
PARIS-SACLAY

k-mer based analysis for cancer transcriptomics at nucleotide resolution

Analyse de k-mers pour la transcriptomique du cancer à la résolution du nucléotide

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 577, Structure et Dynamique des Systèmes Vivants (SDSV)
Spécialité de doctorat: sciences de la vie et de la santé
Unité de recherche: Université Paris-Saclay, CEA, CNRS, Institute for Integrative Biology of the Cell (I2BC), 91198, Gif-sur-Yvette, France.
Graduate School: Sciences de la vie et santé. Référent: Faculté des sciences d'Orsay

Thèse présentée et soutenue à Paris-Saclay, le 13 décembre 2021, par

Haoliang XUE

Composition du jury

Fariza TAHI Professeure, Université Evry Val d'Essonne	Présidente
Tatiana POPOVA Chercheure HDR, Institut Curie	Rapportrice & Examinatrice
Denis PUTHIER MCF HDR, Aix-Marseille Université	Rapporteur & Examinateur
Rayan CHIKHI Chercheur HDR, Institut Pasteur	Examinateur
Alain DENISE Professeur, Université Paris-Saclay	Examinateur

Direction de la thèse

Daniel GAUTHERET Professeur, Université Paris-Saclay	Directeur de thèse

Thèse de doctorat

NNT: 2021UPASL101

Acknowledgements

I would like to firstly express my most sincere thankfulness to my supervisor, Prof. Daniel Gautheret, for his guidance and instructions during my PhD study. He taught me not only much in biology, but also many issues related to computer science and statistics. Also, he introduced me into many collaboration projects, and patiently guided me in article and thesis writing. Without his help, I could never finish this thesis or the articles. Actually, I have been in his guidance since 5 years ago when I did my internship in his group as a master's student. To me, he is at the same time a great scientist with serious and open-minded attitude, and a great teacher with great kindness and patience as well as broad knowledge. Actually, he sets an example for me, and I wish I could do as well as him when I guide students one day in future.

I would also like to express great gratitude to my colleagues Dr. Claire Toffano-Nioche, Dr. Fabrice Leclerc, Dr. Mélina Gallopin, Dr. Gilles Vergnaud, Dr. Christine Pourcel, Dr. Maria Costa, Dr. Thi Ngoc Ha Nguyen, Dr. Yunfeng Wang, Antoine Lainé, Hugues Herrmann, Roy Gonzalez-aleman, Taher Yacoub, and Coline Gardou. Claire and Fabrice helped me a lot in bioinformatics, such as FAIR issues, Linux system manipulation, and cluster computing. I also learnt with them much in French culture about food, festival, and other aspects. Mélina always greatly supported on statistical aspects, and I learnt a lot from her on the statistical and machine-learning models and sequencing data simulation. With Ha, Yunfeng, Antoine, and Hugues, we had great collaborations and discussions in different projects. Coline did an impressive work on application of *KaMRaT* software in her 5-month internship. I enjoyed various fascinating things from different presentations in lab meeting, especially Gilles' talks on bacteria's geographical and historical propagation. I also enjoyed greatly lab's TGIFs. It is really a pleasure memory working in our friendly and motivated team.

Greatly thanks my committee meeting members Dr. Rayan Chikhi and Prof. Alain Denise. They proposed many valuable suggestions to my project, my thesis, and my presentations. I acquired a lot from these.

I would also like to thank a lot our collaborators. In the ANR TranSiPedia project: Prof. Thérèse Commes, Dr. Anthony Boureux, Dr. Florence Rufflé, Dr.

Chloé Bessière, Dr. Benoit Guibert, and Dr. Sébastien Riquier in Université de Montpellier; Dr. Rayan Chikhi, Dr. Mikaël Salson, and Dr. Camille Marchet in Institut Pasteur and Université de Lille. We collaborated in different aspects, including application of *REINDEER*, *Kmerator Suite*, and *KaMRaT*. Also, I enjoyed a lot the meetings in which I learnt a lot. The food during the meetings were also very nice. In the article *Reference-free transcriptome signatures for prostate cancer prognosis*, we collaborated with Dr. Yann Ponty in Ecole Polytechnique, and I learnt many interesting ideas about machine learning from him. We collaborated with Dr. Antonin Morillon and Dr. Marc Gabriel for urine RNA-seq data analysis. We had many informative communications.

Thanks Dr. William Ritchie and Dr. Claudio Lorenzi for their kind help in my understanding and application of *iMOKA* software. They suggested also the using of *MLPack* library for *KaMRaT*, which turned out to be very useful.

Also thanks my thesis reviewers Dr. Tatiana Popova and Dr. Denis Puthier, who provided many great and detailed remarks and suggestions for improving this thesis writing.

Thanks my families and friends for their continuous support. My grandfather told me occasionally the new discoveries related with gene research that he learnt from newspaper, including information about the newly completed human reference genome. My strength always comes from them whenever I meet difficulties.

Finally, thanks ANR for funding my PhD study and thanks CNRS for the 44-day vacation each year. I had a really enjoyable life during the three years.

Contents

Chapter 1

Basic Notions and Concepts

Bioinformatics is a multidisciplinary domain of biology, statistics, and computer science. It uses computational technologies and applies statistical methods for solving biological problems.

This PhD thesis concerns the development and application of novel technologies in the study of transcriptome at nucleotide resolution, including software for retrieval of biological sequences relevant to the research subject, and for arbitrary sequence indexing and querying.

This first chapter aims to give basic notion and concepts in the related fields. Section 1.1 include basic concepts related to gene expression analysis; section 1.2 presents related technology for determining and measuring human genome and transcriptome, as well as The Human Genome Project; section 1.3 is about cancer genomics and transcriptomics; sections 1.4 and 1.5 includes important and related concepts respectively in statistics and in data science; and finally section 1.6 involves related algorithms and data structures from computer science.

1.1 Gene, Gene expression, and Transcriptome

1.1.1 Genome as an array of genes

From a modern point of view, a gene is a segment of genome which is itself a long deoxyribonucleic acid (DNA) sequence formed by 4 types of nucleotide: adenine (A), cytosine (C), guanine (G), and thymine (T). The genome is the hereditary basis of all living organisms. Physically, it is divided into multiple chromosomes (or a single chromosome in most bacteria). Functionally, it is divided into multiple genes which locate linearly on chromosomes. Each gene encodes one or several molecules of ribonucleic acid (RNA) through "transcription"; and ultimately in many cases, polypeptides are further synthesized from these RNAs by "transla-

tion" (see section 1.1 in [Krebs et al., 2017]).

Though the gene's modern definition relies on DNA, it was discovered as early as 1865 - years ahead of DNA's heredity nature being uncovered - by Gregor Mendel. Mendel applied pure statistical methods on the pea phenotype data collected from experiments over eight years, and managed to predict the existence of gene (called as "factor" at the moment) as well as established two fundamental laws of inheritance [Mendel, 1865]. An interesting point is, Mendel's concept is inherited till today to a certain degree: though we now know much more about the molecular basis behind gene expression procedure, the "gene" concept is still largely considered solely as an abstraction of the "functional unit", independent from its material.

In eukaryotes, genes can be split into multiple parts. The parts that remain in the mature RNA are called exons, and the intervening parts are called introns. Introns are usually **spliced** during the transcription step, but they can also be preserved from splicing in some circumstances (intron retention). Messenger RNAs (mRNAs), which contain a coding sequence, also contain non-coding regions at their 5' and 3' termini. The 5' end contains a cap structure that affects mRNA stability, splicing, export, and translation. The 3' end is terminated by poly-A tail which is added post-transcription and is involved in controlling mRNA stability and influencing translation (sections 3.1, 19.2, 19.12, 19.15, 19.16 in [Krebs et al., 2017]).

Besides mRNAs, a large number of genes produce long intergenic non-coding RNAs (lincRNAs). Note that the term "intergenic" here should rather be understood as "inter protein-coding genes", since the genetic units producing them are actual genes. Most lincRNAs have no clear function yet, however some have regulatory functions [Ding et al., 2014]. Coding and non-coding genes occupy only a subset of the whole genome. According to [Francis and Wörheide, 2017], genes only fill 50.2% of genome in human. The remaining parts are thus really "intergenic". These regions also contain important elements, such as proximal (promoters) and distal (enhancer and silencers) regulatory regions [Takai and Jones, 2004, Glinskii et al., 2011, Riethoven, 2010].

1.1.2 Mutations alter gene function

Mutations exist in all organisms, resulting from either normal cellular metabolism or random interactions with environment. Point mutations - alteration of a single DNA base pair - are most often caused by incorrect repair of chemical modifications of DNA or errors introduced during DNA replication. Mutations can also be insertions/deletions of short sequences - caused by DNA repair, incorrect recombination, transposition events, etc. (see sections 1.11, 1.12 in [Krebs et al., 2017]) Point mutations result in Single Nucleotide Variant (SNV), and when a mutation

is shared by a fraction of the population (generally more than 1%), it is considered as Single Nucleotide Polymorphism (SNP). The insertion or deletion of short sequence is often abbreviated as indel.

Mutations affect gene function through complex interaction mechanisms. So called "forward" mutations alter a gene, while "back" mutations restore the original function of an altered gene, and suppression mutations circumvent the effect of mutations in another gene (see section 1.13 in [Krebs et al., 2017]).

1.1.3 The transcriptome comprises a variety of RNA variations

The transcriptome includes the full set of RNA transcripts, no matter coding or non-coding. It summarises all events originating from genetic alterations, transcription initiation, and post-transcriptional modifications (see figure 2 in [Morillon and Gautheret, 2019]). All of these may have potential impacts on human health. For instance SNVs and gene fusions are importantly related to cancer development [Roberts et al., 2013, Mitelman et al., 2007, Sveen et al., 2016], transcription initiation through enhancers can regulate cell fate decision [Xu et al., 2021], alternative splicing is relevant to cancer and Alzheimer's disease [Sveen et al., 2016, Biamonti et al., 2021]. Understanding transcriptome is an absolute requirement for understanding a wide array of biological and medical problems.

1.2 Determination of Human Genome and Transcriptome

1.2.1 Genome sequencing

Genome sequencing targets determination of nucleotide sequence in genome. Up to now in 2021, three generations of sequencing technologies have emerged. Information in the following paragraphs comes mainly from the section 2.7 of [Krebs et al., 2017].

Sanger sequencing Frederick Sanger and his colleagues developed the first widely used method of sequencing - now known as Sanger sequencing - in 1977 [Sanger et al., 1977]. This method required time-consuming steps of gel separation and autoradiography which involved much human labor. Later it was improved using capilarry separation and fluorescent labelling. The typical read length of a Sanger sequencing run varies from 500 to 1,000 bp.

Despite technical improvement to Sanger sequencing, costs remained very high. Sanger sequencing was applied to The Human Genome Project, launched in 1990 and declared complete in 2003. It involved scientific teams from 20 universities and research centers in the US, the UK, Japan, France, Germany, and China, and cost several billion dollars [Collins et al., 2004]. Still, this version had 8% of the genome left unfinished or erroneous. Only recently, in 2021, the T2T Consortium declared completing these gaps using the latest generation of sequencing techniques [Nurk et al., 2021].

Next-generation sequencing (NGS) The Next-Generation Sequencing (NGS) technique, also called second-generation sequencing, was developed from around 2008. The objective was to decrease involved human labor and experiment cost, as well as to increase sequencing speed. A major progress of this technique is that it sequences in a massively parallel way short DNA fragments. This has dramatically decreased sequencing cost - from $100,000,000 (in 2001) to around $1,000 per human genome [Wetterstrand, 2020]. This opened the way to important new projects in genetics and medicine [Pettersson et al., 2009], such as *1000 Genomes Project* [Simpson et al., 2015]. The basic steps of NGS are add-wash-scan, which is presented in detail in the section 1.2.2, paragraph *sequencing by synthesis*. NGS experiment generates relatively short reads. Read lengths vary across sequencing platforms, generally on the 100nc scale.

Third-generation sequencing Third generation techniques were designed for overcoming the main drawbacks of NGS's short reads: misassemblies and gaps in genome assembly tasks, and failure to detect large structural variations [van Dijk et al., 2018].

The major third-generation sequencing technologies include Single-Molecule Real-Time (SMRT) sequencing developed by Pacific Biosciences in 2011, and Nanopore sequencing developed by Oxford Nanopore Technologies in 2014. The SMRT technology can generate sequence reads 10-15k bp long, whereas Nanopore sequencing's read length are dependent on the DNA molecules to be sequenced, which may reach as long as up to about 1M bp [van Dijk et al., 2018].

Another advantage of both technologies is that they avoid the Polymerase Chain Reaction (PCR) step in the NGS library preparation which may result in regions of extreme GC% being inefficiently amplified [van Dijk et al., 2018].

Third-generation sequencing technologies still suffer from more frequent sequencing errors than short read NGS. SMRT has about 13% of single-pass error rate, though this can be alleviated by sequencing the molecules multiple times. Oxford nanopore suffers from around 15% of error rate, and does not support sequencing the same strand more than once. However, Oxford Nanopore error rates

are considerably reduced after multiple alignment of reads from the same locus [van Dijk et al., 2018]

1.2.2 Transcriptome profiling

Genome sequences are only "blueprints" for potential gene-expression. Although a recent deep learning study indicates that gene expression prediction from DNA sequence alone could be possible [Avsec et al., 2021], analyzing gene expression activity usually requires transcriptome profiling, i.e., detecting all mRNAs and non-coding RNAs (ncRNAs) and then measuring their abundance in the organism.

DNA microarray A DNA microarray [Schena et al., 1995] identifies and measures mRNAs through hybridization. The basic idea is to attach a series of individual DNA sequences of interest on a chip (microarray) for capturing target mRNAs. mRNAs extracted from a specimen are firstly converted into complementary DNAs (cDNAs) by reverse transcription with labelled nucleotides. One labelling strategy - called direct labeling - involves fluorophores. Labelled cDNAs are then hybridized to the microarray, followed by washing. Fluorescent signals are measured at each microarray spot and used as a proxy for gene expression level (section 2.10 of [Krebs et al., 2017]).

NGS RNA-seq experiment From 2008, DNA microarrays were gradually (but not entirely) superseded by the NGS RNA-seq method. RNA-seq enabled a more complete and precise capture of the transcriptome. Rather than relying on predefined list of target sequences, RNA-seq captures the whole set of polyadenylated or total RNAs in given samples. Also, it operates at single base resolution, by really sequencing transcripts nucleotide by nucleotide instead of identifying them via hybridization.

RNA-seq can be applied to bulk tissue samples or to single cells. Bulk RNA-seq sequences a mixture of cells of each sample, while the recently developed single-cell RNA-seq applies cell separation techniques to capture and sequence RNA in individual cells. This thesis focus on bulk RNA-seq analysis.

Experimental design is essential prior to any RNA-seq experiment. One important but sometimes ignored point is that the experiment should avoid confounding factors if the samples are processed in multiple batches. This means each batch should contain every experimental condition. This provides necessary information for downstream computational methods to reduce these artifactual differences across batches, though in some cases this information is still not sufficient to remove all the batch effects or may impair proper analysis of the data. Another point to consider is the allocation of budgets to number of replicates and the depth of

sequencing. The budget may be better spent on replicates when performing differential expression analysis [Liu et al., 2014]. The work by [Schurch et al., 2016] suggests replicate number for each condition should be no less than six, and would be ideally as many as twelve to have a complete identification of significantly differentially expressed genes for any fold changes. Furthermore, when samples are heterogeneous, such as when they come from mixed biopsies or individuals with distinct genetic backgrounds, much larger sample sizes are required. When reconstruction of genomic structure is targeted, however, it may be favorable to spend budget for better sequence depth.

The first step of an RNA-seq experiment is RNA extraction and purification. RNAs are first separated from DNAs and proteins. This total RNA fraction contains rRNAs and tRNAs that are usually not relevant to the gene-expression research. Therefore, two methods - polyadenylated (polyA+) (that favors mRNAs) and ribosomal RNA-depleted (ribo-) (that captures all mRNAs and ncRNAs) - are usually applied for purification. Currently, the polyA+ method is more generally used, but it misses some relevant RNA species, especially for ncRNAs [Cui et al., 2010].

After extraction and purification, RNAs are fragmented and the fragmented RNAs are reverse-transcribed to double-stranded cDNAs. At the cDNA preparation step, adapter sequences are ligated to the 3' and 5' ends. Finally, the cDNA library is amplified by PCR for enhancing signals.

For Illumina sequencing, the amplified cDNAs are bound on the sequencing support (a "flow cell") to short oligonucleotides complementary to the ligated adapter sequences, and then sequenced with fluorescently labeled deoxynucleoside triphosphate (dNTP), in a stepwise fashion. After a dNTP is added, the fluorescent label acts as a terminator and thus prevents other dNTPs from being appended. Then, an image is taken for capturing fluorescent signals, and inferring the layer of newly added dNTP types. Labels are then cleaved for adding another layer of dNTPs.

The sequencing step introduces errors. Termination by fluorescent labels is not perfect. It is not rare that more than one dNTPs are added inside a single step. The pairing itself is not perfect either. So, each fluorescent image contains noises. However, since cDNAs are amplified into clusters before sequencing, this error is largely reduced since each dNTP is inferred based on a cluster of signals. Also in this way, a sequencing score can be evaluated and recorded for each base, allowing downstream computational filtering. Besides, sequence error occurs more easily towards the end of each fragments.

Illumina sequencing is performed in two different modes: single-end and paired-end. In single-end sequencing, each cDNA is sequenced from only one end; whereas in paired-end mode, it is sequenced by both ends. There is also two different

protocols of RNA-seq: unstranded or stranded. The former ignores information about the coding strand while the latter preserves it. The stranded information is sometimes required, for example it may help in detecting antisense RNAs or distinguishing between an extended 5' region or a TSS-associated antisense transcript.

1.3 Cancer Genomics and Transcriptomics

Early microarray cancer transcriptomics In the previous era of microarrays, transcriptome analysis was already decisive in understanding cancer pathways and defining cancer subtypes, as examplified by the seminal work by [Golub et al., 1999] for leukemia subtype classification. These authors measured the expression profiles of 6817 genes using DNA microarrays, based on which they targeted two types of problems in leukemia subtype classification: (i) class discovery for identification of previously unrecognized tumor types, and (ii) class prediction for assigning particular tumor samples to already-defined classes. This work established a 50-gene predictor that successfully diagnosed leukemia known subtypes, and a two-cluster self-organizing map that grouped leukemia patients into two subgroups without using information about sample condition, from which the samples were accurately clustered to the known class labels. These early studies led to a booming field with multiple applications in cancer diagnosis and prognosis, up to the commercial breast cancer tests *MammaPrint* [Van't Veer et al., 2002], and *Oncotype DX* [Paik et al., 2004].

RNA-seq for cancer transcriptomics RNA-seq, with its comprehensive and accurate capture of RNAs, promised to improve cancer transcriptome analysis. Apart from the information retrieved from gene-level analysis, researchers have found, with RNA-seq data, multiple types of local event signals relevant to cancer. These include but not limit to: some SNVs, indels, gene fusions, and alternative splicing sites that can all act as driver events [Seo et al., 2012]. RNA-seq was considered so valuable that it became a major component, together with DNA sequencing, of all major cancer genomics projects.

The Cancer Genome Atlas (TCGA) The most important cancer genomics project to date is TCGA, funded by NIH. It aims at understanding of the molecular basis of cancer in a pan-cancer perspective, to identify genomic similarities across tumors regardless of tissue or organ of origin [Cline et al., 2013]. An array of methods are applied to each cancer sample, including RNA-seq, whole exome sequencing, proteomics, methyl-array or methyl-seq and microscopy. In 2021, TCGA had sequenced 33 cancer types over 20,000 samples.

Cancer Cell Line Encyclopedia (CCLE) The CCLE project performed systematic genomic profiles of over 1000 cell lines [Barretina et al., 2012]. Overall, 1019 cell lines were analyzed by RNA-seq, 326 by whole-exome sequencing, and 329 by whole-genome sequencing [Ghandi et al., 2019]. The CCLE dataset has helped finding drugs matching the molecular features of each cell type, for precision medicine applications [Sheng et al., 2015].

1.4 Important Statistical Concepts for Transcriptomics

Statistics can be divided into two broad categories: (i) descriptive statistics which summarizes information inside a data set, and (ii) inferential statistics which infers the general properties beyond the given data set.

Statistics is closely joined with biological data since the very beginning. As mentioned in 1.1 Gene, Gene expression, and Transcriptome, Mendel already applied statistics (more precisely, descriptive statistics) to predict the existence of genes without relying on their molecular basis. Today in transcriptome research, one largely applied method is hypothesis testing (which belongs to inferential statistics). This thesis concerns mainly inferential, rather than descriptive, statistics.

1.4.1 Fundamental notions

Population and sample A population is the full set of individuals that are relevant to a certain study. For instance, when studying prostate adenocarcinoma in human, the complete set of patients with this disease is considered as the population.

The definition of a sample varies between biologists and statisticians. In biology, a sample is **an individual** extracted from the population of interest, while in statistics, a sample is **a collection of individuals** that obtained from the population. To avoid ambiguous term usage in this thesis, we always take the biologist's sample definition regardless of the context, and we use plural form of the word qualified by "group" or "condition" when statistician's version is required. As an example, we say "a sample of prostate adenocarcinoma" to indicate an individual patient, and "the group of prostate adenocarcinoma samples" to indicate all patients.

Population parameters Population parameters describe properties in a population. Since obtaining data from all the population from the past to now would

never be possible, the true population property (e.g. 10-year survival rate for lung adenocarcinoma) is intractable. Statisticians estimate parameters from an obtained group of samples, and use these as proxy to represent population properties. The parameters can be statistics such as mean, median, etc., as well as adjusted coefficients in a model (e.g. logistic regression's coefficients).

Inferential statistics Inferential statistics attempts to estimate properties or uncover patterns in a population by studying a number of samples from this population. Keeping with the above example, researchers may apply inferential statistics for searching differential genes between prostate adenocarcinoma patients and healthy people (two populations), by comparing gene expression profiles of two groups sampled from the two populations. The finding is a list of genes that could be relevant to prostate adenocarcinoma oncogenesis.

1.4.2 Some common probability distributions

Binomial distribution The binomial distribution is related with discrete variables. It describes the number of success occurrence among a known total number of Bernoulli trials (can be success or failure), given that the success occurs independently at a constant probability. Its formula is shown as equation 1.1.

$$P(x; n, p) = \frac{n!}{x!(n-x)!} p^x (1-p)^{(n-x)}, \ (x = 0, 1, 2, ..., n) \tag{1.1}$$

, where parameter p is the probability of success, and n is the total number of experiments.

Poisson distribution The Poisson distribution is related to discrete variables. It describes the number of events occurring in a given period or volume, given that the events occur independently with a constant probability. The Poisson distribution can be formulated as equation 1.2.

$$P(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!}, \ (x = 0, 1, 2, ...) \tag{1.2}$$

, where parameter λ is the mean number of events occurring in a fixed time which equals the variance.

Negative binomial distribution The negative binomial distribution is related to discrete variables. It describes the number of failure trials before obtaining a target number of successes in a set of Bernoulli trials (can be success or failure) with a same probability of success. The formula is described by equation 1.3.

$$P(x; r, p) = \frac{(r+x-1)!}{(r-1)!x!} p^r (1-p)^x, \ (x = 0, 1, 2, ...) \tag{1.3}$$

, where parameters r and p respectively denote the target number and the probability of success trial.

Also, the negative binomial distribution can be used to inversely describe the number of successes before a certain number of failures.

The negative binomial distribution is widely applied in RNA-seq data modeling, for example in *DESeq2* [Love et al., 2014] and *edgeR* [Robinson et al., 2010].

Normal distribution The normal distribution is related to continuous variables. It is widely applied in statistical inference. Its formula follows as equation 1.4 (see section 5.3 in [McClave and Sincich, 2018]).

$$f(x; \mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{1.4}$$

According to the **Central Limit Theorem** (see Theorem 6.2 in section 6.3 of [McClave and Sincich, 2018]), in a large sampling (i.e., sample number is large) from **any** population where a parameter's mean and standard deviation are respectively μ_p and σ_p, the distribution of $\bar{x}$ (mean value of the concerned parameter estimated from the sample set) will follow a normal distribution $f(x; \mu_p, \sigma_p)$.

Student's t-distribution The Student's t-distribution is related to continuous variables. The Student's t-distribution is applied in Student's t-tests (see section 1.4.3 Hypothesis testing between two sample groups).

1.4.3 Hypothesis testing between two sample groups

A widely applied approach in gene-expression analysis is to compare two sample groups labelled with different conditions, and then select a list of genes that distinguish one condition from the other. These genes are then termed "differentially expressed". This involves hypothesis testing between these two condition groups.

Statistical hypotheses A statistical hypothesis is a statement about the numerical value of a population parameter. In the hypothesis testing methodology, one sets a **null hypothesis** assumed to be true unless the data provide convincing evidence against it. The testing task is applied on this null hypothesis. One also needs an **alternative hypothesis** which can be the negation of the null hypothesis, and thus will be accepted if the null hypothesis is rejected (see section 8.1 in [McClave and Sincich, 2018]).

Test statistic To estimate the "convincing" level of evidence for rejecting the null hypothesis, a **test statistic** is computed. Information in this paragraph comes from section 8.5 in [McClave and Sincich, 2018].

In the situation where sample number is limited, a simple example of hypothesis testing is to use Student's t-statistic, formulated in equation 1.5, to test whether the parameter's mean equals to a supposed value.

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}} \tag{1.5}$$

, where $\bar{x}$ is the mean value of parameter x, μ_0 is the population mean under the null hypothesis, s is the parameter's standard deviation across samples, and n is the sample number.

The t-statistic by definition estimates the mean value of a parameter x which follows normal distribution in population. However, the Central Limit Theorem does not apply here, since the number of samples is limited as relatively small. It has been shown that in this situation, t-statistics follows the t-distribution.

Rejection of null hypothesis with p-value When the computed statistic value drops into the zone of "unlikely happening", i.e., the rejection region, the null hypothesis is rejected. A measurement of the confidence of rejecting the null hypothesis is called p-value. It indicates the probability of observing a test statistic value at least as extreme as the one computed from the samples, under the assumption that the null hypothesis is true. The smaller is the p-value, the more confident one is to reject the null hypothesis (see sections 8.2 and 8.3 in [McClave and Sincich, 2018]).

The maximum threshold of p-value for rejecting the null hypothesis is called "significance level" and often denoted as α. One rejects the null hypothesis when the p-value $< \alpha$. As a generally accepted convention, one chooses $\alpha = 0.05$. Still, whether one should accept this one-fit-all value mindlessly is a long standing debate [Yaddanapudi, 2016].

Two types of errors Rejection of the null hypothesis may introduce two types of errors. Type I error occurs when the null hypothesis is rejected while it is actually true. Type II error occurs when it is accepted but is actually false (see section 8.1 in [McClave and Sincich, 2018]).

Multiple testing problem The above discussion concerns a single comparison. In the actual practice of gene-expression analysis, however, multiple genes are considered in parallel. This causes a multiple testing problem where the type I error dramatically augments. By fixing an α, we control that the probability of

the occurrence of type I error equals α for a single test. However, when multiple comparisons are involved, for example $c > 1$ comparisons, this probability becomes $(1 - (1 - \alpha)^c) > \alpha$ (see section 10.3 of [McClave and Sincich, 2018]).

The greater the number of comparison, the more likely the type I error occurs. The probability of this occurrence can even approaching 1 when the comparison number grows very large. Here we show in Figure 1.1 the relationship between the probability of type I error and the number of comparisons, under $\alpha = 0.05$.

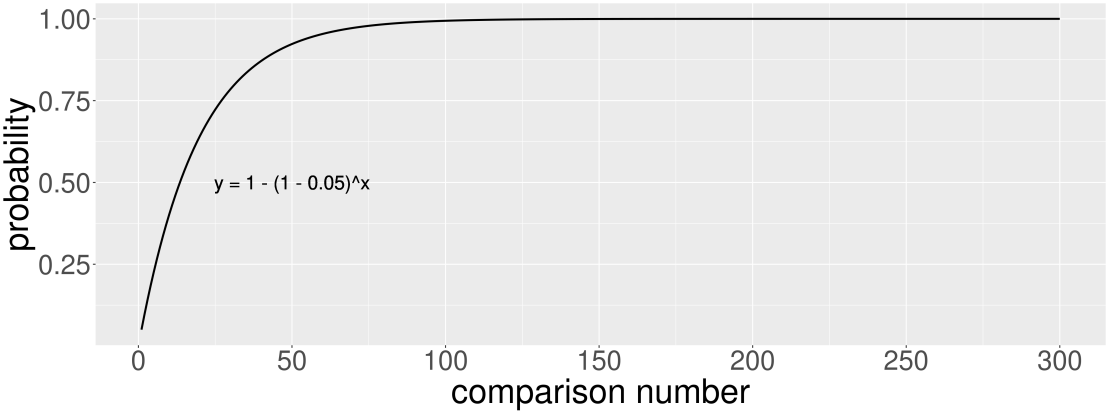


Figure 1.1: Relationship between occurrence of type I error and number of comparisons, from 1 to 300, under the situation of $\alpha = 0.05$.

Benjamini-Hochberg procedure for multiple testing correction Different strategies exist for alleviating the problem of multiple testing. A popular one is to follow the Benjamini-Hochberg procedure [Benjamini and Hochberg, 1995].

1. Rank the testing hypothesis $H^{(1)}, H^{(2)}, ..., H^{(m)}$ with their raw p-values from low to high ($P_1 \leq P_2 \leq ... \leq P_m$);
2. Find the k subject to $P_k \leq \frac{k}{m}\alpha$;
3. Reject all $H^{(i)}$ with $i = 1, 2, ..., k$.

They proved that the probability of type I error occurrence by this procedure is controlled under α [Benjamini and Hochberg, 1995].

Application in gene-expression analysis In gene-expression analysis with consideration about expression difference between two sample groups, we suppose that the expression levels of a given gene g in populations of the two conditions

are respectively μ_{g1} and μ_{g2}, and we want to know if $\mu_{g1} = \mu_{g2}$ for the gene [Jeanmougin et al., 2010].

Obviously, as genes are compared one by one, this involves a multiple comparison problem with the comparison number c equal to gene number (e.g., around 50,000 in human). So, the control of type I error by for instance Benjamini-Hochberg procedure is required.

Various strategies allow hypothesis testing in gene-expression data, such as Welch's t-test and Wilcoxon signed-rank test. There are also some programs, including *edgeR* [Robinson et al., 2010], *DESeq2* [Love et al., 2014], and *Limma* [Ritchie et al., 2015] that implement various strategies for this task.

1.5 Important Data Science Concepts for Transcriptomics

Data science is a much younger domain existing since several decades, comparing to statistics which dates back to centuries ago (though multiple testing correction is a recent topic developed in the 20th century). Data science is actually closely related to statistical science, but with extensive use of computational methods instead of statistical theories. Machine learning plays a key role in data science applications, whereby an algorithm attempts to automatically retrieve patterns from data.

Machine learning methods mainly have two strategies: (i) supervised learning where the algorithm trains a model from samples' independent input variables and their known dependent output variables, and uses this model for prediction when novel samples without known output variables arrive, (ii) unsupervised learning where the algorithm aims to identify patterns or commonalities according to samples' independent input variables without knowing their dependent outputs. Using the example given by [Golub et al., 1999] (also cf. section 1.3, paragraph *Early microarray cancer transcriptomics*), the supervised learning strategy corresponds to the "class prediction" problem, and the unsupervised learning strategy correspond to the "class discovery" problem. A supervised learning problem can be divided into two categories: (i) a regression problem where the outputs are quantitative values, (ii) a classification problem where outputs are qualitative values (section 2.2 in [Hastie et al., 2010]). This thesis mainly concerns classification problems under a supervised strategy.

1.5.1 Fundamental notions

Feature, feature vector, and feature space A feature is one characteristic or property of an individual (a sample). A feature may have a value, and values of

all features compose a feature vector. The space generated by all feature vectors is called the feature space. Each sample is presented as a point in feature space. In the example of gene-expression analysis, each gene is a feature. Therefore, given that human has 50,000 genes (not exactly 50,000, but just a temporary simplification here), each sample can be modelled with a feature vector of 50,000 values. The dimensionality of feature space is also 50,000.

Supervised learning in classification As mentioned above, a supervised learning algorithm trains a model from data with known output values for future prediction. As a classification example, we have a list of samples labelled either as normal tissue or prostate adenocarcinoma tissue, and each sample is associated with a feature vector of gene expression values. In this situation, supervised-learning classification takes the gene expression matrix (a set of gene expression vectors of samples) with the sample label, and attempts to train a model that distinguishes tumor from normal tissue (so, "supervised"). Then the model can be used for predicting tissue labels when new data without known labels come in future.

1.5.2 Compositional data analysis

Information of this section comes mainly from the publication [Quinn et al., 2018].

Another important consideration is that current transcriptome research concerns usually compositional data - especially for NGS techniques of which the library size depends on the chemistry of the assay rather than the input material. Irrelevant sizes of specimens always require scale transformations, and the results of these transformations are relative values or portions. Compositional data are associated with two unique properties: (i) the sum of all values in each library is an arbitrary artifact and (ii) the difference between these values is meaningful only proportionally.

Some consequences of compositional data include: (i) distance between two features (e.g. genes) can be erratically sensitive to the presence/absences of other features (e.g., other genes), which introduces noises in classification; (ii) correlation may indicate false association between irrelevant features [Lovell et al., 2015]; (iii) multivariate statistics may be problematic, since the variables as portions are not independent from each other.

Normalization Compositional data analysis requires a normalization step. In the simplest situation, this is done by rescaling counts by library size; however, this rescaling manipulation does not change the compositional nature of data (also said as "it cannot reopen the closed data"). Other methods attempt to reopen the data

by inferring an ideal reference from a subset of features across conditions. Methods for computing this reference value include trimmed mean of M-values [Robinson and Oshlack, 2010] and median over the transcripts [Anders and Huber, 2010]. Still, given that identifying a truly unchanged reference is difficult, this approach may not be a prior way in general practice to remove data's compositional nature. Besides, normalization may significantly impact analysis results.

The log-ratio transformation Simply applying a log-ratio transformation is an approach for mapping compositional data into real space, thereby making measurements such as Euclidean distances meaningful. One type of log-ratio transformation is the centered log-ratio (clr) transformation, represented as in equation 1.6.

$$clr(\mathbf{x_j}) = \left[\ln \frac{x_{1j}}{g(\mathbf{x_j})}, \ln \frac{x_{2j}}{g(\mathbf{x_j})}, ..., \ln \frac{x_{mj}}{g(\mathbf{x_j})}\right] \tag{1.6}$$

, where $\mathbf{x_j}$ is the j^{th} sample's feature vector, x_{ij} with $i = 1, 2, ..., m$ are m component features of the sample j, and $g(\mathbf{x_j})$ is the geometric mean among components of the vector $\mathbf{x_j}$.

In some context, this transformation acts equivalently as a normalization.

1.5.3 Batch effect correction

As mentioned in section 1.2.2 Transcriptome profiling, artifactual differential signals across batches may severely impact downstream analysis by increasing signal variability, decreasing detection power of real signals, generating false discoveries, and misleading biological/clinical conclusions, even in a perfectly designed study. Therefore, statistical methods are required for removing these batch effects as a preprocessing step [Leek et al., 2010].

Batch effect correction methods summarized in this section come from [Nygaard et al., 2016]. Here, for being clearer, we slightly adapted the article's original notations, modeling a feature value under batch effect as Y_{gbs}, where s indicates a sample, b and g are the sample's corresponding batch and group (condition), respectively, as shown in equation 1.7.

$$Y_{gbs} = \alpha + \beta_g + \gamma_b + \epsilon_{gbs} \tag{1.7}$$

, where α is a constant independent from batch, condition, or sample, β_g relates to sample's condition group, γ_b relates to sample's batch, and ϵ_{gbs} is sample's individual variation.

A naïve method for batch effect removal is zero-centering or one-way ANOVA adjustment. It simply subtracts the mean value of feature measurements among all

samples in the corresponding batch from the feature's raw measurement, expressed as equation 1.8.

$$\tilde{Y}^0_{gbs} = Y_{gbs} - \bar{Y}_b + \bar{Y} \tag{1.8}$$

, where $\bar{Y}_b = \frac{1}{n_b} \sum_{s \in b} Y_{gbs}$ with n_b the number of sample in batch b; $\bar{Y}$ is the mean value of all samples, for readding α.

This method allows removal of the most, but not necessarily all, batch signals, in the ideal situation where all conditions are evenly assigned to all batches. However, when the batch-condition is unbalanced, this may reduce condition differences and reduce statistical power.

An alternative method is to use a two-way ANOVA model, estimating $\hat{\gamma}_b$ by simultaneously considering batch and group condition, and subtract the term from equation 1.7. While this method alleviates the problem of one-way ANOVA in unbalanced batch-condition case, this adjustment may increase differences between condition groups, and lead to an over-confident estimation of group differences.

Other methods were specifically developed for gene-expression data. These include: *ComBat* [Johnson et al., 2007] which implements an empirical Bayes method to microarray expression data, surrogate variable analysis [Leek and Storey, 2007] which is able to use various heterogeneous signal sources, *RUVseq* [Risso et al., 2014] which controls spike-ins from the External RNA Control Consortium, and the recent *ComBat-seq* [Zhang et al., 2020] which extends the original *ComBat* framework for RNA-seq data using negative binomial regression.

1.5.4 Feature dimensionality reduction

Curse of dimensionality An important challenge in the machine learning field is the "curse of dimensionality", which describes the situation where the feature number largely exceeds sample numbers. This is typically true in gene/transcript features of which the number can be in the order of 10^4 or 10^5 in human, but with often less than 100 samples. Even more seriously, in the new k-mer (successive substrings of fixed length k extracted from sequence reads) based approach (see section 2.3.4 k-mer analysis), this feature number can reach 10^9.

Too few sample in the high-dimensional feature space makes the point distribution rather sparse. This creates many problems. For instance the low density of points largely increases the inter-point distance, and poses problems for example in nearest-neighbor methods (section 2.5 in [Hastie et al., 2010]).

Feature selection Feature selection methods are often applied for reducing dimensionality of feature space. According to selection strategies, possible methods are: (i) A filter that applies a univariate examination feature by feature, and keeps

only the most relevant ones for further analysis. In our *KaMRaT* software (see chapter 3), methods implemented in the *rank* module are all of this type. (ii) A wrapper that iteratively evaluates different combination of features with a machine learning algorithm. One typical example of this type is the genetic algorithm. (iii) An embedded method incorporated within the model building step, a typical example of which is random forest based feature selection. [Nguyen, 2020] These feature selection methods only fit into a supervised-learning strategy.

Principal component analysis As an unsupervised-learning method, Principal Component Analysis (PCA) searches the transformation of features that contributes mostly to the variation of data [Clarke et al., 2008].

1.5.5 Common models for classification problems

Information in this section comes from sections 4.4 (logistic regression), 12.2 (SVM), 9.2 (classification tree), and 15.2 (random forest) of [Hastie et al., 2010], and [Zhang, 2004] for naïve Bayes with the formula being equivalently transformed for coherence with others.

Logistic regression Logistic regression applies a linear model to a classification problem. In a binary classification case, the model is specified as in equation 1.9.

$$\log \frac{Pr(G=0|\mathbf{X}=\mathbf{x})}{Pr(G=1|\mathbf{X}=\mathbf{x})} = \beta_0 + \sum_{i=1}^{p} \beta_i x_i \tag{1.9}$$

, where G is the condition of a sample, $\mathbf{X}$ is its feature vector, p is the feature number (dimension of feature space), $\mathbf{x} = [x_1, x_2, ..., x_p]$ is a given known vector of feature values, $\beta_i, i = 0, 1, 2, ..., p$ is a list of parameters.

With the additional fact that the two probabilities should add to 1, this equation 1.9 leads to equation 1.10.

$$Pr(G=0|\mathbf{X}=\mathbf{x}) = \frac{\exp(\beta_0 + \sum_{i=1}^{p} \beta_i x_i)}{1 + \exp(\beta_0 + \sum_{i=1}^{p} \beta_i x_i)} \tag{1.10}$$

, where the symbols have same meaning as above.

Logistic regression can be generalized to a multiple condition classification problem, see section 4.4 of the reference [Hastie et al., 2010] for more detail.

In a supervised classification task, the training procedure aims to estimate parameters $\beta_i, i = 0, 1, ..., p$ through regression, with a list of known samples' feature vectors and their group label $(\mathbf{x_1}, G_1), (\mathbf{x_2}, G_2), ..., (\mathbf{x_n}, G_n)$, where n is number of *a prior* known samples (also called "observations").

Naïve Bayes classifier A Naïve Bayes classifier applies Bayesian inference to perform predictions based on *a prior* known parameters. The term "naïve" assumes that features are independent from each other. Though in theory this is a very strong assumption that rarely holds true in the real world, the method works surprisingly well even when features are interdependent. An explanation is that, dependencies among variables may distribute evenly in each class, or may cancel each other when considered altogether [Zhang, 2004].

The formula of the naïve Bayes method is presented in equation 1.11.

$$\begin{aligned} Pr(G = g|\mathbf{X} = \mathbf{x}) &= \frac{Pr(\mathbf{X} = \mathbf{x}|G = g) \cdot Pr(G = g)}{Pr(\mathbf{X} = \mathbf{x})} \\ &= \frac{\prod_{j=1}^{p} Pr(X_j = x_j|G = g) \cdot Pr(G = g)}{Pr(\mathbf{X} = \mathbf{x})} \end{aligned} \tag{1.11}$$

, where G is the condition of a sample, $G = g$ means the sample belongs to group g; $\mathbf{X}$ is the sample's feature vector, p is the feature number (dimension of feature space), $\mathbf{x} = [x_1, x_2, ..., x_p]$ is a given known vector of feature values.

The denominator of the equation 1.11 is a constant independent from the group labels, while its numerator can be estimated with the given list of samples.

Naïve Bayes classifier fits to multi-condition classification by nature.

Support vector machine A Support Vector Machine (SVM) searches the best hyperplane in the feature space to separate feature points of binary conditions one from the other. In the simplest case where feature points are linearly separable, the hyperplane should locate as far as possible to the points at the boundary of each group (these points are called as "support vectors"). The model can be described as an optimization problem as in equation 1.12.

$$\text{minimize } ||\boldsymbol{\beta}|| \text{ s.t. } y_i(\beta_0 + \sum_{j=1}^{p} x_{ji}\beta_j) \geq 1, i = 1, 2, ..., n \tag{1.12}$$

, where $\boldsymbol{\beta} = [\beta_0, \beta_1, ..., \beta_p]$ are the parameters, $y_i \in \{-1, 1\}$ is the group label of sample i, x_{ji} is the j^{th} component of the feature vector of sample i.

In more complex cases where samples are non-separable linearly, one can either introduce slack variables and slightly modify the constraint (still searching a linear boundary by tolerating error classification, see section 12.2 of [Hastie et al., 2010] for detail), or use Kernel methods (searching a non-linear boundary, see section 12.3 of [Hastie et al., 2010] for detail).

SVM can also be generalized to multi-condition classification, usually via a series of classifications under a $G = g$ vs $G \neq g$ fashion, where $g = 1, 2, ...$ varies across all conditions.

Classification trees and random forest The objective of tree-based classification is to divide the feature space into a set of rectangles with a list of criteria about feature values. One major problem of this method is that it generates high variances across predictions. Because the split criteria is done with a threshold, a slight fluctuation of feature values around the threshold at the top level may cause huge changes in the final classification result.

A solution to the classification tree's high variance problem is to use bagging methods (see section 8.7 in [Hastie et al., 2010]) for reducing this variance, thereby creating a random forest model. The bagging procedure generates a series of noisy but approximately unbiased trees, and final prediction is made by averaging all trees' predictions, thus alleviating the variance of single trees.

1.5.6 Assessment of model's prediction performance

Here we discuss the assessment of model's prediction in a simple binary classification problem. For the multi-classification problem, section 5 of reference [Powers, 2020] presents some generalized ideas.

In the binary classification problem, we consider the sample label as either positive or negative.

Confusion matrix A confusion matrix summarizes the comparison between prediction and reality. The matrix is presented as in Table 1.1. It consists of 4 cases: True Positive (TP) where both the reality and prediction are positive, True Negative (TN) where both reality and prediction are negative, False Positive (FP) where the prediction is positive but the reality is negative, and False Negative (FN) where the prediction is negative but the reality is positive. The TP and TN correspond to correct prediction (blue cases), and the FP and FN correspond to incorrect prediction (orange cases).

Table 1.1: Confusion Matrix

		reality	
		positive	negative
prediction	positive	TP	FP
	negative	FN	TN

Accuracy Accuracy is a simple and straightforward assessment method of prediction performance. It is actually the ratio of correctly classified samples over the total sample number, as shown in equation 1.13.

$$\text{accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{1.13}$$

As a naïve method, accuracy does not perform well with imbalanced data sets, for example, when almost all samples are labeled as positive.

Accuracy can easily be generalized for evaluation of multiple condition classifiers.

Precision and recall Precision is the ratio of true positives over **predicted positives**, as described in equation 1.14.

$$\text{precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{1.14}$$

Recall, also called as sensitivity, is the ratio of true positives over **real positives**, as shown in equation 1.15.

$$\text{recall} = \text{sensitivity} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{1.15}$$

Precision and recall both evaluate how well a classifier handles positive cases (since the numerator is always TP), but do not evaluate the handling of negative cases. Still, these two measurements are widely applied, for instance used in F1-score and precision-recall curves.

F1-score F1-score is the harmonic mean of precision and recall, as shown in equation 1.16.

$$\text{F1} = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} \tag{1.16}$$

A problem of this metric is that it does not have meaning if both precision and recall are 0 (i.e., when $\text{TP} = 0$).

Precision-recall curve A precision-recall curve helps evaluate a classifier's prediction performance by plotting on the x axis recall, and on the y axis precision. The closer is the curve to upper right corner, the better is the classifier. In practice, one can calculate the Area Under the Precision-Recall Curve (PR AUC) for a numerical evaluation.

Sensitivity and specificity Sensitivity and specificity are another pair of metrics that is often used for classifier evaluation. As mentioned above, sensitivity is just another name of recall (equation 1.15). Specificity measures the ratio of true negatives over real negatives, as shown in equation 1.17.

$$\text{specificity} = \frac{\text{TN}}{\text{TN} + \text{FP}} \tag{1.17}$$

Sensitivity and specificity also led to a series of methods for classification evaluation, such as balanced accuracy and Receiver Operating Characteristic curve (ROC curve).

Balanced accuracy Balanced accuracy is the arithmetic mean between sensitivity and specificity, as shown in equation 1.18. This metric can be validly used even when positive and negative sample counts are very imbalanced.

$$\text{balanced accuracy} = \frac{\text{sensitivity} + \text{specificity}}{2} \tag{1.18}$$

ROC curve A ROC curve also combines sensitivity and specificity, showing (1 - specificity) on the x axis and sensitivity on the y axis. The closer is the curve to the upper left corner, the better is the corresponding classifier. An Area Under the ROC Curve (ROC AUC) can be calculated for numerical evaluation. However, a ROC curve can be over-optimistic when sample numbers are imbalanced between conditions.

1.5.7 Split data set for fair model evaluation

Circularity in analysis, equally known as 'double-dipping', is one major problem that is often overlooked during model evaluation. This problem occurs when researchers build a model (including feature selection) on a data set, and then evaluate the model on the same one, yielding false high statistical significance and circular logic [Ball et al., 2020].

A simple method for detecting and avoiding double dipping is to randomly divide the data set into a subset for training and the other for testing. Thereby, the model is built and evaluated on independent data sets. This can be done in permutation for decreasing variability of evaluation results, which is known as cross-validation.

In cross-validation, the data set is randomly split into k sub-groups (i.e., k folds) for k iterations of model construction-evaluation. At each iteration, a training set of $(k-1)$ folds is composed exclusively for feature selection and model construction,

and the remaining fold acts as testing set exclusively for model evaluation. A mean performance may be estimated across the k evaluations as the final result.

Though the problem is relatively straightforward, it is very easily overlooked in practice. According to a recent study by [Quinn, 2021], among 102 articles on human gut microbiome classification, only 12% report a faithful consideration of avoiding this problem. According to our experience, one often unnoticeable pitfall occurs when the analysis integrates a feature selection step followed by a machine learning-model construction. Sometimes the cross-validation is only applied at the model construction step, but not during the feature selection.

1.6 Important Computational Methods for Transcriptomics

1.6.1 Useful data structures for transcriptomics

Hash table A Hash table is designed for efficient search-insertion-deletion operations of (key, value) pairs, which establishes an associative array between keys and values. It relies on a hash function, denoted as h, to map the elements (keys) in target universe U into an element array, as shown in equation 1.19

$$h : U \to \{0, 1, ..., m-1\} \tag{1.19}$$

, where m is the hash table size which is typically much less than the universe size $|U|$.

A main difficulty for handling a hash table is to solve the collision, meaning that two different keys are mapped to a same position. This can easily happen since usually $|U| > m$. A simple resolution of collision is to chain the elements in collision, at the price of longer searching complexity in time. At the worst case, all elements are hashed into a same position, and are chained one by one, which makes the hash table useless. Therefore, designing a good hash function is critical in hash table applications. A well-designed hash function should make **simple uniform hashing**, i.e. any key is equally likely to hash into any position, independent of any other keys. There are no way in theory to verify if a hash function satisfies this criteria, but we know several empirically well-performing hash functions in practice (see section 11 of [Cormen et al., 2009] for more detail).

Hash tables are important in bioinformatics wherever a string or word needs to be connected to an array of values, such as word locations in a genome database, or word counts in different samples.

Bloom filter Bloom filter applies hash coding under a space/time trade-off with allowable errors of false discovery. This is widely applied in the situation where a great majority of query does not belong to the given set, and can be used as a primary filter for fast rejecting non-member elements [Bloom, 1970].

One way to construct a Bloom filter is derived naturally from the conventional error-free hashing method, only to reduce the entire information of a key to a smaller code [Bloom, 1970]. Another method for building Bloom filters considers the hash area as an array of individual addressable bits which are initially set as 0, then hashes each element by setting a subset of these bits as 1; thereby, elements are queried by verifying if all its associated bits are set as 1 [Bloom, 1970].

A major application of Bloom filters on biological sequences is approximate membership query, examining if a sequence belongs to a given set, with a certain amount of false positive and no false negative, i.e., if $x \in \mathcal{X}$, the result must be true; if $x \notin \mathcal{X}$, the result may be incorrectly returned as true with some probability, where x is an element sequence, and $\mathcal{X}$ is the set in query. Bloom filters and their derivatives are widely used in k-mer counters, for dealing with non-informative but resource-consuming k-mers related with sequencing errors. [Marçais et al., 2019b]

de Bruijn Graph A de Bruijn Graph (DBG) is a data structure proposed for solving the "superstring problem": with a given alphabet, finding a shortest circular "superstring" containing all k-mers. The basic idea is to represent each k-mer prefix or suffix as a node, and associate two nodes with a directed edge for each k-mer, then the problem is abstracted as traversing the graph passing through each edge exactly once, which is actually a classical Eulerian cycle finding problem.

When applied to biological sequences, one of the main challenges of this simple model is related with repeats in DNA, since a same k-mer appears multiple times in repeats, and these cannot be modelled by an Eulerian cycle. This problem may be solved in part with paired-end reads. [Compeau et al., 2011]

When multiple samples are involved in the assembly task, a deviation of DBG - colored de Bruijn Graph (cDBG) is introduced, whereby colors are associated to samples. This data structure can be used for tasks such as variant calling and novel sequence detection. [Iqbal et al., 2012]

1.6.2 Sublinear data structures used in transcriptomics

Genomics data generation undergoes a dramatic growth thanks to the greatly reduced sequencing cost (see section 1.2.1). This requires technologies for storing, indexing, and searching these data in a sublinear scale [Marçais et al., 2019b]. This section discusses a list of data structures with this aim. Information here mainly comes from the article [Marçais et al., 2019b].

Compressed string indexes For tasks such as sequence alignment, these structures address the problem of searching in a long sequence a position where a short one is exactly matched. Data structures for compressed string index include: suffix tree, suffix array, and FM-index.

String indexes are applied for the seed-and-extend methods in sequence alignment, i.e., firstly search some exact matches between sequences (seeds), and then extend alignments between the seeds. In genome assembly task, these techniques can also be applied to speed up DBG construction.

Locality sensitive hashing Locality sensitive hashing is related with the nearest neighbor problem, i.e., to search from a set of points in a high-dimensional metric space the one that is closed to a given point. This search can be very expensive if the space dimension is high, and the locality sensitive hashing can quickly solve the problem in an probabilistic way - to return a point that is not too far from the closest one.

This technique can also be applied in read alignment, to find firstly an approximate candidate location for alignment, then to refine it if possible.

Minimizers A minimizer of a k-mer is selected as the minimum m-mer along a k-mer, with $m < k$. Minimizers are used for sketching a collection of sequences. It is a commonly applied strategy in k-mer counters. It can also be used to efficiently summarize information in sparse data structures.

1.6.3 RNA-seq Data Simulation

Simulation plays an essential role in many computational domains, especially when a study aims to benchmark or evaluate certain methods. The basic idea is to generate an artifactual data set with known ground truth, then launch the algorithm on the data set to compare results with the artifact reality.

***polyester* for RNA-seq read simulation** One method for simulating RNA-seq reads is the *polyester* R package [Frazee et al., 2015]. It supports simulation with replicates and differential expression. Its *simulate_ experiment* function takes as input the *FASTA* file of a reference transcriptome and parameters including fold-change between condition, read length, read number per transcript, replicate number, sequencing error model and rate.

polyester is used in chapters 3 and 6.

***compcodeR* for gene expression matrix simulation** *compcodeR* [Soneson, 2014] is an R package that simulates a differential expression matrix.

The simulation is performed with the *generateSyntheticData* function, whose inputs include the numbers of differentially expressed features, total features, and samples per condition. *compcodeR* then generates gene expressions based on a Negative Binomial distribution. Furthermore, it supports including outlier up/down signals for a random set of samples for each gene feature.

compcodeR is used in chapter 3.

Chapter 2

Transcriptome Analysis with RNA-seq Data

This chapter gives an overview of methods and approaches for RNA-seq transcriptome analysis. The main part of this thesis (Chapter 3 - 7) involves this type of data.

2.1 RNA-seq Data Quality Control

Sequence reads files are usually stored in the FASTQ format, a text file format containing a read identifier describing the sequencing lane of origin, the read sequence itself and a line providing quality scores estimated for each base by the sequencing device.

2.1.1 Sequence read quality evaluation

FASTQC [Andrews et al., 2010] evaluates FASTQ files' sequencing quality under various aspects, including sequence quality per base, N content per base, sequence length distribution, sequence duplication levels, overrepresented sequences, etc.

When dealing with a multi-sample data set, *multiQC* [Ewels et al., 2016] can be used on *FASTQC*'s outputs, for summarizing all *FASTQC* reports into a single one.

2.1.2 Sequence read quality control

Read trimming A direct way of improving sequence read quality is to remove low quality parts from each read. This method is called read trimming. *Cutadapt* [Martin, 2011] is an example software applying this strategy. It was originally

used for trimming the artifact adapter sequences at both ends of a read, but it can also be used for trimming low quality bases at each end. Besides, it supports removal of short reads after trimming with a given length threshold. Alternative trimming software include *Trimmomatic* [Bolger et al., 2014], and BBDuk [`http://sourceforge.net/projects/bbmap/`].

Read correction Active read correction goes beyond mere quality trimming. Various methods were developed for DNA-seq read correction. *BLESS* [Heo et al., 2014] and *BFC* [Li, 2015] evaluate k-mers' confidence levels by their occurrence with respect to a given threshold. *SHREC* [Schröder et al., 2009] relies on a suffix tree for replacing low occurrence substrings in a read. *Coral* [Salmela and Schröder, 2011] corrects reads using a multiple sequences alignment approach - firstly clustering the reads by their k-mer overlaps, and using that to guide read correction.

Read correction is harder for RNA-seq than for DNA-seq, due to the much higher variability in read coverage in RNA-seq. Due to this variablity, k-mers with low frequency may also be correct, preventing a direct application of DNA-seq read correction methods [Song and Florea, 2015]. To our knowledge, the first software for RNA-seq read correction was *SEECER* [Le et al., 2013], which follows the multiple sequences alignment strategy. Another software, *Rcorrector* [Song and Florea, 2015], achieved higher efficiency in memory usage by correcting reads according to their k-mer occurrences, using flexible local thresholds of k-mer counts to overcome the problem of coverage variability.

2.2 Conventional RNA-seq Data Analysis

The contents summarized in this section are mainly from [Van den Berge et al., 2019, Martin and Wang, 2011].

2.2.1 Read alignment for gene/transcript mapping

Since NGS platforms generate short reads, identifying reads' source (i.e., which read comes from which gene) is usually necessary for gene-expression estimation. A straight-forward solution is to align the reads to a reference which summarizes sequences of all genes. This alignment-based approach is currently widely adopted, and is further divided into two categories based on the techniques behind.

Spliced alignment to a reference genome A straightforward strategy is to use a reference genome for mapping. Since genes are interrupted by long introns that do not form part of the sequenced RNA product, many reads are split between

two distant exons. This kind of alignment require splice-aware aligners which are able to identify those reads with one part from an exon and the other part from another exon. Splicing awareness allows some aligners - such as *STAR* [Dobin et al., 2013] - to discover novel non-annotated splicing junctions based on known ones. Still, this category of aligners may miss some cases, especially for those when a read has only a small portion aligned to one of the exons.

Unspliced alignment to a reference transcriptome Alternatively, instead of taking reference genome, one can also take reference transcriptome for alignment. Since the transcriptome contains transcript sequences after splicing, reads should be able to align continuously on them. Therefore, aligners no longer need to allow for splitting reads. However an important problem with transcriptome-level alignment is that many genes have multiple isoforms that share common exon sequences. This create ambiguities in read assignment. The first software in this category was *RSEM* [Li and Dewey, 2011]. *RSEM* implements an Expectation-Maximization algorithm to infer the origin of ambiguous reads through likelihood estimation [Li and Dewey, 2011, Pachter, 2011]. Recent transcript-level mapping software *Kallisto* [Bray et al., 2016] and *Salmon* [Patro et al., 2017], by applying pseudo-alignment, largely improved quantification speed. Still, this type of aligner/quantifier does not support discovery of novel splicing or expression patterns, due to the dependence on reference transcriptome.

Gene/transcript quantification Genes/transcripts can be quantified based on alignment results. One point to clarify is that, the two stages alignment-quantification can be either integrated in a single software, such as *Kallisto* [Bray et al., 2016], or implemented separately, such as firstly *STAR* [Dobin et al., 2013] for alignment and then *featureCounts* [Liao et al., 2014] for quantification.

Depending on the alignment strategy, there are also two types of quantification. *RSEM* [Li and Dewey, 2011], *Kallisto* [Bray et al., 2016], and *Salmon* quantify transcripts, while *featureCounts* [Liao et al., 2014] quantifies overall gene expression. The discussion in [Soneson et al., 2016] shows that though transcript-level quantification provides necessary information in some types of study, its estimation accuracy is not as good as for whole genes, and loses advantages in downstream differential expression analysis. However, when aggregating transcript-level quantification into gene-level, final differential gene expression results are improved. These authors provided an R package *tximport* to estimate gene-level quantification from transcript level.

2.2.2 Read assembly for transcript retrieval

An alternative way of RNA-seq read processing is to assemble them for retrieving original transcripts. Assembled transcripts then allows quantification [Trapnell et al., 2010]. Read assembly protocols can be separated into two categories: reference-based and *de novo* (i.e., reference-free) strategies.

Reference-based assembly With reference-based assemblers, sequence reads are firstly aligned to a reference genome with a splice-aware software. Then, a graph is constructed based on the reads clustered on each locus, summarizing all possible isoforms. Finally, the graph is traversed for individual isoform resolving. The most widely used software in this class is *Cufflinks* [Trapnell et al., 2010].

Reference-based assembly transforms a large assembly problem into a set of smaller ones, since the assembly is done only inside each overlapping locus. It generally provides accurate and sensitive detection of transcripts. Moreover, since the mapping is done with splice-aware methods, it allows to discover novel transcripts. However, it also has drawbacks. An obvious one is that it can be applied only on organisms with a reference genome (though this can be sometimes "solved" using a closely related species). Also, while it allows for detection of novel transcripts, some events are still missed, such as those associated by spliced reads that span very large introns, repeats or rearranged genome regions. Here the drawback of splice-aware mapping still holds, i.e., the reads are required to align sufficiently well to each location in the genome in order to be considered.

***de novo* assembly** The other strategy assembles sequence reads *de novo* without relying on a pre-defined reference. These methods are based on the DBG data structure (see section 1.6.1 Useful data structures for transcriptomics). One example *rnaSPAdes* [Bushmanova et al., 2019] firstly break reads into k-mers which are successive sub-strings along each read. Then the DBG is constructed according to the overlap among these k-mers, followed by removal of chimeric and erroneous edges.

De novo assemblers are free from predefined reference, thus they allows studying any organism. Even when a reference is available, this approach is sometimes still applied, for providing additional insights on unusual or aberrant transcripts [Bushmanova et al., 2019], which are surely not always annotated by the reference.

The disadvantages of this approach are also obvious, due to the lack of reference for read mapping, *de novo* assemblers require more resources since they do assembly task among all sequence reads; it requires also more sequencing depth for reconstructing full-length transcripts; besides, the results are also less accurate due to repeats, non-removed sequencing errors and other artifacts (e.g., adapters) [Steijger et al., 2013, Hayer et al., 2015, Bushmanova et al., 2019].

2.2.3 Evaluation of sample count-condition association

Following gene/transcript quantification, a common processing step is to evaluate the association between counts and conditions for each gene. In situations where samples are classified into two conditions, a series of hypothesis testing methods can be applied, such as t-test, Wilcoxon signed rank test, and others; if samples are classified into multiple conditions, the Analysis of Variance (ANOVA) method is suitable (but one should keep in mind that t-test and ANOVA are based on normal distribution assumption). Besides, whatever the condition number, machine-learning based feature reduction-selection methods are applicable.

Gene expression values harbour two main kinds of variability across samples: (i) variability across technical replicates (resequencing of the same sample), which follows an approximate Poisson distribution; (ii) variability across biological replicates (sequencing of different samples). The aggregation of two types of variability makes the read count of a feature (gene, for example) follow a negative binomial distribution [Marioni et al., 2008]. For details about Poisson and negative binomial distributions, see section 1.4.2 Some common probability distributions.

Normalization Before really entering across-sample analysis, a normalization step is usually required, since sequencing depth vary across libraries. Also, effects from differences in gene/transcript lengths need to be eliminated, since longer genes/transcripts accumulate more reads.

A straightforward answer derived from these two points is to normalize read counts for each gene by two scaling factors: (i) total read number in each sample, (ii) length of genes or transcripts. This led to Reads Per Kilobase Million (RPKM), Fragments Per Kilobase Million (FPKM), and Transcripts Per Million (TPM) measurements. RPKM and FPKM first eliminate factor (i), then factor (ii), and differ just in the application of single-end or paired-end RNA-seq. TPM eliminates factor (ii) before factor (i). TPM tends to replace the older RPKM and FPKM normalization as it describes true biological objects (transcripts) rather than abstract counts, however RPKM and FPKM are still used when counts are computed directly at the gene level and actual transcript sizes are ignored.

More sophisticated normalization methods consider differences resulting from variation in RNA composition across libraries (for more detail, see the section 1.5.2 Compositional data analysis). This includes median-of-ratios method in *DESeq2* [Love et al., 2014] and trimmed mean of M-values method in *edgeR* [Robinson et al., 2010].

Differential expression analysis One major analysis protocol following genes' or transcripts' quantification is differential expression analysis. Generally speaking, differential analysis methods can be categorized into three groups: (i) differen-

tial gene-expression analysis, (ii) differential transcript-/exon-usage analysis, and (iii) differential transcript-expression analysis. The (i) and (iii) respectively study across individual genes and transcripts between conditions, and the (ii) consider the composition of genes' isoforms between conditions [Soneson et al., 2016].

For differential gene or transcript extraction, statistical inference (see sections 1.4.1 Fundamental notions, and 1.4.3 Hypothesis testing between two sample groups) can be applied on the gene/transcript expression data. This is performed by the R packages *DESeq2* [Love et al., 2014] or *edgeR* [Robinson et al., 2010]. Generally speaking, the null hypothesis that "log-fold-change between two conditions is zero" is tested, via a variety of hypothesis testing methods, including likelihood ratio tests implemented both in *DESeq2* and *edgeR*, and Wald tests used by default in *DESeq2*.

For differential transcript or exon usage analysis, the isoform composition of each gene is considered. Software of this category include: cuffdiff [Trapnell et al., 2010], LeafCutter [Li et al., 2018], and kissDE [Lopez-Maestre et al., 2016].

The resulting p-values always require multiple-testing correction for reducing false discoveries. A series of approaches are used for this task, of which the Benjamini-Hochberg procedure is one of the most popular choice (see section 1.4.3 Hypothesis testing between two sample groups for more detail).

Machine-learning based feature selection Another strategy for selecting genes or transcripts that contribute to distinguishing one condition from another is to use machine-learning based models, such as mutual information, correlations, regularized logistic regression, LASSO Cox PH model etc. [Long et al., 2014, Cascianelli et al., 2020, Milanez-Almeida et al., 2020, Erho et al., 2013]. In our study of Prostate cancer prognosis [Nguyen et al., 2021] (Chapter 4) we used a Bayes reduction combined to LASSO stability selection for selecting informative genes, followed by logistic regression for classifier construction.

One-vs-N comparison The recently developed *MINTIE* software applies a "single case *versus* N controls" comparison strategy to *de novo* informative transcript retrieval. The program compares each "case" sample one by one with the group of all normal samples (this can be done in parallel), and summarizes the informative signals across comparisons. The program allows a sensitive detection of a broad range of event types in the transcriptome, including fusions, inversions, tandem duplications, insertions, deletions, splicing variants, etc., with a low false positive rates. [Cmero et al., 2021]

Survival analysis Survival analysis is an essential methodology in cancer research. It compares the elapsed period of time between key events in patients.

Time periods can be from cancer diagnosis to death, from treatment response to recurrence or recovery, etc. Major methods for survival analysis include Kaplan-Meier (KM) plots, logrank tests and Cox regression [Clark et al., 2003].

2.2.4 Inter-cohort gene/transcript query

After obtaining an informative list of genes or transcripts, an important issue is to verify that the retrieved signal still remains informative in another independent cohort. This requires that gene/transcript expression can be obtained in a second cohort. Querying gene/transcript expression in an independent data set is straightforward and can be performed using the same quantification tools used for the first cohort. One flaw of gene/transcript query however is that sequencing technologies and reference sequences evolve, thus introducing quantitative differences among batches processed at a few years intervals. Gene/transcript IDs contain a version suffix for avoiding ambiguity; but still, this may be in some cases a disturbing point of analysis.

2.3 The Third Road: *k*-mer Analysis

While NGS methodologies combined to the above bioinformatics tools have fueled considerable advance in transcriptomics, the causative genetic events remain unidentified in many individual patient samples, thus calling for better achievements. Whether it is possible to retrieve more information from RNA-seq data is an open question. An emerging method to address this question is k-mer signal analysis.

2.3.1 Limitations of conventional methods

One basic drawback of conventional methods is that, both mapping-based and *de novo* assembly protocols target quantification of genes or transcripts while leaving aside the capacity of RNA-seq data to capture exact sequences at single-base resolution. In a way, RNA-seq bioinformatics has retained the same viewpoint as microarray-based methods. Restricting measures to gene/transcript expression ignores a more complex world of local variations in RNAs, including but not limited to SNV, indel, novel splicing sites, transcription starts and terminations. When summarizing these events at the whole gene/transcript level, multiple "up" and "down" features present in specific patient subsets are canceled.

For mapping-based methods, an inevitable question is whether a predefined reference will ever comprise all variants in any arbitrary sample of any condition (e.g., age, disease, sample tissue, etc.). Though splice-aware aligners permit to

identify novel transcripts, they are far from guaranteeing the capture of all non-annotated events. Another limitation of relying on full-length genes or transcripts is noted by [Srivastava et al., 2020]: transcript abundance estimation is subject to alignment and mapping quality, and thus differential expression analysis based on alignment and mapping may not be deterministic.

On the other hand, *de novo* assembly methods easily miss rare events, especially at low sequencing depth. Also, their results contain an unavoidable ratio of mis-assemblies, due to the lack of a reference's guidance, resulting in potential false discoveries [Morillon and Gautheret, 2019].

Software do exist for searching local transcript variations, including *Kissplice* [Lopez-Maestre et al., 2016], *IRFinder* [Middleton et al., 2017], and *LeafCutter* [Li et al., 2018]. However they target only certain types of events (e.g. splicing events, intron retentions) and do not comprehensively capture all event types.

2.3.2 Transcriptome analysis based on *k*-mer count signals

k-mers and canonical k-mers k-mers are successive sub-strings of length k, extracted from sequence reads. For example, a read $AACCGGTT$ can be processed into four 5-mers $AACCG$, $ACCGG$, $CCGGT$, $CGGTT$.

In stranded reads, constituent k-mers are taken directly from the sequence, whereas in non-stranded reads, constituent k-mers are extracted by comparing the k-mer with its reverse-complement and taking only the smaller one in lexicographic order ("canonical k-mer"). For example, when the same read $AACCGGTT$ is sequenced in non-stranded mode, it has only two constituent k-mers: $AACCG$ (representing both $AACCG$ and $CGGTT$) and $ACCGG$ (representing both $ACCGG$ and $CCGGT$) (see section 1.2.2 Transcriptome profiling for sequencing strandedness).

In the example above, one sees that in the non-stranded mode, sometimes two constituent k-mers of a same read may be reverse-complement from one another, and they are merged into a single one with count being doubled. This may introduce some noise into analysis. Though this impact should be minor (since k-mers are analyzed individually and noises are thereby removed), it would be better to consider k-mer orientation when the dataset is stranded.

Choice of k The typical k value is an odd number no larger than 31.

The choice of odd k numbers prevents some independent k-mer features from being confused in stranded RNA-seq data. Let us consider an example of problematic case, where the 6-mer - $AAATTT$ reads the same as its counterpart. k-mers like this cannot distinguish between the original events from anti-sense events since the anti-sense 6-mer of $AAATTT$ is still $AAATTT$. Therefore, with even number

of k, we lose the capability of identifying these events, which is however an advantage when using stranded data. On the contrary, if k is an odd number, no k-mer can be read same as its counterpart (e.g., $AAACTTT$ is read as $AAAGTTT$ in counterpart), and thereby this bias is avoided.

The reason of choosing $k < 32$ is that current major computer systems use a 64-bit architecture. k-mer sequences are coded with each type of nucleotide represented with two binary bits, e.g., A with 00, C with 01, G with 10, and T with 11. Therefore, a 31-mers require 62 bits and can be encoded by a single 64 bit variable.

2.3.3 *k*-mer counting and rare *k*-mer prefiltering

k-mer counting k-mer counting aims to count k-mers with a fixed k among all sequence reads. Though the problem *per se* is relatively simple and straightforward, challenges are related to counting efficiency in time and memory, since billions of reads can be generated by NGS RNA-seq [Manekar and Sathe, 2018]. So, the design of counting algorithms is an essential issue that has been under active discussion and development over the past decade.

k-mer counting tools can be categorized based on their strategy: some programs are based on k-mer sorting, for example *KMC* [Deorowicz et al., 2013]; others are based on a hash table data structure, including *DSK* [Rizk et al., 2013] and *Jellyfish* [Marçais and Kingsford, 2011]. Other strategies also exist, including application of Bloom filter (*Jellyfish2* integrates this to achieve better efficiency). [Manekar and Sathe, 2018]

Besides, programs can be distinguished by the way they store the k-mer index: either on disk or in-memory: *DSK* and *KMC* are disk-based, and *Jellyfish* operates "in-memory" [Manekar and Sathe, 2018].

Classical k-mer counters usually count samples one by one. However, k-mer analysis is based on a k-mer count matrix, require summarizing multiple k-mer count lists into a single matrix. Recently, a novel tool - *kmtricks* - was developed for counting k-mers and forming the matrix efficiently using a Bloom filter [Lemane et al., 2021].

Filtering rare k-mers As k-mer numbers become very large in real-life RNA-seq data analysis, pre-filtering of rare k-mers is often required. A typical k-mer count distribution is shown in Figure 2.1, where most k-mers have a very low count. Though these rare k-mers may also come from interesting rare events, they are much more likely to be related with sequencing errors. A straightforward filtering consists in removing k-mers whose counts are lower than a given threshold, sample by sample (i.e., abundance filter). Additionally, one may consider k-mer

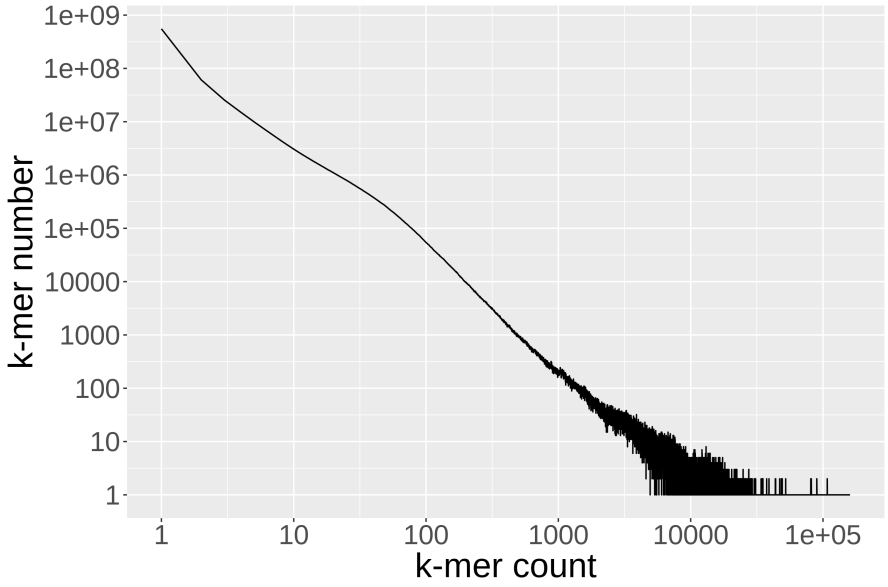


Figure 2.1: k-mer count distribution. The distribution curve is plotted from a real lung adenocarcinoma sample in [Seo et al., 2012]. Both axes are in log scale.

recurrence (recurrence filter). For example, a lowly counted k-mer may still appears recurrently in multiple samples, suggesting they are more likely to come from real biological events than from random sequencing errors. With this recurrence threshold, a more permissive threshold of conventional abundance filter can be applied, and thereby rare events may be "rescued" for further analysis [Lemane et al., 2021]. *DE-kupl* [Audoux et al., 2017] and *kmtricks* [Lemane et al., 2021] retrieve k-mers counted over n times in at least m samples.

2.3.4 *k*-mer analysis

Direct k-mer analysis vs. other k-mer approaches Direct k-mer analysis considers k-mers *per se* as features. Statistical tests or filtering are performed directly following the construction of k-mer count matrix. This is different from the use of k-mers in conventional gene-expression analysis. Certain mapping and transcript quantification algorithms use k-mers only as seeds for read alignment. In assembly approaches, k-mers are utilized for DBG construction. In these conventional methods, however, the features being analyzed are genes/transcripts rather than k-mers.

Advantages Direct k-mer analysis really focuses on local events at single-base resolution, and fully utilizes the capacity of NGS data that measures at this preci-

sion. As a reference-free method, direct k-mer analysis allows measurement of transcriptome without prior knowledge - the reference genome or transcriptome. This offers several benefits: (i) It allows for a comprehensive capture of all novel events without limitation [Audoux et al., 2017, Wang et al., 2021]. Thus the method can be applied to organisms without references, or for detecting non-annotated variants (either due to individual variation or incomplete annotation). (ii) Reference transcriptome and genome vary in time. Results generated from reference-free methods are not impacted by these variations and therefore are more reproducible [Lorenzi, 2021]. (iii) k-mer counts enable a deterministic capture of events, independent from read assignment algorithms, again rendering results more reproducible.

Also, with short sequences typically of length smaller than 32 retrieved from sequence reads, k-mers represent events at single nucleotide resolution. Analyzing individual k-mers without mapping/assembly allows applying statistical inference or machine-learning algorithms on all events individually. Therefore, it prevents the differential signals from cancelling each other when being aggregated to gene/transcript level.

Analyzing k-mer count signals directly offers another gain relative to other reference-free methods such as *KisSplice* [Lopez-Maestre et al., 2016]. K-mer count analysis follows a data-driven logic, examining all signals captured by statistical analysis or machine learning models independently from event identification. This differs from *Kissplice* which implements an expert system (e.g., SNPs relate to bubbles exhibiting two paths of length exactly 2k - 1). Constructing an expert system comprising all possible cases is usually difficult. A data-driven approach is easier for exhaustive event detection.

Finally, I find interesting to note that NGS reads themselves are actually k-mers by nature. One reason that we do not directly process these reads is that they are usually too long (e.g., 101 bp). This makes the feature space (see definition in 1.5.1 Fundamental notions) typically large (up to 4^{101}). Besides, present sequencing technologies are not perfectly error-free. Thus reads are usually trimmed for quality control (see section 2.1.2 Sequence read quality control), which makes the feature space even larger since read length variability is further taken into consideration. Therefore, k-mers can be seen as a way for shortening and fixing feature sequence length, with largely aggravated redundancy as a price. Should there be a possibility of perfect or quasi-perfect short read sequencing in the future, k-mer signal analysis could be applied directly on reads, and it would be a powerful approach to NGS data analysis.

Challenges Basically, there are two main challenges in direct k-mer analysis. Firstly, as k-mers are retrieved by increments of 1 nt, they are highly interdependent and their number quickly explodes compared to genes or transcripts. For

example, a single human RNA-seq sample may contain as many as 10^8 distinct 31-mers, whereas only 10^4 genes or 10^5 transcripts are referenced. Apart from the considerable induced computational complexity both in run-time and memory space, these highly redundant features aggravate the multiple testing problem when estimating statistical significance, and the curse of dimensionality in classification and clustering tasks. The second limitation is that k-mers are typically as short as 31 nucleotides or less. Short sequences lack specificity and thus make downstream interpretation difficult. Besides, this lack of specificity also introduces variability in counts, as some k-mers within a transcript get artificially higher counts. This high variability in counts is a major source of noise when k-mers are used as a proxy for transcript quantification. This point will be discussed in the Chapter 7.

Potential solutions A first way of addressing the above challenges is to extend k-mers into longer sequences based on their sequence overlap (i.e., k-mer contigs). Therefore, the interdependence among k-mers and their number is reduced, and their sequence specificity is enhanced. This kind of k-mer extension is quite different from conventional sequence assembly, where the former stops whenever meeting ambiguities for capturing signals at local-event level, while the latter aims to retrieve the original transcripts (see Chapter 3 for more detail). k-mer extension addresses at once several issues: multiple testing, curse of dimensionality, count variability, and specificity for downstream biological interpretation.

k-mer extension may involves two potentially important remarks: (i) The sequencing depth may have an impact on extension result - low coverage may introduce a mis-extension problem where independent k-mers are merged together only by their good overlap by coincidence, and some intervention is thereby required to control this wrong extension ratio (see Chapter 3 for more detail). (ii) one repetition sequence can be represented by a set of its equivalent elementary substrings - for example, depending on the merging order, the sequence $ACGTACGTACGT$ can be represented by contigs $ACGTAC$, $CGTACG$, $GTACGT$ and $TACGTA$, if choosing $k = 3$.

k-mer extension is usually not sufficient for reducing a k-mer matrix size to manageable dimension. Other strategies for feature dimensionality reduction are needed. Supervised strategies including differential expression filtering [Audoux et al., 2017] and machine-learning algorithms [Lorenzi et al., 2020] are explored in Chapter 3. Some non-supervised strategies, such as the widely used PCA, have a time complexity that is too high for very large matrices. Besides, the compositional nature of k-mer counts may render Euclidean distance meaningless (see section 1.5.2 Compositional data analysis). In the Ph.D. thesis [Nguyen, 2020], application of fast clustering methods such as DBSCAN to k-mer features was examined, but the performance of clustering were not satisfying (unrelated

k-mers could not be accurately sorted out). Recently however, [Sun et al., 2021] used count-based clustering based on locality sensitive hashing for reducing a k-mer matrix in a program aiming at single-cell type classification.

k-mer analysis with DNA-seq data Note that k-mer analysis has already been widely applied to DNA-seq data, notably in large-scale NGS database searches - such as in *BIGSI* [Bradley et al., 2019] - and in genome-wide association studies [Rahman et al., 2018]. Another application aims at identifying mutation events without relying on a pre-defined reference. This analysis involves a one-vs-one design, where a case (mutant) sample is compared to a control (wild-type) sample [Nordström et al., 2013]. I have contributed to such a study during my thesis [Wang et al., 2021] (Annex 1).

k-mer analysis with RNA-seq data RNA-seq data analysis usually involves measurement of gene or transcript expression levels. In the k-mer based approach, this measure is done with k-mer count signals. We present below a selection of software for direct k-mer based RNA-seq analysis that were important for this thesis.

DE-kupl was the first software to apply "direct" k-mer analysis, i.e., with no consideration of gene, assembly or graph, to RNA-seq data. Briefly speaking, *DE-kupl* firstly counts k-mers sample by sample and joins them into a k-mer count matrix where rows are k-mers and columns are samples (i.e., a feature matrix). Then, *DE-kupl* applies differential analysis (t-test, *DESeq2* [Love et al., 2014], or *Limma-Voom* [Ritchie et al., 2015]) for extracting a list of significant k-mer signals. These k-mers are then merged into contigs based on their sequence overlap. Finally, contigs are annotated for biological interpretation. I participated to this project in my *M.Sc.* internship in 2017.

Gecko implements genetic algorithm for selecting k-mers relevant to sample conditions. In the data preparation stage, k-mer counting is followed by successive steps of non-informative and redundant k-mer elimination. Next, a genetic algorithm is used to iteratively search k-mers that classifies samples most accurately. [Thomas et al., 2019]

iMOKA was developed for constructing classifiers using k-mer signals. It uses the recent k-mer counter *KMC3* [Kokot et al., 2017], followed by two levels of reduction: firstly a combination of Bayes classifier and adaptive entropy filter to exclude non-relevant k-mers; secondly an aggregation of k-mers according to their sequence overlap. Differing from *DE-kupl*, the aggregation stage selects a representative k-mer for each overlap group rather than extending k-mers into contigs. *iMOKA* also integrates a classifier builder based on random forests, as well as a sample condition predictor which assigns a label to each newly given

sample based on the trained model. The software also includes a user-friendly graphical interface. [Lorenzi et al., 2020]

In chapter 3, we introduce our new software, *KaMRaT*, which aims to provide a generic toolbox for processing k-mer count tables, including dimensionality reduction and sequence specificity enhancement.

2.3.5 Inter-cohort query of k-mer signals

Inter-cohort query is essential whenever a k-mer or k-mer contig of interest needs to be verified in an independent cohort. In contrast to gene/transcript queries, this task is relatively challenging, since sequences in the query are arbitrary (whatever sequence and whatever length) with no general IDs linking data sets. This results in an infinite dimensionality of the feature space of sequences in the query. Still, thanks to recent developments in the field of k-mer extraction and representation, we now have different software for arbitrary sequence query in an independent cohort.

One way of achieving this goal requires both a transcriptome and a genome reference. It associates the arbitrary sequence with gene and transcript annotation (thus with their IDs for inter-cohort query). One effort in this direction to which I contributed is the *Kmerator Suite* [Riquier et al., 2021] (see chapter 6). *Kmerator* extracts specific k-mers and contigs for genes and transcripts, for quantification using another program, *countTags*. When operating at the gene level, *Kmerator* outputs the k-mers/contigs that are present zero or one time in the reference genome, and at least one time in reference transcriptome. This takes into consideration k-mers spanning splice junction sites (zero time in the reference genome) and shared among transcript isoforms from the same gene (multiple times in the reference transcriptome). When operating at the transcript level, the software allows for searching k-mers/k-mer contigs found zero or one time in the reference genome, but only once in the reference transcriptome. Results prove that gene expression can be queried from arbitrary sequences with good accuracy using this specific k-mer extraction proxy [Riquier et al., 2021] (see chapter 6).

Alternatively, in a reference-free fashion, the datasets to be queried are indexed using k-mers, and the query of an arbitrary sequence is done by searching k-mers in the index. A family of such software includes *HowDeSBT* [Harris and Medvedev, 2020], *Mantis* [Pandey et al., 2018], *SeqOthello* [Yu et al., 2018], and *BIGSI* [Bradley et al., 2019]. These aim to detect the presence/absence of a given sequences in a DNA-seq database. *HowDeSBT* makes use of the Sequence Bloom Tree (SBT) data structure for storing the existence of a given k-mer in the query data set. Arbitrary sequences are queried by searching k-mers in the constructed SBT. *Mantis* and *SeqOthello* propose data structures for replacing SBT, achieving faster and more space-saving indexation. *BIGSI* was developed for addressing

indexing of bacterial and viral genomes covering an enormous diversity. A more recent method by our collaborators, *REINDEER* [Marchet et al., 2020] opened the possibility of abundance query in a k-mer index. It utilizes spectrum-preserving string sets for efficient k-mer count index and query. *REINDEER* is presented in more detail in chapter 7, where I analyze its application to gene count query.

Chapter 3

Development of the *KaMRaT* Toolkit for *k*-mer Analysis

3.1 Motivation

Direct analysis of k-mer counts has shown many benefits for reference-free transcriptomics: (i) exhaustive capture of all sequence variations without limitation from a predefined reference; (ii) stable event representation and expression estimation across reference versions; and (iii) consideration of variations at single-nucleotide resolution. At present, however, no real "general purpose" method is available for k-mer analysis. Current methods, such as *DE-kupl* [Audoux et al., 2017] for k-mer based differential analysis, *Gecko* [Thomas et al., 2019] and *iMOKA* [Lorenzi et al., 2020] for classifier construction, all address a specific problem with a fixed workflow. We consider that the lack of a general perspective on the k-mer analysis approach may be an obstacle to the development of this methodology. This motivation led us to propose *KaMRaT* (k-mer matrix reduction toolkit), a general purpose software providing multi-functional and flexible usage for k-mer count signal processing.

3.2 My contribution

As the first author, I developed the *KaMRaT* software, analyzed and evaluated its performance and efficiency, and participated in article writing.

3.3 Article

KaMRaT: a C++ toolkit for k-mer count matrix dimension reduction

Haoliang Xue[1], Melina Gallopin[1], Ha N. Nguyen[1], Yunfeng Wang[1], Antoine Lainé[1], Chloé Bessiere[2], and Daniel Gautheret[1,3,*]

[1]I2BC, Université Paris-Saclay, CNRS, CEA, Gif sur Yvette, France
[2]IRMB, University of Montpellier, INSERM, 80 rue Augustin Fliche, 34295, Montpellier, France
[3]Gustave Roussy, 114 rue Edouard Vaillant, 94800, Villejuif, France
[*]Correspondence: daniel.gautheret@universite-paris-saclay.fr
(Non definitive author list)

Abstract

RNA-seq provides a snapshot of total transcripts in a sample at single-base resolution. Leading methodologies for RNA-seq analysis operate at the level of genes or full-length transcripts. An emerging alternative is to analyze RNA-seq data as k-mer count signals, which comprehensively captures all sequence variation in the data. The main difficulties with k-mer count analysis are the high number of k-mers, their interdependence, and lack of specificity when k is small. All of these are challenges for statistical analysis, machine learning and biological interpretation. Several programs exist using k-mer analysis to serve specific purposes such as differential expression analysis or sample classification. Here we consider the analysis of k-mers from a RNA-seq or other NGS dataset as a set of generic tasks that may be combined to serve different purposes. To this aim, we developed *KaMRaT*, a general C++ toolkit for processing k-mer count tables that includes modules for selecting informative or condition-related k-mers, merging k-mers into contigs and extracting k-mers matching given sequences. Here we benchmark the main *KaMRaT* modules, present typical applications and compare results to those of dedicated k-mer analysis software.

1 Introduction

Gene expression profiling from high-throughput RNA sequencing (RNA-seq) data is now widely used in all areas of biology. A common design for these studies uses a gene expression matrix where each sample is labelled for a biological condition. The matrix can then be used for differential gene expression analysis, sample clustering or development of predictive classifiers. Gene expression is commonly obtained after aligning RNA-seq reads to a reference genome/transcriptome, followed by quantification of aligned reads [Van den Berge et al., 2019]. This reference-based approach is reliable and convenient, but it amounts to ignore a wide sequence diversity present in the original data. For instance, predominant protocols ignore novel mRNA isoforms, RNAs from repeated genomic regions or exogeneous species, as well as small variations such as SNPs and indels.

An emerging strategy to address all possible variations in high throughput sequencing (HTS) data sets is to use a k-mer counter [Marçais and Kingsford, 2011, Rizk et al., 2013, Kokot et al., 2017] that extracts and counts all successive substrings of length k from sequence reads. k-mer counts are then used as proxys for the quantity of the precise sequence represented by each k-mer. This strategy avoids predefined references while capturing all variations at single-base resolution. Representing these variations individually prevents informative signals from canceling each other while being aggregated to their host gene or transcript.

Here we are interested in the analysis of $n \times p$ count matrices built from n labelled samples and p k-mers, generated from RNA-seq data. Our purpose is to extract from this matrix sequence features relevant to the study, while reducing feature interdependence. This process may also apply to other HTS technologies such as ChIP-seq, ribosome profiling or metagenome sequencing. Several studies have already applied k-mer-based strategies to HTS data to discover biomarkers and produce clinical classifiers [Audoux et al., 2017, Audemard et al., 2019, Pinskaya et al., 2019, Thomas et al., 2019, Lorenzi et al., 2020]. However, available software either rely on complex pipelines with multiple dependencies or, in the case of *Gecko* and *iMOKA*, are specialized in predictive model building. We consider the lack of a general purpose and easy to run software to handle large k-mer matrices to be an obstacle to a more widespread adoption of these methods.

Machine learning applications on gene expression matrices require that p is maintained as small as possible with respect to n to alleviate the "curse of dimensionality" [Clarke et al., 2008]. Typical human gene expression matrices have dimensions with p around 20,000 and n between 10-100. However, an NGS sample has in the order of 10^8 distinct k-mers, and multi-sample studies reach billions of k-mers, which is considered a "ultra-high p". Common dimension reduction methods used in transcriptomics such as principal component analysis (PCA) [Clarke et al., 2008, Fan and Lv, 2008, Bourgon et al., 2010] have computing costs that are prohibitive with a ultra-high p (PCA has a term that is solved in $O(p^3)$). A faster alternative approach is to apply univariate feature filtering. This can be done independently of sample labels using variance or Shannon entropy, or dependently of labels with tests that compare means such as Student's t-test or signal-to-noise ratio [Golub et al., 1999]. Machine-learning methods such as SVM classifiers, genetic algorithms or Bayes classifiers have also been used successfully for univariate prefiltering of count matrices [Guyon et al., 2002, Clarke et al., 2008, Haury et al., 2011, Thomas et al., 2019].

Another strategy available for k-mer matrix reduction is to aggregate k-mers according to their sequence overlaps, either by extending k-mers into contigs [Audoux et al., 2017], or by selecting one representative k-mer from a group of overlapping k-mers [Lorenzi et al., 2020]. The k-mer contig extension or "merging" strategy has the extra benefit of an improved alignment specificity of contigs and thus, easier interpretation for downstream analysis [Audoux et al., 2017].

Here we introduce *KaMRaT* (k-mer Matrix Reduction Toolkit), a lightweight and multi-functional toolkit implemented in C++ for k-mer matrix reduction, offering fast and user-friendly methods for k-mer count matrix reduction and related utilities. It introduces a new aggregation procedure where k-mers are merged only when their counts across samples are similar. Besides, it can be used to search for condition-specific k-mers/contigs or as a feature selection tool to select k-mers/contigs for classifier development. We evaluated *KaMRaT*'s aggregation correctness and selection effectiveness with simulated data sets, and applied *KaMRaT* to reference-free classifier construction and condition-specific k-mer contig extraction using real cancer datasets.

2 Methods

2.1 *KaMRaT* and its modules

KaMRaT takes as input a k-mer count matrix and produces a reduced matrix where features are less interdependent and more relevant to the study, as shown in the generalized workflow in Figure 1. The k-mer count matrix is produced from individual RNA-seq samples with companion scripts using *Jellyfish* [Marçais and Kingsford, 2011] and *DE-kupl joinCounts* C program [Audoux et al., 2017]. *KaMRaT*'s speed and modular design allow user to quickly implement and test any workflow.

KaMRaT is implemented in C++ with dependencies on *MLpack* [Curtin et al., 2018] and *Armadillo* [Sanderson and Curtin, 2016] libraries.

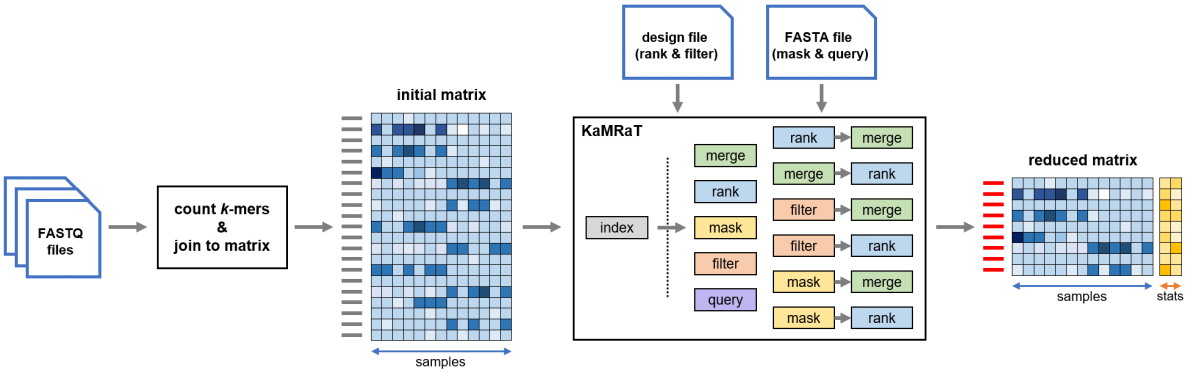


Figure 1: *KaMRaT* workflows: starting from FASTQ files, a k-mer counter is applied for each sample; the samples' k-mer counts are then joined together as a matrix (initial matrix), with columns as samples and rows as k-mers; *KaMRaT* then reduces k-mer features' dimensionality and enhancing their sequence specificity, to output a smaller matrix where features are less interdependent and more relevant to the study. The functional modules *rank*, *merge*, *mask*, *filter* permit flexible single or combined usage in various order.

KaMRaT includes the following six modules. Hereafter, the term "feature" can be k-mer, k-mer contig or any type of quantified element such as gene or transcript.

- *index* constructs a binary index of the input matrix;
- *rank* sorts features by evaluating the association between sample counts and conditions;
- *merge* extends k-mers into longer sequences (contigs) based on sequence overlap;
- *filter* extracts/eliminates features according to their counts;
- *mask* reserves/removes k-mers matching an input sequence list;
- *query* estimates count vectors of a given list of sequences (k-mers or contigs) from k-mer matrix.

KaMRaT index is the first command to be used in any *KaMRaT* application. It converts a text input matrix into binary index files, allowing random access to features' count vectors. All feature names and their sample count vectors are indexed into a single file, with the index positions being stored separately. Downstream modules then only rebuild in memory the association between features and their indexed positions for sample counts, avoiding repetitive processing of the large k-mer count matrix at each subsequent step.

This module also provides a normalization step via count scaling presented by equation 1.

$$X_{f,s}^{norm} \leftarrow \frac{X_{f,s}^{raw}}{\sum_{f_x \in \mathcal{F}} X_{f_x,s}^{raw}} \cdot C \tag{1}$$

where $X_{f,s}^{norm}$, $X_{f,s}^{raw}$: normalized or raw sample count of feature f and sample s; $\mathcal{F}$: universe of all features; C: constant scaling factor provided by user.

KaMRaT merge partially inherits from *DE-kupl mergeTags* module [Audoux et al., 2017] that iteratively extends contigs based on sequence overlap. For each element (i.e., k-mer or contig) to be extended, other elements are sought iteratively, from overlap by $(k-1)$nt down to a given minimum value (by default $\lfloor k/2 \rfloor$nt). Extension stops whenever ambiguities (more than one equally overlapped possibilities) are encountered or no more overlapping is available. *KaMRaT merge* implements an original refinement of the extension procedure - sample count intervention - that measures count compatibility before executing extension: overlapping elements are merged only if both the k-mers adjacent to merging point (prefix-suffix overlap) have coherent sample count vectors. Coherence is evaluated by one of the three different methods: Pearson distance, Spearman distance, and mean absolute contrast (MAC) introduced previously in [Nguyen et al., 2021]. The distances are defined by equation set (2), with which all result values are scaled between 0 and 1. By default, extension is executed when $d_{\text{Pearson}} < 0.20$ (see Results).

$$\begin{cases} d_x(\boldsymbol{c_1}, \boldsymbol{c_2}) = \frac{1}{2} \times (1 - \rho_x(\boldsymbol{c_1}, \boldsymbol{c_2})), & x = \text{Pearson}, \text{Spearman} \\ d_x(\mathbf{c_1}, \mathbf{c_2}) = \text{mean}_{s \in \mathcal{S}} \left(\left| \frac{c_{1,s} - c_{2,s}}{c_{1,s} + c_{2,s}} \right| \right), & x = \text{MAC} \end{cases} \tag{2}$$

where d_x: Pearson/Spearman/MAC distance; $\mathbf{c_1}$ and $\mathbf{c_2}$: sample count vectors of the two k-mers adjacent to merging point, with $c_{1,s}$ and $c_{2,s}$ being the components of sample s; $\mathcal{S}$: universe of all samples; ρ_x: Pearson/Spearman correlation coefficient.

The output of *KaMRaT merge* is a contig count matrix. For each contig, the mean or median counts of all constituent k-mers are calculated for each sample, according to user's preference.

KaMRaT rank scores each feature by evaluating the association between sample counts and conditions. The sample conditions are provided by an extra tabular file (-design) containing (sample, condition) pairs. Features are sorted next based on evaluated scores from the best association to the worst. Table 1 summarizes the currently available scoring methods, their acceptable sample condition number, and whether they support batch effect (BE) removal. More detailed information about the scoring methods are provided in supplementary document.

KaMRaT query estimates count vectors of an extra input list of sequences, based on their constituent k-mers' counts. This is useful when a set of sequences need to be quantified in an independent dataset. The module queries with two modes: mean query and median query, that

Table 1: Scoring methods in *KaMRaT rank*

scorer	# condition	BE removal	note
ttest.padj	2	no	t-test adjusted p-value with B-H procedure
ttest.pi	2	no	t-test π-value in [Xiao et al., 2014]
snr	2	no	signal-to-noise ratio in [Golub et al., 1999]
dids	$\geq$2	no	DIDS in [de Ronde et al., 2013], adapted
lr	2	yes	logit regression's accuracy
bayes	$\geq$2	yes	Bayes classifier's accuracy
sd	$\geq$0	no	standard deviation, no condition considered

compute respectively the mean and median count vector of input sequences' constituent k-mers found in the index. If a sequence has no constituent k-mer found, it can be either omitted or returned with an all-zero vector, according to user's preference.

KaMRaT filter filters features according to their expression level - for instance those counts over n in at least m samples of condition c. It supports both retaining or removing these features. This module requires also an extra input design file, indicating the "UP" samples of which features should have high expression, and the "DOWN" ones of which feature expression should be low.

KaMRaT mask retains or removes k-mers matching an extra list of sequences.

2.2 A simple use case

The *KaMRaT* repository includes a toy dataset (`https://github.com/Transipedia/KaMRaT/tree/master/toyroom/data/`) composed of 89,768 k-mers $\times$ 20 samples from 10 lung tumor and 10 normal lung samples from a public dataset [Seo et al., 2012] (kmer-counts.subset4toy.tsv.gz), and a (sample, condition) file (sample-condition.toy.tsv). A typical use case with these data could be:

```
kamrat index -intab kmer-counts.subset4toy.tsv.gz -outdir kamrat.idx \
             -klen 31 -unstrand -nfbase 1000000000
kamrat merge -idxdir kamrat.idx -outpath merged-kmers.bin
kamrat rank -idxdir kamrat.idx -rankby ttest.padj \
            -design sample-condition.toy.tsv \
            -with merged-kmers.bin -outpath top-ctg-counts.tsv -withcounts
```

This command combination reduces the table from 89,768 k-mer features to 3,066 contig features in about 3 seconds. The first 46 features in the ranked matrix have a t-test adjusted p-value $<$ 0.05. *KaMRaT rank* provides a -seltop option to keep only the top-ranking contigs in the output matrix, which is useful when the results are to be fed to a machine learning pipeline.

2.3 Benchmark datasets

Case 1: simulated error-free RNA-seq data set. A 20-sample data set was simulated by the *simulate_experiment* function in *polyester* R package version 1.22.0 [Frazee et al., 2015], based

on *GENCODE* v34 reference transcriptome, with parameters *num_reps* = c(10, 10), *readlen* = 100, *reads_per_transcript* = round($10 \times \mathbf{L}/100$) where $\mathbf{L}$ is the vector of transcript lengths, *error_model* = "uniform", and *error_rate* = 0. The simulated fold change between conditions for each transcript is a random value between 0 and 2 generated under uniform model (*runif* function in R).

Case 2: simulated differential expression matrix. Four gene expression matrices with genes' differential status known *a prior* were simulated using the *generateSyntheticData* function in *compcodeR* R package version 1.22.0 [Soneson, 2014], with parameters *samples.per.cond* = 150, *n.diffexp* = 500, and *fraction.upregulated* = 0.5, as well as *n.vars*, *effect.size* and *random.outlier.high.prob* set as listed below to create difference among matrices. For the definition of "unusual" over-expression, see [Soneson and Delorenzi, 2013].

- 20,000 features with effect size being 10 between conditions;
- 20,000 features with effect size being 1.5 between conditions;
- 20,000 features with effect size being 1.5 between conditions, and each with 20% unusually over-expressed samples;
- 200,000 features with effect size being 1.5 between conditions, and each with 20% unusually over-expressed samples.

The simulated matrix has gene features as the first column, which can be directly fed into *KaMRaT rank* module as we previously did in [Nguyen et al., 2021].

Cases 3 and 4: real RNA-seq datasets. Real RNA-seq data sets were : (i) 154 matched lung adenocarcinoma RNA-seq samples (77 tumors, 77 adjacent normal tissues, LUADseo TvsN) retrieved with SRA accession *ERP001058* [Seo et al., 2012]. FASTQ files were processed by *Cutadapt* [Martin, 2011] version 2.10, with parameters *-q 12,12 -m 31.* (ii) 78 prostate adenocarcinoma samples from the TCGA project [Abeshouse et al., 2015] (36 relapse, 42 non-relapse, PRADtcga RvsNR) obtained from dbGAP accession *phs000178.v9.p8* with permission. Biochemical relapse labels are assigned same as previously described [Nguyen et al., 2021], based on the clinical information provided in [Liu et al., 2018]. According to our quality check, no *Cutadapt* processing was required for this dataset.

2.4 Evaluation of *KaMRaT merge*

Jellyfish count (version 2.2.10) [Marçais and Kingsford, 2011] was run on each pair of paired-end samples simulated by *polyester*, with parameters *-m 31*, *-s 1000000*, *-C*, *-F 2*. Binary outputs were then dumped (*Jellyfish dump*) into 2-column text files (*-c*). k-mer count lists were then joined as a matrix using *DE-kupl joinCounts* C program [Audoux et al., 2017], without k-mer recurrence and abundance prefiltering (*-r 1 -a 1*). *KaMRaT index* was then run on the joined matrix (*-klen 31 -unstrand -nfbase 1000000000*), followed by *KaMRaT merge* with *-overlap 30-15* and different interventions (*-interv*) of none, pearson, spearman, and mac, each under different thresholds. Before each extension, a variable percentage (from 0 to 60%) of k-mers was randomly removed from the matrix to simulate incomplete k-mer sets.

rnaSPAdes (version v3.14.0) [Bushmanova et al., 2019] was run both on read and k-mer FASTA files. At read level, all samples' FASTA files were firstly mixed together into two files (paired-end)

and fed to *rnaSPAdes* (arguments *-1* and *-2*). At k-mer level, all k-mers after *Jellyfish count-dump* were collected as a single FASTA file regarded as unpaired reads (*--s 1*) for *rnaSPAdes*.

The resulting contigs were aligned to the same reference transcriptome by *BLASTn* (version 2.6.0) [Camacho et al., 2009], with parameters *-max_hsps 1 -max_target_seqs 1 -dust no*, under the default megablast task. Extension/assembly correctness was then evaluated by perfect alignment ratio (percentage of contigs that are **perfectly aligned** to a transcript in the reference) and identity ratio (percentage of the contigs **identical** to a transcript in the reference). The median length of contigs was computed to evaluate extension completeness.

Reduction ratios were computed for different intervention modalities as the ratio of k-mer number before extension divided by contig number after extension.

2.5 Evaluation of *KaMRaT rank*

KaMRaT index was run with option *-nfbase 30000000* (without *-klen*) and *rank* was run with all ranking methods except sd on each *compcodeR* simulated matrix. Pearson distances for comparing feature ranks

The ability of ranking methods to identify differentially expressed genes was evaluated by comparison with simulated ground truth. Similarities among ranking methods were evaluated by Pearson distance between feature ranks.

2.6 Comparison of *KaMRaT merge-rank*, *rank-merge* and *iMOKA* for feature preselection

Our goal here was to evaluate *KaMRaT* as a preselection tool for random forest (RF) classification. *KaMRaT merge-rank*, *rank-merge*, and the *reduce-aggregate* modules of *iMOKA* (version 1.1) [Lorenzi et al., 2020] were used with the same input and the same RF prediction/evaluation procedure. For each input data set, matrices were produced with a 5-fold cross-validation scheme by dividing samples into 5 subsets and iteratively using 4/5 of them for training with the remaining 1/5 for testing. Initial counts were all produced by *Jellyfish* with the same parameters as above.

iMOKA reduce-aggregate modules was run with default parameters, except:

- at the *reduce* stage: *-c 100* (default value) and *-c 1* were used both, separately;
- at the *aggregate* stage: -m nomap.

For *KaMRaT* runs, *Jellyfish* counts were joined using *DE-kupl joinCounts* for each training or testing set, considering k-mers present in at least one sample with counts over 5 (*-r 1 -a 5*, so as to be equivalent to the default setting in *iMOKA reduce*). *KaMRaT index* was run (*-klen 31 -unstrand -nfbase 2000000000*) on each training matrix, followed by separate application of *merge-rank* and *rank-merge* workflows. *KaMRaT merge* was run with *-overlap 30-15* and *-interv pearson:0.20*. *KaMRaT rank* was run with all ranking methods except sd, and with selection of top features (*-seltop*) to retain numbers of *KaMRaT* reduced features similar to those after *iMOKA aggregation*. For both workflows, contig count vectors were computed with mean counts across constituent k-mers.

All RF classifers were built using the *iMOKA random_forest* module (parameter *-m 100*). *iMOKA extract* and *KaMRaT query* (*-toquery median -withabsent*) were run to estimate trained features' counts in testing sets, respectively for *iMOKA* and *KaMRaT* outputs. Prediction balanced

accuracies were estimated using *iMOKA predict*. When comparing features between *KaMRaT* and *iMOKA*, *KaMRaT* contigs and *iMOKA* k-mers were considered as equivalent if the k-mer was present within the contig.

2.7 Application of *KaMRaT filter-merge* for retrieving condition-specific k-mer contigs

K-mer matrices were built from *Jellyfish* outputs using *DE-kupl joinCounts* (*-r 1 -a 5*), followed by *KaMRaT index* without normalization (*-klen 31 -unstrand*). At the *filter* step, LUADseo TvsN data set was processed by selecting specific k-mers with counts ≥ 1 in at least half of tumor samples (*-upmin 1:39*) and =0 in all normal samples (*-downmax 0:77*). At the *merge* step, the same parameters as before were applied: *-overlap 30-15*, *-interv pearson:0.20*, and *-withcounts mean*. Contigs were annotated using *BLASTn* against *GENCODE* v34, using the same parameters as in section 2.4 except for *-task blastn*.

3 Results

3.1 Evaluating *KaMRaT merge* for k-mer extension

The process of extending k-mers into contigs is subject to a significant mis-extension rate, due in a large part to the size of the permitted overlap between k-mers which is smaller than many genome repeats [Audoux et al., 2017]. We implemented an intervention procedure whereby contigs are extended only with k-mers having similar count profiles (see Methods). We evaluated extension correctness and completeness using a simulated read data set built from human transcripts.

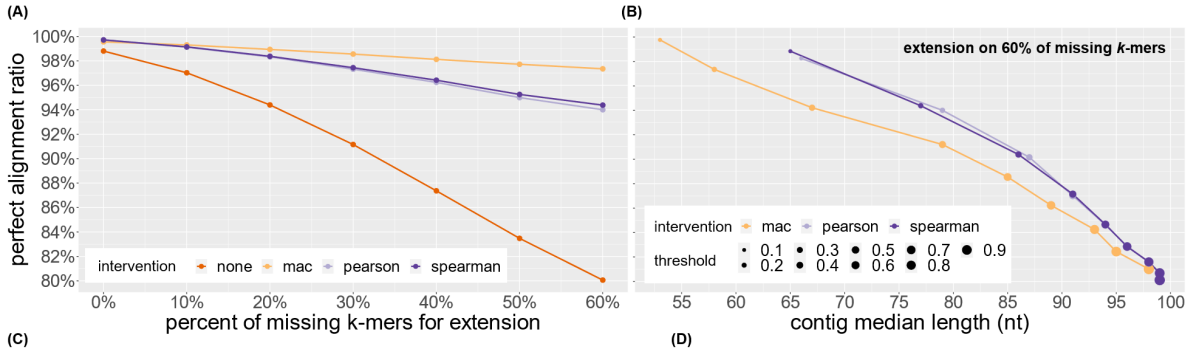
We firstly calculated the ratio of extended contigs that perfectly aligned - from first to last nucleotide, without any gap or mismatch - to the original transcripts (perfect alignment ratio). This ratio evaluates extension correctness, since the reads were extracted exactly from the reference transcriptome and the resulting contigs should be perfectly aligned to the reference if the extension is correct. To simulate a common situation where extension is executed on incomplete k-mer sets - e.g., differential k-mers after t-test or k-mers extracted from FASTQ files with uneven read coverage - we extended subsets of randomly selected k-mers from the initial set. We tested different intervention methods (Pearson, Spearman and MAC) on each dataset. Results show that all intervention methods remarkably improve extension correctness (Figure 2A). Over 94% of contigs are perfectly aligned to the original transcripts with any intervention method vs 80% in the absence of intervention, in the worst case scenario where 60% of k-mers are missing.

Next, we examined the effect of varying maximal thresholds for each intervention method, from 0.1 (stringent) to 0.9 (permissive), on contigs' correctness (perfect alignment ratio) and completeness (contig median length) (Figure 2B). As expected, stricter thresholds improve contig correctness at the price of completeness. Also, the MAC intervention, which is sensitive to absolute count deviation, is considerably stricter than both correlation-based methods. Under the same threshold, more contigs are correct, albeit more fragmented, with MAC. According to this simulation, a threshold of 0.2 for Pearson and Spearman distances and 0.3 for MAC guarantees 94% of contigs being correct with a median length above 67 nt.

KaMRaT merge differs from a read assembler in that it stops the extension process whenever two different k-mers or contigs equally overlap the extending one. Therefore it does not tolerate sequence polymorphism (e.g., SNP, indel) within its constitutive elements, unlike a read assembler

such as *rnaSPAdes* [Bushmanova et al., 2019]. *KaMRaT merge* only aims at local extension where each contig represents a unique sequence variant present in the dataset. To illustrate this difference, we compared *KaMRaT merge* contigs with assemblies produced by the read assembler *rnaSPAdes* on the above errorless simulated data set (Figure 2C). As expected, *rnaSPAdes* fed with the same set of k-mers produces longer contigs than *KaMRaT merge* at the price of a higher rate of assemblies among k-mers aligning with mismatches. *KaMRaT merge* produces near perfect, but much shorter contigs with a median length of 60 nt vs. 376 nt for *rnaSPAdes*.

We examined the reduction ratio enabled by *KaMRaT merge* - k-mer number before extension over contig number after extension. While about 100-fold reductions were obtained on the error-free simulated data, reduction ratios on real data sets were 13 to 21-fold (Figure 2D).



(C)

software	input set	% perfect alignment	% perfect identity	# contigs	contig median length (nt)
rnaSPAdes	all reads	31.22%	5.16%	152647	1461
rnaSPAdes	all k-mers	76.88%	1.50%	190044	376
KaMRaT-merge none	all k-mers	98.80%	0.57%	1001598	60

(D)

intervention	threshold	reduction ratio		
		error-free simulation	LUADseo	PRADtcga
none	-	118.1	~20.9	~15.8
MAC	0.3	114.2	~19.9	~14.7
Pearson	0.2	110.0	~19.7	~14.0
Spearman	0.2	109.7	~18.9	~13.3

Figure 2: *KaMRaT merge* evaluation. (A) Perfect alignment ratio of contigs extended in different sets of k-mers, by fixing all intervention thresholds as 0.20. (B) Perfect alignment ratio of contigs vs contig median length, extended on the situation with 60% k-mers missing. (C) Comparison of *KaMRaT merge* and *rnaSPAdes* on the same set of k-mers extracted from simulated transcripts. Results of an *rnaSPAdes* run using the original simulated reads is shown for reference. (D) A table comparing *KaMRaT merge*'s reduction ratio across different data sets (numbers with $\sim$ symbol are mean values among folds).

3.2 Evaluating *KaMRaT rank* for selecting differential features

We benchmarked *KaMRaT rank* using simulated gene-expression matrices containing 300 samples with varying numbers of features and levels of differential expression (Figure 3A). Note that ranking methods in *KaMRaT* are not necessarily intended to detect differential expression. While t-tests estimate the difference between group means (and the differential counts were simulated with different group means), SNR, Bayes and LR are intended for classification, which is a different purpose evaluated in the next section.

Expectedly, feature ranking by t-test adjusted p-values and π-value performed best for retrieving differential features (Figure 3B). In the most complicated case, PR AUCs were 0.811 and 0.801,

respectively (Supplementary Table S1). DIDS in principle should detect differential features with outlier samples but did not perform well here, possibly due to the way outlier samples are produced by the *compcodeR* simulation procedure, independently from differential feature generation [Soneson and Delorenzi, 2013]. Therefore, non-differential features also have outliers which can be detected by DIDS but are associated with "non-differential" labels. Still, it appears from our simulation that t-tests and SNR are more robust to outliers than other methods and should be useful to exclude irrelevant heterogeneous signals.

KaMRaT's ranking methods can be highly divergent. Dendrograms with Pearson distance of feature rankings (Figure 3C) show that ranks based on t-tests are congruous, while LR, Bayes and SNR are somewhat grouped and DIDS always stands out. The distance between ranking methods increases with the complexity of the data and this has a very strong effect when ranking is used to select the top N features in the count matrix. When comparing the top 500 features by each ranking, the ratio of features shared by all five methods ranged from 74% in the simplest case (20,000 features, no outlier, high fold change) to only 2% in the most complex case (200,000 features, 20% outliers, low fold change) (Supplementary Figure S1). It is important to keep this in mind when selecting a ranking statistic.

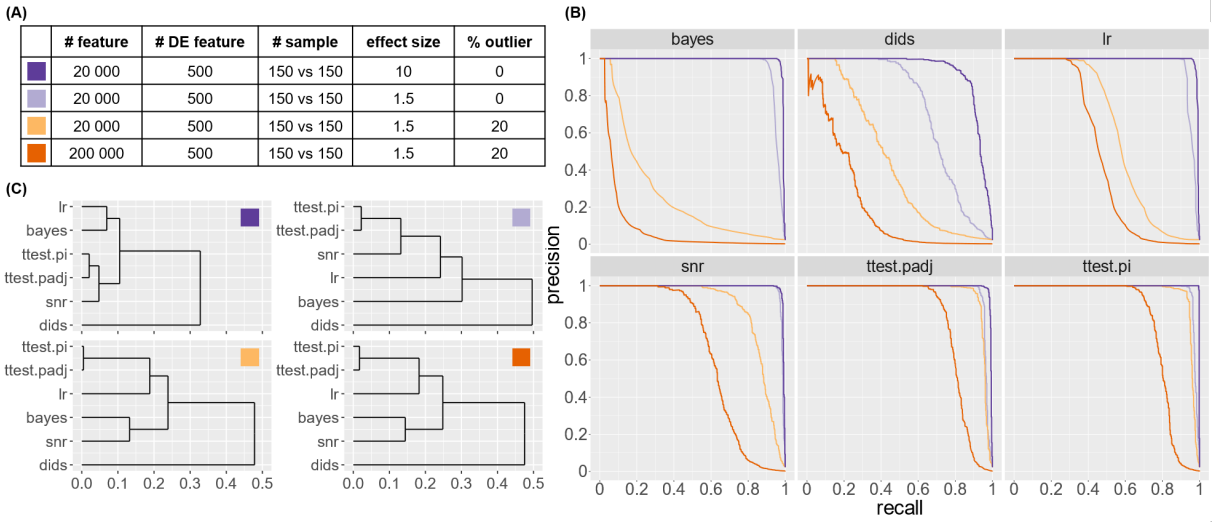


Figure 3: Evaluation of methods in *KaMRaT rank*. (A) A table summarizing simulation parameters in each simulated situation. (B) Precision-recall curves of all ranking methods in each simulated situation. (C) Dendrograms indicating similarities among ranking methods in each simulated situation.

3.3 *KaMRaT* as a feature preselection tool in classifiers

We evaluated *KaMRaT* as a preselection tool for two classification problems: (i) diagnosis of lung adenocarcinoma vs normal lung samples in a dataset of 154 biopsies (LUADseo TvsN), and (ii) prediction of relapsing/non-relapsing prostate adenocarcinoma from 78 biopsy samples (PRADtcga RvsNR). The *KaMRaT merge-rank* and *rank-merge* workflows were compared to the preselection modules implemented in the recent k-mer based classifier *iMOKA* [Lorenzi et al., 2020] which starts with a k-mer reduction step using a combination of naïve Bayes classification and entropy filtering,

followed by selection of one representative k-mer per cluster of overlapping k-mers. The resulting k-mers are used to fit a random forest (RF) model with 100 final features. All procedures of feature preselection, model training, and prediction evaluation were performed under a 5-fold cross-validation scheme as explained in Methods.

Depending on training sets, *KaMRaT index* generated 224-232 Gb indexes from 442-458 million k-mers of LUADseo TvsN, and 35-38 Gb indexes from 129-134 million k-mers of PRADtcga RvsNR (Figure 4A). For comparison to *iMOKA*, we applied *KaMRaT merge-rank* and *rank-merge* workflows to reduce the initial k-mer matrix to a comparable number of features (Figure 4A), before fitting the RF model.

Figure 4B shows prediction performance of the RF models obtained with the different preselection strategies for the two problems, evaluated by balanced accuracy. In the simpler problem of tumor vs normal classification, both *KaMRaT* workflows and *iMOKA* achieved balanced accuracy very close to 1. In the more difficult task of relapse prediction, balanced accuracy ranged from 0.55 to 0.65. Still, *KaMRaT* performed similarly (sometimes slightly better or worse) as *iMOKA* as a feature preselection tool. Comparisons of *merge-rank* and *rank-merge* strategies did not show a clear winner, supporting a flexible design such as *KaMRaT*'s to enable users to select different pipelines to meet specific needs. Note that *iMOKA*'s default 100-fold cross-validation did not improve prediction accuracy in our 5-fold CV setting (Figure 4B). We thus disabled this option in subsequent comparisons.

Figures 4C and 4D show CPU time and peak RAM usage for *iMOKA reduce-aggregate* and for the two *KaMRaT* workflows. Figures do not take into consideration the indexing step by either (*KaMRaT index* and *iMOKA create*). Only the fastest *iMOKA* parameters (without 100 repetition CV) are used. *KaMRaT*'s *rank-merge* scenario generally took less time and less memory than *merge-rank*, except when using a Bayes classifier with the larger data set and logistic regression with both data sets. *KaMRaT*'s *merge-rank* and *rank-merge* workflows were both faster than *iMOKA*, especially with the *rank-merge* workflow (14-19 time faster, except with Bayes and logistic regression rankings, vs. 6 to 7 times faster with *merge-rank*). The two exceptions in the *rank-merge* show that the "rank-first" strategy is heavily impacted by a slower ranking method, as expected. In the smaller data set, *KaMRaT merge-rank* takes similar or slightly longer time compared with *iMOKA*; its *rank-merge* workflow are 3 to 4 times faster than *iMOKA*, still with exception of Bayes and logistic regression rankings (respectively, 17% faster and twice slower).

Comparisons of peak memory (Fig. 4D) is not informative as *iMOKA* implements a maximum memory control (set at 100Gb in our tests) whereas *KaMRaT* does not. *iMOKA* used about three times more RAM than *KaMRaT* with the smaller dataset, but RAM usages were about the same for the larger dataset, indicating that *iMOKA* had triggered its memory limitation. This fact may explain the significantly slower run time of *iMOKA* on the large dataset.

On this problem again, features selected by the various *KaMRaT-rank* methods diverged strongly from each other and from those selected by *iMOKA*. Most *KaMRaT* selections shared with *iMOKA* between 4% and 13% of features before RF, and between 2% and 9% after RF (Supplementary Figure S2). The DIDS ranking method shared fewest features with *iMOKA*. As expected, the more complex problem problem (PRADtcga RvsNR) generally yield more divergent features than the simpler problem LUADseo TvsN, in spite of a smaller set of selected features ($\sim$1,500 vs $\sim$200,000 features). Among *KaMRaT* ranking methods, DIDS was also an outlier, either before or after RF selection (Supplementary Figure S3).

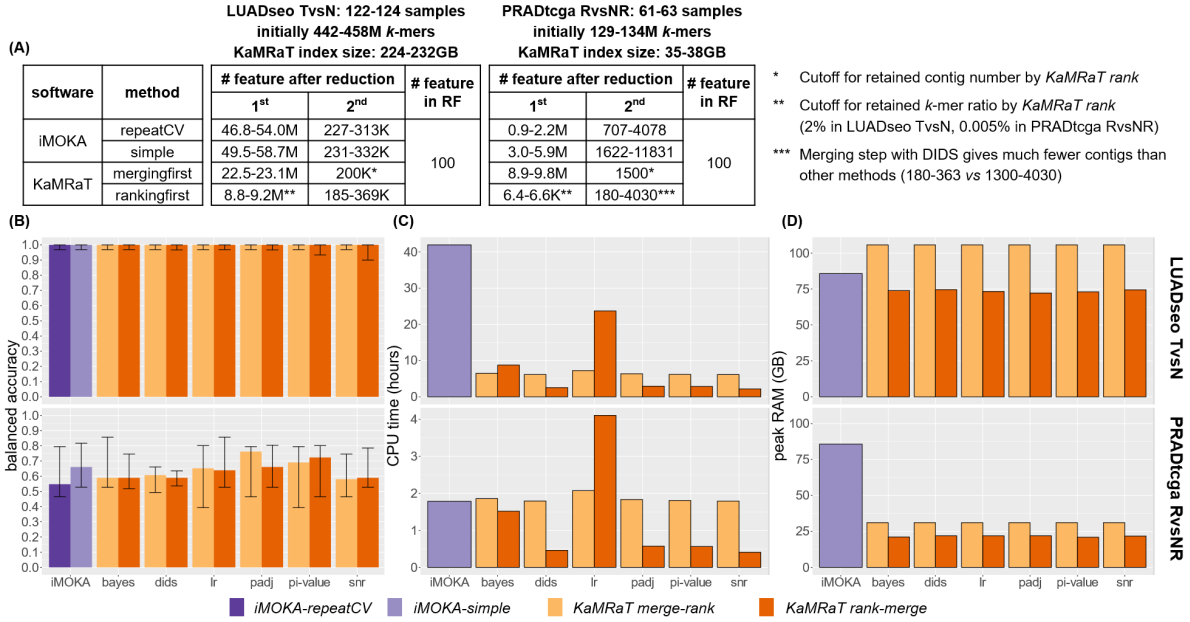


Figure 4: Characteristics of classifiers based on *KaMRaT* and *iMOKA reduce-aggregate*. (A) Training sets at initial state and during the reduction procedure. Values given as (x-y) correspond to the ranges obtained in each fold of the 5-fold sub-sampling. (B) Prediction performance evaluated by balanced accuracy for different *KaMRaT* workflows and for *iMOKA*. Bars represent median values and error bars represent minimum and maximum values. (C-D) CPU time and peak RAM usage for *iMOKA*, *KaMRaT merge-rank* and *rank-merge*. Bars represent mean values for CPU time and maximum values for peak RAM across folds.

3.4 Other *KaMRaT* applications

An attractive application of *KaMRaT* is the identification of contigs expressed exclusively in samples from one condition. As contigs are resolved at nucleotide precision, this can be useful for instance for identifying sources of tumor specific antigens. When applying a *KaMRaT filter-merge* workflow (see Methods for parameters), we could retrieve 970 tumor-specific contigs in the LUADseo TvsN dataset with runtimes below 1h after index construction (Table 2 and Figure S4).

Table 2: *KaMRaT index-filter-merge* CPU Time

	#samples	***#k*-mers**	***index***	***filter***	***merge***
LUADseo TvsN	77 vs 77	487M	3h19min	58min	<1s

Another interesting application of *KaMRaT* is *k*-mer selection by variance, followed by merging. This allows to reduce a large *k*-mer matrix into a matrix small enough to be submitted to PCA analysis for unsupervised discovery of sample groups. We ran such a *rank-merge* pipeline on the LUADseo TvsN dataset to determine whether the resulting contig set could distinguish tumor and normal samples in an unsupervised fashion. Unfortunately, this analysis could not be completed in time for this thesis submission. This result will be included before submitting this chapter for

publication.

4 Discussion

We developed *KaMRaT* to enable selection of condition-related k-mers while addressing challenges of k-mers' high interdependence and high dimension. An advantage of *KaMRaT* is that it follows a modular design, and thus allows for different workflows according to users' requirement. In contrast to other software, *KaMRaT* offers k-mer extension as an independent method from differential analysis or sample count-condition association evaluation, and includes mis-extension control in the module. To our knowledge, *KaMRaT* is the first k-mer analysis software that allows to perform k-mer extension before differential analysis [Audoux et al., 2017] or feature reduction [Lorenzi et al., 2020]. Besides, though designed primarily for k-mer matrix reduction, *KaMRaT*'s *rank* and *filter* modules apply to any generic count matrix such as gene-/transcript-expression matrices. This enables building classifiers from reference-free features (k-mers, contigs) and reference-based features (genes, transcripts) in a consistent, comparable way [Nguyen et al., 2021].

We examined different *KaMRaT* workflows and applications: (i) a *merge-rank* workflow where extension is applied on the whole set of k-mers and the resulting contigs are selected for condition association; (ii) a *rank-merge* workflow where k-mers are firstly evaluated-ranked and the extension is made only on the selected k-mers; (iii) a *filter-merge* workflow for extracting case-specific k-mer contig signals and, finally (iv) an *merge-rank* workflow for unsupervised sample clustering (to be completed). Other potential uses of *KaMRaT* not explored herein include (v) inter-cohort sequence search, for validating a list of biological events retrieved from one data set in another (using an index-query or index-mask-merge workflow) and (vi) analysis of novel biological events non-annotated in reference transcriptome, using an index-mask-merge workflow.

An important lesson from our comparison of ranking methods for classification purposes is that several ranking methods reach similar performance when used for feature selection in classifiers, yet have few features in common. This illustrates the importance of allowing users to test different ranking strategies.

Certain limitations of *KaMRaT* must be acknowledged. The code does not support parallel computing and still has a large memory footprint with real life data sets (e.g., about 100G RAM for a matrix of 450M k-mers and 154 samples). This is due to the duplication of contig sequence and its member k-mer indexed position. Potentially useful ranking functions are yet to be implemented, such as ranking based on continuous variables (e.g. survival). Also, unsupervised feature ranking is currently limited to a variance filter. One may consider other unsupervised means of reduction, such as count-based clustering, as for instance in [Sun et al., 2021].

In conclusion we think *KaMRaT*'s benefits mainly lie on three points: (i) a flexible approach and multi-functional usage. (ii) lightweight and fast ranking methods, (iii) performances in feature preselection on a par with state-of-the-art software. We hope this software will open new possibilities for applying statistical methods on k-mer signals and to advance cancer or other disease research by driving insights into the "subtranscript" level.

5 Availability

KaMRaT is open source under MIT license. Source code is available on GitHub `https://github.com/Transipedia/KaMRaT`, and a Docker image is available on Docker Hub `https://hub.docker.`

com/repository/docker/xuehl/kamrat. Scripts for producing results in this article are included in the same GitHub repository.

6 Acknowledgement

This work was supported by a grant by the Agence Nationale de la recherche for the project "Transipedia" [ANR-18-CE45-0020]. We express our great thankfulness to Rayan Chikhi who provided many useful suggestions for this work, and to Claudio Lorenzi who kindly helped on iMOKA software application.

References

[Abeshouse et al., 2015] Abeshouse, A., Ahn, J., Akbani, R., Ally, A., Amin, S., Andry, C. D., Annala, M., Aprikian, A., Armenia, J., Arora, A., et al. (2015). The molecular taxonomy of primary prostate cancer. *Cell*, 163(4):1011–1025.

[Audemard et al., 2019] Audemard, E. O., Gendron, P., Feghaly, A., Lavallée, V.-P., Hébert, J., Sauvageau, G., and Lemieux, S. (2019). Targeted variant detection using unaligned rna-seq reads. *Life science alliance*, 2(4).

[Audoux et al., 2017] Audoux, J., Philippe, N., Chikhi, R., Salson, M., Gallopin, M., Gabriel, M., Le Coz, J., Drouineau, E., Commes, T., and Gautheret, D. (2017). De-kupl: exhaustive capture of biological variation in rna-seq data through k-mer decomposition. *Genome biology*, 18(1):243.

[Bourgon et al., 2010] Bourgon, R., Gentleman, R., and Huber, W. (2010). Independent filtering increases detection power for high-throughput experiments. *Proceedings of the National Academy of Sciences*, 107(21):9546–9551.

[Bushmanova et al., 2019] Bushmanova, E., Antipov, D., Lapidus, A., and Prjibelski, A. D. (2019). rnaspades: a de novo transcriptome assembler and its application to rna-seq data. *GigaScience*, 8(9):giz100.

[Camacho et al., 2009] Camacho, C., Coulouris, G., Avagyan, V., Ma, N., Papadopoulos, J., Bealer, K., and Madden, T. L. (2009). Blast+: architecture and applications. *BMC bioinformatics*, 10(1):421.

[Clarke et al., 2008] Clarke, R., Ressom, H. W., Wang, A., Xuan, J., Liu, M. C., Gehan, E. A., and Wang, Y. (2008). The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature reviews cancer*, 8(1):37–49.

[Curtin et al., 2018] Curtin, R. R., Edel, M., Lozhnikov, M., Mentekidis, Y., Ghaisas, S., and Zhang, S. (2018). mlpack 3: a fast, flexible machine learning library. *Journal of Open Source Software*, 3:726.

[de Ronde et al., 2013] de Ronde, J. J., Rigaill, G., Rottenberg, S., Rodenhuis, S., and Wessels, L. F. (2013). Identifying subgroup markers in heterogeneous populations. *Nucleic acids research*, 41(21):e200–e200.

[Fan and Lv, 2008] Fan, J. and Lv, J. (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):849–911.

[Frazee et al., 2015] Frazee, A. C., Jaffe, A. E., Langmead, B., and Leek, J. T. (2015). Polyester: simulating rna-seq datasets with differential transcript expression. *Bioinformatics*, 31(17):2778–2784.

[Golub et al., 1999] Golub, T. R., Slonim, D. K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J. P., Coller, H., Loh, M. L., Downing, J. R., Caligiuri, M. A., et al. (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537.

[Guyon et al., 2002] Guyon, I., Weston, J., Barnhill, S., and Vapnik, V. (2002). Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1):389–422.

[Haury et al., 2011] Haury, A.-C., Gestraud, P., and Vert, J.-P. (2011). The influence of feature selection methods on accuracy, stability and interpretability of molecular signatures. *PloS one*, 6(12):e28210.

[Kokot et al., 2017] Kokot, M., Długosz, M., and Deorowicz, S. (2017). Kmc 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761.

[Liu et al., 2018] Liu, J., Lichtenberg, T., Hoadley, K. A., Poisson, L. M., Lazar, A. J., Cherniack, A. D., Kovatich, A. J., Benz, C. C., Levine, D. A., Lee, A. V., et al. (2018). An integrated tcga pan-cancer clinical data resource to drive high-quality survival outcome analytics. *Cell*, 173(2):400–416.

[Lorenzi et al., 2020] Lorenzi, C., Barriere, S., Villemin, J.-P., Bretones, L. D., Mancheron, A., and Ritchie, W. (2020). imoka: k-mer based software to analyze large collections of sequencing data. *Genome Biology*, 21(1):1–19.

[Martin, 2011] Martin, M. (2011). Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12.

[Marçais and Kingsford, 2011] Marçais, G. and Kingsford, C. (2011). A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770.

[Nguyen et al., 2021] Nguyen, H. T., Xue, H., Firlej, V., Ponty, Y., Gallopin, M., and Gautheret, D. (2021). Reference-free transcriptome signatures for prostate cancer prognosis. *BMC cancer*, 21(1):1–12.

[Pinskaya et al., 2019] Pinskaya, M., Saci, Z., Gallopin, M., Gabriel, M., Nguyen, H. T., Firlej, V., Descrimes, M., Rapinat, A., Gentien, D., de La Taille, A., et al. (2019). Reference-free transcriptome exploration reveals novel rnas for prostate cancer diagnosis. *Life science alliance*, 2(6).

[Rizk et al., 2013] Rizk, G., Lavenier, D., and Chikhi, R. (2013). Dsk: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653.

[Sanderson and Curtin, 2016] Sanderson, C. and Curtin, R. (2016). Armadillo: a template-based c++ library for linear algebra. *Journal of Open Source Software*, 1(2):26.

[Seo et al., 2012] Seo, J.-S., Ju, Y. S., Lee, W.-C., Shin, J.-Y., Lee, J. K., Bleazard, T., Lee, J., Jung, Y. J., Kim, J.-O., Shin, J.-Y., et al. (2012). The transcriptional landscape and mutational profile of lung adenocarcinoma. *Genome research*, 22(11):2109–2119.

[Soneson, 2014] Soneson, C. (2014). compcoder—an r package for benchmarking differential expression methods for rna-seq data. *Bioinformatics*, 30(17):2517–2518.

[Soneson and Delorenzi, 2013] Soneson, C. and Delorenzi, M. (2013). A comparison of methods for differential expression analysis of rna-seq data. *BMC bioinformatics*, 14(1):1–18.

[Sun et al., 2021] Sun, Q., Peng, Y., and Liu, J. (2021). A reference-free approach for cell type classification with scrna-seq. *bioRxiv*.

[Thomas et al., 2019] Thomas, A., Barriere, S., Broseus, L., Brooke, J., Lorenzi, C., Villemin, J.-P., Beurier, G., Sabatier, R., Reynes, C., Mancheron, A., et al. (2019). Gecko is a genetic algorithm to classify and explore high throughput sequencing data. *Communications biology*, 2(1):1–8.

[Van den Berge et al., 2019] Van den Berge, K., Hembach, K. M., Soneson, C., Tiberi, S., Clement, L., Love, M. I., Patro, R., and Robinson, M. D. (2019). Rna sequencing data: Hitchhiker's guide to expression analysis. *Annual Review of Biomedical Data Science*, 2(1):139–173.

[Xiao et al., 2014] Xiao, Y., Hsiao, T.-H., Suresh, U., Chen, H.-I. H., Wu, X., Wolf, S. E., and Chen, Y. (2014). A novel significance score for gene selection and ranking. *Bioinformatics*, 30(6):801–807.

7 Supplementary methods: Ranking method description

The scores of different ranking methods in *KaMRaT rank* are calculated and sorted as described below:

ttest.padj ranks features with samples in binary conditions. Firstly a $\log_2(x+1)$ transformation is applied to sample counts. Then each feature's association between sample counts and conditions is evaluated by p-value based on t-test, adjusted by Benjamini-Hochberg procedure for controlling false discovery rate. The scores are sorted from lowest value to the highest.

ttest.pi ranks features with samples in binary conditions. It is calculated with the formula given in [Xiao et al., 2014] as shown in equation 3. The scores are sorted from the highest value to the lowest.

$$\pi = -\log_{10}(p) \times |mean_{i \in G_1}[\log_2(S_i + 1)] - mean_{j \in G_2}[\log_2(S_j + 1)]| \tag{3}$$

where p is the non-adjusted; G_1 and G_2 are two sample groups; S_i and S_j are sample counts of two groups, respectively.

snr ranks features with samples in binary conditions. It is calculated by dividing the difference between group means by the sum of group standard deviations, followed by what proposed in [Golub et al., 1999], as shown in equation 4. The scores are sorted with their absolute value from the highest to the lowest.

$$snr = \frac{|mean_{i \in G_1}(S_i) - mean_{j \in G_2}(S_j)|}{stddev_{i \in G_1}(S_i) + stddev_{j \in G_2}(S_j)} \tag{4}$$

where G_1 and G_2 are two sample groups; S_i and S_j are sample counts of two groups, respectively.

dids ranks features both with binary sample conditions and multiple sample conditions. It is a generalized version of what proposed initially in [de Ronde et al., 2013], where the authors had estimated the sum of the extent to which outlier samples in the second group exceeds the first group. Here we generalized this to multi-group situations: we estimated for each group sum of the extent that outlier samples exceeds samples in other group(s), and returned the maximum value as the final score, as shown below. The scores are sorted from the highest to the lowest value.

$$\begin{aligned} \hat{S}_g &= max_{i \in g}(S_i) \\ dids &= max_g \left(\sum_{j \notin g} \sqrt{|S_j - \hat{S}_g|^+} \right) \end{aligned} \tag{5}$$

where g is a certain group; S_i is the count of sample i; $|x|^+$ is x if $x > 0$ or 0 otherwise.

lr ranks features with samples in binary conditions. It estimates classification accuracy of logistic regression, calculated by *MLPack* [Curtin et al., 2018] library. "lr" ranking method applies a standarization preprocess to feature counts, i.e., minus all components of sample count vector by their mean value, and then divide them by the standard deviation. It contains a functionality provided by *MLPack* that distributes samples into n (given by user) folds, and estimate the accuracy with n-fold cross-validation. As a machine-learning based method, we provided thereby a functionality for batch effect removal, providing simultaneously sample count vector and batch label vector as the input object for predicting the output sample conditions for each feature. The scores are sorted from the highest to the lowest value.

bayes ranks features both with binary sample conditions and multiple sample conditions. It estimates classification accuracy of Bayes classifier, calculated by *MLPack* [Curtin et al., 2018]

library. "bayes" ranking method applies a standarization preprocess to feature counts, i.e., minus all components of sample count vector by their mean value, and then divide them by the standard deviation. It contains a functionality provided by *MLPack* that distributes samples into n (given by user) folds, and estimate the accuracy with n-fold cross-validation. As a machine-learning based method, we provided thereby a functionality for batch effect removal, providing simultaneously sample count vector and batch label vector as the input object for predicting the output sample conditions for each feature. The scores are sorted from the highest to the lowest value.

sd ranks features without considering sample conditions (i.e., in non-supervised fashion). It estimates each features' standard deviation across samples, and sort them from the largest to the smallest.

8 Supplementary Figures and Table

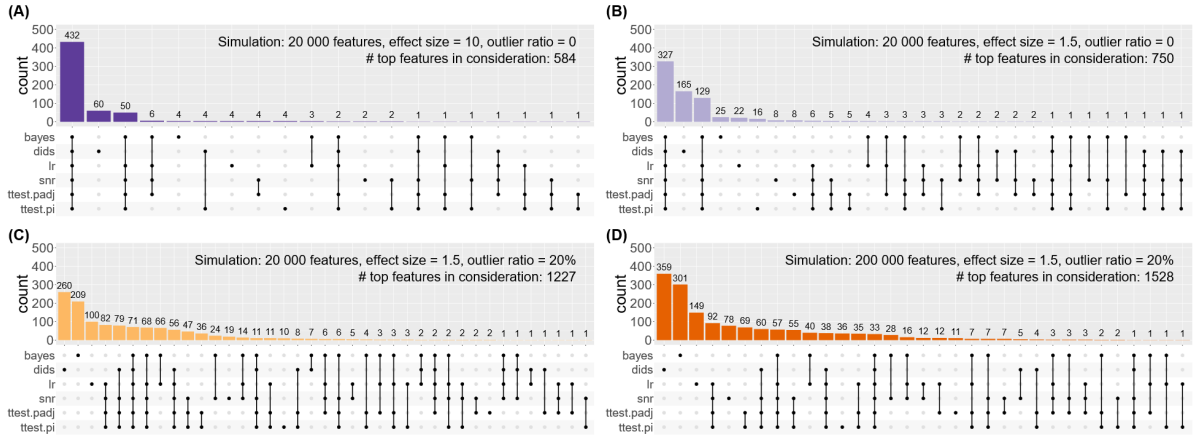


Figure S1: Intersection among top 500 features by different ranking methods in *compcodeR* simulated matrices, plotted via *ggupset* package in R. Each sub-figure correspond to a simulated situation.

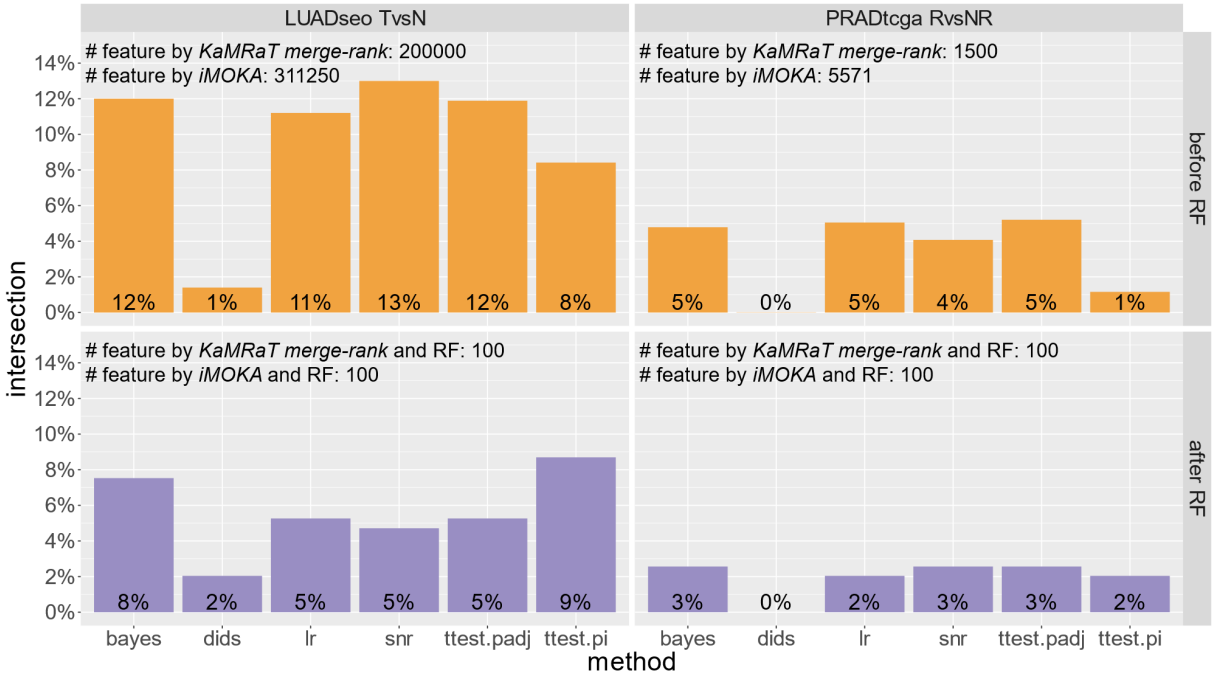


Figure S2: Feature comparison between *KaMRaT merge-rank* approach and *iMOKA reduce-aggregate* on real data sets among different ranking methods. The bars represent percentages of features in the intersection parts.

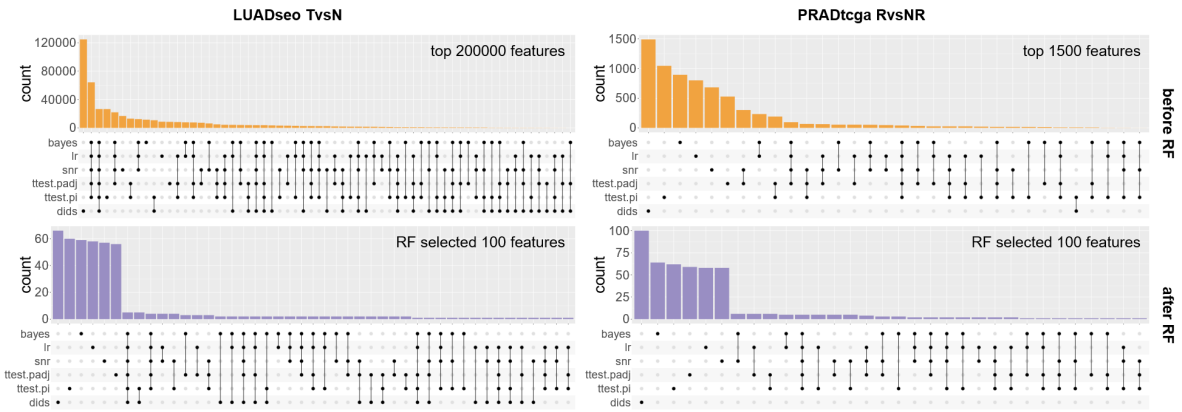


Figure S3: Feature comparison across ranking methods in *KaMRaT merge-rank* approach on both real data sets, plotted via *ggupset* package in R.

Table S1: PR AUC values for different ranking methods with *compcodeR* simulation

#feature	**effect size**	**%outlier**	**bayes**	**dids**	**lr**	**snr**	**ttest.padj**	**ttest.pi**
20,000	10	0	0.990	0.929	0.991	0.992	0.992	0.998
20,000	1.5	0	0.958	0.725	0.960	0.986	0.968	0.978
20,000	1.5	20	0.265	0.435	0.597	0.872	0.958	0.961
200,000	1.5	20	0.088	0.206	0.470	0.637	0.811	0.801

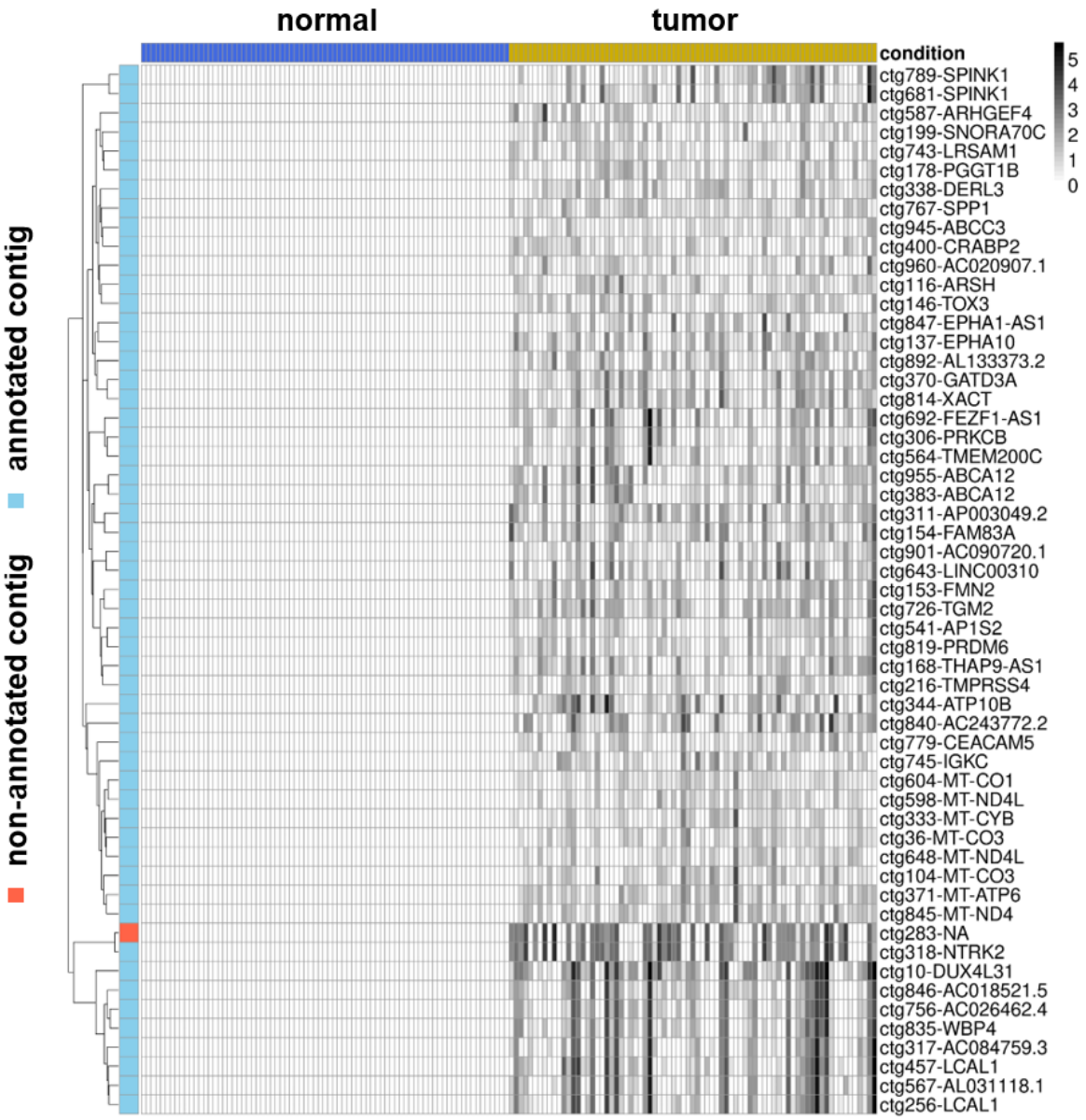


Figure S4: Heatmaps of top *filter-merge* contigs in both real data sets. Counts are not normalized, but scaled with $\log(x+1)$ transformation.

Chapter 4

k-mer Classifiers for Cancer Prognosis

4.1 Motivation

Realizing that RNA-seq had a capacity of completely capturing transcriptome signals and that state-of-the-art classifiers were all reference-based and ignored non-canonical RNAs produced in disease tissues, we attempted to build a classifier based on reference-free features, i.e., k-mer contigs. We tested our classifier in a real problem of prostate cancer prognosis and compared it to a conventional transcriptome classifiers.

Thereby, we constructed a reference-free classifier using k-mer contigs, and benchmarked it against a conventional reference-based classifier using gene expression signals. We guaranteed that the benchmarking workflows were as similar as possible between the two approaches:

- Same feature reduction (Bayes) and feature selection (LASSO + stability selection) method;
- Same machine-learning model (logistic regression) for classifier construction;
- Same evaluation metrics (ROC-AUC) on same independent validation data sets.

4.2 My contribution

In this project, I helped implement the different steps for gene and k-mer contig classifier construction into a series of C++ programs (merge k-mers into contigs, normalization, feature ranking with Bayes classifier, and k-mer masking with a

given list of sequences), which were actually predecessors of the KaMRaT software. In the merging task, I proposed mean absolute contrast as a way of improving merging correctness (Fig.2). I participated in writing sections *Reduction of k-mer matrix via contig extension*, *Count normalization*, and *Univariate features ranking*, as well as in responding to reviewers comments.

4.3 Article

Nguyen *et al. BMC Cancer* (2021) 21:394
https://doi.org/10.1186/s12885-021-08021-1

BMC Cancer

RESEARCH ARTICLE Open Access

Reference-free transcriptome signatures for prostate cancer prognosis

Ha T.N. Nguyen[1], Haoliang Xue[1], Virginie Firlej[2], Yann Ponty[3], Melina Gallopin[1] and Daniel Gautheret[1*]

Abstract

Background: RNA-seq data are increasingly used to derive prognostic signatures for cancer outcome prediction. A limitation of current predictors is their reliance on reference gene annotations, which amounts to ignoring large numbers of non-canonical RNAs produced in disease tissues. A recently introduced kind of transcriptome classifier operates entirely in a reference-free manner, relying on k-mers extracted from patient RNA-seq data.

Methods: In this paper, we set out to compare conventional and reference-free signatures in risk and relapse prediction of prostate cancer. To compare the two approaches as fairly as possible, we set up a common procedure that takes as input either a k-mer count matrix or a gene expression matrix, extracts a signature and evaluates this signature in an independent dataset.

Results: We find that both gene-based and k-mer based classifiers had similarly high performances for risk prediction and a markedly lower performance for relapse prediction. Interestingly, the reference-free signatures included a set of sequences mapping to novel lncRNAs or variable regions of cancer driver genes that were not part of gene-based signatures.

Conclusions: Reference-free classifiers are thus a promising strategy for the identification of novel prognostic RNA biomarkers.

Keywords: Reference-free transcriptomic, Supervised learning, Prostate cancer signature

Introduction

The outcome of human cancer can be predicted in part through gene expression profiling [1–3]. Outcome prediction is particularly important in prostate cancer (PCa), where distinguishing indolent from aggressive tumors would prevent unnecessary treatment and improve patients' quality of life. However, currently there is no reliable signature of aggressive prostate cancer. Pathologists classify prostate tumor biopsies using scoring systems such as the Gleason score that evaluates tumor differentiation and Tumour, Node, Metastasis (TNM) staging that evaluates tumor extent and propagation. Gleason, TNM and Prostate-specific antigen (PSA) levels can be combined into a low, medium or high risk status [4]. Several studies used gene expression profiles to derive predictors of Gleason score or risk [5–8]. Other studies predicted actual clinical progression (tumor recurrence or metastasis) after several years of patient followup. Clinical progression can be evaluated either indirectly through monitoring of PSA levels (BCR=biochemical relapse) [9–12] or upon direct clinical observation [13–16]. Gene expression predictors usually take the form a of signature, that is a set of genes or transcripts and associated coefficients of a model that can be used to predict risk or outcome from a patient sample. Commercial tests such as Decipher and Oncotype DX predict prostate cancer risk based on gene expression. However these are still not recommended for routine use [17]. In general, the prostate cancer community has progressed pretty well at identifying low and high risk patients, but men with mid-range

*Correspondence: daniel.gautheret@universite-paris-saclay.fr
[1]Institute for Integrative Biology of the Cell, UMR 9198, CEA, CNRS, Université Paris-Saclay, Gif-Sur-Yvette, France
Full list of author information is available at the end of the article

risk face more uncertainty and would most benefit from improved tests.

Gene expression profiling of prostate biopsies is performed either using DNA microarrays [13–16] or high throughput RNA sequencing (RNA-seq) [5–8]. An important advantage of RNA-seq is its ability to identify novel genes or transcripts, which can in principle be incorporated into predictive signatures. However, RNA-seq analysis is usually performed in a "reference-based" fashion, ie. by using RNA-seq reads to quantify a predetermined set of transcripts. This amounts to using RNA-seq in the same way as a microarray that only quantifies a predetermined set of probes. Yet, there is abundant evidence that non-reference RNAs are frequent in disease tissues and may constitute clinically useful biomarkers [18]. Therefore one may expect that prognostic models incorporating non-reference RNAs may carry substantial benefits.

Our group [19, 20] and others [21] introduced new k-mer based strategies to analyse RNA-seq data in a "reference-free" manner, that is without mapping sequence reads to a predefined set of genes or transcripts. K-mers are sub-sequences of fixed length which are extracted and quantified from sequence files. When applied to medical RNA-seq datasets using appropriate statistical methods, this strategy identifies any subsequence whose increased abundance is associated to a given clinical label. This may include novel splice variants, long non-coding RNAs (lncRNAs) or RNAs from repeated retroelements [19, 20] which are ignored by conventional protocols based on reference gene annotations.

Although attractive in principle, k-mer derived prognostic signatures pose two major challenges. First, a single RNA-seq dataset commonly contains tens to hundreds of millions distinct k-mers. Therefore false positive and replicability issues encountered with gene expression profiles [22–25] are expected to worsen with k-mer count matrices. The second challenge is related to the transfer of a k-mer signature across independent datasets. Signatures inferred from an initial discovery set are expected to generalize to any independent dataset. In the absence of a unifying gene concept, independent validation requires matching signature k-mers to read sequences from the new dataset. This may cause significant signal loss if sequencing or library preparation technologies differ.

Our main objective here was to compare the characteristics and performances of reference-based and reference-free classifiers for PCa risk and relapse prediction. We built both types of classifiers using the same discovery dataset and assessed their performances in independent datasets using equivalent pipelines and parameters. For the reference-free approach, this required special developments to reduce the number of variables and to transfer expression measures between datasets. We present below a detailed analysis of the relative performances and sequence contents of the different classifiers and discuss possible future developments to improve performances of models.

Materials and methods

Data acquisition and outcome labelling

We used tumor samples from the TCGA-PRAD data collection [26] (N=505) for signature discovery. The resulting classifiers were then assessed in two independent datasets, from the Canadian Prostate Cancer Genome Network (ICGC-PRAD-CA) [27] (N=148) and from the Portuguese Oncology Institute's "Porto" cohort, analyzed in Stelloo et al. [28] (N=91). All three datasets were produced from radical prostatectomies and used similar technologies for library preparation (frozen samples, poly(A)+ RNA selection) and Illumina sequencing, however they differed by read-size, read depth, strandedness and use of single or paired ends sequencing (Table 1).

TCGA-PRAD RNA-seq data were retrieved from dbGAP accession phs000178.v9.p8 with permission. ICGC-PRAD-CA RNA-seq data (EGAD00001004424) were downloaded from the European Genome-Phenome Archive (EGA) with permission. The RNA-seq files from the "Porto" cohort [28] were retrieved from GEO, under accession GSE120741. Clinical information was retrieved from Liu et al. [29] for TCGA-PRAD, from Fraser et al. [27] for ICGC-PRAD and from sample metadata of GEO accession GSE120741 for Stello et al. [28].

We built predictors for risk and relapse using two-class prediction models. To achieve a clear separation between the two classes, we only focused on high risk (HR) samples versus low risk (LR) samples, ignoring the medium risk, and we focused on relapse prior to a given year and non-relapse after a given year. For this reason, only a fraction of samples could be labelled for a given class in each set. Risk information was not available in the Stelloo dataset and relapse labelling on the ICGC dataset led to a small validation set (only 7 relapse samples).

We classified tumor specimens into low-risk and high-risk groups using an adaptation of d'Amico's classification which does not take into account the PSA rate but only the anatomo-pathological data on the basis of Gleason and TNM features as performed previously [20]. Tumors with Gleason score 6/7 (3+4) and TNM stage pT1/2 were classified as low risk. Tumors with Gleason score 8/9 and/or TNM stage pT3b/4 were defined as high-risk. Tumors classified as pT3a, pT1 or (pT2 and Gleason (4+3)) were considered as intermediate and excluded from the analysis. 374 TCGA-PRAD tumors and 63 ICGC-PRAD-CA tumors could be labelled for LR or HR. We could not obtain Gleason/TNM scores for Stelloo et al, hence we did not annotate risk for this cohort.

Table 1 Characteristics of prostate tumor RNA-seq datasets

Study	RNA-seq library type	Reads/sample	#Tumor samples	Risk		Relapse	
				LR	HR	NO	YES
TCGA-PRAD	Poly(A)+ unstranded 2x50nt	130M	505	134	240	56	58
ICGC-PRAD	Poly(A)+ stranded 2x100nt	313M	148	40	23	49	7
STELLOO	Poly(A)+ stranded 1x65nt	20M	91			43	48

For relapse analysis, we distinguished patients with biochemical relapse (BCR) and time to BCR <2yr and patients with no BCR after 5 years or longer, except for Stelloo et al. where only precomputed relapse data was available with cutoffs at 5yr and 10yr, respectively (Table 2). BCR information was obtained from Table S1 of Liu et al. [29] for TCGA-PRAD and from table S1 (PFS field) of Fraser et al. [27] for ICGC-PRAD. Precomputed relapse data for Stelloo et al. was taken from SRA accession PRJNA494345.

A generic framework to infer reference-based and reference-free signatures

Risk and relapse predictors were derived using a combination of feature selection and supervised learning (Fig. 1). The predictive model was tuned over a discovery (or training) dataset and its performance was then evaluated on an independent validation (or testing) dataset, to avoid selection bias [30]. The same procedure was used for reference-based and reference-free models, however two extra steps were included to obtain and validate reference-free signatures. First a procedure was implemented to reduce the k-mer matrix using a sequence assembly-like algorithm to merge k-mers into contigs based on their sequence overlap and on the similarity of their count vectors. This step led to a contig count table an order of magnitude smaller than the initial k-mer count table (see "Results" section below). Feature selection and model fitting were performed over this contig table. A second adaptation was necessary to validate the reference-free signature in an independent dataset. This required extracting k-mers from both the signature and the sequence files of the independent set, and compute the signature expression in the independent set based on counts of matching k-mers. The pipeline is detailed in Methods. Note that we select features and train a predictive model only on the discovery dataset. The model is then applied to the validation set with no retraining (i.e. with the same coefficients) for an unbiased evaluation of the signature.

Gene and k-mer count matrices

DEkupl-run [19] was used to produce gene and k-mer count matrices for each dataset. DEkupl-run converts FASTQ files to k-mer counts using Jellyfish [31], joins individual sample counts into a single count table and filters out low count k-mers. K-mer size was set to 31, lib_type to unstranded, and parameters min_recurrence and min_recurrence_abundance were set for each dataset as in Additional file 4: Table S1. K-mer size was set to 31 as commonly adopted for human transcriptome applications [19, 32]. Note that contrary to TCGA-PRAD, ICGC-PRAD uses stranded RNA-seq libraries. However we could not use this information as signatures were produced from unstranded libraries. We thus built all k-mer tables in canonical mode, which amounts to consider all libraries as unstranded. Gene expression was computed using Kallisto v0.43.0 [32] with Gencode V24 as a reference transcriptome. Gene-level counts were obtained by summing counts for all transcripts of each gene. Gene expression matrices were submitted to the same recurrence filters as k-mer tables to remove low expression genes. After count tables were generated and filtered, the k-mer merging and differential expression analysis module of DEkupl-run were not used. Instead, tables were further processed as explained below.

Reduction of k-mer matrix via contig extension

k-mer occurence tables were converted into contig occurence tables using an extension procedure similar

Table 2 Relapse group definitions

Relapse group	TCGA-PRAD	ICGC-PRAD	STELLOO
Relapse (YES)	PFS = 1 and PFS.time <2yr	BCR = "Yes" and BCR.time <2yr	BCR = "Yes" and BCR.time <5yr
Non relapse (NO)	PFS = 0 and PFS.time >5yr	BCR = "No" and BCR.time >5yr	BCR = "No" and BCR.time >10yr

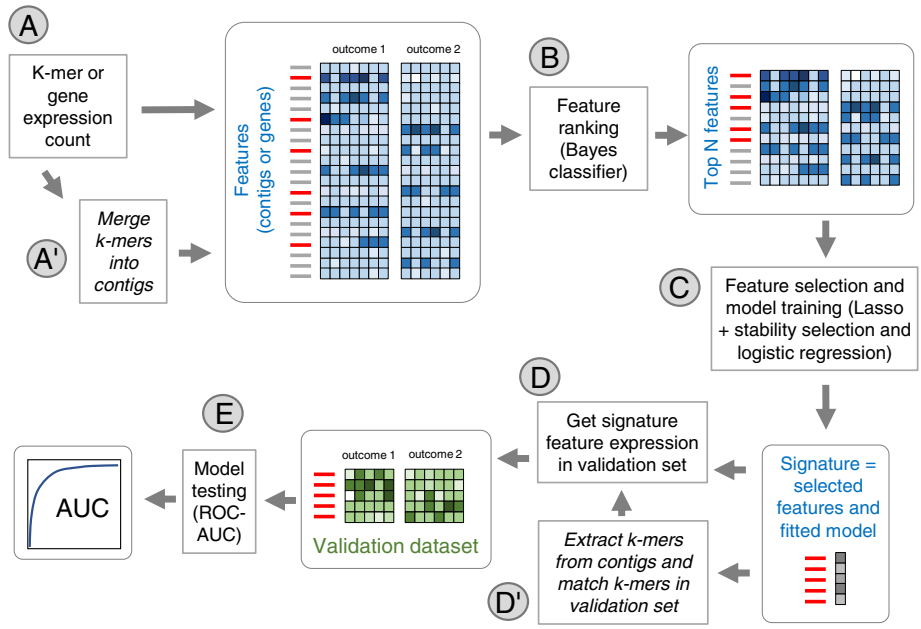


Fig. 1 Uniform procedure for signature inference based on k-mer or gene expression. **a** The discovery matrix is built from normalized k-mer counts or gene expression counts. Samples are labelled by their outcome (risk or relapse) status. Normalization is performed as count per billion for k-mers or count per million for genes. **b** Features are ranked according to their F1-score computed by cross validation using a Bayes classifier (BC). The top 500 features are retained. **c** Among the top 500, features are selected using lasso logistic regression combined with stability selection. A logistic regression is tuned on the selected features. **d** Features from the signature are measured in the count matrix from an independent dataset. **e** Performance of the signature (selected features + tuned logistic regression) is evaluated using Area Under ROC Curve (AUC) on the validation dataset. To deal with the specificity of k-mer matrices, extra steps A′ and D′ are introduced: **a**′ the k-mer matrix in converted into a much smaller contig matrix by merging overlapping k-mers with compatible counts. **d**′ k-mers are extracted from the signature contigs and their counts in the validation matrix are aggregated

to that described in Audoux et al. [19]. We define here as contig any sequence produced by merging 1 or more k-mers. Briefly, contigs overlapping by (k-1) to (k-15) nucleotide were iteratively merged into longer contigs till any of the following condition was encountered. In a straightforward case, extension stops when no more overlapping contig is available. Alternatively, extension stops when ambiguity is introduced i.e. when competing extension paths occur. Lastly, we applied here an intervention not included in Audoux et al. [19] by considering sample count compatibility between contigs, as shown in Fig. 2. Sample count compatibility is measured by the mean value of absolute contrast (MAC) between the counts of the two contigs across all samples, i.e.

$$\mathrm{MAC}\,(\mathbf{c_1}, \mathbf{c_2}) = mean_{s\in\{samples\}}\left(\left|\frac{c_{1,s} - c_{2,s}}{c_{1,s} + c_{2,s}}\right|\right)$$

where $\mathbf{c_1}$ and $\mathbf{c_2}$ are count vectors of two contigs to be merged, and $c_{1,s}$ and $c_{2,s}$ are counts in sample s from the corresponding count vectors. The extension is rejected if MAC $>$ 0.25. In this way, all contigs are guaranteed to have member k-mers with consistent sample count vectors. After the merging procedure, the new contig's sample count vector is set to the mean of composite k-mer's sample count vectors.

Count normalization

To account for differences in sequencing depth among samples, we applied a normalization step on feature counts (genes or contigs) in discovery and validation datasets. Each feature count in a sample is divided by the sum of all feature counts in this sample, then multiplied by a constant base number:

$$e_{f,s} \leftarrow \frac{e_{f,s}}{\sum_{f\in\{features\}} e_{f,s}} \cdot C_b,$$

where $e_{f,s}$ refers to count of feature f in sample s, and C_b is the base constant. For genes, $C_b = 10^6$ resulting in a conventional count per million (CPM) normalization, while for contigs, we used $C_b = 10^9$, or count per billion (CPB). For contigs, normalization is applied on the contig count table produced after contig extension and for genes it is applied on the recurrence filtered gene expression matrix.

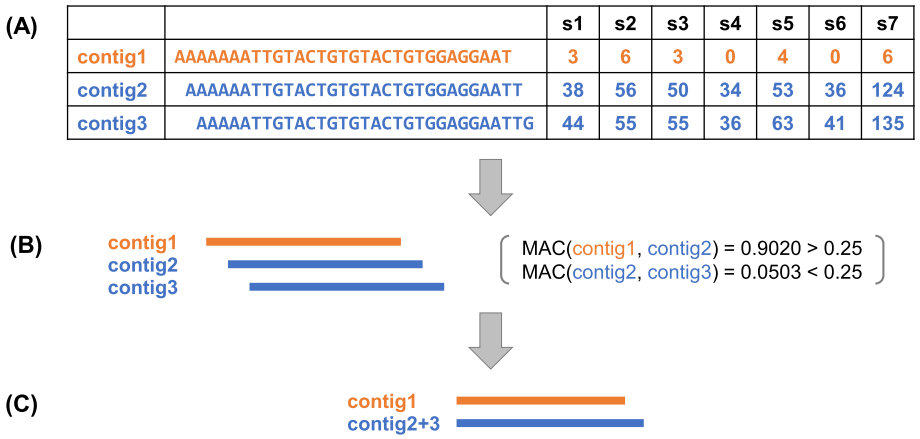


Fig. 2 Merging procedure of 3 example contigs: **a** Count table of contigs in samples. Both pairs *(contig1, contig2)* and *(contig2, contig3)* have good overlaps shifting by only one nucleotide, but the sample count vectors of *contig1* and *contig2* are not compatible. **b** Merging intervention considering sample count compatibility between contigs. The mean absolute contrast (MAC) is calculated for each pair, and merging of *(contig1, contig2)* is rejected due to a MAC value exceeding threshold. **c** The resulting contigs are the initial *contig1* and the merged contig from the initial *(contig2, contig3)* pair

Univariate features ranking

Given the limited number of samples, it was necessary to reduce the number of features (genes or contigs) in the dataset. We discarded irrelevant features to focus on a subset of 500 top candidates for subsequent feature selection. To rank features, we selected a Bayes classifier because the C++ implementation of this classifier was the fastest to run among several available feature ranking tools. We did not try to optimize this part to avoid biasing the comparison towards gene-based or gene-free methods. In detail, we performed prediction of status (risk/relapse) using a Bayes classifier on each independent feature, after log transformation of the normalized counts (after adding an offset 1 to avoid numerical problem). To assess the quality of the prediction, we computed the average f_1 score by 5-fold cross validation ($f_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, where precision $= TP/(TP + FP)$ and recall $= TP/(TP + FN)$ and FP, TP, FN are respectively the False Positive, True Positive and False Negative). In cases where 5-fold cross-validation returned an undefined value, f_1 score was set to 0 (the worst). The average f_1 score was used to rank features. The Bayes classifier implementation was taken from the `MLPack` library [33].

Feature selection, model fitting and predictor evaluation

To select a subset of non-correlated features (genes or contigs) among the top 500 candidates, we performed penalized logistic regression using the implementation from the `glmnet` R package [34]. We implemented stability selection [35]: only features selected with a frequency of being selected above 0.5 upon 2000 resamples of the input dataset were retained. To evaluate the performance of the selected features on the discovery (training dataset), we fitted a logistic regression and computed the area under the ROC curve (AUC) using a 10-fold cross validation scheme, repeated 20 times, as implemented in the `caret` package [36]. To handle imbalanced datasets, we included optional oversampling and downsampling in our evaluation procedures [37]. We also computed the Precision-Recall AUC, a more informative metric than the ROC AUC when evaluating binary classifiers on imbalanced datasets [38]. To assess the performance of the signature on the external validation datasets, we fitted a logistic regression on the whole discovery dataset and applied the predictor to the validation datasets. In the reference-free approach, some features present in the signature were not found in the validation (see below). In this case, the coefficient of the logistic regression corresponding to missing features were set to zero. Signature contigs were annotated through BLAST alignment *vs.* Gencode V34 transcripts. HGNC symbols for signature genes were obtained from the Ensembl EnsDb.Hsapiens.v79 R package [39].

Matching signature contigs in the validation cohort

To measure contig expression in the validation cohort we implemented the procedure schematized in Fig. 3. The procedure comprises two main steps: (1) all k-mers from signature contigs were extracted and identified in the k-mer count matrix generated from the validation cohort and (2) the resulting sub-matrix was used to estimate each contig's expression in the validation cohort, measured for each sample as the median of extracted k-mer counts.

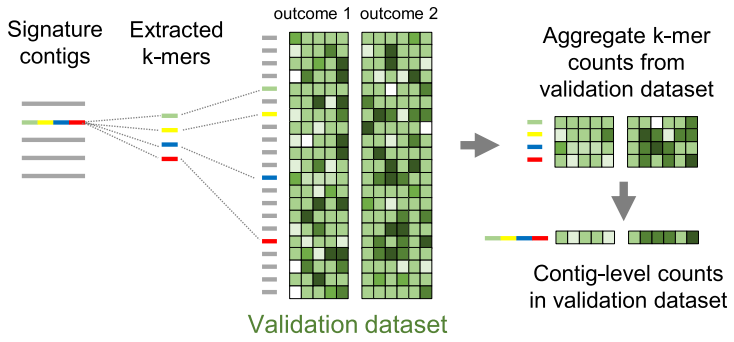


Fig. 3 Procedure for inferring signature contig expression in an independent validation dataset. The colored contig from the signature is quantified in the validation cohort by extracting all its constituent k-mers and retrieving the corresponding k-mer counts from validation k-mer count matrix. The count vector of the contig in each sample of the validation dataset is taken as the median of counts for k-mers in this sample

Results

A reference-free risk signature for prostate cancer

We first applied the gene-free and gene-based signature discovery procedures detailed above to infer PCa risk signatures. The k-mer table for 374 TCGA-PRAD risk-labelled samples (Fig. 4a) had 94M k-mers after low count filtering. The merging step reduced it to 5.2M contigs, i.e. achieving a considerable 18-fold reduction in size (Fig. 4b). Contig sizes (mean=49nt, median=34nt, Table 3) were small relatively to a typical human RNA, which is characteristic of the adopted contig extension procedure [19] (see "Reduction of k-mer matrix via contig extension" section).

The 5.2M contig matrix and the 38k gene expression matrix were submitted to screening using univariate Bayes classification and the top scoring 500 features were retained for feature selection and model fitting. Interes-

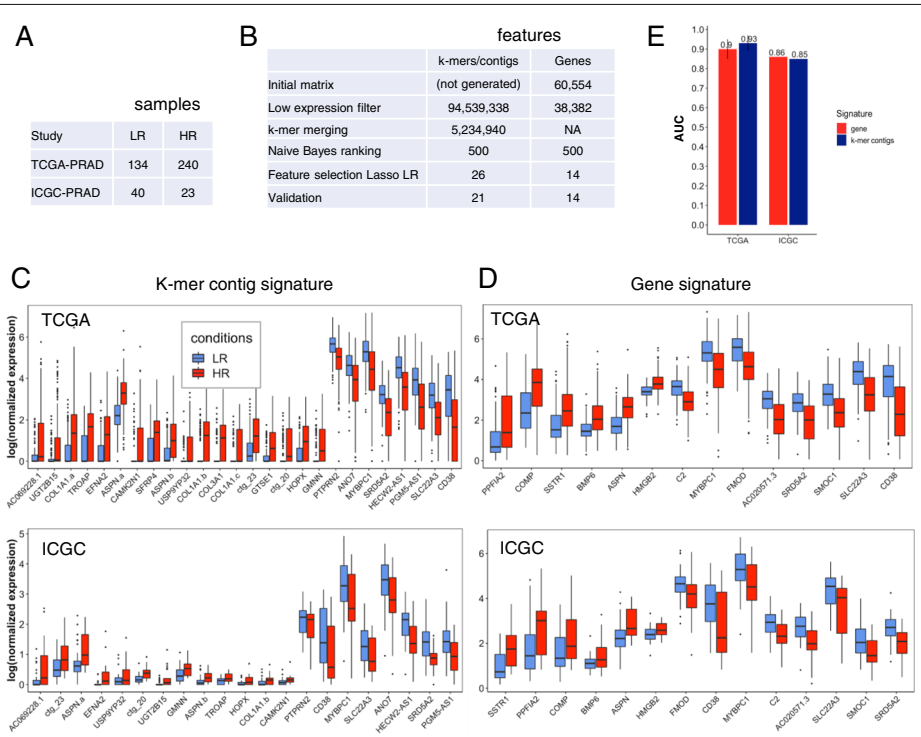


Fig. 4 Risk signatures generation and analysis. **a** Characteristics of prostate tumor RNA-seq datasets. **b** Result of filtering procedure on the k-mer and gene matrices for risk analysis. Expression of risk signature elements in LR and HR samples in the TCGA-PRAD and ICGC-PRAD cohorts **c** k-mer contig signature; **d** Gene signature. **e** Signature performances for risk prediction in the TCGA-PRAD and ICGC-PRAD cohorts

Table 3 Contig sizes (Risk model)

	After k-mer merging	After Bayes classifier ranking
Mean contig size (nt)	49.1	189
Median contig size (nt)	34	61

tingly, the 500 top scoring contigs were significantly longer than prior to selection (median 61nt vs. 34nt, Table 3), suggesting the procedure tended to eliminate spurious short contigs.

Finally, Lasso logistic regression produced a reference-free signature of 26 contigs and a reference-based signature of 14 genes (Fig. 4b). Ten-fold cross validation performances of both signatures were very high on the discovery dataset (0.90 and 0.93 for genes and k-mers, respectively) (Fig. 4e), which is an over-estimated performance since features here were tested on the same dataset used to select features [30]. PR-AUC and ROC-AUC on different sampling techniques to adjust the class distribution of a dataset are also presented in Additional file 4: Table S2. These results lead to the same conclusion as the ones presented in (Fig. 4e).

Figure 4c shows the 26 contigs in the reference-free risk signature and their abundance distribution in LR and HR samples. 24/26 contigs mapped Gencode transcripts from 21 unique genes (Additional file 1). Eleven of the 21 genes were also found in a list 180 genes compiled from published PCa outcome signatures (Additional file 2), which is a highly significant enrichment (*P*-value = 7.9e-9, Fisher's exact test), especially when considering that no gene information was used to infer our signature. The gene and contig signatures involved five shared genes: MYBPC1, ASPN, SLC22A3, SRD5A2 and CD38 (Additional file 2, Fig. 4c and d). The first four genes are part of published prostate risk signatures. CD38 is particular in that it is the most downregulated in both signatures and it is not part of previous signatures. However, downregulation of this gene has been associated with poor outcome in prostate cancer [40], supporting its status as a high risk biomarker. Risk signature contigs mapped at least five other genes with established driver roles in PCa or other cancers: CAMK2N1 [41], COL1A1 [42], GTSE1 [43] and PTPRN2 [44], supporting the relevance of these sequence contigs in PCa etiology.

Of the two contigs that did not map any Gencode transcript, one aligned to an intron of GMNN (ctg_20), a gene also mapped by an exonic contig, the other an intron of LDLRAD4 (ctg_23). Contig ctg_23 corresponds to a 1.29 kb spliced transcript located between exons 4 and 5 of LDLRAD4 and is strongly upregulated in HR samples, as displayed in the Integrative Genomics Viewer (IGV) [45] in Additional file 4: Figure S1. Although ctg_23 partly maps short annotated LDLRAD4 isoforms, its expression seems unrelated to that of the longer LDLRAD4 transcripts whose coverage in flanking exons is 4-6 times lower than ctg_23 (Additional file 4: Figure S2.) Therefore ctg_23 likely comes from an independent lncRNA. The host gene LDLRAD4 is a negative regulator of TGF-beta signaling with roles in proliferation and apoptosis and was recently associated to negative outcome in other tumor types [46, 47]. Lastly, one contig (ctg_11, EFNA2) was probably misassigned to the EFNA2 gene since it maps to a highly expressed discrete area just 3' of EFNA2 while EFNA2 seems silent. Thus ctg_11 probably comes from an independent lncRNA as well (Additional file 4: Figure S3).

To assess the replicability of risk signatures, we evaluated their performance in the ICGC-PRAD independent dataset. To this aim, we developed a specific procedure to estimate the expression of an arbitrary sequence contig across datasets using matched k-mers (see "Materials and methods" section). The 26 contigs represented 1444 k-mers, of which 97% were present in the ICGC-PRAD validation dataset. Overall 5 contigs (SFRP4, GTSE1, COL3A1, COL1A1.a, COL1A1.c) could not be quantified in the validation set due to lack of supporting k-mers (see Fig. 4b and c). In spite of this, the reference-free signature had similar performance in the validation set as the reference-based signature (0.85 and 0.86 respectively, Fig. 4e), although the later did not sustain any loss when transferred to the independent cohort (Fig. 4b). High prediction AUCs observed in the independent validation cohorts indicate a strong replicability of both the reference-free and reference-based risk signatures.

Relapse signatures contain key PCa drivers

For relapse prediction, we distinguished patients with biochemical relapse within less than 2 years and patients with no BCR after 5 years or longer. Application of the gene-free and gene-based signature discovery procedures to relapse prediction produced a 14-contig reference-free signature and a 10-gene reference-based signature (Additional file 2, Fig. 5b, c and d). The reference-free signature was populated by obvious PCa drivers. Strikingly, 3 contigs matched KLK2, AR and KLK3, which are among the most important genes in PCa onset and progression [48], the androgen receptor (AR) and two of its main targets, KLK2 and KLK3, the later encoding the PSA protein (Fig. 5c). Another contig matched SPDEF, a gene whose loss is associated to PCa metastasis [49].

Contigs matching KLK2 and AR were overexpressed 23-fold and 7-fold, respectively in relapsed patients while the contig matching KLK3 was depleted 1.8 fold. The AR contig matches exon 1 of AR and contains an non-templated poly-A end but no visible polyadenylation signal. The

A

samples

Study	NO	YES
TCGA-PRAD	56	58
ICGC-PRAD	49	7
STELLOO	43	48

B

features

	k-mers/contigs	Genes
Initial matrix	(not generated)	60,554
Low expression filter	97,731,857	36,006
k-mer merging	6,184,108	NA
Naive Bayes ranking	500	500
Feature selection Lasso LR	14	10
ICGC-PRAD validation	12	10
STELLOO validation	3	10

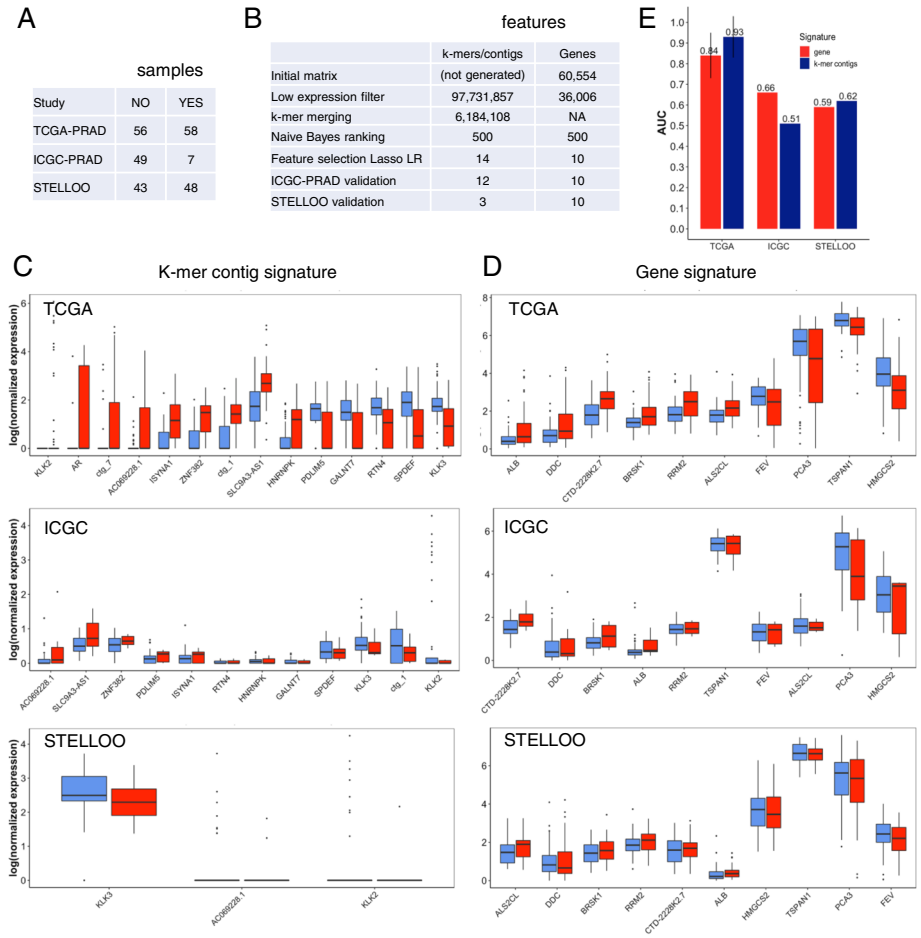


Fig. 5 Relapse signatures generation and analysis. **a** Characteristics of prostate tumor RNA-seq datasets. **b** Result of filtering procedure on the k-mer and gene matrices for relapse analysis. Expression of relapse signature elements in LR and HR samples in the TCGA-PRAD, ICGC-PRAD and STELLOO cohorts **c** k-mer contig signature; **d** Gene signature. **e** Signature performances for relapse prediction in the TCGA-PRAD, ICGC-PRAD and STELLOO cohorts

KLK2 contig is intronic and harbours a common SNP (rs62113074). The KLK3 contig is located in a distal part of the 3' UTR region present only in longer isoforms of KLK3. Its lower expression in relapsed patients was unexpected as low expression of PSA is usually associated to a lower risk. It is possible though that only this longer isoform is depleted in relapsing samples. The expression boxplot shows the KLK2 contig occurs only in a few outlier patients while the AR and KLK3 contigs are common (Fig. 5c). The contig matching SPDEF is a special variant of the 3' exon including two nonsynonymous SNPs. The SPDEF gene as a whole was highly expressed in both relapse and non-relapse samples but the contig expression was twice lower in average in relapse samples. Two contigs matched no known transcript: ctg_7 is a low complexity sequence of unknown origin and ctg_1 matches an intron of RPL9.

The contig matching lncRNA AC069228.1 also raised our attention since AC069228.1 is the only gene mapped by contigs in both relapse and risk signatures. The AC069228.1 lncRNA is antisense of PPFIA2, a protein tyrosine phosphatase that is itself an alleged urine biomarker of PCa [50]. The contigs from risk and relapse models match different regions of AC069228.1 (Figure S4). One is spliced, the other is a continuous 864 bp segment of a long exon. In both cases, a negative outcome (HR or relapse) is associated to a clearly higher expression of the contig, while the antisense gene PPFIA2 does not appear to follow the same trend (Figure S4).

Of note, the 10 genes in the reference-based signature were also clearly PCa-related: one was the major PCa biomarker PCA3 [51] and 5 others (DDC, RRM2, FEV, TSPAN1, HMGCS2) are involved in PCa etiology [52–56]. Therefore both gene-based and gene-free relapse signa-

tures were significant in terms of PCa related functions of their component genes or contigs.

Relapse signatures do not accurately classify independent cohorts

Contrary to the risk signatures, relapse signatures showed little overlap with each other and with published PCa signatures (Additional file 2). Only PCA3 and KLK2 were found in prior signatures [16, 57] and the only gene found shared between relapse and risk signatures in this study was AC069228.1. The poor overlap in this study was not unexpected as the discovery samples for risk and relapse information were quite disjointed and not always consistent: for instance only 25% of the high risk samples were labelled for relapse and 28% of these did not relapse. Conversely, 51% of non-relapse patients were labelled as HR. Therefore risk and relapse classifiers were trained to recognize quite different phenotypes.

As in the risk model, both reference-based and reference-free signatures had excellent cross-validation performance on the discovery set (AUC of 0.84 and 0.93 respectively, Fig. 5e). However this should again be considered as an overly optimistic estimation due to the experimental design. Indeed, performances of both relapse signatures on the ICGC-PRAD and Stelloo validation sets were much lower (AUC 0.51 to 0.66), bordering randomness and confirming overfitting of the trained signatures. Substituting the logistic Regression classifier by Random Forest, or Boosted Logistic Regression did not improve performance of either model (Table S3). The reference-based model performed slightly better over ICGC-PRAD, and the reference-free model was slightly better over the Stelloo dataset (Fig. 5e). Furthermore, several genes and contigs in the discovery signatures had inconsistent expression variations in the validation datasets (Fig. 5c and d, Additional file 3). Overall two genes from the reference-based signature (ALB and CTD-2228K2.7) and 5 contigs from the reference-free signature (KLK2, AC069228.1, PDLIM5, RTN4, ctg_1) changed logFC sign between the discovery and either validation cohort. This problem, which was not observed in risk models, underlines the poor replicability of the relapse signatures, whether or not reference-free.

Low replicability of the relapse model may be caused in part by weaknesses in validation datasets: the ICGC dataset had only 7 samples labelled for relapse (Fig. 5a) and the Stello dataset had very low coverage (Fig. 5a) which caused considerable loss when computing contig expression. Only three of the 14 signature contigs (AC069228.1, KLK2 and KLK3) could be quantified in the Stelloo dataset (Fig. 5b and c). Yet, we note that in spite of this loss the reference-free model still outperformed the reference-based model on this set (AUC of 0.62 vs. 0.59, Fig. 5e). Other limitations of the relapse model are addressed in the discussion.

Discussion

Properties of reference-free signatures

We evaluated here a method for building transcriptome classifiers that are totally reference-free, i.e. that do not require prior knowledge of genes or genome. The major interest of this approach lies in its ability to discover and incorporate in models previously unknown RNA biomarkers. Multiple examples exist of such disease-specific RNAs produced by genome alterations or deficient RNA processing and we hypothetized their inclusion in predictive models would be beneficial [18]. Applying a reference-free strategy to PCa outcome prediction, we obtained signatures made of short RNA contigs (median size 33 to 45 nt). These contigs are not full transcript models as can be produced by usual de novo assembly procedures. Instead, they often match SNPs or splice variants thus describing specific genetic or transcriptional events enriched in a patient group. Our strategy thus identifies RNA variations independently instead of lumping them into a full transcript model. Yet, the mapped genes were highly relevant to PCa etiology and included known cancer drivers LDLRAD4, GMNN, COL1A1, CD38, PTPRN2, GTSE1 and CAMK2N1 in the risk signature and KLK2, AR, KLK3, SPDEF in the relapse signature. Furthermore the risk signature comprised contigs matching two potential novel lncRNAs, located within LDLRAD4 and immediately downstream of EFNA2.

To our knowledge the only other software using a reference-free approach for inferring predictive signatures is Gecko [21]. Gecko uses machine learning (genetic algorithm) directly on the k-mer count matrix while we first reduce the matrix by grouping k-mers into contigs, before classification and machine learning. This enabled us to produce a signature composed of sequences larger than k, hence easier to interpret and quantify in an independent dataset.

Transferring a reference-free model to a new dataset is challenging. This requires that important features, such as SNPs, are precisely evaluated in the independent dataset. To this aim, we transferred signatures between datasets based on exact k-mer matches. As k-mer contents vary a lot between library preparation protocols, we expected this strategy to show poor sensitivity when discovery and validation datasets differed substantially. Indeed, transfer of signatures trained on the TCGA-PRAD dataset to the low coverage Stelloo dataset caused the loss of a majority of contigs. However, in this particular case, the remaining contigs were sufficient to maintain a prediction performance at the same level as that of the gene-based signature.

Performances and generalization issues

To compare the reference-free and reference-based strategies, a common evaluation framework was adopted. For both risk and relapse predictions, performances of the reference-free classifiers were on a par with that of reference-based classifiers. However while risk signatures showed satisfying reproducibility, relapse signatures performed poorly in independent datasets.

A possible reason for the low performance of relapse models is our grouping of patients in discrete relapse and non relapse categories as done in other studies [9, 13, 15, 16]. This allowed us to address relapse prediction using the same logistic regression method as for risk, however this meant valuable patient information was left unused. A more accurate prediction of relapse may be achieved using survival models [10, 12, 14, 57, 58]. Adaptation of survival analysis tools to large k-mer matrices require additional developments that are certainly worth considering in the future.

A more general concern with relapse analysis is related to difficulty of predicting an outcome occurring several years after a sample is biopsied and analyzed. There might just be too little information available in the training data to infer a reliable classifier, a problem that is independent of the use of contigs or genes. However, both gene-level and contig-level signatures were highly enriched in PCa driver genes, which suggests information about tumor progression was indeed present in the primary tumor biopsy. The key problem with relapse analysis was more likely related to sample heterogeneity. The diversity of relapse mechanisms was not properly represented in a training set of 100 patients as we used here. Patient stratification have been proposed to deal with sample heterogeneity in omics data [59, 60]. Adaptations of these solutions to large k-mers matrices will also be considered in the future.

Conclusion

For prediction of PCa risk and relapse, reference-free classifiers did not significantly outperform reference-based classifiers, however they incorporated a distinct set of RNA sequences including unannotated RNAs and novel variants of annotated RNAs. It is likely that with other diseases and datasets, novel biomarkers will be identified with an even greater impact on prediction performance. The reference-free approach will be of particular interest in problems where unknown RNAs are expected to play an important role, such as when studying rare diseases, poorly studied tissue types or when analysing dual human-pathogen RNA-seq samples. Our strategy also permits to infer efficient transcriptome classifiers in species lacking an accurate genome or transcriptome reference.

Abbreviations

AUC: Are under the ROC curve; BCR: Biochemical relapse; HR: High risk; lncRNA: Long non-coding RNA; LR: Low risk; MAC: Mean absolute contrast; PCa: Prostate cancer; PSA: Prostate-specific antigen; RNA-seq: RNA sequencing; TNM: Tumour node metastasis

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12885-021-08021-1.

Additional file 1: Contig sequences and mapping locations in the risk and relapse signatures.

Additional file 2: Published PCa risk and relapse signatures. Genes in common between published and this publication's signatures.

Additional file 3: Contents and expression characteristics of all signatures in the discovery and validation datasets.

Additional file 4: Supplementary figures and tables.

Acknowledgements

Not applicable.

Authors' contributions

HTNN and HX developed the software, HTNN generated and analyzed the results, VF analyzed the clinical data, YP, MG and DG designed the experiments, MG and DG wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work was funded in part by Agence Nationale de la Recherche grant ANR-18-CE45-0020 and by a 911 Scholarship Fund from the Vietnamese Government to HTN.

Availability of data and materials

The codes to reproduce the experiments are available on GitHub at: https://github.com/i2bc/PCa-gene-based_vs_gene-free.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

The authors declare that they have no competing interests.

Author details

[1] Institute for Integrative Biology of the Cell, UMR 9198, CEA, CNRS, Université Paris-Saclay, Gif-Sur-Yvette, France. [2] Institute of Biology, Université Paris Est Creteil, Creteil, Creteil, France. [3] LIX CNRS UMR 7161, Ecole Polytechnique, Institut Polytechnique de Paris, Palaiseau, France.

Received: 18 December 2020 Accepted: 9 March 2021
Published online: 12 April 2021

References

1. Perou CM, Sørlie T, Eisen MB, Van de Rijn M., Jeffrey SS, Rees CA, Pollack JR, Ross DT, Johnsen H, Akslen LA, Fluge O, Pergamenschikov A, Williams C, Zhu SX, Lønning PE, Børresen-Dale AL, Brown PO, Botstein D. Molecular portraits of human breast tumours. Nature. 2000;406(6797):747–52. https://doi.org/10.1038/35021093.
2. Singh D, Febbo PG, Ross K, Jackson DG, Manola J, Ladd C, Tamayo P, Renshaw AA, D'Amico AV, Richie JP, Lander ES, Loda M, Kantoff PW, Golub TR, Sellers WR. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell. 2002;1(2):203–9. https://doi.org/10.1016/S1535-6108(02)00030-2.

3. van 't Veer LJ, Dai H, Van de Vijver MJ, He YD, Hart AAM, Mao M, Peterse HL, Van der Kooy K, Marton MJ, Witteveen AT, Schreiber GJ, Kerkhoven RM, Roberts C, Linsley PS, Bernards R, Friend SH. Gene expression profiling predicts clinical outcome of breast cancer. Nature. 2002;415(6871):530–6. https://doi.org/10.1038/415530a.
4. D'Amico AV, Whittington R, Malkowicz SB, Schultz D, Blank K, Broderick GA, Tomaszewski JE, Renshaw AA, Kaplan I, Beard CJ, et al. Biochemical outcome after radical prostatectomy, external beam radiation therapy, or interstitial radiation therapy for clinically localized prostate cancer. Jama. 1998;280(11):969–74.
5. Bibikova M, Chudin E, Arsanjani A, Zhou L, Garcia EW, Modder J, Kostelec M, Barker D, Downs T, Fan JB, Wang-Rodriguez J. Expression signatures that correlated with Gleason score and relapse in prostate cancer. Genomics. 2007;89(6):666–72. https://doi.org/10.1016/j.ygeno.2007.02.005.
6. Penney KL, Sinnott JA, Fall K, Pawitan Y, Hoshida Y, Kraft P, Stark JR, Fiorentino M, Perner S, Finn S, et al. mrna expression signature of gleason grade predicts lethal prostate cancer. J Clin Oncol. 2011;29(17):2391.
7. Sinnott JA, Peisch SF, Tyekucheva S, Gerke T, Lis R, Rider JR, Fiorentino M, Stampfer MJ, Mucci LA, Loda M, et al. Prognostic utility of a new mRNA expression signature of gleason score. Clin Cancer Res. 2017;23(1):81–87.
8. Jhun MA, Geybels MS, Wright JL, Kolb S, April C, Bibikova M, Ostrander EA, Fan J-B, Feng Z, Stanford JL. Gene expression signature of gleason score is associated with prostate cancer outcomes in a radical prostatectomy cohort. Oncotarget. 2017;8(26):43035.
9. Latil A, Bièche I, Chêne L, Laurendeau I, Berthon P, Cussenot O, Vidaud M. Gene expression profiling in clinically localized prostate cancer: a four-gene expression model predicts clinical behavior. Clin Cancer Res. 2003;9(15):5477–85.
10. Long Q, Xu J, Osunkoya AO, Sannigrahi S, Johnson BA, Zhou W, Gillespie T, Park JY, Nam RK, Sugar L, Stanimirovic A, Seth AK, Petros JA, Moreno CS. Global transcriptome analysis of formalin-fixed prostate cancer specimens identifies biomarkers of disease recurrence. Cancer Res. 2014;74(12):3228–37. https://doi.org/10.1158/0008-5472.CAN-13-2699.
11. Ren S, Wei G-H, Liu D, Wang L, Hou Y, Zhu S, Peng L, Zhang Q, Cheng Y, Su H, et al. Whole-genome and transcriptome sequencing of prostate cancer identify new genetic alterations driving disease progression. Eur Urol. 2018;73(3):322–39.
12. Sinha A, Huang V, Livingstone J, Wang J, Fox NS, Kurganovs N, Ignatchenko V, Fritsch K, Donmez N, Heisler LE, et al. The proteogenomic landscape of curable prostate cancer. Cancer Cell. 2019;35(3):414–27.
13. Erho N, Crisan A, Vergara IA, Mitra AP, Ghadessi M, Buerki C, Bergstralh EJ, Kollmeyer T, Fink S, Haddad Z, et al. Discovery and validation of a prostate cancer genomic classifier that predicts early metastasis following radical prostatectomy. PloS ONE. 2013;8(6):66855.
14. Karnes RJ, Bergstralh EJ, Davicioni E, Ghadessi M, Buerki C, Mitra AP, Crisan A, Erho N, Vergara IA, Lam LL, Carlson R, Thompson DJS, Haddad Z, Zimmermann B, Sierocinski T, Triche TJ, Kollmeyer T, Ballman KV, Black PC, Klee GG, Jenkins RB. Validation of a genomic classifier that predicts metastasis following radical prostatectomy in an at risk patient population. J Urol. 2013;190(6):2047–53. https://doi.org/10.1016/j.juro.2013.06.017.
15. Klein EA, Yousefi K, Haddad Z, Choeurng V, Buerki C, Stephenson AJ, Li J, Kattan MW, Magi-Galluzzi C, Davicioni E. A genomic classifier improves prediction of metastatic disease within 5 years after surgery in node-negative high-risk prostate cancer patients managed by radical prostatectomy without adjuvant therapy. Eur Urol. 2015;67(4):778–86. https://doi.org/10.1016/j.eururo.2014.10.036.
16. Shahabi A, Lewinger JP, Ren J, April C, Sherrod AE, Hacia JG, Daneshmand S, Gill I, Pinski JK, Fan J-B, Stern MC. Novel gene expression signature predictive of clinical recurrence after radical prostatectomy in early stage prostate cancer patients. Prostate. 2016;76(14):1239–56. https://doi.org/10.1002/pros.23211.
17. Eggener SE, Rumble RB, Armstrong AJ, Morgan TM, Crispino T, Cornford P, Van der Kwast T, Grignon DJ, Rai AJ, Agarwal N, Klein EA, Den RB, Beltran H. Molecular biomarkers in localized prostate cancer: ASCO guideline. J Clin Oncol. 2020;38(13):1474–94. https://doi.org/10.1200/JCO.19.02768.
18. Morillon A, Gautheret D. Bridging the gap between reference and real transcriptomes. Genome Biol. 2019;20(1):1–7.
19. Audoux J, Philippe N, Chikhi R, Salson M, Gallopin M, Gabriel M, Le Coz J, Drouineau E, Commes T, Gautheret D. DE-kupl: exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. Genome Biol. 2017;18(1):243. https://doi.org/10.1186/s13059-017-1372-2.
20. Pinskaya M, Saci Z, Gallopin M, Gabriel M, Nguyen HTN, Firlej V, Descrimes M, Rapinat A, Gentien D, De La Taille A, Londoño-Vallejo A, Allory Y, Gautheret D, Morillon A. Reference-free transcriptome exploration reveals novel RNAs for prostate cancer diagnosis. Life Sci Alliance. 2019;2(6):1–12. https://doi.org/10.26508/lsa.201900449.
21. Thomas A, Barriere S, Broseus L, Brooke J, Lorenzi C, Villemin J.-p., Beurier G, Sabatier R, Reynes C, Mancheron A, Ritchie W. GECKO is a genetic algorithm to classify and explore high throughput sequencing data. Commun Biol. 2019;2(1):222. https://doi.org/10.1038/s42003-019-0456-9.
22. Michiels S, Koscielny S, Hill C. Prediction of cancer outcome with microarrays: a multiple random validation strategy. Lancet. 2005;365(9458):488–92. https://doi.org/10.1016/S0140-6736(05)17866-0.
23. Ein-Dor L, Zuk O, Domany E. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. Proc Natl Acad Sci. 2006;103(15):5923–8. https://doi.org/10.1073/pnas.0601231103.
24. Michiels S, Koscielny S, Hill C. Interpretation of microarray data in cancer. Br J Cancer. 2007;96(8):1155–8. https://doi.org/10.1038/sj.bjc.6603673.
25. Venet D, Dumont JE, Detours V. Most random gene expression signatures are significantly associated with breast cancer outcome. PLoS Comput Biol. 2011;7(10):1002240. https://doi.org/10.1371/journal.pcbi.1002240.
26. Abeshouse A, Ahn J, Akbani R, Ally A, Amin S, Andry CD, Annala M, Aprikian A, Armenia J, Arora A, et al. The molecular taxonomy of primary prostate cancer. Cell. 2015;163(4):1011–25.
27. Fraser M, Sabelnykova VY, Yamaguchi TN, Heisler LE, Livingstone J, Huang V, Shiah Y-J, Yousif F, Lin X, Masella AP, et al. Genomic hallmarks of localized, non-indolent prostate cancer. Nature. 2017;541(7637):359–64.
28. Stelloo S, Nevedomskaya E, Kim Y, Schuurman K, Valle-Encinas E, Lobo J, Krijgsman O, Peeper DS, Chang SL, Feng FY-C, et al. Integrative epigenetic taxonomy of primary prostate cancer. Nat Commun. 2018;9(1):1–12.
29. Liu J, Lichtenberg T, Hoadley KA, Poisson LM, Lazar AJ, Cherniack AD, Kovatich AJ, Benz CC, Levine DA, Lee AV, et al. An integrated tcga pan-cancer clinical data resource to drive high-quality survival outcome analytics. Cell. 2018;173(2):400–16.
30. Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. Proc Natl Acad Sci. 2002;99(10):6562–6. https://doi.org/10.1073/pnas.102102699.
31. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics. 2011;27(6):764–70.
32. Bray NL, Pimentel H, Melsted P, Pachter L. Near-optimal probabilistic rna-seq quantification. Nat Biotechnol. 2016;34(5):525–7.
33. Curtin RR, Edel M, Lozhnikov M, Mentekidis Y, Ghaisas S, Zhang S. mlpack 3: a fast, flexible machine learning library. J Open Source Softw. 2018;3:726. https://doi.org/10.21105/joss.00726.
34. Friedman J, Hastie T, Tibshirani R. Regularization paths for generalized linear models via coordinate descent. J Stat Softw. 2010;33(1):1–22.
35. Meinshausen N, Bühlmann P. Stability selection. J R Stat Soc Ser B Stat Methodol. 2010;72(4):417–73. https://doi.org/10.1111/j.1467-9868.2010.00740.x.
36. Kuhn M. Building predictive models in r using the caret package. J Stat Softw Artic. 2008;28(5):1–26. https://doi.org/10.18637/jss.v028.i05.
37. Menardi G, Torelli N. Training and assessing classification rules with imbalanced data. Data Min Knowl Disc. 2014;28(1):92–122.
38. Saito T, Rehmsmeier M. The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. PLOS ONE. 2015;10(3):0118432. https://doi.org/10.1371/journal.pone.0118432.
39. Rainer J. EnsDb.Hsapiens.v79: Ensembl based annotation package. R package version 2.99.0. 2017.
40. Liu X, Grogan TR, Hieronymus H, Hashimoto T, Mottahedeh J, Cheng D, Zhang L, Huang K, Stoyanova T, Park JW, et al. Low cd38 identifies progenitor-like inflammation-associated luminal cells that can initiate human prostate cancer and predict poor outcome. Cell Rep. 2016;17(10):2596–606.
41. Wang T, Liu Z, Guo S, Wu L, Li M, Yang J, Chen R, Xu H, Cai S, Chen H, et al. The tumor suppressive role of camk2n1 in castration-resistant prostate cancer. Oncotarget. 2014;5(11):3611.

42. Liu J, Shen J-X, Wu H-T, Li X-L, Wen X-F, Du C-W, Zhang G-J. Collagen 1a1 (col1a1) promotes metastasis of breast cancer and is a potential therapeutic target. Discov Med. 2018;25(139):211–23.
43. Wu X, Wang H, Lian Y, Chen L, Gu L, Wang J, Huang Y, Deng M, Gao Z, Huang Y. Gtse1 promotes cell migration and invasion by regulating emt in hepatocellular carcinoma and is associated with poor prognosis. Sci Rep. 2017;7(1):1–12.
44. Chen C-L, Mahalingam D, Osmulski P, Jadhav RR, Wang C-M, Leach RJ, Chang T-C, Weitman SD, Kumar AP, Sun L, et al. Single-cell analysis of circulating tumor cells identifies cumulative expression patterns of emt-related genes in metastatic prostate cancer. Prostate. 2013;73(8): 813–26.
45. Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP. Integrative genomics viewer. Nat Biotechnol. 2011;29(1):24–6. https://doi.org/10.1038/nbt.1754.
46. Xie W, Xiao H, Luo J, Zhao L, Jin F, Ma J, Li J, Xiong K, Chen C, Wang G. Identification of low-density lipoprotein receptor class a domain containing 4 (ldlrad4) as a prognostic indicator in primary gastrointestinal stromal tumors. Curr Probl Cancer. 2020;44(6):100593.
47. Mo S, Zhang L, Dai W, Han L, Wang R, Xiang W, Wang Z, Li Q, Yu J, Yuan J, et al. Antisense lncrna ldlrad4-as1 promotes metastasis by decreasing the expression of ldlrad4 and predicts a poor prognosis in colorectal cancer. Cell Death Dis. 2020;11(2):1–16.
48. Chen CD, Welsbie DS, Tran C, Baek SH, Chen R, Vessella R, Rosenfeld MG, Sawyers CL. Molecular determinants of resistance to antiandrogen therapy. Nat Med. 2004;10(1):33–39.
49. Chen W-Y, Tsai Y-C, Yeh H-L, Suau F, Jiang K-C, Shao A-N, Huang J, Liu Y-N. Loss of spdef and gain of tgfbi activity after androgen deprivation therapy promote emt and bone metastasis of prostate cancer. Sci Signal. 2017;10(492):6826.
50. Leyten GH, Hessels D, Smit FP, Jannink SA, de Jong H, Melchers WJ. Identification of a candidate gene panel for the early diagnosis of prostate cancer. Clin Cancer Res. 2015;21(13):3061–70.
51. Bussemakers MJG, van Bokhoven A, Verhaegh GW, Smit FP, Karthaus HFM, Schalken JA, Debruyne FMJ, Ru N, Isaacs WB. DD3: a new prostate-specific gene, highly overexpressed in prostate cancer. Cancer Res. 1999;59(23):5975–9.
52. Koutalellis G, Stravodimos K, Avgeris M, Mavridis K, Scorilas A, Lazaris A, Constantinides C. L-dopa decarboxylase (ddc) gene expression is related to outcome in patients with prostate cancer. BJU Int. 2012;110(6b):267–73.
53. Mazzu YZ, Armenia J, Chakraborty G, Yoshikawa Y, Si'Ana AC, Nandakumar S, Gerke TA, Pomerantz MM, Qiu X, Zhao H, et al. A novel mechanism driving poor-prognosis prostate cancer: overexpression of the dna repair gene, ribonucleotide reductase small subunit m2 (rrm2). Clin Cancer Res. 2019;25(14):4480–92.
54. Zhong W-D, Liang Y-X, Liang Y-K, Zhuo Y-J, Ye J-H, Zhu X-J, Cai Z-D, Lin Z-Y, Zhu J-G, Wu S-L, et al. Tumor suppressor role and clinical implication of the fifth ewing variant (fev) gene, an ets family gene, in prostate cancer. In: Prostate Cancer; 2019. SSRN: https://ssrn.com/abstract=3372417.
55. Munkley J, McClurg UL, Livermore KE, Ehrmann I, Knight B, Mccullagh P, Mcgrath J, Crundwell M, Harries LW, Leung HY, et al. The cancer-associated cell migration protein tspan1 is under control of androgens and its upregulation increases prostate cancer cell migration. Sci Rep. 2017;7(1):1–11.
56. Wan S, Xi M, Zhao H-B, Hua W, Liu Y-L, Zhou Y-L, Zhuo Y-J, Liu Z-Z, Cai Z-D, Wan Y-P, et al. Hmgcs2 functions as a tumor suppressor and has a prognostic impact in prostate cancer. Pathol Res Pract. 2019;215(8): 152464.
57. Klein EA, Cooperberg MR, Magi-Galluzzi C, Simko JP, Falzarano SM, Maddala T, Chan JM, Li J, Cowan JE, Tsiatis AC, Cherbavaz DB, Pelham RJ, Tenggara-Hunter I, Baehner FL, Knezevic D, Febbo PG, Shak S, Kattan MW, Lee M, Carroll PR. A 17-gene assay to predict prostate cancer aggressiveness in the context of gleason grade heterogeneity, tumor multifocality, and biopsy undersampling. Eur Urol. 2014;66(3):550–60. https://doi.org/10.1016/j.eururo.2014.05.004.
58. Witten DM, Tibshirani R. Survival analysis with high-dimensional covariates. Stat Methods Med Res. 2010;19(1):29–51. https://doi.org/10.1177/0962280209105024.
59. de Ronde JJ, Rigaill G, Rottenberg S, Rodenhuis S, Wessels LFA. Identifying subgroup markers in heterogeneous populations. Nucleic Acids Res. 2013;41(21):200. https://doi.org/10.1093/nar/gkt845.
60. Campos-Laborie FJ, Risueño A, Ortiz-Estévez M, Rosón-Burgo B, Droste C, Fontanillo C, Loos R, Sánchez-Santos JM, Trotter MW, De Las Rivas J. DECO: decompose heterogeneous population cohorts for patient stratification and discovery of sample biomarkers using omic data profiling. Bioinformatics. 2019;35(19):3651–62. https://doi.org/10.1093/bioinformatics/btz148. https://academic.oup.com/bioinformatics/article-pdf/35/19/3651/30061524/btz148.pdf.

Publisher's Note

Chapter 5

Analyzing Differential Tumor vs. Normal *k*-mers across Independent Cohorts

5.1 Motivation

DE-kupl is our lab's previous software for k-mer count signal analysis. It aims to retrieve k-mer contigs that are differentially expressed between two biological conditions [Audoux et al., 2017]. We analyzed here the replicability of *DE-kupl*'s findings across independent cohorts on three publicly available data sets: two lung adenocarcinoma data sets [Seo et al., 2012, Collisson et al., 2014] and a prostate adenocarcinoma data set [Abeshouse et al., 2015]. Our goal was to identify in each set differential k-mer contigs between normal and tumor tissues and to compare the findings from the three data sets. We expected that comparison of the two lung adenocarcinoma studies would yield a set of shared events of high biological value. The final results largely met this expectation.

5.2 My Contribution

I participated in the algorithm design for shared event identification, by proposing seeking cliques in a graph structure for shared k-mer contig retrieval (Figure S1). I proposed PCA assessing of the classification value of differentially expressed genes (the last panel in Figure 7). I also participated in responding reviewers comments.

5.3 Article

RESEARCH

The contribution of uncharted RNA sequences to tumor identity in lung adenocarcinoma

Yunfeng Wang[1,3], Haoliang Xue[1], Marine Aglave[2], Antoine Lainé[1], Mélina Gallopin[1] and Daniel Gautheret[1,2*]

*Correspondence: daniel.gautheret@universite-paris-saclay.fr
[1]Institute for Integrative Biology of the Cell (I2BC), Université Paris-Saclay, CNRS, CEA, 1 avenue de la Terrasse, 91190, Gif-sur-Yvette, France
[2]Gustave Roussy, 114 rue Edouard Vaillant, 94800, Villejuif, France
Full list of author information is available at the end of the article

Abstract

Background: Transcriptome analysis of cancer tissues has been instrumental in defining tumor subtypes, diagnostic signatures and cancer regulatory networks. Cancer transcriptomes are still predominantly analyzed at the level of gene expression. Few studies have addressed transcript-level variations, and most of these only looked at splice variants. Previously we introduced a k-mer based, reference-free method, DE-kupl, that performs differential analysis of RNA-seq data at the k-mer level, which enables distinguishing RNAs differing by a single nucleotide. Here we evaluate the significance of differential events discovered by this method in two independent lung adenocarcinoma RNA-seq datasets (N=583 and N=154).

Results: Focusing on differential events in a tumor vs normal setting, we found events in endogenous repeats, alternative splicing and polyadenylation sites, long non-coding RNAs, retained introns and unmapped RNAs. Replicability was highly significant for most event classes (assessed by comparing to events shared between unrelated tumors). Overall about 160,000 differential k-mer contigs were shared between datasets, including a large set of sequences from hypervariable genes such as immunoglobulins, $SFTP$ and mucin genes. Most interestingly, we identified a set of novel tumor-specific long non-coding RNAs in intergenic and intronic regions. We found that expressed endogenous transposons defined two major groups of patients (high/low repeat expression) with distinct clinical characteristic. A number of repeats, intronic RNAs and lincRNA achieved strong patient stratification in univariate or multivariate survival models. Finally, using antigen presentation prediction, we identified 55 contigs predicted to produce recurrent tumor-specific antigens.

Conclusions: K-mer based RNA-seq analysis enables description of cancer transcriptomes at nucleotide precision, independently of prior transcript annotation. Application to lung cancer data uncovered events stemming from a wide variety of transcriptional and postranscriptional mechanisms. Among those events, a significant subset was replicable between cohorts, thus constituting novel RNA hallmarks of cancer. The code is available at: https://github.com/Transipedia/dekupl-lung-cancer-inter-cohort.

Keywords: k-mers; contigs; repeats; LUAD; mapping-free; replicability

Background

Over a period of 20 years, cancer transcriptomics has transformed our understanding of tumor biology and led to improved tools for tumor typing and outcome prediction [1, 2]. While first generation transcriptome analysis was based on DNA microarrays with a focus on protein-coding genes, the current generation relies on RNA-seq

data, which promises to deliver a more comprehensive view of gene expression. However, in spite of its potential for transcript discovery, cancer RNA-seq data is still utilized mostly to quantify the expression of annotated genes listed in a reference transcriptome. This ignores a wide array of mRNA isoforms, non-coding RNAs, endogenous retroelements and transcripts from exogenous viruses and bacteria [3]. The quantity of information left unexploited in non-canonical transcripts remains unknown. A number of studies have started to address this question using publicly available cancer RNA-seq data, focusing on specific transcript classes such as splice variants [4, 5], lncRNAs [6], snoRNAs [7], repeats [8], bacterial RNA [9], or viral RNA [10]. Other neglected sources of RNA diversity are the so-called blacklisted regions of the genome that are too variable or repeated to be properly analyzed by conventional approaches [11]. To our knowledge, no attempt has been made to extract and evaluate at once all this non-standard RNA information from tumor RNA-seq data. We think this approach could be particularly valuable in cancer since every individual tumor harbors a unique transcriptome that departs from that of normal tissues in multiple, unpredictable ways.

Previously we introduced a computational method, DE-kupl [12], that performs differential analysis of RNA-seq data at the k-mer level. As this method is reference-free and mapping-free, it identifies any novel RNA or RNA isoform present in the data at nucleotide resolution, including poorly mapped transcripts such as RNAs from repeats and chimeric RNAs. Here we set ourselves to evaluate all non-reference events discovered by DE-kupl in a comparison of normal vs. tumor samples using lung adenocarcinoma as a test case. To mitigate false positives events inherent to any gene expression profiling [13, 14], we focused on events that were replicated in two independent datasets. This required the development of a dedicated protocol to identify shared events in unmapped RNA sequences. Results revealed a collection of novel tumor-specific unannotated lincRNAs, intron retentions, and splicing events. Most strikingly, a collection of endogenous retroelements form a major class of tumor defining transcripts and constitute potent survival signatures. We also identified a subset of events with no expression in normal tissues which could be potential neoantigens sources. We would like to suggest DE-kupl as a promising, comprehensive approach to cancer transcript profiling.

Methods

Datasets

LUAD-TCGA: 582 lung RNA-seq samples from the LUAD-TCGA project were downloaded from the dbgap repository with permission, including 524 lung adenocarcinoma (LUAD) tissues and 58 adjacent normal tissues [15]. LUAD-SEO: The LUAD RNA-seq dataset of Seo et al. [16] was downloaded from the SRA database (accession: ERP001058). This dataset contains fastq files of 87 LUAD and 77 adjacent normal tissues. Only the 77 paired normal and tumor samples were analyzed. PRAD-TCGA: For control, 557 PRAD-TCGA prostate RNA-seq datasets were downloaded from dbgap with permission, including 505 prostate adenocarcinoma (PRAD) and 52 normal controls [17]. Bam format files from the TCGA datasets were converted to fastq format using Picard tools version 2.18.16 (http://broadinstitute.github.io/picard).

DE-kupl pipeline

DE-kupl (version 5.3.0) was applied to the three datasets with the same parameters: in the filtering steps, k-mers with abundance fewer than 5 (min_recurrence_abundance) and present in no more than 10 samples (min_recurrence) were ruled out. In order to focus on non-canonical transcripts, we masked all k-mers pertaining to the main transcript of each Gencode gene as in [12]. Normalization factors for k-mer counts were computed by DE-kupl as medians of the ratios of sample counts by counts of a pseudo-reference obtained by taking the geometric mean of each k-mer across all samples. Herein we will use these counts as a proxy to represent the expression of the corresponding RNA fragment.

For differential expression analysis, the version of DESeq2 available at the time of the experiment was too slow for dealing with hundreds of samples and we found the faster "T-test" option to lack sensibility. Hence we used instead Limma [18], adapted to millions of k-mers using a chunk-based strategy (suppl. methods). This was found to perform 10 times faster than DESeq2. The performances of DESeq2, Limma and T-test for differential expression evaluation have been evaluated before [19]. Evaluations of k-mer counts were log-transformed and Limma was used to calculate log fold-changes and P-values. Retention thresholds for log2 fold changes and P-values were 1 and 0.05, respectively. All k-mers passing the filtering process above were merged into contigs and the contig table was saved as output. GC-contents in "up" and "down" contigs in the PRADtcga dataset were verified and did not present any bias (Additional file2: Table S1). High-quality contigs ("top contigs") were contigs with counts>10 in at least 15% of the smaller class (Normal or Tumor).

Gene-level expression was measured using Kallisto v0.43.02 [20] and Gencode v31 transcripts, followed by summing TPM values of transcripts from the same gene. Gene-level differential expression analysis was performed using Limma and the same normalization procedure as above. Downstream analyses were conducted using R version 3.5.2. Heatmaps were drawn using the ComplexHeatmap package (version 2.4.3) [21].

Shared event identification

Contigs from distinct DE-kupl analyses were decomposed into their constituent k-mer lists and a graph was constructed using the NetworkX Python package (version 2.3) [22], with k-mers as nodes and shared k-mers as edges. Contigs corresponding to the same local event are expected to form a fully connected subgraph or clique (Additional file 1: Fig. S1). We thus extracted all cliques to identify shared contigs. Hereafter we use the $\cap$ operator to represent contigs shared between two datasets.

Contig annotation

A uniform annotation procedure was applied to contigs from each independent analysis (LUADtcga, LUADseo, PRADtcga) and to shared contigs (LUADtcga $\cap$ LUADseo and LUADtcga $\cap$ PRADtcga). Initially, differential contigs were mapped and annotated with DE-kupl annotation (https://github.com/Transipedia/dekupl). Briefly, DE-kupl annotation maps contigs to the human genome and reports intronic, exonic or intergenic status, CIGAR string, IDs of mapped or neighboring

genes, differential usage status. A new repeat annotation field ("rep_type") was added based on Blast [23] alignments of contigs to the DFAM repeat database [24] (see Suppl. Methods). The results of DEkupl-annot were then loaded into R and submitted to further filtering and annotation. Firstly, a count filter was applied to retain only contigs with a count of 10 in at least 15% of the smaller class (Normal or Tumor). Contigs meeting this criterion were classified into event classes comprising SNV, intronic, splices, split, lincRNA, polyA, repeat and unmapped, as described in Additional file2: Table S3. Classes were non exclusive, meaning that a contig can belong to several classes. Since the TCGA datasets are unstranded, antisense events were not called. Differential usage (i.e. the relative change in expression of a local event relative to the expression of the host gene) was evaluated for each event mapped to an annotated gene. Intergenic contigs were further aligned with Blast against MiTranscriptome V2 [6] retrieved at http://mitranscriptome.org/ and converted to fasta using gffread (https://github.com/gpertea/gffread). Finally, we defined a new category called "neoRNAs", which includes contigs that are expressed in tumor tissues but silent in normal tissues.

Functional enrichment of intronic events

Candidate intronic events were identified based on the DE-kupl differential usage P-value (computed by comparing the expression or the contig with that of the host gene). Gene Ontology biological process enrichment of host genes was assessed using the clusterProfiler R package (version 3.16.0) [25].

Sample clustering based on repeats

We used the K-means algorithm [26] to cluster LUAD patients into two main subgroups based on the expression of contigs matching AluSx, L1P1_orf2 and L1P3_orf2 repeats. Clusters were then analyzed for enrichment in clinical features, immune infiltration, tumor mutational burden and copy number variants. LUAD driver genes were retrieved from the COSMIC Cancer Gene Census (CGC) list [27]. Oncoplots were drawn using the maftools R package (version 2.4.10) [28]. The estimated tumor mutational burden (TMB) for each patient was computed using the total number of non-synonymous mutations from the Mutation Annotation Format (MAF) file, divided by the estimated size of the whole exome. Copy number variation (CNV) data was downloaded by the TCGAbiolinks R package (version 2.16.3) [29], which provides a mean copy number estimate of segments covering the whole genome (inferred from Affy SNP 6.0). The ratio of gain and loss for each patient was estimated by the fraction of segments indicating CNVs. Heatmap representations were produced with ComplexHeatmap [21].

Correlation with immune infiltration

Immune infiltration analysis was performed on the LUADseo dataset. Relative proportions of infiltrating immune cells were determined using CIBERSORT [30]. Relationships between immune cell types and shared contigs (grouped by annotation category) were computed as the Spearman correlation between the contig expression and the relative proportion of the cell type in all samples. Any contig with an absolute Spearman correlation coefficient above 0.5 with at least one immune cell type was retained.

Neoantigen prediction

For prediction of recurrent tumor-specific antigen, we selected contigs absent in all normal tissues but present in at least 15% of tumor tissues. We translated contig sequences using EMBOSS transeq over 6 frames [31]. Sequences with stop codons were ruled out and candidate peptides were submitted to netMHCpan 4.0 [32] to predict binding affinity to MHC-class-I molecules. Peptide–MHC Class I interactions with strong binding levels (by default 0.5%) were reported.

Survival analysis based on event classes

Since the LUADseo dataset does not include survival information, we only performed the survival analysis on the LUADtcga dataset. Overall survival time and status was downloaded from the GDC portal (https://portal.gdc.cancer.gov/projects/TCGA-LUAD). We performed both univariate Cox regression and multivariate Cox regression on each event class to assess the prognosis value of the differential events. Survival analysis was performed using the survival (version 3.2.3) and survminer (version 0.4.7) R packages [33, 34]. Hazard ratios (HR) and P-values were calculated for each contig. Contigs with HR>1 and P-value<0.05 were considered as potential risk factors. For multivariate Cox regression, contigs were initially selected by cox-lasso regression using the glmnet R package (version 4.0.2) [35] applied independently to each contig class. The multivariate model was then constructed using selected contigs. Patients were divided into high and low-risk groups based on the median value of all risk scores for representation in Kaplan–Meier (KM) curves [36].

Unsupervised clustering analysis

We applied Principal Component Analysis (PCA) and hierarchical clustering to each event class. PCA analysis was performed with the factoextra R package (version 1.0.7) [34]. Heatmap views were obtained using ComplexHeatmap [21].

Sequence alignment views

We created "metabam" alignment files for tumor and normal tissues from each cohort. To this aim, we randomly sampled 1M reads from each fastq file of each subcohort using seqtk (https://github.com/lh3/seqtk) and aligned the aggregated reads to the genome (GRCh38) using STAR (version 2.7.0f) [37] with default parameters. BAM files were visualized using Integrative Genomics Viewer (IGV 2.6.2) [38].

Results

Gene-level *vs.* contig-level differential events

We performed tumor *vs.* normal differential expression (DE) analysis on two independent Lung adenocarcinoma RNA-seq datasets from TCGA (LUADtcga) and Seo et al. (LUADseo) and on a prostate adenocarcinoma dataset from TCGA (PRADtcga) as a control. Each dataset was submitted to a conventional, gene-level, differential expression analysis and a k-mer level differential expression analysis where all k-mers from annotated genes were first removed and the resulting differential k-mers were assembled into contigs (Fig 1A). For simplification, we shall hereafter use term "expression" when referring to either gene expression or contig

k-mer counts. While the number of DE genes in the three comparisons ranged from 6,000 to 9,000, the number of DE k-mers was about a thousand times larger (2 to 12 millions). Assembly of k-mers into contigs reduced this number to about 400,000 DE contigs in each analysis (Fig 1B).

We next compared the DE genes and contigs discovered in independent datasets to identify shared DE events. While this process is trivial for genes, it is not for contigs, since contigs found in each dataset have no standard identifier that could be used to relate them. We thus implemented a graph analysis procedure that identified shared contigs based on their common k-mers (Fig 1A, Additional file 1: Fig. S1). A final annotation step assigned contigs to non exclusive categories based on their mapping characteristics or expression (repeats, lincRNAs, splice variant, polyadenylation variants, split RNAs, tumor-specific RNAs) as described in Additional file2: Table S3 and Methods. The numbers of shared elements slightly differ between LUADtcga and LUADseo because a minority of elements are in a 2-to-1 or 1-to-2 relationship in the contig graph. If not otherwise specified, numbers of elements are given for the LUADtcga cohort.

Overall 160,610 differential contigs were shared between the two LUAD analyses (Fig 1C). Over these, 120,822 contigs were considered of sufficient quality based on counts and occurrence in a minimal number of samples (see Methods). 83% of shared contigs were overexpressed in tumors vs. only 17% underexpressed (Fig 1C).

Event replicability

The replicability of differential events was generally lower for k-mer or contigs than for genes. Fig 1D shows the number of differential expression genes and contigs shared by the two independent LUAD analyzes, with contigs binned by annotation class. About 41% of differential expression genes (3032 genes) were shared by the two LUAD analyses, compared to an average of 14% for differential expression contigs (repeats: 3.7%, unmapped RNAs: 10%, alternative polyAs: 13%, lincRNAs: 14%, alternative splices: 20%, retained introns: 20%). Although the ratio of shared events was relatively low for k-mer analysis, it was considerably higher than when comparing two unrelated pathologies (LUADtcga $\cap$ PRADtcga, Fig 1D), and this applied to all event classes except repeats. This indicates that, although k-mer based differential expression events are noisy, a significant subset is replicable in independent studies. Furthermore, we observed a strong correlation between the fold-change value of differential expression contigs and the likelihood to be shared between cohorts (Additional file 1: Fig. S2), demonstrating the non-randomness of high scoring, non-reference events.

DE contig localization, hypervariable genes

The majority of shared contigs are genic (83%), 45% are intronic and 32% carry SNVs or indels (Fig 2A). These characteristics are induced by the initial filter that removed all k-mers matching reference transcripts, retaining any intronic or SNV-carrying k-mer. Therefore a large number of SNV and intronic contigs are just "passenger" events of DE genes. We confirmed this by analyzing the correlation between numbers of DE contigs and host gene expression. We found a significant correlation (Pearson CC=0.45), but this correlation was reduced (Pearson CC=0.28) in shared

DE contigs, indicating shared contigs contain fewer passenger events (Additional file 3).

More than 400 genes were matched by 35 or more contigs. We classified these genes into two categories: for 296 genes, most contigs matched introns and were up-regulated in tumors (Fig 2A, B, Additional file 2: Table S5). These mostly correspond to the aforementioned passenger events. The second category is composed of 107 genes we refer to as "hypervariable" as they tend to yield a large number of contigs carrying SNVs, indels and larger rearrangements (Fig 2A, C, Additional file 2: Table S5). The largest sets of hypervariable genes are IGK, IGL and IGH immunoglobulin genes. This is not surprising given immunoglobulins (i) are highly variable due to V(D)J segment recombination and (ii) are expressed by plasma B-cells which are abundant in the tumor immune infiltrate [39], hence these genes are seen as up-regulated in tumors. Interestingly, those IG sequence variants are found expressed in different patients and across the two cohorts, suggesting our approach can be used to profile immunoglobulin repertoires, as performed recently with other RNA-seq datasets [40]. To evaluate the accuracy of DE-kupl contigs assembled from IG genes, we selected all contigs mapped to one arbitrary IG gene (IGHV: 100 contigs) and aligned them to IGHV contigs from the IMGT database [41]. Ninety out of 100 contigs had significant matches in the corresponding IMGT category extending over 90% of the contig length (Additional file2: Table S6).

Other hypervariable loci were found in surfactant protein ($SFTP$) and Mucin genes which are known to harbor a high level of polymorphism [42, 43]. We observed polymorphism not only in the form of SNPs, but also in the form of splicing variations. Five $SFTP$ genes alone combine over 9000 SNVs and 800 splice sites contigs, while 12 Mucin genes harbour 1324 contigs including 42 splice variants (Additional file 1: Fig. S3A-B, Additional file 2: Table S5). While $SFTP$ contigs were all underexpressed in tumors, Mucin contigs were mostly overexpressed (Additional file 2: Table S5). Mucins are immunogenic [43] and are important biomarkers for prognosis [44] and drug resistance [45]. The existence of recurrent mucin variants overexpressed in tumors may be relevant for these therapeutic and biomarker developments. We also observed hypervariability in $CEACAM5$ and $KR19$, two other prognostic biomarkers and/or immunotherapy targets [46, 47] (Additional file 1: Fig. S3C, Additional file 2: Table S5).

Intron retention and other intronic events

We found intronic contigs with differential usage (DU) in 313 host genes, 290 (93%) of which were up-regulated in tumors (Additional file 2: Table S4). 70% of the host genes were also up-regulated, thus the apparent overexpression of these intronic sequences may have been confounded by overexpression of host genes. However, 30% of host genes were not overexpressed, and in 103 cases, intron and host gene expressions varied in opposite directions (93 introns up and 10 introns down). Our annotation pipeline did not differentiate intron retentions (as shown for example in Additional file 1: Fig. S4A) from transcription units occurring within introns (example in Additional file 1: Fig. S4B). We observed intron retention events in lung cancer drivers $EGFR$ and MET (Additional file 1: Fig. S4C and Additional file 1: Fig. S4D). In $EGFR$, the retained intron was located between exons 18 and

19, just upstream of the principal oncogenic *EGFR* mutations located in exons 19-21. Intron retention before exon 19 would likely produce a truncated form of *EGFR* compatible with oncogenic activation.

Additional file 1: Fig. S5A shows the 20 intronic events with the most significant differential usage P-values. All show opposite directions of intron and gene expression. Gene Ontology enrichment analysis indicates host genes are enriched for inflammation and immune response pathways involving neutrophil and T cells (additional file 1: Fig. S5B), suggesting these events may come from regulations in the tumor microenvironment rather than in the tumor itself.

Novel lincRNAs

Contigs that do not map any Gencode annotated gene are of particular interest as they potentially represent novel lincRNA biomarkers of lung tumors. Overall we identified shared DE contigs in 885 intergenic regions, which we labelled as lincRNAs. As genic regions already included annotated lncRNAs and pseudogenes from Gencode, the actual number of DE contigs in lncRNAs and pseudogenes was much higher (N=2892) but we focus here on unannotated regions. lincRNA contigs were mostly overexpressed in tumors (83% of contigs) and often contained a known repeat element (73% of contigs). Their average length was 137 nt, however actual transcription units were generally longer as most units were composed of multiple contigs, as shown in examples in Additional file 1: Fig. S6. Most intergenic contigs (793 out of 823) were already annotated in the independent Mitranscriptome lncRNA database [6], which was expected since this database was also produced from TCGA RNA-seq data. Less than one third of the flanking genes of intergenic contigs were differentially expressed, indicating that novel lincRNA expression was most often independent from that of flanking genes.

Expressed repeats delineate patient subgroups with distinct clinical properties

The dominant model for endogenous retroelements (EREs) expression is that EREs are mainly expressed in germline and embryonic stem cells while they are repressed in differentiated somatic cells. However recent studies have shown expression of EREs in somatic cells is more common and heterogeneous than expected[48]. Repeat-containing reads are difficult to analyze by RNA-seq standard pipelines due to ambiguity in the alignment process. We thus questioned whether our alignment-free procedure could help reveal these events. From the initial set of 50572 contigs annotated as repeats (Fig 1C), we selected a high quality subset of 10341 contigs over 60 bp in size and with expression above a set threshold (see Methods). Of these, 87.7% were overexpressed in tumors (Additional file 2: Table S4).

Fig 3A shows the distribution of contigs per repeat family. Most repeats correspond to Line 1 and Alu family sequences. The most frequent repeat overall is L1P1, a Line 1 of the L1Hs family which is the only retrotransposition-competent EREs in the human genome [49]. L1P1/L1Hs elements, as well as human endogenous retrovirus (HERV), were almost exclusively over-expressed in tumors, suggesting tumor-specific activation of these elements. In contrast, Alu elements, which are often expressed as part of protein coding genes, were either over- or under-expressed in

tumors. Fig 3A shows the top 20 repeat types that contribute more contigs. Fig 3B-C shows the expression heatmap of the 60 repeats contributing more contigs. For each type of repeats, we selected the contig with the highest absolute fold-change.

Repeat contigs also included a group annotated as "simple repeats", containing microsatellites and other low complexity elements. Contrarily to EREs, these do not have the capacity to be expressed independently. Indeed, in over 70% of cases, these contigs were uniquely mapped to genic sequences. In addition to annotated repeats and simple repeats, DE-kupl identified 4762 contigs (4497 up, 265 down) with multiple genome hits but no match in the DFAM repeat database (Additional file 2: Table S4). Many of these repeats were from Mucins, immunoglobulins and multicopy gene families such as $NBPF$ and $TBC1$. These repeats are shared between two cohorts and thus represent robust events of (mostly) overexpressed RNA fragments in tumors that would hardly be noticed in regular RNA-seq analysis due to their low mappability.

To investigate repeat-based patient subgroups, we performed clustering of tumors based on the most frequent repeat elements in Fig 3A: AluSx, L1P1_orf2, and L1P3_orf2 (as FLAM repeats are a family of Alu-like monomers that give birth to the left arms of the Alu elements, we did not account for FLAM_C_1_143). K-means clustering with k varying from 2 to 4 groups consistently found two major subgroups: subgroup 1 ("repeat-low") displayed generally low expression of Alu and L1 repeats compared to subgroup 2 ("repeat-high") (Fig 4A).

We then related the two repeat subgroups with somatic alterations observed in TCGA patients. Patients in the repeat-high group were more frequently mutated in LUAD drivers $CSMD3$, $TP53$, $PTPRD$, $PTPRT$, $GRIN2A$, $EPHA3$, and $MB21D2$ (Fig 4B, Fisher P<0.05). Patients in the repeat-high group had a significantly higher TMB (Wilcoxon P=1.5e-07) and a higher ratio of CNVs than other patients (Wilcoxon P=5.5e-05 for gain; P=0.019 for loss) (Fig 4C).

We observed no difference between subgroups in terms of age, gender, tumor stage, overall survival (OS), and vital status, but found more smokers in the repeat-high group (Wilcoxon P=0.02). We then assessed the immune cell contents of samples estimated by gene expression deconvolution. The repeat-high subgroup had lower proportions of dendritic cells, M2 macrophages, mast cells, monocytes and CD4+ T cells and overall immune content than the repeat-low subgroup (Fig 4D). In summary, "repeat-high" tumors associate with higher genome instability, more frequent smoking and lower immune infiltration.

Immune cell-associated contigs

We sought which contigs best correlated with tumor immune cell contents estimated by gene expression deconvolution. Sixty five contigs were found correlated with at least one type of immune cell (Additional file 1: Fig. S7). Most of these were uniquely mapped to genic introns or exons and underexpressed in tumors. Positive correlations were mostly observed with M2/M0 macrophages or resting CD4+ T cells, *i.e.* with a generally repressive or quiescent immune environment. However, a few contigs were associated to immune active M1 macrophages, including two contigs matching $GBP5$ (a marker of activated macrophages) and $CXCR2P1$ (a pseudogene expressed in an intron of $RUFY4$, a gene expressed in dendritic cells).

Overall, immune cell-associated contigs mapped leukocyte-specific or immunity-related genes, suggesting most contigs originated from the immune cell themselves (Additional file 2: Table S11).

Perhaps the most intriguing set of immune cell-associated contigs was that correlated to naive CD4+ T-cells. These cells are not especially enriched in tumor or normal samples, yet they correlate with six DE contigs. One contig was strongly repressed in tumors and corresponded to *Klebsiella pneumoniae* large subunit rRNA. Indeed, *Klebsiella* is a common lung bacterium against which cross-reactive T-cells are present in the naive CD4+ T-cell repertoire [50]. Our results thus suggest the joint occurrence of *Klebsiella* and matching CD4+ T-cell in normal lungs, and their disappearance in tumors. Of note, this *Klebsiella* contig also correlates positively with multiple contigs in the *SFTP* gene (Additional file 2: Table S12), in line with *SFTP* roles in defense against respiratory pathogens [51].

The other five contigs associated with naive CD4+ T-cells were all overexpressed in tumors. These included two intergenic repeats related to HERV (human endogenous retrovirus): HERV-E and MER9. The HERV-E contig was expressed from the *env* gene of a near full-length retroelement. One may hypothesize that expression and antigen production by the *env* gene trigger recruitment of CD4+ T-cells, as observed already in breast cancer [52]. Alternatively, reactivation of HERV elements could be an intrinsic feature of the CD4+ T-cells [53]. This analysis illustrates how non-reference RNA quantification can illuminate the interplay between cell types and specific RNA elements including exogenous elements in a bulk tissue.

Novel sources of shared neoantigens enriched in lincRNAs

Tumors express a large diversity of transcripts that are not usually expressed in normal tissues. When translated, these transcripts can produce peptides recognized as non-self by the epitope presentation machinery, triggering antitumor immune response [54]. These tumor-specific antigens or neoantigens are the object of active investigation for immunotherapy and tumor vaccine development. Protocols for neoantigen discovery usually start from a list of nonsynonymous somatic mutations identified from WES or WGS libraries and whose expression is confirmed by RNA-seq. Candidate mutated peptides are then submitted to an epitope presentation prediction pipeline [55]. This protocol predicts potential neoantigens from annotated and mappable regions. However, neoantigens can be produced from any transcript, including repeats and supposedly non-coding lncRNAs [56, 57]. Therefore we thought our reference-free approach could be a good source for such elements.

We considered contigs with no expression in normal tissues as potential neoantigen sources. To focus on shared neoantigens, we further requested contigs to be expressed in at least 15% of tumor samples. This selected 2375 contigs in the LUADtcga dataset (Fig 5.A). About 20% of these contigs (N=472) where also silent in normal tissues of the LUADseo cohort (Fig 5.B). We evaluated the potential of these "strictly tumoral" contigs for neoantigen presentation. Fifty five strictly tumoral contigs produced peptides predicted to be strong MHC-class-I binders by netMHCpan (Additional file 2: Table S10). Although potential neoantigen-producing contigs were found in several categories and locations, intergenic location was the most

significantly enriched category (Additional file 1: Fig. S8). Overall, contigs from intergenic regions, non-coding RNAs and pseudogenes contributed 58% of predicted neoantigens (Additional file 2: Table S10), consistent with previous reports of abundant neoantigen production from non-coding regions in other cancers [57].

Repeats, intronic RNAs and lincRNA as survival predictors

To identify RNA elements associated with outcome, we retrieved overall survival (OS) data for the TCGA cohort and performed univariate Cox regression with the different classes of contigs. Thirty nine contigs were significantly related to OS after multiple testing correction (Additional file 2: Table S7). Outcome-related contigs are mostly enriched in repeats (Additional file 2: Table S8), especially HERV elements (4 out of the 10 top repeats) and Alu/L1 family elements (AluSx and L1P3_orf2). While HERV elements expression was always negatively related to OS, the trend for other repeats was variable, with different Line1 and Alu elements having either positive or negative relation to OS (Additional file 2: Table S7). Another interesting OS-related element was a novel splice variant in ELF1, a transcription factor of the ETS family involved in multiple cancers (Additional file 2: Table S7)[58].

We then performed multivariate Cox regression using sets of contigs selected by lasso regression within each contig category and using differentially expressed genes (Additional file 2: Table S9). Models based on annotated and simple repeats had the best prognostic power (log-rank P=2e-16, 2e-13, respectively, Fig 6). The "annotated repeat" model was based on 12 contigs, including six L1 and three HERV elements, reinforcing the relevance of these repeats for prognosis. The "simple repeat" model included 12 contigs with microsatellite-like repeats, of which 11 were uniquely mapped to the genome (Additional file2: Table S9). Other strong outcome predictors were obtained using lincRNA, intronic and unmapped contigs, all of which achieved a better patient stratification than a model based on DE genes (Fig 6).

Unsupervised sample clustering based on non-reference RNAs

To investigate the capacity of non-reference RNAs to distinguish tumor and normal tissues in an unsupervised fashion, we performed PCA clustering of samples using contigs from each class (Fig 7). Tumor and normal tissues can be distinguished based on SNV, splice, intron, and lincRNA event classes as clearly as based on differentially expressed genes ("DEG" in Fig 7). This capacity is consistently observed in both cohorts. However, while many repeats are important with respect to tumor subclasses and survival, repeats altogether do not permit a clear separation of tumor and normal tissues in unsupervised clustering. Classes "polyA", "split" and "unmapped" did not achieve clear separation either, which was more expected as these sets were much smaller in size.

Discussion

Using reference-free analysis of LUAD RNA-seq data, we identified a large set of differential RNA elements that were present in two independent LUAD cohorts. We classified these elements based on their genomic location, mapping characteristics and repeat contents. We did not analyze in detail all contig classes but focused

instead on contigs mapping to hypervariable genes, repeats, lincRNAs and intronic elements. Besides these, a number of splice variants, chimeras, exogenous (non-human) sequences were found differentially expressed and could be pursued further.

A defining class of differential events involved endogenous repeats. The expression of L1 and Alu repeats defined two major tumor subgroups. The subgroup with higher L1/Alu expression was associated with more frequent mutations in $P53$, a higher mutational and copy number burden and a reduced immune cell infiltrate. This is consistent with previous observations that retrotransposition events can be controlled by $P53$ [59], correlate with a repressed immune environment [59, 60] and can lead to genome instability [61]. Expressed repeats also had significant prognostic power. Multivariate signatures composed of HERV and L1 elements, or simple repeats, stratified patients into distinct survival groups. Of note, HERV expression has been sporadically involved in various cancer types [62] and has recently been associated with poor prognosis in colorectal cancer [63].

A limitation of k-mer approaches for TE analysis is that transcripts are not fully assembled and thus the nature of repeats, whether expressed as functional retroelements or as part of mRNA or lncRNAs cannot be systematically established. Nonetheless, the majority of DE contigs are long enough to enable unambiguous mapping on the human genome, hence their origin could be further explored, including when coming from novel insertion events.

An attractive aspect of reference-free RNA-seq analysis is the capacity to identify novel forms of known cancer drivers or biomarkers. Indeed, we identified novel intron retention events in $EGFR$ and MET and multiple new variants of $CEACAM5$ and $KR19$. Perhaps even more interesting is the ability to detect potential neoantigen sources in variant transcripts. Tumor-specific neoantigens have previously been identified from repeats and non-coding regions using mapping-based strategies [54, 57]. However, our approach casts a wider net as it collects all events independently of their origin, including when arising from unmappable or profoundly rearranged regions. Indeed we identified about 500 strictly tumoral contigs shared by patients from the two independent cohorts, 55 of which were predicted to produce MHC-class-I neoantigens. These shared neoantigen candidates are of particular interest since their targeting by antitumor therapy would potentially benefit groups of multiple patients.

The wealth of information uncovered in the present study is a strong incentive to explore other applications of reference-free transcriptomics. One such application is the identification of patient-specific abnormal transcripts under a 1 *vs* n experimental design, which is addressed by the Mintie software [64]. Reference-free strategies can also be used for building predictive models. We [65] and others [66, 67] are exploring this kind of approach to classify cancer RNA-seq samples with promising results. Finally, reference-free differential analysis of the type used in this study could be of particular interest in meta-transcriptomics projects where RNAs are sequenced from an environment containing unknown bacterial, archaeal or eukaryotic species. Our protocol guarantees that any RNA that is specific to a sample subset will be captured independently of its origin. We hope the present analysis will encourage others to explore other data sources in a reference-free manner.

List of abbreviations
SNV: Single-Nucleotide Variants
CNV: Copy Number Variant
SV: Structural Variant
AS: Alternative Splicing
TCGA: The Cancer Genome Atlas
LUAD: Lung Adenocarcinoma
PRAD: Prostate Adenocarcinoma
EREs: endogenous retroelements

Declarations
Ethics approval and consent to participate
Not applicable

Consent to publish
Not applicable.

Availability of data and materials
Not applicable.

Competing interests
The authors declare that they have no competing interests.

Funding
This work was funded in part by Agence Nationale de la Recherche grant ANR-18-CE45-0020 and by a PhD studentship to YW by Annoroad Gene Technology, Beijing.

Authors' contributions
YW and DG designed the workflow and analyzed the results, YW downloaded and processed the datasets, YW and DG wrote the manuscript, MA and MG assisted in statistical analysis, HX assisted in coding scripts. AL annotated the repeat types.

Acknowledgements
The results shown in this work are in whole or part based upon data generated by the TCGA Research Network: https://www.cancer.gov/tcga.

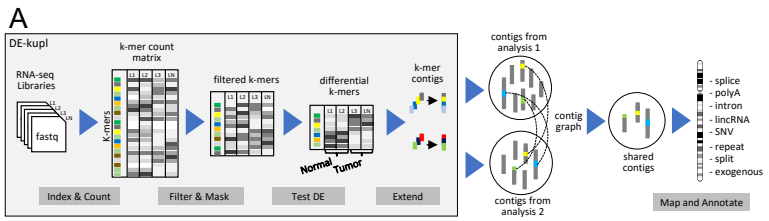
Author details
[1]Institute for Integrative Biology of the Cell (I2BC), Université Paris-Saclay, CNRS, CEA, 1 avenue de la Terrasse, 91190, Gif-sur-Yvette, France. [2]Gustave Roussy, 114 rue Edouard Vaillant, 94800, Villejuif, France. [3]Annoroad Gene Technology Co., Ltd,, 100176, Beijing, China.

References
1. Gollub, M.J., Prowda, J.C.: Primary melanoma of the esophagus: radiologic and clinical findings in six patients. Radiology **213**(1), 97–100 (1999)
2. Parker, J.S., Mullins, M., Cheang, M.C., Leung, S., Voduc, D., Vickery, T., Davies, S., Fauron, C., He, X., Hu, Z., *et al.*: Supervised risk predictor of breast cancer based on intrinsic subtypes. Journal of clinical oncology **27**(8), 1160 (2009)
3. Morillon, A., Gautheret, D.: Bridging the gap between reference and real transcriptomes. Genome biology **20**(1), 1–7 (2019)
4. Kahles, A., Lehmann, K.-V., Toussaint, N.C., Hüser, M., Stark, S.G., Sachsenberg, T., Stegle, O., Kohlbacher, O., Sander, C., Caesar-Johnson, S.J., *et al.*: Comprehensive analysis of alternative splicing across tumors from 8,705 patients. Cancer cell **34**(2), 211–224 (2018)
5. Vitting-Seerup, K., Sandelin, A.: Isoformswitchanalyzer: analysis of changes in genome-wide patterns of alternative splicing and its functional consequences. Bioinformatics **35**(21), 4469–4471 (2019)
6. Iyer, M.K., Niknafs, Y.S., Malik, R., Singhal, U., Sahu, A., Hosono, Y., Barrette, T.R., Prensner, J.R., Evans, J.R., Zhao, S., *et al.*: The landscape of long noncoding rnas in the human transcriptome. Nature genetics **47**(3), 199–208 (2015)
7. Gong, J., Li, Y., Liu, C.-j., Xiang, Y., Li, C., Ye, Y., Zhang, Z., Hawke, D.H., Park, P.K., Diao, L., *et al.*: A pan-cancer analysis of the expression and clinical relevance of small nucleolar rnas in human cancer. Cell reports **21**(7), 1968–1981 (2017)
8. Solovyov, A., Vabret, N., Arora, K.S., Snyder, A., Funt, S.A., Bajorin, D.F., Rosenberg, J.E., Bhardwaj, N., Ting, D.T., Greenbaum, B.D.: Global cancer transcriptome quantifies repeat element polarization between immunotherapy responsive and t cell suppressive classes. Cell reports **23**(2), 512–521 (2018)
9. Ouchenir, L., Renaud, C., Khan, S., Bitnun, A., Boisvert, A.-A., McDonald, J., Bowes, J., Brophy, J., Barton, M., Ting, J., et al.: The epidemiology, management, and outcomes of bacterial meningitis in infants. Pediatrics **140**(1) (2017)
10. Zapatka, M., Borozan, I., Brewer, D.S., Iskar, M., Grundhoff, A., Alawi, M., Desai, N., Sültmann, H., Moch, H., Cooper, C.S., *et al.*: The landscape of viral associations in human cancers. Nature genetics **52**(3), 320–330 (2020)

11. Amemiya, H.M., Kundaje, A., Boyle, A.P.: The encode blacklist: identification of problematic regions of the genome. Scientific reports **9**(1), 1–5 (2019)
12. Audoux, J., Philippe, N., Chikhi, R., Salson, M., Gallopin, M., Gabriel, M., Le Coz, J., Drouineau, E., Commes, T., Gautheret, D.: De-kupl: exhaustive capture of biological variation in rna-seq data through k-mer decomposition. Genome biology **18**(1), 1–15 (2017)
13. Ioannidis, J.P.: Microarrays and molecular research: noise discovery? Lancet (London, England) **365**(9458), 454–455 (2005)
14. Michiels, S., Koscielny, S., Boulet, T., Hill, C.: Gene expression profiling in cancer research. Bulletin du cancer **94**(11), 976–980 (2007)
15. Network, C.G.A.R., *et al.*: Comprehensive molecular profiling of lung adenocarcinoma. Nature **511**(7511), 543 (2014)
16. Seo, J.-S., Ju, Y.S., Lee, W.-C., Shin, J.-Y., Lee, J.K., Bleazard, T., Lee, J., Jung, Y.J., Kim, J.-O., Shin, J.-Y., *et al.*: The transcriptional landscape and mutational profile of lung adenocarcinoma. Genome research **22**(11), 2109–2119 (2012)
17. Abeshouse, A., Ahn, J., Akbani, R., Ally, A., Amin, S., Andry, C.D., Annala, M., Aprikian, A., Armenia, J., Arora, A., *et al.*: The molecular taxonomy of primary prostate cancer. Cell **163**(4), 1011–1025 (2015)
18. Ritchie, M.E., Phipson, B., Wu, D., Hu, Y., Law, C.W., Shi, W., Smyth, G.K.: limma powers differential expression analyses for rna-sequencing and microarray studies. Nucleic acids research **43**(7), 47–47 (2015)
19. De Paepe, K.: Comparison of methods for differential gene expression using rna-seq data (2015)
20. Bray, N.L., Pimentel, H., Melsted, P., Pachter, L.: Near-optimal probabilistic rna-seq quantification. Nature biotechnology **34**(5), 525–527 (2016)
21. Gu, Z., Eils, R., Schlesner, M.: Complex heatmaps reveal patterns and correlations in multidimensional genomic data. Bioinformatics **32**(18), 2847–2849 (2016)
22. Hagberg, A., Swart, P., S Chult, D.: Exploring network structure, dynamics, and function using networkx. Technical report, Los Alamos National Lab.(LANL), Los Alamos, NM (United States) (2008)
23. Madden, T.: The blast sequence analysis tool. In: The NCBI Handbook [Internet]. 2nd Edition. National Center for Biotechnology Information (US), ??? (2013)
24. Hubley, R., Finn, R.D., Clements, J., Eddy, S.R., Jones, T.A., Bao, W., Smit, A.F., Wheeler, T.J.: The dfam database of repetitive dna families. Nucleic acids research **44**(D1), 81–89 (2016)
25. Yu, G., Wang, L.-G., Han, Y., He, Q.-Y.: clusterprofiler: an r package for comparing biological themes among gene clusters. Omics: a journal of integrative biology **16**(5), 284–287 (2012)
26. MacQueen, J., *et al.*: Some methods for classification and analysis of multivariate observations. In: Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, vol. 1, pp. 281–297 (1967). Oakland, CA, USA
27. Sondka, Z., Bamford, S., Cole, C.G., Ward, S.A., Dunham, I., Forbes, S.A.: The cosmic cancer gene census: describing genetic dysfunction across all human cancers. Nature Reviews Cancer **18**(11), 696–705 (2018)
28. Mayakonda, A., Lin, D.-C., Assenov, Y., Plass, C., Koeffler, H.P.: Maftools: efficient and comprehensive analysis of somatic variants in cancer. Genome research **28**(11), 1747–1756 (2018)
29. Colaprico, A., Silva, T.C., Olsen, C., Garofano, L., Cava, C., Garolini, D., Sabedot, T.S., Malta, T.M., Pagnotta, S.M., Castiglioni, I., *et al.*: Tcgabiolinks: an r/bioconductor package for integrative analysis of tcga data. Nucleic acids research **44**(8), 71–71 (2016)
30. Newman, A.M., Liu, C.L., Green, M.R., Gentles, A.J., Feng, W., Xu, Y., Hoang, C.D., Diehn, M., Alizadeh, A.A.: Robust enumeration of cell subsets from tissue expression profiles. Nature methods **12**(5), 453–457 (2015)
31. Madeira, F., Park, Y.M., Lee, J., Buso, N., Gur, T., Madhusoodanan, N., Basutkar, P., Tivey, A.R., Potter, S.C., Finn, R.D., *et al.*: The embl-ebi search and sequence analysis tools apis in 2019. Nucleic acids research **47**(W1), 636–641 (2019)
32. Jurtz, V., Paul, S., Andreatta, M., Marcatili, P., Peters, B., Nielsen, M.: Netmhcpan-4.0: improved peptide–mhc class i interaction predictions integrating eluted ligand and peptide binding affinity data. The Journal of Immunology **199**(9), 3360–3368 (2017)
33. Therneau, T.M., Lumley, T.: Package 'survival'. R Top Doc **128**(10), 28–33 (2015)
34. Kassambara, A., Kosinski, M., Biecek, P., Fabian, S.: Package 'survminer'. Drawing Survival Curves using 'ggplot2'.(R package version 0.3. 1.) (2017)
35. Friedman, J., Hastie, T., Tibshirani, R.: Regularization paths for generalized linear models via coordinate descent. Journal of statistical software **33**(1), 1 (2010)
36. Kaplan, E.L., Meier, P.: Nonparametric estimation from incomplete observations. Journal of the American statistical association **53**(282), 457–481 (1958)
37. Dobin, A., Davis, C.A., Schlesinger, F., Drenkow, J., Zaleski, C., Jha, S., Batut, P., Chaisson, M., Gingeras, T.R.: Star: ultrafast universal rna-seq aligner. Bioinformatics **29**(1), 15–21 (2013)
38. Robinson, J.T., Thorvaldsdóttir, H., Winckler, W., Guttman, M., Lander, E.S., Getz, G., Mesirov, J.P.: Integrative genomics viewer. Nature biotechnology **29**(1), 24–26 (2011)
39. Thorsson, V., Gibbs, D.L., Brown, S.D., Wolf, D., Bortone, D.S., Yang, T.-H.O., Porta-Pardo, E., Gao, G.F., Plaisier, C.L., Eddy, J.A., *et al.*: The immune landscape of cancer. Immunity **48**(4), 812–830 (2018)
40. Mandric, I., Rotman, J., Yang, H.T., Strauli, N., Montoya, D.J., Van Der Wey, W., Ronas, J.R., Statz, B., Yao, D., Petrova, V., *et al.*: Profiling immunoglobulin repertoires across multiple human tissues using rna sequencing. Nature communications **11**(1), 1–14 (2020)
41. Lefranc, M.-P., Giudicelli, V., Ginestoux, C., Jabado-Michaloud, J., Folch, G., Bellahcene, F., Wu, Y., Gemrot, E., Brochet, X., Lane, J., *et al.*: Imgt®, the international immunogenetics information system®. Nucleic acids research **37**(suppl_1), 1006–1012 (2009)
42. Imielinski, M., Guo, G., Meyerson, M.: Insertions and deletions target lineage-defining genes in human cancers. Cell **168**(3), 460–472 (2017)
43. Swallow, D.M., Gendler, S., Griffiths, B., Corney, G., Taylor-Papadimitriou, J., Bramwell, M.E.: The human

tumour-associated epithelial mucins are coded by an expressed hypervariable gene locus pum. Nature **328**(6125), 82–84 (1987)
44. Ning, Y., Zheng, H., Zhan, Y., Liu, S., Zang, H., Luo, J., Wen, Q., Fan, S., *et al.*: Comprehensive analysis of the mechanism and treatment significance of mucins in lung cancer. Journal of Experimental & Clinical Cancer Research **39**(1), 1–10 (2020)
45. Aithal, A., Rauth, S., Kshirsagar, P., Shah, A., Lakshmanan, I., Junker, W.M., Jain, M., Ponnusamy, M.P., Batra, S.K.: Muc16 as a novel target for cancer therapy. Expert opinion on therapeutic targets **22**(8), 675–686 (2018)
46. Wang, X.-M., Zhang, Z., Pan, L.-H., Cao, X.-C., Xiao, C.: Krt19 and ceacam5 mrna-marked circulated tumor cells indicate unfavorable prognosis of breast cancer patients. Breast cancer research and treatment **174**(2), 375–385 (2019)
47. Thistlethwaite, F.C., Gilham, D.E., Guest, R.D., Rothwell, D.G., Pillai, M., Burt, D.J., Byatte, A.J., Kirillova, N., Valle, J.W., Sharma, S.K., *et al.*: The clinical efficacy of first-generation carcinoembryonic antigen (ceacam5)-specific car t cells is limited by poor persistence and transient pre-conditioning-dependent respiratory toxicity. Cancer Immunology, Immunotherapy **66**(11), 1425–1436 (2017)
48. Larouche, J.-D., Trofimov, A., Hesnard, L., Ehx, G., Zhao, Q., Vincent, K., Durette, C., Gendron, P., Laverdure, J.-P., Bonneil, É., *et al.*: Widespread and tissue-specific expression of endogenous retroelements in human somatic tissues. Genome medicine **12**, 1–16 (2020)
49. Rangwala, S.H., Zhang, L., Kazazian, H.H.: Many line1 elements contribute to the transcriptome of human somatic cells. Genome biology **10**(9), 1–18 (2009)
50. Cassotta, A., Goldstein, J.D., Durini, G., Jarrossay, D., Baggi Menozzi, F., Venditti, M., Russo, A., Falcone, M., Lanzavecchia, A., Gagliardi, M.C., *et al.*: Broadly reactive human cd4+ t cells against enterobacteriaceae are found in the naïve repertoire and are clonally expanded in the memory repertoire. European journal of immunology **51**(3), 648–661 (2021)
51. Wright, J.R.: Host defense functions of pulmonary surfactant. Neonatology **85**(4), 326–332 (2004)
52. Wang-Johanning, F., Radvanyi, L., Rycaj, K., Plummer, J.B., Yan, P., Sastry, K.J., Piyathilake, C.J., Hunt, K.K., Johanning, G.L.: Human endogenous retrovirus k triggers an antigen-specific immune response in breast cancer patients. Cancer research **68**(14), 5869–5877 (2008)
53. White, C.H., Beliakova-Bethell, N., Lada, S.M., Breen, M.S., Hurst, T.P., Spina, C.A., Richman, D.D., Frater, J., Magiorkinis, G., Woelk, C.H.: Transcriptional modulation of human endogenous retroviruses in primary cd4+ t cells following vorinostat treatment. Frontiers in immunology **9**, 603 (2018)
54. Smith, C.C., Selitsky, S.R., Chai, S., Armistead, P.M., Vincent, B.G., Serody, J.S.: Alternative tumour-specific antigens. Nature Reviews Cancer **19**(8), 465–478 (2019)
55. Gopanenko, A.V., Kosobokova, E.N., Kosorukov, V.S.: Main strategies for the identification of neoantigens. Cancers **12**(10), 2879 (2020)
56. Ouspenskaia, T., Law, T., Clauser, K.R., Klaeger, S., Sarkizova, S., Aguet, F., Li, B., Christian, E., Knisbacher, B.A., Le, P.M., et al.: Thousands of novel unannotated proteins expand the mhc i immunopeptidome in cancer. bioRxiv (2020)
57. Laumont, C.M., Vincent, K., Hesnard, L., Audemard, É., Bonneil, É., Laverdure, J.-P., Gendron, P., Courcelles, M., Hardy, M.-P., Côté, C., et al.: Noncoding regions are the main source of targetable tumor-specific antigens. Science translational medicine **10**(470) (2018)
58. Sizemore, G.M., Pitarresi, J.R., Balakrishnan, S., Ostrowski, M.C.: The ets family of oncogenic transcription factors in solid tumours. Nature Reviews Cancer **17**(6), 337–351 (2017)
59. Jung, H., Choi, J.K., Lee, E.A.: Immune signatures correlate with l1 retrotransposition in gastrointestinal cancers. Genome research **28**(8), 1136–1146 (2018)
60. Zhang, X., Zhang, R., Yu, J.: New understanding of the relevant role of line-1 retrotransposition in human disease and immune modulation. Frontiers in Cell and Developmental Biology **8**, 657 (2020)
61. Lee, E., Iskow, R., Yang, L., Gokcumen, O., Haseley, P., Luquette, L.J., Lohr, J.G., Harris, C.C., Ding, L., Wilson, R.K., *et al.*: Landscape of somatic retrotransposition in human cancers. Science **337**(6097), 967–971 (2012)
62. Bannert, N., Hofmann, H., Block, A., Hohn, O.: Hervs new role in cancer: from accused perpetrators to cheerful protectors. Frontiers in microbiology **9**, 178 (2018)
63. Golkaram, M., Salmans, M.L., Kaplan, S., Vijayaraghavan, R., Martins, M., Khan, N., Garbutt, C., Wise, A., Yao, J., Casimiro, S., *et al.*: Hervs establish a distinct molecular subtype in stage ii/iii colorectal cancer with poor outcome. NPJ genomic medicine **6**(1), 1–11 (2021)
64. Cmero, M., Schmidt, B., Majewski, I.J., Ekert, P.G., Oshlack, A., Davidson, N.M.: Mintie: identifying novel structural and splice variants in transcriptomes using rna-seq data. bioRxiv (2020)
65. Nguyen, H.T., Xue, H., Firlej, V., Ponty, Y., Gallopin, M., Gautheret, D.: Reference-free transcriptome signatures for prostate cancer prognosis. BMC cancer **21**(1), 1–12 (2021)
66. Lorenzi, C., Barriere, S., Villemin, J.-P., Bretones, L.D., Mancheron, A., Ritchie, W.: imoka: k-mer based software to analyze large collections of sequencing data. Genome Biology **21**(1), 1–19 (2020)
67. Thomas, A., Barriere, S., Broseus, L., Brooke, J., Lorenzi, C., Villemin, J.-P., Beurier, G., Sabatier, R., Reynes, C., Mancheron, A., *et al.*: Gecko is a genetic algorithm to classify and explore high throughput sequencing data. Communications biology **2**(1), 1–8 (2019)

Figures



B

Study	LUADtcga	LUADseo	PRADtcga
# Normal samples	58	77	52
# Tumor Samples	524	77	506
DE genes	9 058	6 567	6 292
DE k-mers	6 069 791	12 102 802	2 459 131
DE contigs	368 257	449 860	439 062

C

$LUAD_{tcga} \cap LUAD_{seo}$	N	%
Shared contigs	160610	–
High quality	120822	100.0
Multi hit or repeat	50572	44.1
Up	100405	83.1
Down	20417	16.9
Genic	104762	86.6
Intronic	56146	44.9
SNV or indel	54581	32.1

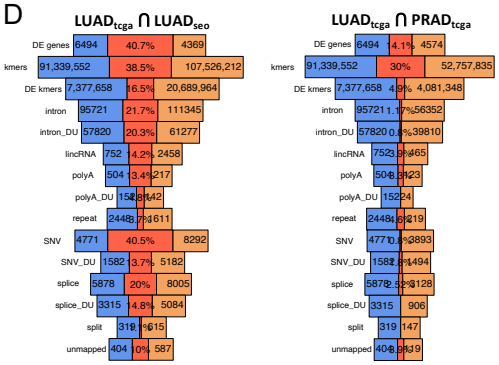
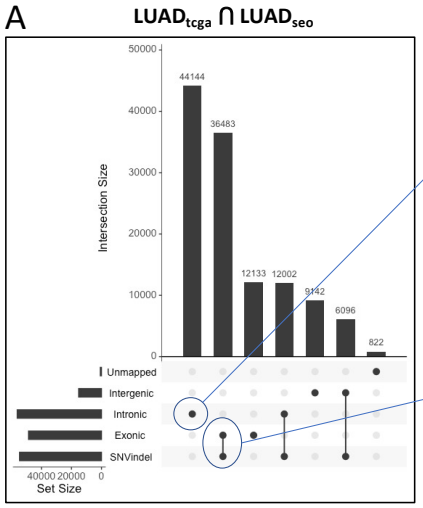


Figure 1 Overall analysis procedure and properties of identified contigs. (A) Computational pipeline for inferring differential contigs in each tumor/normal cohort, extraction of shared contigs and annotation. (B) Sizes of RNA-seq cohorts analyzed and numbers of differential events observed. (C) Summary statistics of differential contigs identified as shared between the LUADtcga and LUADseo analyzes. (D) Number of differential genes, k-mers and contigs in each independent analysis and shared between analyzes. On each row, lateral areas represent differential genes/k-mers/contigs found in each independent analysis and the central area represents shared differential genes/k-mers/contigs. Contigs are classified into different annotation groups.



B

ABCC3	162	CP	94	LMO3	105
AC019117.1	281	CU633906.2	84	LRIG3	85
AC243967.1	86	CU633967.1	98	MGST1	106
AGBL4	115	DUXAP9	166	MIR663AHG	78
AGR2	80	ESRP1	79	MUC4	127
AKR1C1	139	FAM3C	79	MYO6	78
ASPH	156	FNDC3B	90	PVT1	210
ATP1B1	121	FP671120.2	185	RNF19A	126
C11orf80	104	GALNTL6	96	RUNX1	261
AMACR	117	GLS	108	SDHAP3	136
CAPN13	94	HIF1A-AS2	97	SFTPA2	84
CAPN8	154	LAMB3	104	SSR3	83
CD46	109	LINC00342	84	SYTL2	109
CIT	77	LINC00624	102	TRA2A	98
CLPTM1L	85	LINC01004	108	XPR1	77

C

AC244205.1	111	IGHV1-69	196	IGKV3-20	351
B2M	255	IGHV3-11	158	IGKV3OR2	432
COL1A1	203	IGHV3-23	217	IGKV4-1	209
IGHA1	846	IGHV3-30	258	IGLC2	1082
IGHA2	292	IGHV3-7	239	IGLC3	181
IGHG1	1843	IGHV4-39	173	IGLL5	441
IGHG2	1094	IGHV4-4	155	IGLV1-47	111
IGHG3	196	IGKC	1974	IGLV3-1	121
IGHG4	1440	IGKJ1	128	IGLV3-21	263
IGHGP	313	IGKV1-33	145	MT-ND6	184
IGHJ4	143	IGKV1-39	238	MUC3A	135
IGHJ5	236	IGKV1-5	345	SFTPA1	1043
IGHV1-18	129	IGKV1OR1-1	118	SFTPA2	1664
IGHV1-2	379	IGKV3-11	155	SFTPB	220
IGHV1-3	138	IGKV3-15	383	SFTPC	3087

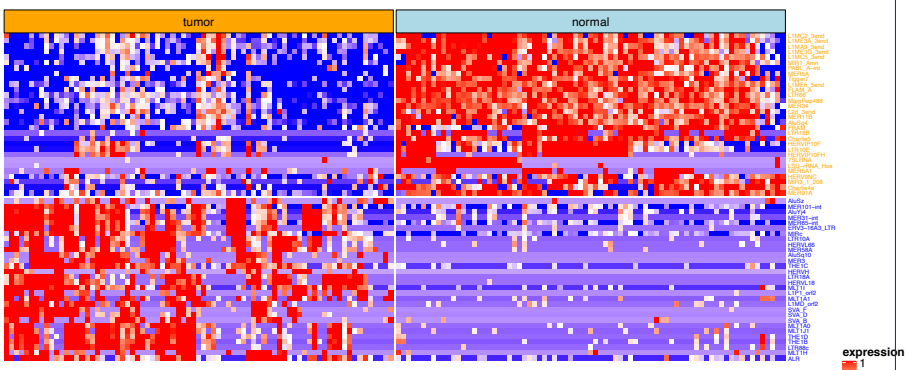
Figure 2 General properties of shared differential expression contigs in LUAD. (A) UpsetR plot of major contig categories based on mapping location and presence of SNV or indels. (B) 45 top genes by number of mapped contigs in the circled intronic category. (C) 45 top genes by number of mapped contigs in the circled exonic+SNVindel category. Numbers of contigs mapped to each gene are indicated.

A

Top 20 repeat types with the most contigs from LUADtcga

rep.type	contigs	Up in tumor	Down in tumor	SNVs	protein.coding	lncRNA
L1P1_orf2	755	754	1	568	233	186
AluSx	455	369	86	250	299	60
FLAM_C_1_143	316	252	64	128	216	38
L1P3_orf2	302	288	14	189	109	82
AluJb	276	227	49	119	189	28
LSU-rRNA_Hsa	264	259	5	249	51	185
AluSz6	174	143	31	74	115	25
AluSp	147	105	42	84	85	18
PRIMA4-int	123	92	31	38	92	8
L1PB_orf2	118	116	2	44	52	26
L1P1_5end	115	115	0	70	39	32
MIR_1_262	111	98	13	10	90	15
AluJr4	110	90	20	47	69	20
AluY	110	79	31	68	59	12
L1HS_5end	109	109	0	65	31	19
L1PREC2_orf2	106	103	3	37	29	32
HERVH	105	96	9	30	19	52
AluSz	104	91	13	61	62	13
L1M3_orf2	89	88	1	5	49	16
AluJr	86	66	20	30	48	14

B



C

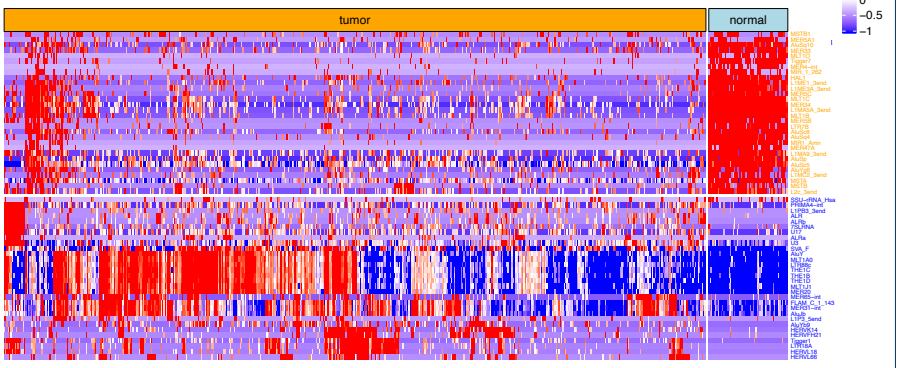


Figure 3 Tumor identity based on top repeats (A) Top 20 repeat types with the most contigs in LUADtcga dataset. (B/C) Expression heatmap of top repeat-containing contigs (ranked by fold change) in the LUADseo (A) and LUADtcga (B) datasets. Contig expression level in heatmaps is represented from blue (lowest) to red (highest).

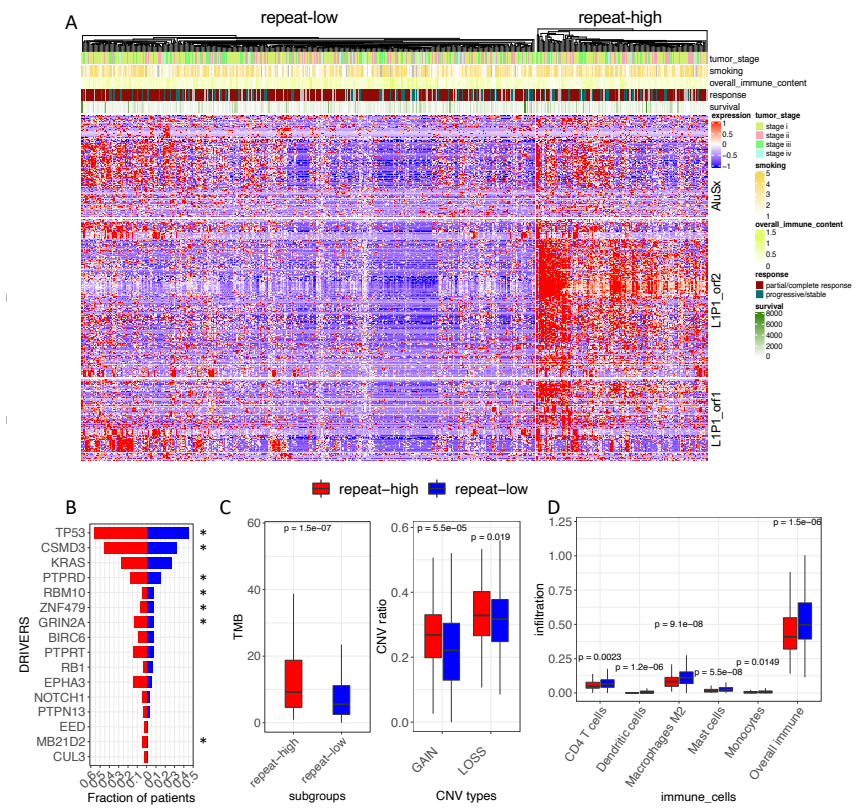


Figure 4 Characterization of patient subgroups based on repeat-containing contigs. (A) Clustering of LUADtcga patients into two subgroups based on Alu and L1P1 repeat expression. Subgroups were defined by K-means. (B) Fraction of patients with driver mutations for 16 COSMIC LUAD drivers. Drivers with Fisher P value < 0.05 were marked with star. (C) Mutational burden and CNV frequency distribution between two subgroups. (D) Variation of immune features between subgroups. The red and blue represent the repeat-high and repeat-low subgroups, respectively. P-values are computed by Wilcoxon test.

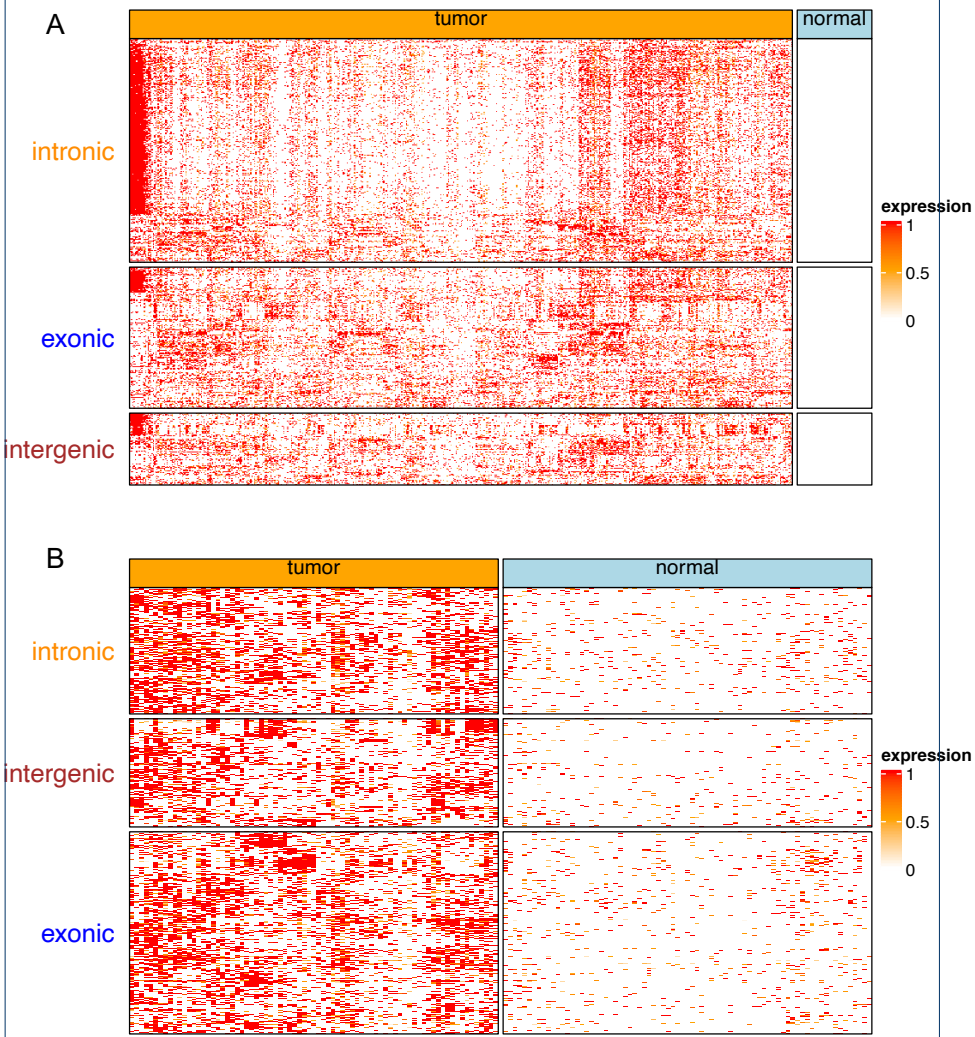


Figure 5 Expression heatmap of potential neoantigen sources in the two LUAD datasets. Tumor-specific contigs were first selected in LUADtcga (A) and validated in the LUADseo dataset (B).

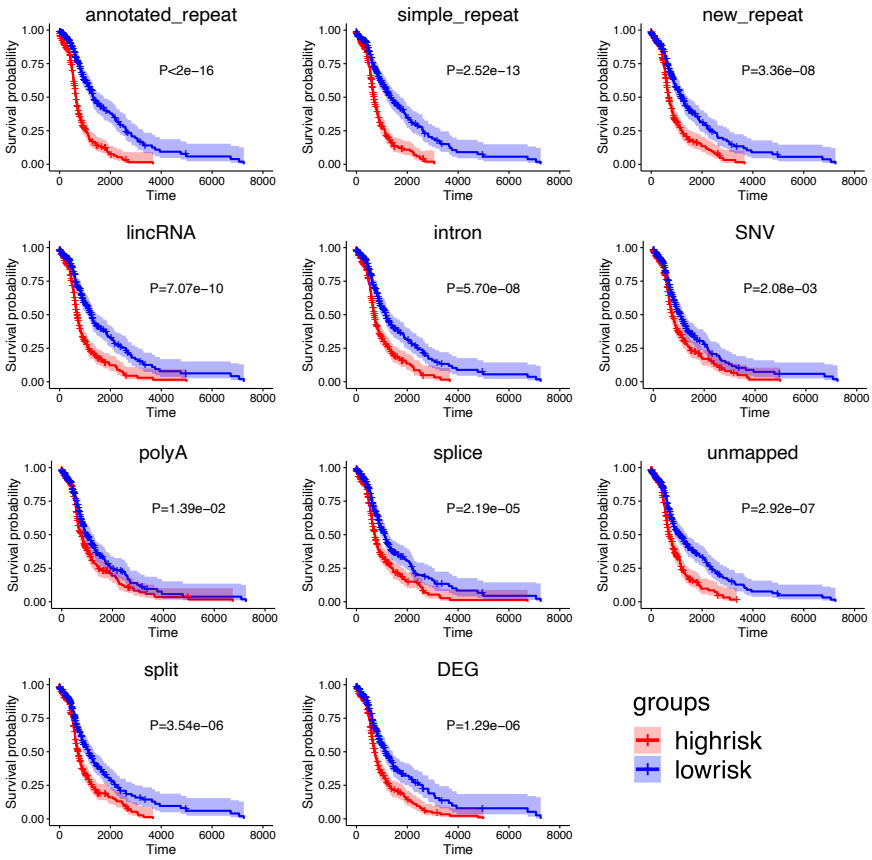


Figure 6 Kaplan-Meyer curves of multivariate survival models per class of event. Patients in high and low-risk groups are shown in red and blue, respectively. Repeat events were separated into annotated, new and simple repeats. The other categories with more lasso-selected contigs were also included (Additional file 3: Table S8).

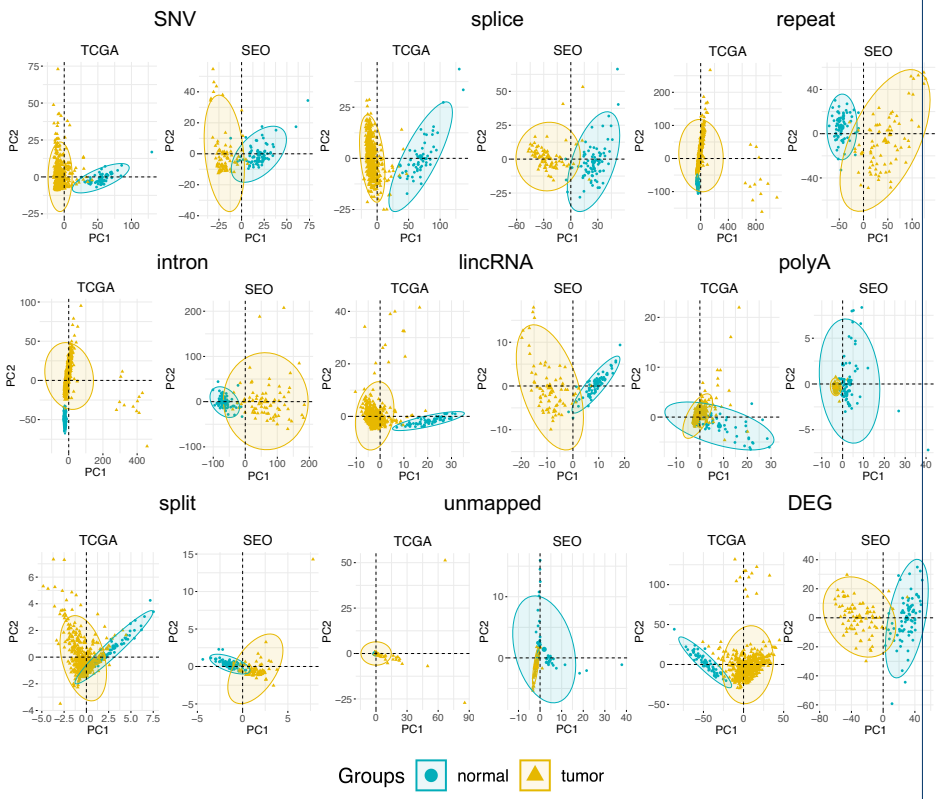


Figure 7 Principal component analysis of samples based on DE contigs and genes. Each panel represents PCA performed with one class of contigs and with differentially expressed genes (DEG), for the LUADtcga (TCGA) and LUADseo (SEO) datasets. Normal and tumor samples are marked using blue and yellow, respectively. Confidence ellipses are drawn with package factoextra for each group.

Additional files

Additional file 1 — Figure S1
The graph-based protocol detecting shared contigs between TCGA and SEO datasets. (A) Contigs from each dataset. The bars marked with the same color represent the same k-mers. (B) Cliques construction based on the common k-mers. (C) Shared contigs identification based on the cliques.

Additional file 1 — Figure S2
Enrichment analysis of shared DEGs and contigs between TCGA and SEO datasets. The x axis represents the ranked DEGs or contigs based on log2FC in ascending order. The red vertical dotted line represents the position of log2FC cutoff.

Additional file 1 — Figure S3
Hypervariable genes in our analysis. Upset graph shows the overlap between different categories, including intronic, exonic, spliced and SNV or indel.

Additional file 1 — Figure S4
IGV views of intronic events. Each frame shows a metabam file composed of randomly sampled reads corresponding to the subcohort indicated on the left panel. The lower panel shows DE contigs and Gencode annotation. A: multiple intron retention in CEACAM5; B: lncRNA element expressed in an intron of MBD5; C: intron retention in EGFR; D: intron retention in MET.

Additional file 1 — Figure S5
Intronic event analysis. (A) Log2FC values of the top 20 intronic events (DU). Red and blue colors represent the expression fold change of intronic contigs and host genes, respectively. (B) Gene Ontology functional enrichment. Color represents the P-values and size represents the ratio of genes.

Additional file 1 — Figure S6
IGV views of lincRNA elements overexpressed in tumors. Each frame shows a metabam file composed of randomly sampled reads corresponding to the subcohort indicated on the left panel. The lower panel shows DE contigs and Gencode annotation.

Additional file 1 — Figure S7
Heatmap of Spearman correlation coefficient (CC) of contig counts and abundance of immune cell types evaluated by CIBERSORT. All contigs with a CC$>$0.5 with at least one immune cell type are shown. Immune cells not correlated with at least one contig are not shown. Row names show gene symbols and repeat types of contigs, whenever applicable. Row name colors indicate different contig categories. The log2FC sidebar shows expression fold change of contigs between normal and tumor samples.

Additional file 1 — Figure S8
Fractions of event types in strictly tumoral contigs predicted to produce neoantigens ("neo", N=472) and total shared DE contigs ("all", N=2375). Intergenic contigs are significantly over-represented in "neo" contigs (Fisher's exact P=1.2e-20).

Additional file 2 — Table S1-S12
Table S1: Nucleotide contents of DE-kupl contigs for the TCGA LUAD dataset. Table S2: Description of event categories extracted from DE-kupl-annot tables. Table S3: General characteristics of contigs shared between LUADtcga and LUADseo. Table S4: Summary statistics for all event categories in contigs shared between LUADtcga and LUADseo. Table S5: Genes with more than 35 mapped contigs (shared LUAD contigs. Colored columns indicate ratio of contigs in said categories). Table S6: Blast results of 100 contigs mapped to IGHV genes. Table S7: Univariate Cox regression results of all categories. Table S8: Enrichment of OS-related events. Table S9: Multivariate Cox regression results of all categories. Table S10: Peptides of strong binding levels predicted by netMHCpan 4.0 from "neoRNA" contigs. Table S11: GO enrichment using host genes of immune related contigs. Table S12: Contigs correlated with the $klebsiella$ contig.

Additional file 3
Correlation analysis of number of contigs and host gene expression.

Chapter 6

Association of Reference-free *k*-mer Signals to Genes and Transcripts

6.1 Motivation

The *Kmerator Suite* comprises three software: *Kmerator*, *countTags*, and *KmerExploR* (cf. Figure 1A in the attached article). *Kmerator* aims at extracting k-mer signatures from references transcriptome and genome at the gene, transcript, or chimera (fusion transcript) level (see Figure 1B for definitions of signature k-mers). The *countTags* program uses these signature k-mers as proxys for quantification of gene features across RNA-seq reads. *KmerExploR* is an example usage of *Kmerator* and *countTags* for a rapid characterization and quality control of any input RNA-seq dataset, with a user-friendly graphical interface. The work described in this chapter was conducted in collaboration with the Bio2M team led by Thérèse Commes at Université de Montpellier.

6.2 My Contribution

I contributed to test *Kmerator*'s ability to retrieve specific k-mers in genome and transcriptome, as well as participated in the communication with *Kmerator*'s developer about the initial algorithm and data structure design. I used different references for the human genome and transcriptome, ran *Kmerator* and analyzed results. I also used the *Kmerator* in chapter 7 of this thesis.

6.3 Article

Published online 23 June 2021

NAR Genomics and Bioinformatics, 2021, Vol. 3, No. 3 **1**
https://doi.org/10.1093/nargab/lqab058

Kmerator Suite: design of specific *k*-mer signatures and automatic metadata discovery in large RNA-seq datasets

Sébastien Riquier[1,†], Chloé Bessiere[1,†], Benoit Guibert[1], Anne-Laure Bouge[2], Anthony Boureux[1], Florence Ruffle[1], Jérôme Audoux[2], Nicolas Gilbert[1], Haoliang Xue[3], Daniel Gautheret[3] and Thérèse Commes[1,*]

[1]IRMB, University of Montpellier, INSERM, 80 rue Augustin Fliche, 34295, Montpellier, France, [2]SeqOne, 34000, Montpellier, France and [3]Institute for Integrative Biology of the Cell, CEA, CNRS, Université Paris-Saclay, 91198, Gif sur Yvette, France

Received November 17, 2020; Revised May 10, 2021; Editorial Decision May 31, 2021; Accepted June 17, 2021

ABSTRACT

The huge body of publicly available RNA-sequencing (RNA-seq) libraries is a treasure of functional information allowing to quantify the expression of known or novel transcripts in tissues. However, transcript quantification commonly relies on alignment methods requiring a lot of computational resources and processing time, which does not scale easily to large datasets. *K*-mer decomposition constitutes a new way to process RNA-seq data for the identification of transcriptional signatures, as *k*-mers can be used to quantify accurately gene expression in a less resource-consuming way. We present the Kmerator Suite, a set of three tools designed to extract specific *k*-mer signatures, quantify these *k*-mers into RNA-seq datasets and quickly visualize large dataset characteristics. The core tool, Kmerator, produces specific *k*-mers for 97% of human genes, enabling the measure of gene expression with high accuracy in simulated datasets. KmerExploR, a direct application of Kmerator, uses a set of predictor gene-specific *k*-mers to infer metadata including library protocol, sample features or contaminations from RNA-seq datasets. KmerExploR results are visualized through a user-friendly interface. Moreover, we demonstrate that the Kmerator Suite can be used for advanced queries targeting known or new biomarkers such as mutations, gene fusions or long non-coding RNAs for human health applications.

INTRODUCTION

Publicly available human RNA-sequencing (RNA-seq) datasets are precious resources for biomedical research. RNA-seq data are widely used to identify actively transcribed genes, quantify gene or transcript expression, identify new fusion transcripts or identify alternative splicing or mutation events. The search for specific transcriptional events or RNAs across large-scale data has become essential in precision medicine. Advanced tools such as recount2 (1) have achieved transcript counts in large datasets, available in an online resource. However, these tools are reference based and only provide counts for precomputed transcripts. An increasing number of studies attempt to analyze in a retrospective fashion the vast repository of RNA-seq data, including normal and pathological conditions, to discover or validate RNA biomarkers for disease diagnosis (2,3).

For this purpose, it is important to select relevant RNA-seq datasets with homogeneous characteristics and sufficient samples among thousands of publicly available files. The reanalysis of RNA-seq datasets poses two major challenges. The first challenge is to filter data series and select the most homogeneous and reliable set of libraries for exploration in the context of incomplete metadata (4). The second challenge is to perform RNA biomarker quantification in reasonable time and with sufficient accuracy to extract biological information in such datasets. Alignment-based methods like STAR (5) and CRAC (6) require significant computational resources, making them inadequate for querying datasets on the order of 100–1000 files for a specific biomarker. Pseudo-alignment algorithms like Kallisto (7) and Salmon (8) are much faster but most commonly use a reference transcriptome far from the real complex biological RNA diversity. This highlights the need for tools enabling fast and specific quantification of candidate se-

*To whom correspondence should be addressed. Tel: +33 4 67330190; Email: therese.commes@inserm.fr
†These authors contributed equally to this work.

quences in a large set of RNA-seq data. Recently, approaches relying on *k*-mers from raw sequence files have emerged and are used for the query of transcriptomic data. These methods require less time and computational resources than common ones and are suited to various biological questions, including the analysis of unannotated and atypical RNA transcriptional events. For instance, Okamura and Kinoshita proposed an ultrafast mRNA quantification method, based on unique *k*-mers, that outperforms conventional approaches (9). Yu *et al*. (10) investigated gene fusion queries of all tumor samples from The Cancer Genome Atlas project using *k*-mer sets. The DEkupl pipeline developed by Audoux *et al*. (11) finds differential events between two groups of RNA-seq data at the *k*-mer level.

Moreover, classical methods fail to interrogate the whole transcriptome complexity as each RNA is the result of a complex chain of events that combines genetic variation, transcription regulation and RNA processing combined with pathological alterations (12). The *k*-mer approach we propose is not an equivalent method compared to the above-mentioned ones, but a new way to explore RNA-seq data that could also be used for in-depth exploration outside the reference.

Although any transcript sequence can be decomposed into *k*-mers, only a subset of these *k*-mers is specific for the transcript. We call this subset the *k*-mer signature. These specific *k*-mers can then be quantified in RNA-seq raw data, making it quick and easy to measure the candidate transcript expression level in a wide range of RNA-seq datasets.

In this paper, we present the Kmerator Suite, a set of three tools designed to (i) extract *k*-mer signatures from transcripts, (ii) quantify these *k*-mers into RNA-seq datasets and (iii) visualize large RNA-seq dataset characteristics using precomputed signatures. The core of this suite is Kmerator, which generates *k*-mer signatures specific for genes or transcripts. The second tool, countTags, is used to quantify selected *k*-mers across raw RNA-seq files. We first tested the performance of Kmerator + countTags over the whole transcriptome and showed that *k*-mer signature quantification results were close to simulated count data. The third tool, KmerExploR, demonstrates the capacity of the Kmerator + countTags pipeline combined to a set of predefined *k*-mer signatures, to perform metadata extraction from raw RNA-seq data. KmerExploR extracts sample characteristics related to the sequencing protocol (ribosomal depletion, polyA+, strand-specific protocol, 5′/3′ bias, etc.), tissue origin (sex) and possible contaminations (mycoplasma, virus, other species or cell lines). Such high-level quality control procedures are valuable as a screening tool before analyzing datasets of uncertain quality, such as public datasets. KmerExploR can also be used in advanced applications to look for user-defined transcripts resulting from mutated alleles or gene fusions in RNA-seq datasets.

MATERIALS AND METHODS

Kmerator: *k*-mer signature identification

An overview of the Kmerator Suite is provided in Figure 1A. Kmerator is a tool designed for the prediction of specific *k*-mers from input sequences, considering a reference genome and an Ensembl-like fasta transcriptome (see Figure 1A and Supplementary Figure S1A). It is implemented in Julia programming language (https://julialang.org) and distributed with GitHub (https://github.com/Transipedia/kmerator). Kmerator strictly depends on a reference genome [fasta or Jellyfish (13) index format] and on an Ensembl fasta format transcriptome, to define a *k*-mer as specific or not, depending on the number of occurrences on each reference. The reference genome and transcriptome fasta, used in this paper, have been downloaded here: https://www.ensembl.org/info/data/ftp/index.html. The procedure also needs a list of gene/transcript Ensembl IDs (or gene symbols) or sequences in fasta format from which Kmerator will extract specific *k*-mers. As shown in Supplementary Figure S1A, Kmerator first uses the Jellyfish software to index and count *k*-mers from the reference genome and transcriptome. For both genome and transcriptome fasta files, Jellyfish produces a hash table including all possible *k*-mers and their number of occurrences. These hash tables are stored for further querying. Second, using Jellyfish query, Kmerator generates, for each input gene/transcript, the list of *k*-mers derived from this sequence and their corresponding genome and transcriptome counts. These *k*-mers are then filtered according to the following criteria: (i) only *k*-mers associated with a biological event (transcript or gene, splice variant, chimeric RNA, circular RNA, etc.) are retained and (ii) *k*-mers must be specific according to Kmerator rules (see Figure 1C and Supplementary Figure S1A). Indeed, Kmerator includes three different levels of specificity (–level option), ‘gene’, ‘transcript’ and ‘chimera’, detailed below:

- Gene level specific *k*-mers are found zero (to include *k*-mers containing splicing junctions) or one time in the reference genome. They are also present in the reference transcriptome in at least one isoform transcript sequence. If we want to select only *k*-mers matching at least *n* isoforms on a total of *N*, a threshold can be set to the proportion of isoforms n/N the *k*-mer has to be specific to, using the –threshold option.
- Transcript level specific *k*-mers are found zero or one time in the reference genome. They also match the reference transcriptome only once (transcript specificity). If the candidate transcript is not annotated, the –unannotated option must be added. In this case, *k*-mers found zero or one time in the reference genome and that do not map to the reference transcriptome are retained.
- Chimera level specific *k*-mers are found neither in the reference genome nor in the reference transcriptome. This level must be combined to the –unannotated option. Kmerator outputs the list of specific *k*-mers (also called *k*-mer gene/transcript signature) according to the chosen parameters in fasta format, for each input sequence.

Kmerator command line options. The *k*-mer length can be set using the –length option. In the present study, we used the default 31 nt *k*-mer length according to the literature (11). The level of specificity is chosen among ‘gene’, ‘transcript’ and ‘chimera’ with the –level option. When using the gene level, the APPRIS database (http://appris.bioinfo.cnio.es) can be queried to identify the ‘PRINCIPAL’ transcript,

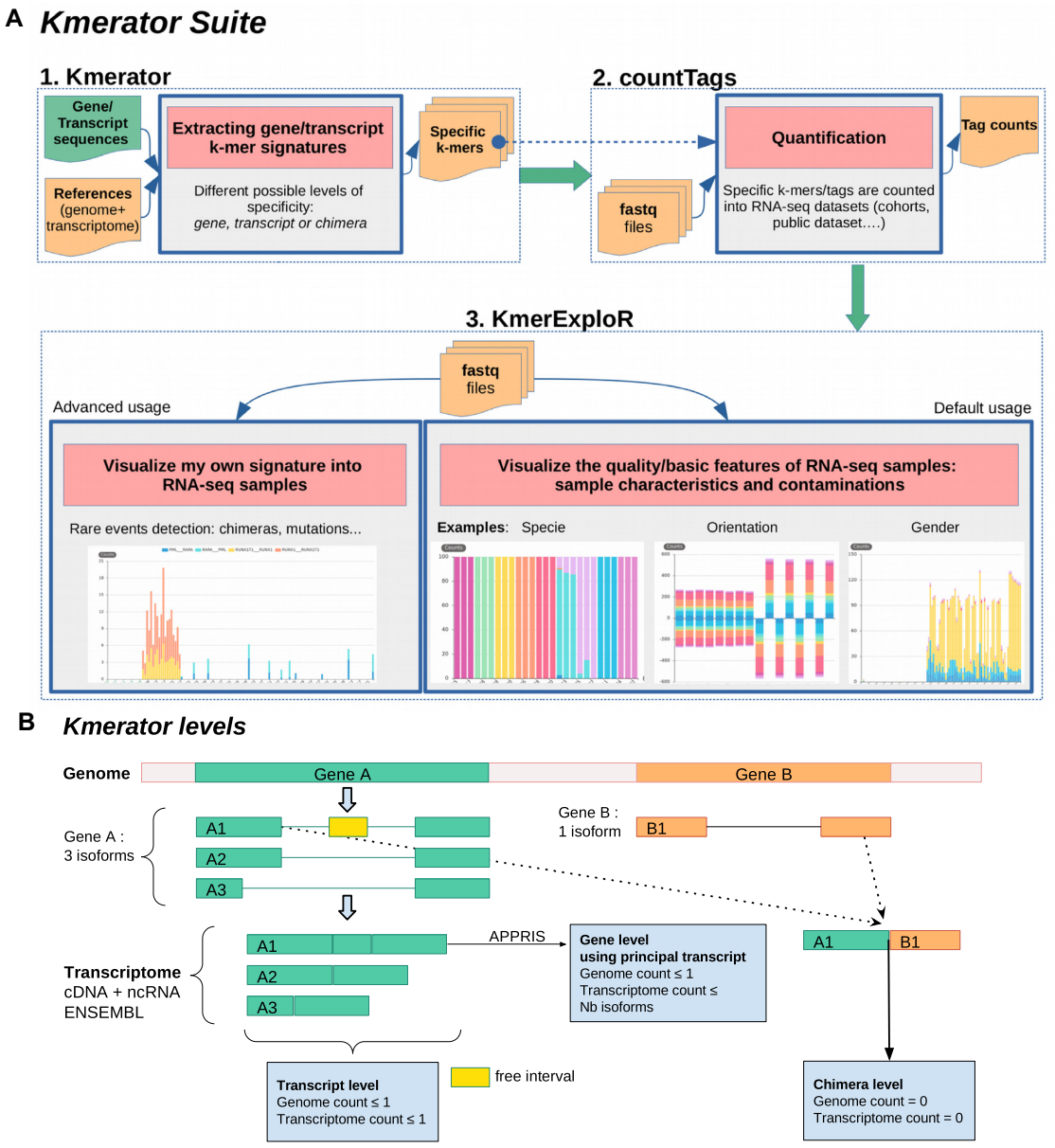


Figure 1. Kmerator Suite and Kmerator levels: definitions. **(A)** The Kmerator Suite is a set of three tools: (1) Kmerator extracts gene/transcript *k*-mer signatures. It takes as input a reference genome and a reference transcriptome + a list of gene or transcript sequences to extract specific *k*-mers from. The output is a set of fasta files (one per input gene/transcript sequence) with the specific *k*-mers. (2) countTags quantifies input *k*-mers in a set of input sequencing raw files (fastq files) and outputs a count table. (3) KmerExploR is a particular application of Kmerator/countTags to visualize input RNA-seq dataset (set of fastq files) characteristics. The default usage includes characteristics related to the sequencing protocol (ribosomal depletion, polyA+, strand-specific protocol, 5′/3′ bias), tissue origin (sex) and possible contaminations (mycoplasma, virus, other species or HeLa cell line). Users can also visualize their own signatures with the advanced usage. Details are given in the text and Supplementary Figure S1. **(B)** Kmerator extracts gene/transcript *k*-mer signatures with three possible levels of stringency. This figure describes how the different levels are defined (transcript, gene or chimera) for two example genes A and B. Example gene A has three isoforms: A1, A2 and A3. A1 is the only one with a free interval, i.e. a region not covered by other isoforms, and is defined as the principal transcript (APPRIS database). Therefore, at the transcript level, each transcript has its own specific *k*-mer set, depending on its coverage with other isoforms. At the gene level, the principal transcript defined with the APPRIS database is used, and specific *k*-mers can be common to several isoforms. At the chimera level (example of A1–B1 fusion), the *k*-mer is not described in annotations.

using the –appris option. APPRIS defines as the 'PRINCIPAL' isoform a CDS (coding sequence) variant for each gene, based on the range of protein features. When this option is not used or no principal sequence is given by APPRIS [i.e. for long non-coding RNA (lncRNA)], the isoform with the longest sequence is kept. In this study, we always used the gene level in combination with the –appris option.

Kmerator usage on the entire transcriptome for performance assessment. Kmerator was tested to extract *k*-mer signatures from the whole human Ensembl transcriptome (combination of cDNA and ncRNA fasta files, version 91). The Ensembl reference transcriptome was filtered to remove any transcript with alternate loci (labels with '_alt') and have been processed by Kmerator at both transcript (i.e. 199 181 transcripts) and gene (54 874 genes) levels with the –appris option previously described. At the transcript level, 62 transcripts have been ignored due to their length inferior to the *k*-mer length (31 nt). The processing to generate the specific *k*-mers on the whole transcriptome has been completed in <3 days at the gene level (88 003 855 *k*-mers) and 24 h at the transcript level (69 760 957 *k*-mers), using a LINUX server with 30 computing cores and 20 GB hard disk space. This step has to be done only one time for one chosen reference transcriptome. Once we have all the annotated transcript *k*-mer signatures, we can rapidly quantify them in any RNA-seq data.

K-mer counting and expression quantification

Simulated data. To test the precision of *k*-mer quantification, we created a set of 10 simulated RNA-seq data for which we have the exact counts. We first used the R *compcodeR* package (14) and the 'generateSyntheticData' function to simulate a count matrix with two conditions with five samples in each (samples.per.cond = 5). Each line of this matrix corresponds to a transcript of the Ensembl v91 annotation. Counts of transcripts with a length equal or inferior to 200 nt were not simulated. To highlight the quantification process, we increased the number of differentially expressed genes (n.diffexp = 10 000) with balanced over- and underexpressed fractions (fraction.upregulated = 0.5) and with authorized different dispersions between the conditions (between.group.diffdisp = TRUE, fraction.non.overdispersed = 0). Besides, we set the sequencing depth by RNA-seq file to 100 million reads (seq depth = 100 000 000) and we did not filter low counts (filter.threshold.total = 0). Providing this data frame and the Ensembl reference transcriptome, we used the 'simulate_experiment_countmat' function, from *polyester* R package (15), to generate paired-end and strand-specific (fr fashion) RNA-seq reads in fasta format. Finally, the fasta files have been converted to fastq.gz format using seqtk (https://github.com/lh3/seqtk).

countTags. *K*-mers designed by Kmerator on the whole transcriptome were counted into the 10 simulated RNA-seq data. For this purpose, the list of *k*-mers was submitted to countTags (https://github.com/Transipedia/countTags), a tool written in C language (see Figure 1A). countTags searches for short sequences (<32 nt) and their reverse complement with an exact match in fastq files and counts their occurrences. We used a *k*-mer length of 31 nt (–*k* 31) and the paired-end option (–paired), and we also used the countTags normalization option to normalize *k*-mer counts per billion of *k*-mers present in the dataset, using the –kbp option. As many specific *k*-mers are associated with one single transcript/gene, we computed the mean *k*-mer count by transcript/gene.

Comparison with Kallisto. We compared the Kmerator + countTags pipeline with Kallisto regarding the performances in transcript/gene expression quantification on simulated data detailed above. As our pipeline cannot quantify genes/transcripts without specific *k*-mers, we limited Kallisto quantification to the genes/transcripts having specific *k*-mers. Kallisto 0.43.1 (7) was run using the –fr-stranded option with the Ensembl v91 annotation file. For each pipeline, TPM (transcripts per million) counts were compared to true normalized TPM using the Spearman's correlation, either at the transcript level or at the gene level. Counts estimated by Kallisto were merged at the gene level by summing normalized transcript counts.

KmerExploR: exploring large RNA-seq datasets

KmerExploR is a command line tool powered by the backend pipeline Kmerator + countTags. KmerExploR provides *k*-mer quantification results in RNA-seq samples as a graphical and user-friendly html interface (see Figure 1A). To deal with data heterogeneity and the weaknesses of RNA-seq technology, we developed a turnkey application using KmerExploR. Characterization of a requested RNA-seq dataset can be improved with the quantification of selected genes (predictor genes) via the Kmerator + countTags pipeline. Predictor genes and their corresponding specific *k*-mers are included in KmerExploR and have been selected based on the literature to answer specific biological questions:

- Are my RNA-seq data based on polyA selection protocol or ribo-depletion?
- Are my RNA-seq libraries stranded or not?
- What is/are the sex corresponding to my samples?
- Is there a read coverage bias from 5′ to 3′ end along my dataset transcripts?
- Are my RNA-seq data contaminated by HeLa (presence of HeLa-derived human papillomavirus 18), mycoplasmas or other viruses such as hepatitis B virus?
- What is/are the species present in my samples?

Implementation. KmerExploR is a command line tool written in python 3. It can be installed on a server or on a personal computer from GitHub or with pip command (see https://github.com/Transipedia/kmerexplor). No additional modules are required. KmerExploR does not need a lot of memory and can be launched from a laptop. Indeed, for a common analysis of 36 paired-end samples (80 GB of fastq files), it takes 250 MB of memory (RAM per core) and 24 min. In comparison, the popular useful and complementary QC tool fastQC (https://qubeshub.org/resources/fastqc) takes 3300 MB of memory (RAM per core) and 15 min. KmerExploR includes countTags, described above.

From input fastq files, KmerExploR runs countTags, with a multithreading option, to quantify built-in *k*-mer selection associated with each predictor gene. The detailed diagram is shown in Supplementary Figure S1B. KmerExploR can also directly take countTags output files, as for large datasets it could be useful to separately run countTags on a cluster, for example. KmerExploR outputs an html file with css and javascript in separate files, using the echartsjs library to display user-friendly and graphical information (https://echarts.apache.org/en/index.html). Categories to show are described either in the built-in config file or in the user personal config file. KmerExploR also produces a tabulated text file with mean counts for each predictor gene in each category (rows) and in each sample (columns).

Predictor gene selection. We selected a subset of housekeeping genes from the list previously published by Eisenberg and Levanon (16) as well as some widely expressed histone genes that produce non-polyadenylated transcripts barely detected in polyA+ RNA-seq (see Table 1). We also selected specific genes from chromosome Y that have a ubiquitous expression, from Maan *et al.*'s publication (17). For these different sets of genes, we designed specific *k*-mers using Kmerator at the gene level and also computed the *k*-mer reversed complementary counterparts for the orientation category. Housekeeping genes' ubiquitous expression profile in various tissues, chromosome Y genes' specific expression pattern in male tissues and histone genes' low expression in polyA+ RNA-seq samples have been validated by exploring the GTEx database (https://www.gtexportal.org) (see Supplementary Figure S2).

For the detection of 5′/3′-end biases, we used the specific *k*-mers from ubiquitous genes (orientation set) and individually attributed them to their corresponding region, 5′ untranslated region (UTR), 3′ UTR or CDS, depending on their position in the principal transcript, according to the APPRIS database. For that purpose, we used Ensembl annotations with the biomaRt R package that gives the information of the UTR and CDS regions for each transcript. We searched the *k*-mers in transcript CDS and UTR sequences to label them by region. For mycoplasma tag selection, we first selected the most frequent mycoplasma found in cell contamination according to Drexler and Uphoff (18). We then downloaded ribosomal RNA (rRNA) sequences of the six selected mycoplasma species from the SILVA database v132 (19), which provides updated and curated rRNA sequences from Bacteria, Archaea and Eukaryota. Some species have several associated strains and therefore, several rRNA sequences. We have included them all for the *k*-mer design. For HeLa detection, we selected HPV-18 transcripts reported to be expressed in HeLa cells (20). Using UGENE software (21), we manually modified these transcripts to match the mutations reported as HeLa specific in the Cantalupo *et al.* study (20). We then defined sequences taking 30 nt on both sides of each mutation, before passing them to Kmerator to keep only *k*-mers not present in the human genome and transcriptome. For species identification, we selected those principally found in the SRA database. We then downloaded mitochondrially encoded cytochrome *c* oxidase I (MT-CO1) human gene sequence and its orthologs in each of the selected species, using the corresponding animal reference genome and transcriptome sequences (Ensembl v91 for each). Finally, sequences of virus genomes have been downloaded from RefSeq using the common virus list provided by Uphoff *et al.* (22). All these potential contamination sequences were used to produce specific *k*-mers using Kmerator at the chimera level, to select tags that can be found neither in the human reference genome nor in the transcriptome. For the advanced application of KmerExploR, we designed *k*-mers corresponding to new or rare transcriptional events detected in the Leucegene dataset (https://leucegene.ca/). For chimera detection, we used two well-known fusion RNA examples associated with chromosomal translocation and their reciprocal counterparts [RUNX1–RUNXT1 t(*x*,21) RUNXT1–RUNX1, PML–RARA t(15,17) and RARA–PML]. Specific *k*-mers are designed with Kmerator on 60 bp sequences spanning the junction. For mutation detection, we manually designed 31 bp *k*-mers centered on the mutation for reference and alternative sequences of three genes currently used in acute myeloid leukemia (AML) diagnosis: TET2, KRAS and CEBPA. We finally designed *k*-mers with Kmerator at the transcript level for a new lncRNA previously published in (23) as NONE 'chr2-p21'.

RNA-seq dataset. In this paper, we illustrated KmerExploR output on several datasets, depending on the biological question, all described in Supplementary Table S1. Characteristics related to RNA-seq protocol, which we call basic features, are tested on 103 paired-end samples from ENCODE (Dataset-FEATURES). For the contaminations part, we used the 33 single-read samples from the PRJNA153913 study (24) previously described as highly contaminated by mycoplasma (Dataset-MYCO) (25). We also selected three public RNA-seq samples by species to check the relevance of our species-specific *k*-mers (Dataset-SPECIES). HeLa contamination was tested in three cervical cancer CCLE (Cancer Cell Line Encyclopedia) cell lines: one HeLa and two negative controls (Dataset-HELA-CLE). Finally, for virus detection we used 19 samples from the CCLE dataset reported by Uphoff *et al.* (22) as contaminated by viruses and three control non-contaminated cell lines also included in the Uphoff *et al.* study (Dataset-VIRUS-CCLE).

RESULTS

Kmerator performances

To assess the Kmerator methodology, we first extracted *k*-mer signatures from all the human Ensembl transcriptome (i.e. 199 181 transcripts) and genes (i.e. 54 874 coding and non-coding genes). We were able to identify specific *k*-mers ($k = 31$ nt) for 83% of human transcripts and 97% of human genes as shown in Figure 2A and B.

This way, the transcriptome information has been almost entirely summarized by 69 760 957 *k*-mers at the transcript level and 88 003 855 *k*-mers at the gene level, corresponding to 23.8% and 30% of the total number of *k*-mers in the reference transcriptome, respectively. The attribution of specific *k*-mers at the gene and transcript levels is fundamentally different: whereas the gene level (–appris option) accepts specific *k*-mers shared with other isoforms, the transcript

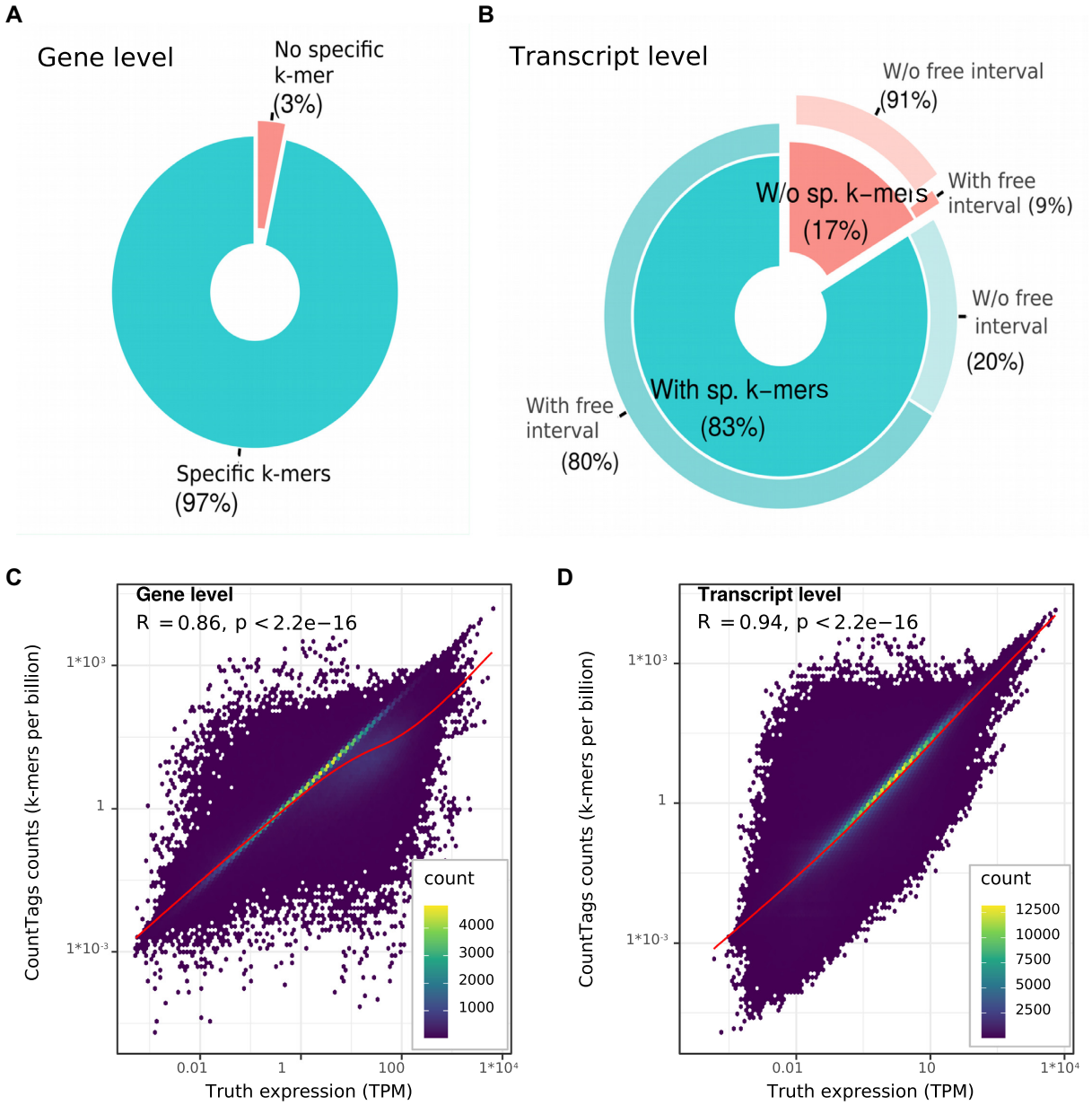


Figure 2. Kmerator performances on the whole transcriptome. We extracted *k*-mer signatures from all the human Ensembl transcriptome v91 at both gene (54 874 coding and non-coding genes, left) and transcript (i.e. 199 181 transcripts, right) levels. **(A)** The first pie chart represents the proportion of genes having specific *k*-mers (turquoise) versus those without specific *k*-mers (red). **(B)** In the same way, we represented the proportion of transcripts having specific *k*-mers (turquoise) or not (red). For these two classes, we looked at the percentage having free intervals, i.e. regions in the transcript not shared with other isoforms (secondary pie). Most of the transcripts lacking specific *k*-mers do not have free intervals (91%). We tested Kmerator sensitivity to quantify simulated data, at both gene **(C)** and transcript **(D)** levels. We represented the *k*-mer counts normalized per billion of *k*-mers in the sample (*Y*-axis) as a function of the true expression in TPM (*X*-axis), on the whole simulated dataset. *R* is the Spearman's correlation coefficient between *k*-mer counts and TPM. Each point on the graph is a transcript and the color scale depends on the transcript density on the graph.

level is more stringent and eliminates each *k*-mer shared by other ones. This explains the higher percentage of transcripts without specific *k*-mer compared to the gene level. To explain the absence of specific *k*-mers for some transcripts, we used BiomaRt genomic intervals to calculate the part of each transcript not covered by other isoforms, considering the strand, and named it 'free interval' (see Figure 1B). As expected, 91% of transcripts without specific *k*-mer have no 'free interval', which means that they are completely covered by other transcripts, thus confirming the validation of the Kmerator process. The set of specific *k*-mers designed with Kmerator strongly depends on the input sequence and on the level of selection. At the gene level, we observed that the length of the input sequence was corre-

lated with the number of designed specific *k*-mers ($R = 0.91$, $P < 2.2e{-}16$; see Supplementary Figure S3A) but not at the transcript level ($R = 0.22$, $P < 2.2e{-}16$; see Supplementary Figure S3B). On the contrary, the transcript level depends on the overlap between the input transcript and the different isoforms. A high number of isoforms is correlated to a low number of specific *k*-mers ($R = 0.79$, $P < 2.2e{-}16$; see Supplementary Figure S3C) and, in addition, the length of free intervals is strongly correlated to the number of specific *k*-mers ($R = 0.94$, $P < 2.2e{-}16$; see Supplementary Figure S3D). Finally, *k*-mer design differs between biotypes and selection levels: the biotypes without specific *k*-mers mainly correspond to small RNAs (miRNAs, rRNA) at the gene level (see Supplementary Figure S3E) and to coding and pseudo-genes at the transcript level (see Supplementary Figure S3F).

The Kmerator Suite has been designed as a new way to explore RNA-seq data and rapidly quantify some chosen sequences called predictors. Kmerator, the first key element of this suite, can extract unique *k*-mers from any sequence. In combination with countTags, it is used to generate large *k*-mer count tables. To situate our tool in relation to a widely used, referenced and benchmarked quantification tool, we tested the Kmerator + countTags pipeline accuracy to estimate gene and transcript expression using simulated data (see the 'Materials and Methods' section). Indeed, using a simulated dataset, for which we have the exact counts, even if it fails to capture the complexity of real data, is the best way to proceed to illustrate our purpose (26). We have run Kmerator and countTags to search for all human gene and transcript expression levels in a set of 10 simulated data. We assessed Spearman's correlation between normalized *k*-mer counts and the ground truth. We used countTags *k*-mer mean count per transcript reported to the total of *k*-mers contained in the input fastq. As shown in Figure 2, the Spearman's correlation factor comparing Kmerator + countTags results to the truth is 0.86 for the gene level (Figure 2C) and 0.94 for the transcript level (see Figure 2D), indicating a highly positive relationship with normalized counts ($P < 2e{-}16$).

Quantification results are comparable when using the Kallisto pseudo-alignment method, despite slightly higher correlation factors (gene and transcript $R = 0.97$; see Supplementary Figure S4A and B). This result is consistent with the recent paper describing Matataki (9), another quantification tool based on *k*-mers. Our pipeline being not specifically dedicated to gene quantification but for rapid exploration of large datasets is accurate enough to evaluate gene and transcript expression levels in RNA-seq data. Interestingly, the precision of Kallisto quantification decreases strongly with transcripts/genes not covered by Kmerator (see Supplementary Figure S4C and D), showing that each protocol using the *k*-mer principle struggles to correctly quantify sequences that do not possess distinctive *k*-mers.

Finally, we tested speed performance of countTags processing time on random subparts of sample simulated data (10 million, 101 nt paired-end reads), while increasing the number of quantified *k*-mers (1/1000/1 million). It appears that processing time remains low compared to alignment-based protocols (~1 min for 10 million reads) and depends on the number of *k*-mers quantified (see Supplementary Figure S4E). These results support an optimized usage of the Kmerator Suite protocol for its primary usage: the research of a limited number of signatures in large RNA-seq datasets.

KmerExploR for inspecting large RNA-seq datasets

We developed KmerExploR to improve the characterization of large RNA-seq datasets using the quantification of selected predictor genes. Predictor genes have been selected based on the literature to answer specific questions (see Table 1). As described in the 'Materials and Methods' section, we first extracted with Kmerator sets of specific *k*-mers from gene sequences and use KmerExploR to count the *k*-mer occurrences in RNA-seq datasets and visualize the results. Here, we present the results obtained with specific datasets (Table 1 and Supplementary Table S1) selected to highlight the rapid control of biological and technical parameters using KmerExploR. The results of the basic features, including sample sex, polyA or ribo-depletion, orientation and 5′/3′ bias, are presented in Figure 3.

As previously described, sample sex is determined by searching for *k*-mers corresponding to genes located on the Y chromosome. The *k*-mer signature clearly separates samples depending on the sex. To help the user classify his samples, we defined, in KmerExploR, a threshold of five *k*-mers per billion, above which we expect with confidence that it is a male. Moreover, Y chromosome gene expression variance between the samples can be explained by the variability of cell types and public RNA-seq experiment parameters, including sequencing depth and methods of RNA extraction and selection. For instance, the four male samples with the lowest expression (ENCFF232KGN, ENCFF434EMO, ENCFF831HCD and ENCFF992HBZ) come from a unique study (ENCSR999ZCI). However, the sex classification is more complicated in case of cancerous data. When we are looking at cancerous RNA-seq cell lines, some samples with male metadata show low Y chromosome-specific gene expression (data not shown). This extreme downregulation of chromosome Y gene expression has already been described in previous studies and strongly associated with cancer risk in men (27).

Gene abundance can be measured in RNA-seq data through sequencing of mRNA or ribo-depleted total RNA samples. The mRNA protocol relies on polyA selection, when the total RNA method is based on rRNA depletion (Ribozero protocol). However, non-polyadenylated transcripts should only be found in data produced using this procedure, when they should barely be detectable in mRNA samples. As the majority of histone transcripts are known to be non-polyadenylated, we used this characteristic first to detect sample contamination by non-polyadenylated RNA, and second to infer from the result the RNA preparation procedure. We first investigated the expression level of all histone genes and retained the most highly expressed according to the literature. Second, we analyzed their expression pattern using the GTEX resource. As RNA-seq from GTEX are exclusively produced from polyA selected RNA samples, we used this database to select histone genes showing the lowest expression levels (see Supplementary Figure S2B). We used this set of histone genes to test a se-

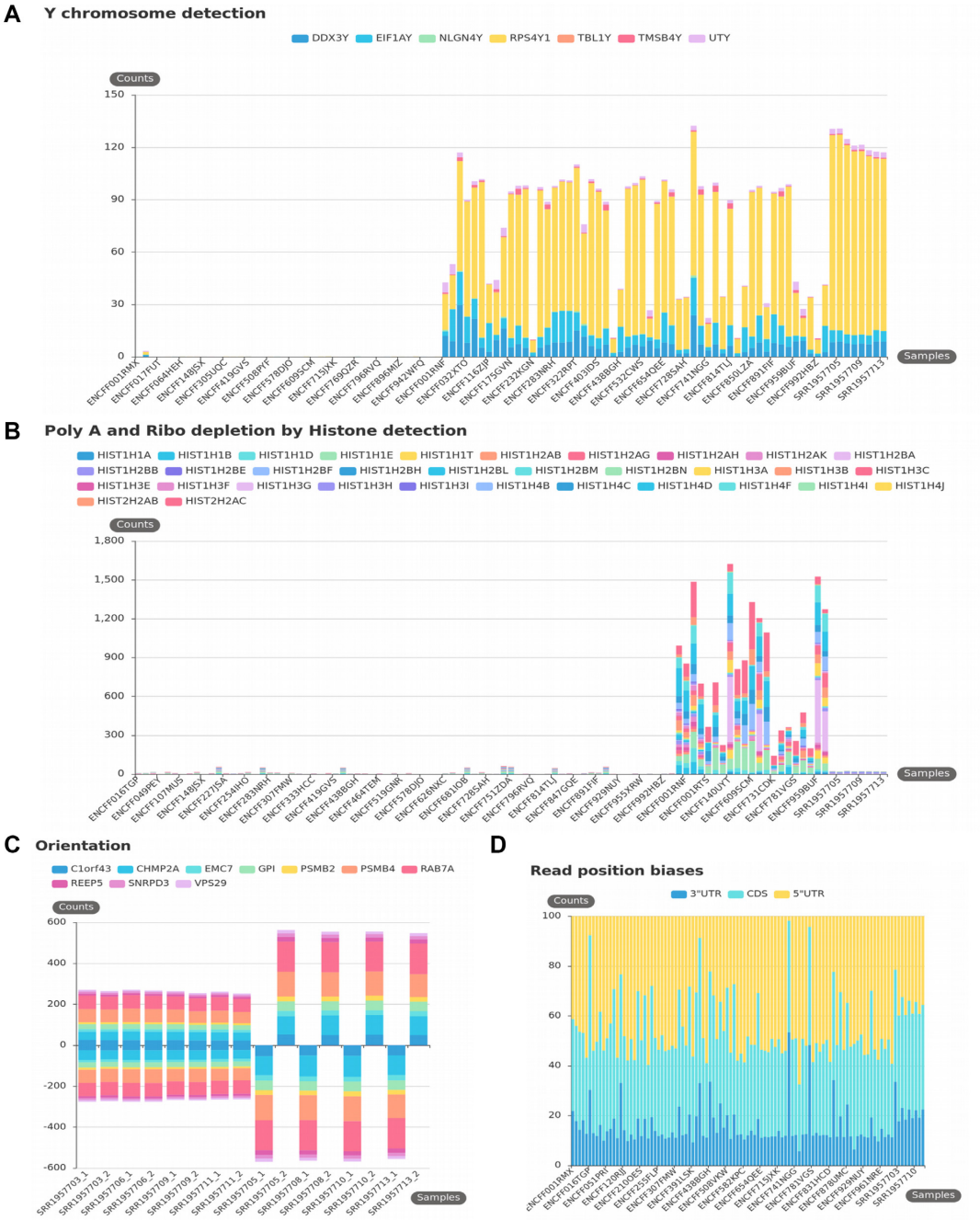


Figure 3. KmerExploR default usage: basic features. All presented bar plots are direct output of KmerExploR and they are generated from the Dataset-FEATURES described in Supplementary Table S1 (103 paired-end ENCODE samples) except for the orientation (C), which is a subset of eight RNA-seq from the Dataset-FEATURES. For each bar plot, the legend lists the set of predictor genes for which *k*-mer mean counts are computed (see also Table 1). Samples are on the *X*-axis. Panels (A), (B) and (C) have the mean *k*-mer counts by gene normalized per billion of *k*-mers on the *Y*-axis. **(A)** Sex determination. Samples are sorted by sex in the order female, then male. **(B)** PolyA+ selection versus ribo-depletion by histone detection. Samples are sorted by protocol in this order: polyA, ribo-depletion, unknown. **(C)** Stranded versus unstranded sequencing protocol. For this category, both fastq files by sample are shown. The first four samples are unstranded and the last four samples are stranded. **(D)** Read position biases along 5′ UTR, 3′ UTR and CDS regions. After computing *k*-mer mean counts by gene, they are summed up by 5′ UTR, 3′ UTR or CDS regions and converted in % (*Y*-axis).

Table 1. List of predictor genes, by category, included in KmerExploR and associated RNA-seq dataset names used in this paper

	Datasets	Predictor genes	Total *k*-mer number	References and details
PolyA/RiboD	Dataset-FEATURES	HIST2H2AC, HIST2H2AB, HIST1H4J, HIST1H4I, HIST1H4F, HIST1H4D, HIST1H4C, HIST1H4B, HIST1H3I, HIST1H3H, HIST1H3G, HIST1H3F, HIST1H3E, HIST1H3C, HIST1H3B, HIST1H3A, HIST1H2BN, HIST1H2BM, HIST1H2BL, HIST1H2BH, HIST1H2BF, HIST1H2BE, HIST1H2BB, HIST1H2BA, HIST1H2AK, HIST1H2AH, HIST1H2AG, HIST1H2AB, HIST1H1T, HIST1H1E, HIST1H1D, HIST1H1B, HIST1H1A	24 512	Supplementary Figure S2
Orientation	Dataset-FEATURES	VPS29, SNRPD3, REEP5, RAB7A, PSMB4, PSMB2, GPI, EMC7, CHMP2A, C1orf43, VPS29_rev, SNRPD3_rev, REEP5_rev, RAB7A_rev, PSMB4_rev, PSMB2_rev, GPI_rev, EMC7_rev, CHMP2A_rev, C1orf43_rev	36 638	Supplementary Figure S2 (16)
Sex	Dataset-FEATURES	UTY, TMSB4Y, TBL1Y, RPS4Y1, NLGN4Y, EIF1AY, DDX3Y	21 996	Supplementary Figure S2 (17)
5′/3′ bias	Dataset-FEATURES	VPS29, SNRPD3, REEP5, RAB7A, PSMB4, PSMB2, GPI, EMC7, CHMP2A, C1orf43	12 705	Supplementary Figure S2 (16)
Mycoplasma	Dataset-MYCO	Mycoplasma_orale, Mycoplasma_hyorhinis, Acholeplasma_laidlawii, Mycoplasma_hominis, Mycoplasma_arginini, Mycoplasma_fermentans	363 025	(18)
Virus	Dataset-VIRUS-CCLE	Human_gammaherpesvirus_4, Human_herpesvirus_4, Human_herpesvirus_8, Murine_leukemia_virus, Hepatitis_C_virus_genotype, Human_immunodeficiency_virus_1, Human_T_lymphotropic_virus_1, Squirrel_monkey_retrovirus, Human_T_lymphotropic_virus_2, Human_papillomavirus_type_92, Hepatitis_B_virus_strain, Human_immunodeficiency_virus_2, MuLV_related_virus_22Rv1/CWR, Bovine_viral_diarrhea_virus	516 882	(22)
HeLa	Dataset-HELA-CCLE	L1_mut7486, L1_mut7258, L1_mut6842, L1_mut6625, L1_mut6460, L1_mut6401, L1_mut5875, E7_mut806, E7_mut751, E6_mut549, E6_mut485, E6_mut287, E6_mut104, E1_mut2269, E1_mut1994, E1_mut1843, E1_mut1807, E1_mut1353, E1_mut1012	589	(20)
Species	Dataset-SPECIES	Homo_sapiens_MT_CO1, Danio_rerio_mt_co1, Zea_mays_COX1, Saccharomyces_cerevisiae_COX1, Rattus_norvegicus_Mt_co1, Mus_musculus_mt_Co1, Gallus_gallus_MT_CO1, Drosophila_melanogaster_mt_CoI, Caenorhabditis_elegans_ctc_3_MTCE, Arabidopsis_thaliana_COX1	12 119	MT-CO1 (and orthologs)
Chimeras	Dataset-LEUCEGENE	PML–RARA, RARA–PML, RUNX1T1–RUNX1, RUNX1–RUNX1T1	724	
lncRNA	Dataset-LEUCEGENE	NONE	78	(23)
Mutations	Dataset-LEUCEGENE	TET2, KRAS, CEBPA	10	

The samples included in each dataset and some metadata are detailed in Supplementary Table S1.

lection of ENCODE samples that metadata indicates either polyA or ribo-depletion protocol (Supplementary Table S1). The results clearly demonstrate differences between libraries prepared by ribo-depletion versus polyA selection for most of the chosen histone genes. We observe histone gene expression variability between the samples demonstrating again the disparity of public data. To help users categorize their RNA-seq data, we defined in the KmerExploR tool a threshold of 200 *k*-mer counts per billion for this category, above which we expect to have only the ribo-depleted samples and not the polyA ones.

Strand-specific and unstranded library preparation are two commonly used preparation protocols that differ by their ability to retain or not RNA strand information. To detect this characteristic from RNA-seq data, we designed *k*-mers, specific for a set of ubiquitous genes (Table 1) and their reverse complement counterparts. *K*-mers on the forward strand are counted as positive and their reverse complement as negative, permitting to determine the orientation of the library. If forward and reverse tags are found in equivalent proportions in the same fastq file, data are considered as 'unstranded'. This leads graphically to a balanced distribution between positive and negative counts. As shown in Figure 3, using this property we are able to clearly separate unstranded and stranded libraries. 5′ to 3′-end bias is a difference of reads' repartition along the tran-

scripts, classically linked to library preparation: incomplete retrotranscription or specific protocols. A comparison between polyA selection and ribo-depletion protocols has previously shown coverage differences across transcripts with a poor 5′-end coverage with the polyA selection method (28). Knowing whether an RNA-seq sample possesses a read repartition bias is critical for isoform detection, or simply to give an indication on the library construction protocol used in large-scale analysis of public data. Using previously described housekeeping genes (Table 1), we have selected different sets of specific *k*-mers depending on their position in the regions defined as 5′ UTR, 3′ UTR and CDS. Figure 3C shows the repartition in percent of these *k*-mers across the Dataset-FEATURES samples. Representing the mean *k*-mer counts as a percentage allows us to evaluate the distribution homogeneity across 5′ UTR, 3′ UTR and CDS regions between the 103 ENCODE samples. This global representation grouping together several genes allows us to identify samples for which one region has a very little coverage. Here, four samples have <10% 5′ UTR coverage (ENCFF734ZAD, ENCFF770NYA, ENCFF419GVS and ENCFF016TGP). We can also notice a better homogeneity of coverage for ribo-depleted samples.

Detection of potential contamination

Different microorganisms like mycoplasma and virus can contaminate samples and cell cultures, modifying the metabolism of the cell and therefore biasing the results of ensuing analysis. Moreover, cancer research has shown that viruses are responsible for ∼20% of human cancers (29). To detect contaminants in RNA-seq data, tools relying on alignment like DecontaMiner (30) or viGEN (31) have been widely used, but the alignment step is time and memory consuming. Exact alignment of *k*-mer-based approaches like Kraken (32) and Taxonomer (33) is an alternative for taxonomic classification. However, these tools are complex and involve data cleaning from adaptors (trimming), the use of internal and external databases and/or probabilistic models for contaminant classification. Using a specific and reduced set of *k*-mers, we have seen an advantage to quickly detect principal contaminants of human cells in RNA-seq datasets, free from alignment methods.

Because mycoplasma is a common source of cell culture sample contamination and could affect host gene expression (25), we choose to control its presence in RNA-seq data. Mycoplasma contamination is evaluated through the detection of specific *k*-mers corresponding to 16S rRNA sequences according to the literature. In fact, Olarerin-George and Hogenesch showed that 90% of the specific mycoplasma-mapped reads from human RNA-seq samples mapped to mycoplasma rRNA. We selected six species that have the highest record rate of detection in cell culture samples (i.e. *Acholeplasma laidlawii*, *Mycoplasma fermentans*, *Mycoplasma hominis*, *Mycoplasma hyorhinis*, *Mycoplasma orale* and *Mycoplasma arginini*) (18) to design our *k*-mers. We used part of an RNA-seq data series previously described as highly contaminated (25) (PRJNA153913 study) to test the relevance of our approach. As shown in Figure 4, we can easily detect the six selected mycoplasma species in some samples, with a prevalence for the *M. hyorhinis* species. Comparing our results with the Olarerin-George and Hogenesch study that used Bowtie 1 alignment and BLAST+ to filter non-specific reads, we were able to confirm mycoplasma rRNA presence for the same samples (see Supplementary Figure S5A). Moreover, we observe a high proportionality between our *k*-mer counts and their read counts on the 33 single-read samples (Dataset-MYCO described in Supplementary Table S1), for each of the six common *Mycoplasma* species.

Viruses are a significant cause of human cancers. Several studies interrogate for the presence of major viruses known to infect human and other mammalian cells (22,34,35). Recently, Uphoff *et al*. screened >300 CCLE RNA-seq data using the Taxonomer interactive tool and compared the results to virus-specific polymerase chain reaction (PCR) analysis, revealing 20 infected cell lines with different viruses (22). To rapidly explore the potential presence of viruses in RNA-seq datasets with our *k*-mer-based approach, we used the same virus reference genomes as described in the Uphoff *et al*. study. Using Kmerator at the chimera level (absent from human annotations), we designed specific *k*-mers for each virus and searched them in a subset of contaminated CCLE data according to Uphoff *et al*. (19 CCLE paired-end samples) and in negative controls (3 CCLE paired-end samples), to validate our protocol ability to detect viruses. Among the contaminated samples, we were able to detect the main viruses in the same samples as in the Uphoff *et al*. study, except for the SRR8615677 sample where we do not detect any virus, as the bovine polyomavirus is not included in our list of common viruses. Our results are shown in Figure 4B and Taxonomer results from the Uphoff *et al*. study are presented in Supplementary Figure S5B. Epstein–Barr virus (EBV) is a very common virus detected in most of the samples; we have therefore analyzed it in more detail in Supplementary Figures S5C (our approach) and S5D (Taxonomer quantification). Indeed, our EBV quantification is correlated with the one from Taxonomer (Pearson's and Spearman's correlation coefficients are 0.99 and 0.89, respectively).

HeLa is the first immortal human cell line, coming from Henrietta Lacks' cancerous tissue samples. Her cancer was triggered by an infection with human papillomavirus type 18 (HPV-18). Nowadays, this cell line is largely used in medical research. Looking for several viruses in public RNA-seq cancer-related databases revealed the presence of HPV-18 sequences in many cancers (36) that closely resemble the HPV-18 viral sequence that is integrated into HeLa cells, suggesting a contamination. Three segments of HPV-18 are integrated into the HeLa genome on chromosome 8 and include the long control region, the E6, E7 and E1 genes, and partial coding regions for the E2 and L1 genes (20). These genes are expressed in HeLa cells, and mutations have been found specifically in HeLa cells. Thus, selecting these mutated HeLa HPV-18 gene-specific *k*-mers and counting them into three CCLE RNA-seq datasets (one positive sample and two negative controls), we validated the accuracy of our selection as we are able to find our *k*-mer selection specifically in HeLa cells. We also checked the results in other HeLa samples from the PRJNA639358 study (see Supplementary Figure S5E).

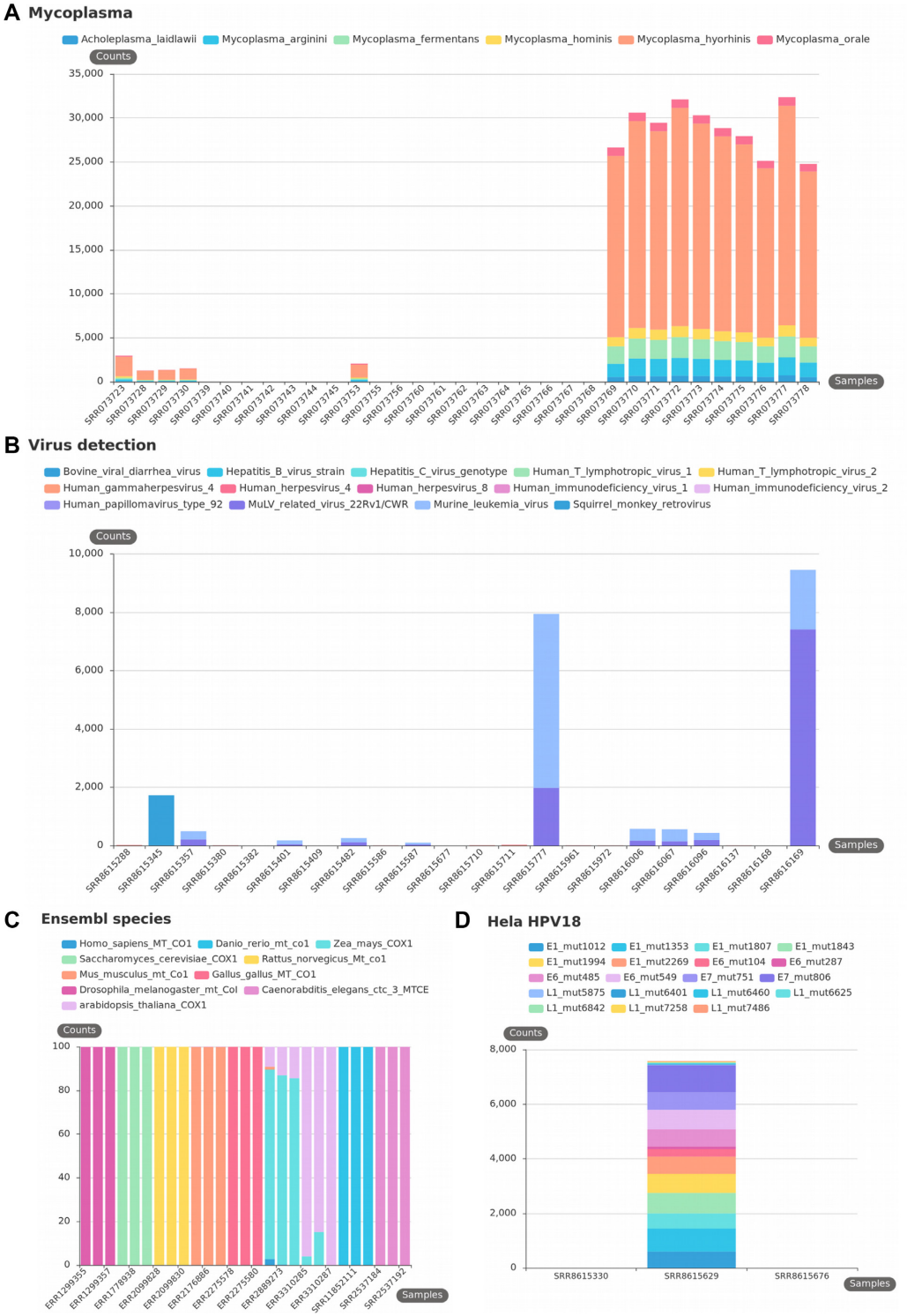


Figure 4. KmerExploR default usage: contaminations. All presented bar plots are direct output of KmerExploR and all bar plot datasets are described in Supplementary Table S1. For each bar plot, the legend lists the set of predictors for which *k*-mer mean counts are computed (details in Table 1). Samples are on the *X*-axis. Panels (A), (B) and (D) have the mean *k*-mer counts by gene normalized per billion of *k*-mers on the *Y*-axis. **(A)** Mycoplasma contamination on the Dataset-MYCO (33 single-read samples). **(B)** Virus detection on the Dataset-VIRUS-CCLE (22 paired-end samples). **(C)** Species determination on the Dataset-SPECIES (27 paired-end samples). For this category, after computing *k*-mer mean counts by species, they are converted in % (*Y*-axis) to avoid big expression differences between species. **(D)** HeLa determination on the Dataset-HELA-CCLE (three paired-end samples). The sample in the middle is a HeLa cell line and the two others are negative controls (SF767 and SiHa cells).

As for HeLa cells, cross-species contamination remains a documented 'danger' for the interpretation of results in molecular biology (37). The probability of mixed cell lines in sample preparation, usage of PCR that can accidentally amplify the wrong piece of DNA, and an unknown probability of error in metadata assignation motivated us to create a quality check to determine the species of an RNA-seq sample. In (38), the usage of mitochondrial DNA for phylogenetic and taxonomic inference was discussed and two extreme viewpoints emerged: using exclusively the mitochondrial DNA or fully excluding it. It appears that mitochondrial DNA does not fully answer or impairs the perspectives of advanced phylogenetics. However, the 'mitochondrial barcode' approach does show an interesting gene marker, MT-CO1 (39), that could be sufficient for a quick check of the species of RNA-seq data. Indeed, this gene is highly expressed and reference sequences from many distinct species of animals are available. Thus, we selected specific *k*-mers with Kmerator, at the gene level, for MT-CO1. We repeated the procedure for MT-CO1 orthologs in different species, principally found in the SRA database, using the appropriate species reference genome and transcriptome. These *k*-mers have been then quantified in three public data by species to check the efficiency of their usage. As shown in Figure 4C, the research of MT-CO1 *k*-mers alone can discriminate most of the common Ensembl species and can be usable for a quick quality check. However, without proper experiments we cannot support its usage with phylogenetically close species.

To conclude, we developed KmerExploR to rapidly control RNA-seq raw data quality and filter samples on unusual profiles or presence of contaminations. KmerExploR is a tool that provides a modular set of analyses like fastQC (https://qubeshub.org/resources/fastqc). It can be used in a complementary way to fastQC analysis to complete missing metadata in public datasets or to give a quick profile of the RNA-seq contents. The modular analysis is based on a *k*-mer selection from predictor genes, included in KmerExploR. The tool can be used to control any human RNA-seq dataset, and it can also be easily modified adding any other modular function.

KmerExploR, an advanced usage for the detection of genomic or transcriptomic events

The above 'checking application' of KmerExploR demonstrated all its potential in the rapid exploration of large public RNA-seq datasets before performing any biological query. However, the KmerExploR tool can also be used in a more advanced way such as biomarker search or discovery in human health. This application is a powerful one as it can compensate for the lack of completeness in genomic or transcriptomic references and we currently know that much important information may be missed by ignoring the unreferenced RNA diversity (12). As a proof of concept, we used a set of *k*-mers designed with Kmerator to identify events outside reference annotations including fusion or chimeric RNA, oncogene mutations and new lncRNA expression. We then applied *k*-mer quantification in a tumoral and a non-tumoral dataset to evaluate the specificity and performance of the approach. The results obtained with a part of the Leucegene cohort are presented in Figure 5.

The selection includes different AML subtypes and normal $CD34^+$ cells as control (Dataset-LEUCEGENE described in Supplementary Table S1). The results obtained with two well-known fusion RNAs associated with chromosomal translocation, RUNX1–RUNXT1 t(x,21) and PML–RARA t(15,17), and their reciprocal counterparts RUNXT1–RUNX1 and RARA–PML are presented in Figure 5A. In this case, the *k*-mers, once designed by Kmerator, are restricted to those spanning the fusion junction with at least 10 nucleotides in gene 1 or gene 2 of the fusion. All the normal $CD34^+$ cells are negative and we only observe an expression in corresponding positive AML subtypes. Figure 5B illustrates the results obtained for mutations in TET2, KRAS and CEBPA genes currently used in AML diagnosis. Once again, we only observe the presence of these mutations in positive samples, demonstrating the high specificity of the approach by *k*-mers. The expression of a new lncRNA was also quickly searched in the Leucegene dataset (see Figure 5C); we observe a homogeneous and low expression in CD34 normal cells compared to a heterogeneous one in AML subtypes. This lncRNA candidate was already described in (23), using for the first time the '*k*-mer concept' for checking new biomarker candidates, and we have demonstrated a restricted expression of the NONE 'chr2-p21' lncRNA in the hematopoietic lineage using the Leucegene and ENCODE datasets. Hence, for lncRNA candidates, following their discovery in a tissue/disease type, their specificity could be easily evaluated through quantification in a wide range of RNA-seq data including normal and pathological conditions as recently described by Riquier *et al*. (40).

In conclusion, the high specific expression of transcriptional events may lead them to be used as biomarkers for biological and health applications, including cell therapy, diagnosis, prognosis or patient follow-up as it is already done with fusion RNAs and mutations.

DISCUSSION

Considering the growing number of RNA-seq data, the use of raw data sequences is an important step to check with RNA-seq protocols or bioinformatic pipelines bias. Here, we demonstrated that the Kmerator Suite is an efficient and useful set of tools to verify RNA-seq quality and control intrinsic method and biological characteristics that often failed in technical description. We also showed that the Kmerator Suite can be used to quantify gene/transcript-specific expression as well as to explore sequence variations at the transcriptional level. In this first version, the tool is adapted to human data Ensembl entry, as main public data are available for this species (164 000 RNA-seq with >30 million reads for *Homo sapiens* in the SRA database). A new implementation with adapted predictors is necessary for other species.

The meta-analyses performed in the present study with KmerExploR are a proof of concept of the procedure potential and could be extended to other biological RNA-seq questioning: (i) to extend the application to an enlarged set of microorganisms including new ones like SARS-Cov2

Figure 5. KmerExploR advanced usage: quantification of transcriptomic events outside the annotations. All presented bar plots are direct output of KmerExploR and they are all generated from the Dataset-LEUCEGENE described in Supplementary Table S1 (131 paired-end samples). This dataset includes normal CD34$^+$ cells as control (in green on the *X*-axis) and different AML subtypes (in black on the *X*-axis). For each bar plot, the legend lists the set of predictors for which *k*-mer mean counts normalized per billion (*Y*-axis) are computed. **(A)** Chimera detection. Two well-known fusion RNAs associated with chromosomal translocation and their reciprocal counterparts are presented: RUNX1–RUNXT1 t(*x*,21) and PML–RARA t(15,17). **(B)** Mutation detection. TET2, KRAS and CEBPA genes are used in AML diagnosis. The bar plot shows four different mutations for these genes, detected specifically in some AML samples. The reference allele for each of these mutations is detected in almost all samples. **(C)** New lncRNA detection: NONE 'chr2-p21' lncRNA described in (23). This transcript is expressed in the whole dataset but shows different levels of expression depending on AML subtype.

detection and (ii) to search for immunophenotyping profile in cancer datasets as already published by Mangul *et al*. (41,42). Considering advanced applications, we also demonstrated the potential of *k*-mers to explore gene expression in RNA-seq to reinforce biological questions or biomarker usage and discovery. Moreover, many other requests could be easily considered for annotated gene exploration like gene co-expression, or to compensate the lack of completeness in genomic or transcriptomic references to cover unreferenced RNA diversity and search for new spliced events, intron retention or new transcript categories including circular RNAs. In order to increase the potential of the *k*-mer approach, access to very large-scale datasets like SRA level (164 000 human samples) could be considered with efficient indexing structure development (43).

Finally, we showed that the Kmerator Suite can be used to quantify gene/transcript expression as well as to explore sequence variations at the transcriptional level. The simplicity of specific *k*-mer extraction principle and quantification provide flexibility of usage. Indeed, Kmerator Suite quantification does not use probabilistic methods or expectation–maximization algorithms like in Kallisto (7), Sailfish (44) or RNA-Skim (45). Therefore, the sets of specific *k*-mers for quantification can be created, merged and updated at will, without consequence on the quantification itself. The principle of user-owned collection of signatures of interest that can be searched broadly among datasets is the core of KmerExploR application.

DATA AVAILABILITY

RNA-seq libraries were downloaded from the European Nucleotide Archive of the European Bioinformatics Institute (46). The reference GRCh38 genome and Ensembl v91 transcripts were downloaded from Ensembl. Kmerator is distributed under the MIT license. The Kmerator, KmerExploR and countTags software, documentation and supplementary material presented herein are available from https://github.com/Transipedia/kmerator, https://github.com/Transipedia/kmerexplor and https://github.com/Transipedia/countTags, respectively.

SUPPLEMENTARY DATA

Supplementary Data are available at NARGAB Online.

ACKNOWLEDGEMENTS

The authors are grateful to Henrietta Lacks, now deceased, and to her surviving family members for their contributions to biomedical research. The HeLa cell line that was established from her tumor cells without her knowledge or consent in 1951 has made significant contributions to scientific progress and advances in human health. The authors are also grateful to Rayan Chikhi for his comments and corrections.

Author contributions: S.R. and T.C. designed the study. S.R., C.B. and T.C. wrote the manuscript. A.-L.B., N.G. and D.G. were contributors in the design of the study and manuscript corrections. S.R. developed the code of Kmerator, selected and downloaded public datasets, and analyzed data. B.G. and C.B. participated in Kmerator code improvements. C.B. analyzed RNA-seq data and generated figures. B.G. developed KmerExploR code, and generated *k*-mer counting and figures. J.A. and A.B. computed and corrected countTags. F.R. validated the RNA-seq data for mutation and chimeric RNAs, and helped in the interpretation of results. H.X. participated in Kmerator testing and checking. All authors read and approved the final manuscript.

FUNDING

Agence Nationale de la recherche [ANR-10-INBS-09]; Canceropole Grand Ouest [2017-EM24]; Region Occitanie [R19073FF].

Conflict of interest statement. None declared.

REFERENCES

1. Collado-Torres,L., Nellore,A., Kammers,K., Ellis,S.E., Taub,M.A., Hansen,K.D., Jaffe,A.E., Langmead,B. and Leek,J.T. (2017) Reproducible RNA-seq analysis using recount2. *Nat. Biotechnol.*, **35**, 319–321.
2. Byron,S.A., Van Keuren-Jensen,K.R., Engelthaler,D.M., Carpten,J.D. and Craig,D.W. (2016) Translating RNA sequencing into clinical diagnostics: opportunities and challenges. *Nat. Rev. Genet.*, **17**, 257–271.
3. Xi,X., Li,T., Huang,Y., Sun,J., Zhu,Y., Yang,Y. and Lu,Z.J. (2017) RNA biomarkers: frontier of precision medicine for cancer. *Non-Coding RNA*, **3**, 9.
4. Hippen,A.A. and Greene,C.S. (2020) Expanding and remixing the metadata landscape. *Trends Cancer*, **7**, 276–278.
5. Dobin,A., Davis,C.A., Schlesinger,F., Drenkow,J., Zaleski,C., Jha,S., Batut,P., Chaisson,M. and Gingeras,T.R. (2013) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics*, **29**, 15–21.
6. Philippe,N., Salson,M., Commes,T. and Rivals,E. (2013) CRAC: an integrated approach to the analysis of RNA-seq reads. *Genome Biol.*, **14**, R30.
7. Bray,N.L., Pimentel,H., Melsted,P. and Pachter,L. (2016) Near-optimal probabilistic RNA-seq quantification. *Nat. Biotechnol.*, **34**, 525–527.
8. Patro,R., Duggal,G., Love,M.I., Irizarry,R.A. and Kingsford,C. (2017) Salmon provides fast and bias-aware quantification of transcript expression. *Nat. Methods*, **14**, 417–419.
9. Okamura,Y. and Kinoshita,K. (2018) Matataki: an ultrafast mRNA quantification method for large-scale reanalysis of RNA-seq data. *BMC Bioinformatics*, **19**, 266.
10. Yu,Y., Liu,J., Liu,X., Zhang,Y., Magner,E., Qian,C. and Liu,J. (2018) SeqOthello: querying RNA-seq experiments at scale. *Genome Biol.*, **19**, 167.
11. Audoux,J., Philippe,N., Chikhi,R., Salson,M., Gallopin,M., Gabriel,M., Le Coz,J., Drouineau,E., Commes,T. and Gautheret,D. (2017) DE-kupl: exhaustive capture of biological variation in RNA-seq data through *k*-mer decomposition. *Genome Biol.*, **18**, 243.
12. Morillon,A. and Gautheret,D. (2019) Bridging the gap between reference and real transcriptomes. *Genome Biol.*, **20**, 112.
13. Marçais,G. and Kingsford,C. (2011) A fast, lock-free approach for efficient parallel counting of occurrences of *k*-mers. *Bioinformatics*, **27**, 764–770.
14. Soneson,C. (2014) compcodeR: an R package for benchmarking differential expression methods for RNA-seq data. *Bioinformatics*, **30**, 2517–2518.
15. Frazee,A.C., Jaffe,A.E., Langmead,B. and Leek,J.T. (2015) Polyester: simulating RNA-seq datasets with differential transcript expression. *Bioinformatics*, **31**, 2778–2784.
16. Eisenberg,E. and Levanon,E.Y. (2013) Human housekeeping genes, revisited. *Trends Genet.*, **29**, 569–574.
17. Maan,A.A., Eales,J., Akbarov,A., Rowland,J., Xu,X., Jobling,M.A., Charchar,F.J. and Tomaszewski,M. (2017) The Y chromosome: a blueprint for men's health? *Eur. J. Hum. Genet.*, **25**, 1181–1188.

18. Drexler,H.G. and Uphoff,C.C. (2002) Mycoplasma contamination of cell cultures: incidence, sources, effects, detection, elimination, prevention. *Cytotechnology*, **39**, 75–90.
19. Yilmaz,P., Parfrey,L.W., Yarza,P., Gerken,J., Pruesse,E., Quast,C., Schweer,T., Peplies,J., Ludwig,W. and Glöckner,F.O. (2014) The SILVA and 'All-species Living Tree Project (LTP)' taxonomic frameworks. *Nucleic Acids Res.*, **42**, D643–D648.
20. Cantalupo,P.G., Katz,J.P. and Pipas,J.M. (2015) HeLa nucleic acid contamination in The Cancer Genome Atlas leads to the misidentification of human papillomavirus 18. *J. Virol.*, **89**, 4051–4057.
21. Okonechnikov,K., Golosova,O. and Fursov,M. and UGENE Team (2012) Unipro UGENE: a unified bioinformatics toolkit. *Bioinformatics*, **28**, 1166–1167.
22. Uphoff,C.C., Pommerenke,C., Denkmann,S.A. and Drexler,H.G. (2019) Screening human cell lines for viral infections applying RNA-seq data analysis. *PLoS One*, **14**, e0210404.
23. Rufflé,F., Audoux,J., Boureux,A., Beaumeunier,S., Gaillard,J.-B., Bou Samra,E., Megarbane,A., Cassinat,B., Chomienne,C., Alves,R. *et al.* (2017) New chimeric RNAs in acute myeloid leukemia. *F1000Res.*, **6**, https://doi.org/10.12688/f1000research.11352.2.
24. Prensner,J.R., Iyer,M.K., Balbin,O.A., Dhanasekaran,S.M., Cao,Q., Brenner,J.C., Laxman,B., Asangani,I.A., Grasso,C.S., Kominsky,H.D. *et al.* (2011) Transcriptome sequencing across a prostate cancer cohort identifies PCAT-1, an unannotated lincRNA implicated in disease progression. *Nat. Biotechnol.*, **29**, 742–749.
25. Olarerin-George,A.O. and Hogenesch,J.B. (2015) Assessing the prevalence of mycoplasma contamination in cell culture via a survey of NCBI's RNA-seq archive. *Nucleic Acids Res.*, **43**, 2535–2542.
26. Mangul,S., Martin,L.S., Hill,B.L., Lam,A.K.-M., Distler,M.G., Zelikovsky,A., Eskin,E. and Flint,J. (2019) Systematic benchmarking of omics computational tools. *Nat. Commun.*, **10**, 1393.
27. Cáceres,A., Jene,A., Esko,T., Pérez-Jurado,L.A. and González,J.R. (2020) Extreme downregulation of chromosome Y and cancer risk in men. *J. Natl Cancer Inst.*, **112**, 913–920.
28. Cui,P., Lin,Q., Ding,F., Xin,C., Gong,W., Zhang,L., Geng,J., Zhang,B., Yu,X., Yang,J. *et al.* (2010) A comparison between ribo-minus RNA-sequencing and polyA-selected RNA-sequencing. *Genomics*, **96**, 259–265.
29. McLaughlin-Drubin,M.E. and Munger,K. (2008) Viruses associated with human cancer. *Biochim. Biophys. Acta*, **1782**, 127–150.
30. Sangiovanni,M., Granata,I., Thind,A.S. and Guarracino,M.R. (2019) From trash to treasure: detecting unexpected contamination in unmapped NGS data. *BMC Bioinformatics*, **20**, 168.
31. Bhuvaneshwar,K., Song,L., Madhavan,S. and Gusev,Y. (2018) viGEN: an open source pipeline for the detection and quantification of viral RNA in human tumors. *Front. Microbiol.*, **9**, 1172.
32. Wood,D.E., Lu,J. and Langmead,B. (2019) Improved metagenomic analysis with Kraken 2. *Genome Biol.*, **20**, 257.
33. Flygare,S., Simmon,K., Miller,C., Qiao,Y., Kennedy,B., Di Sera,T., Graf,E.H., Tardif,K.D., Kapusta,A., Rynearson,S. *et al.* (2016) Taxonomer: an interactive metagenomics analysis portal for universal pathogen detection and host mRNA expression profiling. *Genome Biol.*, **17**, 111.
34. Cao,S., Strong,M.J., Wang,X., Moss,W.N., Concha,M., Lin,Z., O'Grady,T., Baddoo,M., Fewell,C., Renne,R. *et al.* (2015) High-throughput RNA sequencing-based virome analysis of 50 lymphoma cell lines from the Cancer Cell Line Encyclopedia Project. *J. Virol.*, **89**, 713–729.
35. Cantalupo,P.G., Katz,J.P. and Pipas,J.M. (2018) Viral sequences in human cancer. *Virology*, **513**, 208–216.
36. Selitsky,S.R., Marron,D., Hollern,D., Mose,L.E., Hoadley,K.A., Jones,C., Parker,J.S., Dittmer,D.P. and Perou,C.M. (2020) Virus expression detection reveals RNA-sequencing contamination in TCGA. *BMC Genomics*, **21**, 79.
37. Ballenghien,M., Faivre,N. and Galtier,N. (2017) Patterns of cross-contamination in a multispecies population genomic project: detection, quantification, impact, and solutions. *BMC Biol.*, **15**, 25.
38. Rubinoff,D. and Holland,B.S. (2005) Between two extremes: mitochondrial DNA is neither the panacea nor the nemesis of phylogenetic and taxonomic inference. *Syst. Biol.*, **54**, 952–961.
39. Hebert,P. D.N., Cywinska,A., Ball,S.L. and deWaard,J.R. (2003) Biological identifications through DNA barcodes. *Proc. R. Soc. Lond. B: Biol. Sci.*, **270**, 313–321.
40. Riquier,S., Mathieu,M., Bessiere,C., Boureux,A., Ruffle,F., Lemaitre,J.-M., Djouad,F., Gilbert,N. and Commes,T. (2021) Long non-coding RNA exploration for mesenchymal stem cell characterisation. *BMC Genomics*, **22**. 412.
41. Mangul,S., Yang,H.T., Strauli,N., Gruhl,F., Porath,H.T., Hsieh,K., Chen,L., Daley,T., Christenson,S., Wesolowska-Andersen,A. *et al.* (2018) ROP: dumpster diving in RNA-sequencing to find the source of 1 trillion reads across diverse adult human tissues. *Genome Biol.*, **19**, 36.
42. Mandric,I., Rotman,J., Yang,H.T., Strauli,N., Montoya,D.J., Van Der Wey,W., Ronas,J.R., Statz,B., Yao,D., Petrova,V. *et al.* (2020) Profiling immunoglobulin repertoires across multiple human tissues using RNA sequencing. *Nat. Commun.*, **11**, 3126.
43. Marchet,C., Boucher,C., Puglisi,S.J., Medvedev,P., Salson,M. and Chikhi,R. (2021) Data structures based on *k*-mers for querying large collections of sequencing datasets. *Genome Research*, **31**, 1–12.
44. Patro,R., Mount,S.M. and Kingsford,C. (2014) Sailfish enables alignment-free isoform quantification from RNA-seq reads using lightweight algorithms. *Nat. Biotechnol.*, **32**, 462–464.
45. Zhang,Z. and Wang,W. (2014) RNA-Skim: a rapid method for RNA-seq quantification at transcript level. *Bioinformatics*, **30**, i283–i292.
46. Silvester,N., Alako,B., Amid,C., Cerdeño-Tarrága,A., Clarke,L., Cleland,I., Harrison,P.W., Jayathilaka,S., Kay,S., Keane,T. *et al.* (2018) The European Nucleotide Archive in 2017. *Nucleic Acids Res.*, **46**, D36–D40.

Chapter 7

Arbitrary Sequence Query in RNA-seq Data

7.1 Motivation and contribution

We define here "arbitrary sequence query" as the ability to find and quantify unannotated, arbitrary RNA or DNA sequence in raw NGS sequence files (see section 2.3.5 Inter-cohort query of k-mer signals). In k-mer analysis, this task is essential for inter-cohort query of k-mer or k-mer contig counts, analogical to the much simpler query of gene/transcript expressions using their universal IDs. In a biomedical perspective, there is a large unmet need for identifying unannotated, disease-related transcripts in the vast RNA-seq repositories.

The major difficulty of this task is that the sequence in query has arbitrary length and arbitrary order of nucleotides, resulting in infinite possibility space. Therefore, the sequences have no unified ID across cohorts, rendering both indexing and querying difficult, especially when abundance query is required. *REINDEER* [Marchet et al., 2020] is a pioneer software that solves this by using monotigs as index and query element. Our goal here is to validate the accuracy of *REINDEER*'s query results.

My task in this chapter was to assess the capacity of k-mer level queries to determine if the arbitrary sequence query estimates the abundance of a target RNA transcript as well as conventional quantification tools. I did the comparison between *REINDEER* query and *Kallisto-tximport* quantification, and analyzed the comparison results.

This work is part of the ongoing *ANR TranSiPedia* project conducted in collaboration with Chloé Bessière, Benoît Guibert, and Thérèse Commes at Université de Montpelier, who contributed to the study by generating data and taking part in result analysis, and Camille Marchet, Mikaël Salson and Rayan Chikhi at Institut

Pasteur/Université de Lille who developed the *REINDEER* software and helped in our understanding and application of it.

7.2 The *REINDEER* software

REINDEER stands for REad Index for abuNDancE quERy. It is designed for arbitrary sequence query in RNA-seq data sets without predefined reference, allowing for returning the sequence abundance in addition to their presence/absence state [Marchet et al., 2020]. The software includes two stages. (i) index generation: establishing multi-sample index to record k-mer sequences and counts; (ii) sequence query: associate the given arbitrary sequence with a list of monotig counts.

REINDEER index integrates three steps:

- Step 1: construct compacted DBG using *BCalm2* software [Chikhi et al., 2016], sample by sample;
- Step 2: create a union DBG graph of all samples;
- Step 3: associate count values for each **monotig** (k-mers from the a DBG path with identical count vectors and sharing a same minimizer).

REINDEER query decomposes each query sequence into its constituent k-mers, and query them individually in the constructed index. It outputs a list of monotig counts for each query of sequence and for each sample, such as "0-30:4,31-34:*,35-41:4,42-42:*,43-49:4". Each triplet "b-e:q" corresponds to a monotig in *Bcalm2* [Chikhi et al., 2016] index, meaning that the quantification from b^{th} to e^{th} k-mers equals to q. The "*" symbol means that between the position interval of contig in query, there is not enough k-mers (threshold set by the -P argument, see paragraph below) presented in the index for reporting a count value.

REINDEER offers a possibility to indicate a minimum percentage of findable k-mers in indexed monotigs. This is controlled by -P p with p being a number between 0 and 100, i.e., if a contig or one of its substring does not have at least p% of k-mers presented on the same monotig, the query is unsuccessful and a "*" symbol is returned. This parameter allows balancing precision and recall. Too high a value (e.g. near 100) may drop some queries since it does not well tolerate missing k-mers caused by mismatches/gaps. On the contrary, too low a value (e.g. near 0) may introduce much noise, since a single k-mer can be encountered at an independent locus by coincidence. In section 7.3.5 Analysis of *REINDEER* recall, we will test the effect of this parameter.

7.3 *REINDEER* query assessment

7.3.1 Test dataset

We selected RNA-seq from 12 Cancer Cell Line Encyclopedia (CCLE) [Barretina et al., 2012] lung cancer cell lines listed below.

SRR8615893	SRR8615897	SRR8615898	SRR8615899
SRR8615900	SRR8615901	SRR8615904	SRR8615905
SRR8615944	SRR8616205	SRR8616206	SRR8616217

Sequencing was performed with *Illumina HiSeq 2500* in paired-end and non-stranded mode. Samples were processed with *Cutadapt* [Martin, 2011] to trim low-quality bases at either end of sequences (-q 10,10) and exclude those shorter than 31 nt (k-mer length) after trimming (-m 31).

7.3.2 General idea of quantification assessment

To assess the accuracy of *REINDEER* counts for RNA quantification, we compared results with quantifications produced by *Kallisto* [Bray et al., 2016], as follows.

Query sequences We selected 1,000 random genes from *ENSEMBL* release 99. We used *Kmerator* [Riquier et al., 2021] (see chapter 6) to extract gene-specific contigs for each gene (--level gene), respectively with $k = 31$ and $k = 21$. *Kmerator* was able to detect specific k-mers/contigs for 856/1,000 genes (5260 contigs) with $k = 31$, and for 855/1,000 genes (17001 contigs) with $k = 21$.

REINDEER index construction We constructed the *REINDEER* index from the 12 CCLE samples. k-mers occurring less than twice in one sample were ignored (parameter of *BCalm2* [Chikhi et al., 2016]). Two indexes were built with $k = 21$ and $k = 31$.

REINDEER query and result parsing Contigs processed by *Kmerator* were queried by *REINDEER* on the constructed index. As mentioned above, each contig is associated with a series of triplets "b-e:q". We interpreted these counts by calculating mean, median, mode, min, max, and sum, across constituent k-mer counts. All contigs from the same gene were considered together.

Comparison with *Kallisto-tximport* *Kallisto* (version 0.46.1) [Bray et al., 2016] quantification was performed on the same 12 CCLE samples after *Cutadapt* trimming with the *ENSEMBL* release 99 reference transcriptome. Gene-level raw counts and TPM were computed from transcript-level counts using *tximport* [Soneson et al., 2016].

7.3.3 Different interpretation of *REINDEER* results

A basic rule is, if *REINDEER*'s query results are compared to **estimated counts**, the interpreted values should be compared to gene quantification directly, without normalization; whereas if compared to **TPM**, a normalization step is required on these values, for being coherent with the TPM definition. For the latter comparison, we applied a simple scaling normalization, i.e., dividing the count values by each sample's total k-mer count and then multiplied by a scaling factor 10^9. The total k-mer counts was estimated by the *countTags* tool [Riquier et al., 2021] (`https://github.com/Transipedia/countTags`).

Raw *REINDEER* counts vs. *Kallisto-tximport* est-counts Different interpretations of raw *REINDEER* query results were compared to *Kallisto-tximport* estimated counts (Figure 7.1). Interpretation "sum" yield the best correlation with *Kallisto-tximport*'s raw counts (Pearson: 0.818, Spearman: 0.896).

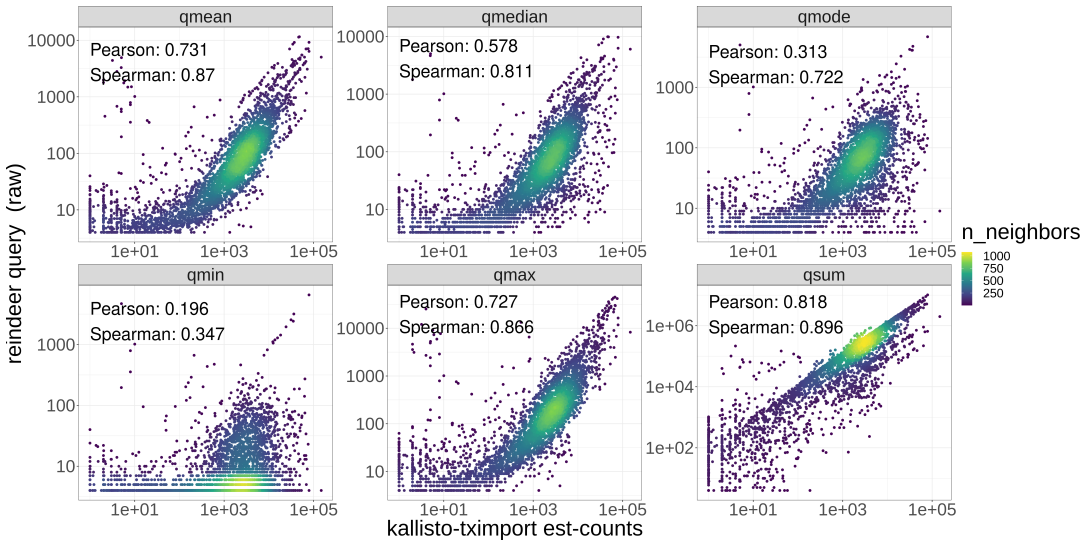


Figure 7.1: Correlation of different *REINDEER* raw query interpretations with *Kallisto-tximport* estimated counts. Each point represents a (gene, sample) pair. Outlier values below 10^{-5} either by *REINDEER* or *Kallisto-tximport* are removed.

Scaled *REINDEER* counts compared to *Kallisto-tximport* TPM The different interpretations of scaled *REINDEER* counts were compared to *Kallisto-tximport* TPM (Figure 7.2). Interpretation "max" gave the best correlation (Pearson=0.858, Spearman=0.912).

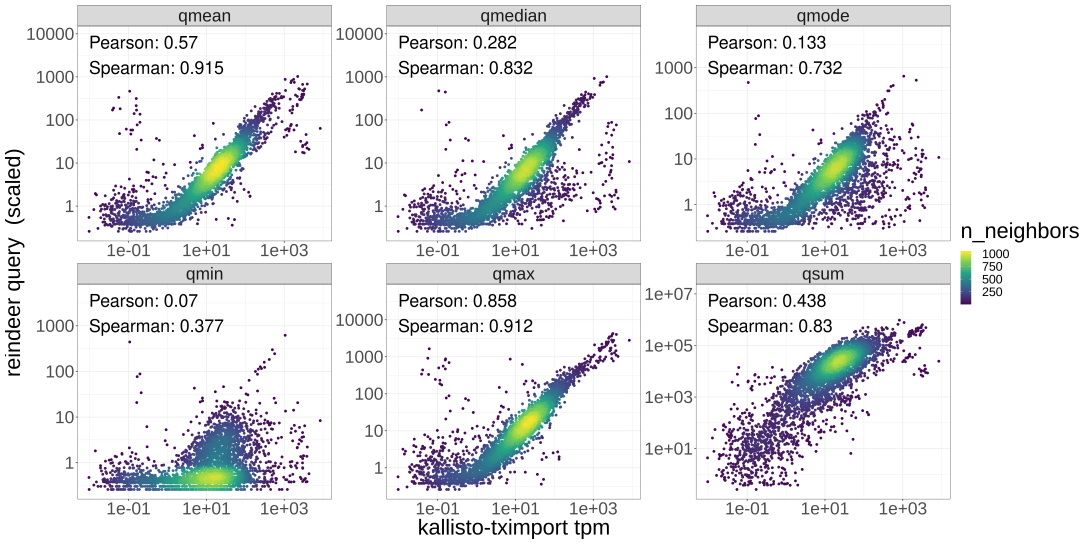


Figure 7.2: Correlation of different *REINDEER* scaled query interpretations with *Kallisto-tximport* TPM. Each point represents a (gene, sample) pair. Outliers with values below 10^{-5} either in *REINDEER* or *Kallisto-tximport* were removed.

In summary, when considering raw counts, the sum interpretation correlates best to conventional quantification, whereas when normalized counts are considered, the max interpretation performs best. This corresponds to our intuition, since a gene's estimated count measures the number of all reads mapped to it, while the TPM measures an average coverage along the gene. For the following validation tasks, we opt to use the sum interpretation and non-normalized counts, as this avoids transformations external to *REINDEER per se*.

7.3.4 Multi-linear relationship between sum interpretation of query vs estimated count of quantification

Here I examine in more detail the linear correlation between the sum interpretation of *REINDEER* query and *Kallisto-tximport* quantification.

Figure 7.3 compares *REINDEER* "sum" counts with *Kallisto-tximport*'s estimated counts, without log transformation on axis scales. The result indicates that

there may exist multiple linear patterns among the points. Fitting the boundary signal gave a slope of around 127.

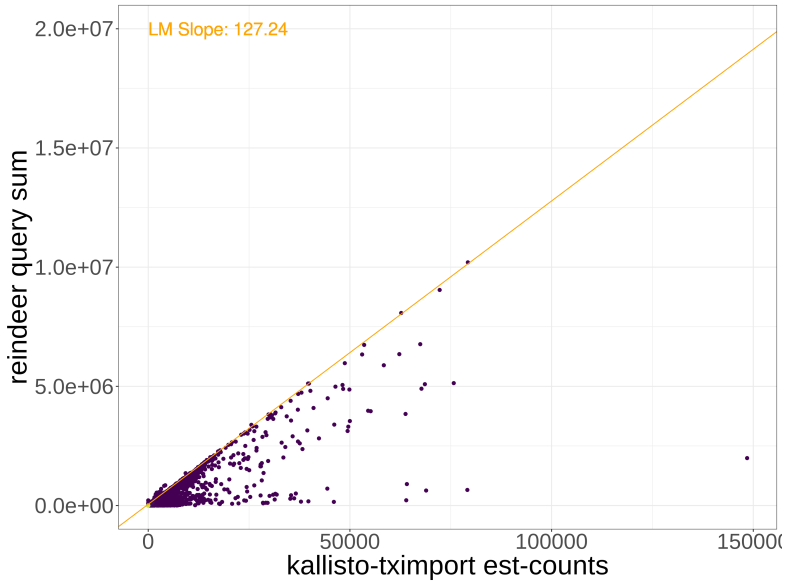


Figure 7.3: Scatter plot of *REINDEER*'s sum interpretation and *Kallisto-tximport*'s estimated counts. Each point is a gene-sample pair. Only values ≥ 1 by both methods are shown.

Here I demonstrate that there are actually multiple linear signals inside Figure 7.3, which are in fact related with different genes (see also Figure 7.4).

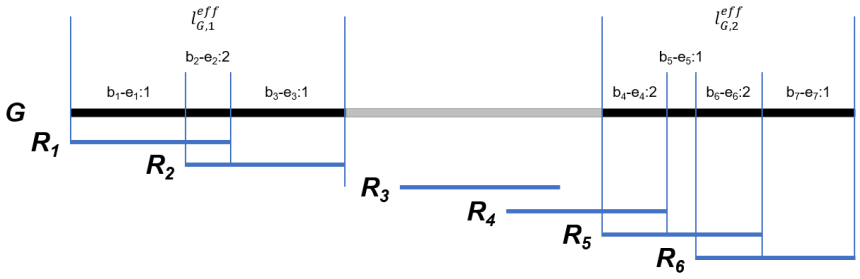


Figure 7.4: A simple artificial example *REINDEER* result of a gene G in query. This gene has two specific parts reserved by *Kmerator* (black ones), and one non-specific part excluded from the query (gray one). Lengths of the two reserved parts denote respectively $l^{eff}_{G,1}$ and $l^{eff}_{G,2}$. $\{R_1, R_2, ..., R_6\}$ are six reads related to G, and b_j-e_j:$q_j, j = 1, 2, ..., 7$ denotes 7 monotigs returned by *REINDEER* query with their counts q_j

Proof.

Let us consider a gene G with length l_G associated with reads $\{R_1, R_2, ..., R_{n(s)}\}$ in a sample s. The read R_i has length $l_i(s), i = 1, 2, ..., n(s)$, and the mean length of all reads is $\bar{l}$. We suppose that $\bar{l}$ is not dependent on sample. In the example of Figure 7.4, $n(s) = 6$.

Suppose the gene G is indexed as m monotigs with samples. These monotigs are only associated with the specific parts (retrieved by *Kmerator*) of the gene sequence, with lengths $l^{eff}_{G,w}$, $w = 1, 2, ..., m$. In Figure 7.4, the specific parts are marked as two black lines, respectively with length $l^{eff}_{G,1}$ and $l^{eff}_{G,2}$; also, $m = 7$.

Suppose query results of the gene G in the sample s across the m monotigs are $\{b_j\text{-}e_j{:}q_j(s)\}$, $j = 1, 2, ..., m$.

The sum interpretation of the monotigs' counts of this gene, denote as $Q_G(s)$, is actually the total number of constituent k-mers from reads (or sub-part of reads) mapped to the specific parts of the gene.

$$\begin{aligned} Q_G(s) &= \sum_{j=1}^{m} (e_j - b_j + 1) \cdot q_j(s) \\ &\approx \sum_{r \in \{r^{eff}(s)\}} (l_r - k + 1) \end{aligned} \tag{7.1}$$

, where $\{r^{eff}(s)\}$ is the read and sub-part read set that mapped to the specific part of the gene G, in the sample s. In the example of Figure 7.4, $\{r^{eff}(s)\} = \{R_1, R_2, R_4, R_5, R_6\}$. Approximation is due to the partially covered reads (R_4 in Figure 7.4).

Admitting an even read coverage, we can continue the deduction as shown in equation 7.2.

$$\begin{aligned} Q_G(s) &\approx \sum_{r \in \{r^{eff}(s)\}} (l_r - k + 1) \\ &= \frac{\sum_w \left(l^{eff}_{G,w} - k + 1 \right)}{l_G - k + 1} \sum_{i}^{n(s)} (l_i(s) - k + 1) \\ &= \frac{\sum_w \left(l^{eff}_{G,w} - k + 1 \right)}{l_G - k + 1} \cdot n(s) \cdot (\bar{l} - k + 1) \\ \frac{Q_G(s)}{n(s)} &= \frac{\sum_w \left(l^{eff}_{G,w} - k + 1 \right)}{l_G - k + 1} (\bar{l} - k + 1) \end{aligned} \tag{7.2}$$

On the other hand, the estimated count of G by *Kallisto-tximport* actually equals to $n(s)$.

Therefore, the sum interpretation of *REINDEER* query is linearly correlated with *Kallisto-tximport* estimated count, with the slope $\frac{\sum_w \left(l^{eff}_{G,w} - k + 1\right)}{l_G - k + 1}(\bar{l} - k + 1)$ dependent on the gene but not on sample. □

To confirm our theoretical deduction, we fitted the scatter plots gene by gene, and summarized the adjusted R-squares and slopes among genes (Figure 7.5).

Most genes present a linear relationship between the sum interpretation of *REINDEER* and *Kallisto-tximport* estimated counts, and with different fitted slopes. This is coherent with our deduction in equation 7.2. However, in Figure 7.5B, a group of fitted slopes are above $(\bar{l} - k + 1)$ which should not happen according to our deduction. A potential explanation is related to *REINDEER*'s indexing strategy with paired-end reads. Further investigation are needed on this point at the time of writing.

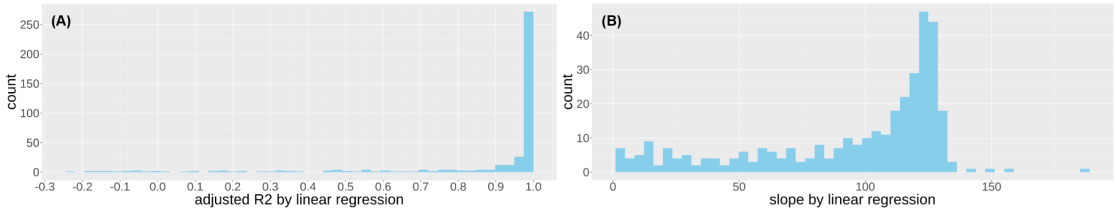


Figure 7.5: Histograms of linear fitting between sum of *REINDEER* query results and *Kallisto-tximport* raw quantification. (A) Fitted adjusted R-square values among genes. Only gene-sample with values ≥ 1 by both methods are considered, and only genes with at least 5 samples remained are fitted and summarized as an histogram. (B) Fitted slopes among genes. In addition to the criteria in panel (A), only slopes with R-square value ≥ 0.6 are summarized in the histogram.

7.3.5 Analysis of *REINDEER* recall

We analyzed *REINDEER* recall for $k = 31$ and the sum interpretation of *REINDEER* query. We examined different values of *REINDEER*'s parameter -P that specifies the minimum ratio of k-mers in a monotig to be found by the query for reporting a count value (by default, $P = 40$).

Table 7.1 presents several indicators: (i) the percentage of dropped result by *REINDEER*, which is defined as the ratio of number of query returned as "*" by *REINDEER* over the number of results returned as a positive count by at least one software (*REINDEER*, *Kallisto*, or both); (ii) and (iii): respectively Pearson and Spearman correlations between *REINDEER* and *Kallisto* counts.

As expected, increasing P results in more dropouts by *REINDEER*. In the strictest situation where all k-mers of a monotig are required for reporting the count

(-P 100), 65% of queries were missed by *REINDEER*. With the most permissive P, where any k-mer found in a monotig is sufficient for reporting a count, 20% queries were missed. We remind here that only gene-specific k-mers are used for *REINDEER*, but not for *Kallisto* counting. Some genes' specific parts may not have enough read coverage to be queried.

Interestingly, changing P of query did not much impact correlations with *Kallisto-tximport* quantification; these stayed around 0.82 (Pearson) and 0.79-0.9 (Spearman). Thus a low value of P can be selected to guarantee reasonable quantification accuracy while controlling the dropout ratio.

Table 7.1: *REINDEER* dropout and accuracy (k=31)

	P = 0	**P = 40**	**P = 70**	**P = 100**
Dropout ratio	20%	30%	37%	65%
Kallisto correlation (Pearson)	0.82	0.82	0.81	0.84
Kallisto correlation (Spearman)	0.91	0.9	0.88	0.79

7.3.6 Effect of *k*-mer length

To assess the effect of k on *REINDEER* performance, we performed the same tests as above with $k = 21$ (Table 7.2).

Table 7.2: *REINDEER* dropout and accuracy (k=21)

	P = 0	**P = 40**	**P = 70**	**P = 100**
Dropout ratio	15%	22%	28%	34%
Kallisto correlation (Pearson)	0.79	0.78	0.78	0.81
Kallisto correlation (Spearman)	0.91	0.9	0.89	0.87

The dropout was lower with $k = 21$ than with $k = 31$ for any value of P. The biggest difference (from 65% to 34%) was observed at $P = 100$. Still, at the most permissive case ($P = 0$), there are still 15% of genes missed. Pearson correlation was generally lower with $k = 21$ than with $k = 31$, but still stable across different values of parameter P. Spearman correlation was improved with $P = 100$ (0.87 with $P = 21$ *versus* 0.79 with $k = 31$).

The lower dropout ratio with $k = 21$ relative to $k = 31$ implies that a smaller value of k improves *REINDEER*'s query sensibility. This may relate to two facts: (i) On the index end, smaller k allows indexing at "higher resolution", since k-mers are the constructing elements of the index. (ii) On the query end, smaller k makes *Kmerator* output more fragmented for query (i.e. from 5260 contigs to 17001 contigs, see section 7.3.2 General idea of quantification assessment). At a same

value of P, more fragmented query contigs are more likely to have a query value reported, since fewer k-mers are required to be present in the indexed monotigs.

7.4 Concluding remarks

We showed here that reference-free arbitrary sequence query with *REINDEER* provides a relatively precise and sensitive estimation of gene expression, especially under the "sum of all monotigs" interpretation. We also provided a tentative theoretical proof to support this good correlation, albeit with some limitation.

Tuning the "minimum percentage of k-mers found in a monotig for reporting a count" parameter (-P) did not much affect correlations with *Kallisto* counts. However, at high values of -P, the combination *Kmerator-REINDEER* missed some genes that *Kallisto-tximport* was able to quantify. This dropout ratio can be as high as 65% when $P = 100$ with *Kmerator*'s $k = 31$.

An index of higher resolution can be achieved by using smaller k, helpful for reducing dropout ratio with a given value of parameter -P. However, using shorter k-mers may increase count variability and make queries less correlated to standard quantification.

An important caveat in this analysis is that, while our validation was made at gene level, *REINDEER per se* is not intended to quantify whole gene expression. In tests of transcript-level queries, *REINDEER* was not as accurate as at the gene level. A possible explanation is that *REINDEER* transcript-level counts do not use Expectation-Maximization for assigning reads to isoforms and thus is not able to resolve alternative transcripts correctly. *Kmerator* preprocessing to remove k-mers shared among isoforms did not improve accuracy. Overall this observation is to some extent coherent with reports that gene-level quantification is considerably more accurate than transcript-level quantification [Soneson et al., 2016].

Chapter 8

Discussion

8.1 Summary of Thesis Discoveries

8.1.1 General logic of the thesis

This thesis proposes and discusses a novel method for transcriptome analysis based on directly analyzing k-mer count signals, which yield insights into the sub-transcript level at single-nucleotide resolution.

Chapter 1 presents the basic notions and concepts used in the thesis, basically from the perspectives of biology (gene, gene expression, transcriptome, sequencing technologies, and cancer), statistics and data science, and computer science (data structures and data simulation).

Chapter 2 summarizes transcriptome analysis methods. This chapter divides the methods basically to two groups: (i) the conventional one including mapping-based quantification and *de novo* transcript assembly, and (ii) the emerging one directly analyzing k-mer count signals. For both groups, the chapter generally follows the logic of feature construction - informative feature extraction - inter-cohort feature querying.

Chapter 3 presents our new software *KaMRaT*, supporting various processing of k-mer count signals. It integrates different strategies for k-mer dimensionality reduction, including:

- a filter module considering their expression level;
- a masking module to extract k-mers with a given sequence list;
- an extension module merging overlapped k-mers;

- a ranking-selecting module by evaluating association between sample counts and conditions.

Chapter 4 explores and evaluates the application of k-mer contig signals on classifier construction for prostate cancer prognosis. It conducts a fair comparison between this emerging type of classifiers with the conventional one based on gene-expression profiles, guaranteeing that the two workflows are as similar as possible.

Chapter 5 conducts an inter-cohort analysis of k-mer contig signals, and examines the replicability of these signals across relevant but independent cohorts.

Chapter 6 explores the possibility of associating reference-free k-mer features to the conventional reference-based gene features with *Kmerator* software, and using these k-mers as a proxy to fast quantify interested features and detect specific signals in RNA-seq data, using *countTags*.

Chapter 7 analyses the relevance of k-mer counts as a proxy for quantifying arbitrary RNA expression. Tests are performed with the REINDEER program, in the context of a broader study of REINDEER applications.

Annex 1 presents a minor contribution to an article describing application of k-mer signal analysis to DNA-seq data.

8.1.2 Advantage of transcriptome analysis with k-mer count signals

Differing from the conventional methods - mapping-based transcript/gene quantification or *de novo* transcript assembly, the emerging direct k-mer analysis approach provides a new possibility for measuring the occurrence of local RNA variations at single nucleotide resolution.

The foundation of k-mer based transcriptome analysis is to represent gene-expression events using short k-mers (typically, with $k \leq 31$). This can be followed by different processing methods, such as evaluation of association between k-mer counts and conditions across samples, and construction of classifiers upon this.

k-mer signals allow a reference-free transcriptome analysis at single nucleotide resolution, and permits capturing events in RNA-seq samples comprehensively. k-mers can represent arbitrary sequences, limited neither by predefined gene or transcript sets, nor by patterns based on prior knowledge (e.g., bubbles in DBG representing SNP, indel, and splicing event). Direct analysis on individual k-mer counts

prevent informative signals from cancelling each other during gene/transcript quantification stage.

Furthermore, as a reference-free method, k-mer count analysis helps enhance the replicability of studies. k-mer signals are insensitive to version changes of reference databases from year to year. Therefore, the findings of k-mer approaches should be more stable than those from conventional gene/transcript expressions. [Lorenzi, 2021]

RNA-seq is predominantly carried out using short sequencing technologies that produce reads which are k-mers by nature. There are mainly two reasons that reads are not counted directly: (i) common read lengths of around 100bp generate a too high feature space dimensionality; (ii) current reads are not perfect in aspect of sequencing quality, and usually a trimming stage is required which increases even more the feature space dimensionality. Therefore, retrieving k-mers from reads is necessary, at the price of greatly introducing feature redundancies. If perfect reads were achieved some day, the k-mer analysis logic could be applied directly on reads, hopefully achieving a better performance. (see section 2.3.4 k-mer analysis)

8.1.3 k-mer analysis on cancer genome

Cancer is developed from the accumulation of random mutations. As a result, a cancer genome usually contains abundant genomic alternations. This may considerably increase k-mer diversity. Technically, this results in a even larger feature dimensionality, as well as a tougher task of k-mer extension. Biologically, however, k-mer analysis should be a favorable analysis methodology in this situation, since they capture mutations individually, and analyze mutations altogether for retrieving informative signals. It should be possible to search informative events such as SNVs related with RNA processing genes and neoantigens.

8.1.4 KaMRaT for k-mer dimensionality reduction

We designed, developed, and validated the KaMRaT software (chapter 3) to address the challenges of k-mer interdependence, tremendous number and lack of specificity.

The tremendous k-mer number should result from two main sources: (i) k-mers are generated by single nucleotide shifts along sequence reads, (ii) k-mer signals contain a large amount of noise related to non-relevant events and artifacts (e.g., sequencing errors, adapters, etc.). We designed the KaMRaT merge and KaMRaT rank modules targeting respectively these two situations.

KaMRaT was designed as a set of modules that can be combined in different ways to fit various applications. Possible application workflows include:

- merge-rank and ran$k$-merge approaches for k-mer preselection prior to classifier construction;
- filter-merge approach for extraction of condition-specific k-mer;
- query for inter-cohort analysis;
- mask for exclusion of non-relevant sequences.

Our article also underlines the difference between overlapping k-mer extension (also called merging) and *de novo* assembly. Our k-mer extension is not intended to retrieve whole transcripts but to focus on local events with even single nucleotide differences.

8.1.5 *k*-mer based signals for classifier construction

Researchers now build reference-free classifiers by applying machine-learning algorithms on k-mer features [Thomas et al., 2019, Lorenzi et al., 2020, Sun et al., 2021]. The major advantages of these classifiers is that they can outperform conventional transcriptome classifiers while allowing detection of novel biomarkers.

Here we contributed to these exciting new developments in different ways. We focused two real problems of prostate cancer prognosis: (i) risk level determined by pathologists, and (ii) relapse events after a defined period. We examined on both problems the reference-free classifier performance using conventional gene classifiers as the benchmark. We conducted a fair comparison of the two types of the features by applying the same selection and model building procedure: reduction of dimensionality with a Bayes classifier, model building with LASSO regression - stability selection - logistic regression. We then evaluated the models and selected features on the same independent data sets.

Results showed that our reference-free classifier performed as well as reference-based one, while detecting multiple novel events such as non-annotated RNAs and novel variants of annotated RNAs. On the difficult task of relapse prediction however, the shortcoming of reference-based classifiers, namely the poor generalization in independent cohorts, held with the reference-free classifier.

8.1.6 Replicability of differential *k*-mer signals in tumors

Here we compared differential k-mer contigs found in different normal vs. tumor analysis (two analyzes in adenocarcinoma, one in prostate). Results showed that the intersection of differential k-mer contig signals between two lung data sets (i.e., relevant and independent data sets) was much larger than that between a lung and a prostate dataset. This indicates that despite the large amount of noisy k-mers

found in each differential analysis, the differential k-mer contig signals are actually replicable across data sets, and this replicability does not occur by coincidence.

Moreover, by examining differential k-mer contigs shared by two lung cohorts, we found a list of interesting biological events, including:

- Intron retention events in lung cancer drivers, such as EGFR and MET;
- Novel lincRNAs;
- Repeat elements related with Line 1 and Alu family that were specifically expressed in tumors and correlated with clinical parameters;
- Contigs associated to immune cells.

Finally, by focusing on the shared contigs expressed exclusively in tumor samples, we defined a list of potential neoantigen sources.

8.1.7 Finding k-mer signatures for genes and transcripts

k-mer signals can be associated to specific genes, transcripts (even to chimeric transcripts). This allows using k-mers for querying the corresponding genes or transcripts in large NGS datasets. Specific k-mers can be defined at different levels as follows:

- Gene-specific: k-mers found zero or one time in the reference genome, and found at least one time in reference transcriptome;
- Transcript-specific: k-mers found zero or one time in the reference genome, and only once in the reference transcriptome;
- Chemira-specific: k-mers found in a chimeric (or fusion) transcript, but neither in the reference genome nor in the reference transcriptome.

These specific k-mers are a way to associate reference-free features to actual reference-based genes or transcripts. This analysis performed as an application of the *Kmerator* tool (chapter 6), showed that most genes (97%) and transcripts (83%) have signature k-mers with $k = 31$.

These signature k-mers can help researchers do a variety of tasks faster and as accurate as the conventional ways. These tasks include:

- Estimation of target gene expression;
- Detection of polyA+ or ribo- sequencing protocols;

- Detection of sequencing strandedness;
- Identification of sample sex;
- Identification of sample species;
- Examination of read coverage bias from 5' to 3' ends;
- Examination of HeLa, mycoplasma, or virus contamination.

8.1.8 Arbitrary sequence quantification with *k*-mers

The *REINDEER* software offers a possibility of indexing RNA-seq data for arbitrary sequence abundance query in large RNA-seq datasets. We analyzed the accuracy of counts reported by *REINDEER* for expression quantification purposes.

As each *REINDEER* query returns a list of counts for different regions of the de Bruijn Graph, the conversion of these counts to an actual transcript or gene expression value was not trivial. We examined different interpretations of *REINDEER*'s query result, and found the 'sum of counts' interpretation correlated well with a standard gene quantification result. We also proposed a proof to explain the linearity between this summed value and conventional gene expression. However, some results were not consistent with our proposed proof, therefore this analysis still needs to be refined. *REINDEER* achieved a quite satisfying recall of up to 80% that was dependent on the fraction of k-mers to be retrieved. Furthermore, accuracy was not strongly influenced by sensitivity, suggesting that a low fraction of k-mers in an RNA sequence can be used in queries, to optimize sensitivity without endangering accuracy.

8.2 Perspectives

8.2.1 *k*-mer count matrix generation

The current version of *KaMRaT* combines *Jellyfish* [Marçais and Kingsford, 2011] and *DE-kupl joinCounts* program [Audoux et al., 2017] for generating k-mer count matrix. This two-step procedure is slow and cumbersome for users. A recent software - *kmtricks* [Lemane et al., 2021] - allows to do these two tasks in a single step. *kmtricks* should also run faster, since it optimizes the counting step using a Bloom filter (see section 1.6.1 Useful data structures for transcriptomics). Moreover, the software can scale the count values into a range given by user (e.g., from 0 to 255). This actually would allow reducing the memory used by counts, since the 32-bit *float* or *int* values can be replaced by 16-bit or even 8-bit integers. We did not

have time to test usage of *kmtricks* as the upstream software of *KaMRaT* during the thesis, but it should be a worthy choice when running *KaMRaT* in the future.

The issue of updating the index and count matrix when new samples are added has not been addressed in my thesis. *iMOKA* indexes each sample separately, and produces the sample count vector when needed. This allows adding new sample indexes without reprocessing the previous ones. However, this technique may to some extent conflict with feature recurrence filtering (see section 2.3.3 k-mer counting and rare k-mer prefiltering), since it involves a sample-wise evaluation.

8.2.2 Improving *KaMRaT*

Due to time limitations, the *KaMRaT* software is still a prototype. It may require further development to attain better performance and offer more functionalities.

Currently, one major limitation of *KaMRaT* is that its modules make intensive calls to *seek*, *read* and *write* functions that saturate I/Os and increase execution time. One potential solution is to implement minimizers (see section 1.6.2 Sublinear data structures used in transcriptomics for definition of minimizers) in the *KaMRaT* index module. This should make overlapping k-mers more likely to be together in the index file, hence reducing reloads of indexed content into memory [this proposition comes from an discussion with Yoann Dufresne in Institut Pasteur in September 2021]. Another potential solution may be the application of HDF5 technologies to optimize I/Os. Also, the current version of *KaMRaT* does not support parallel computing. Some modules, such as *KaMRaT* filter and the evaluation step in *KaMRaT* rank could be performed in parallel.

Currently, *KaMRaT* does not consider compositional data reopening (see section 1.5.2 Compositional data analysis), but uses a straightforward scaling method instead. A possible improvement would be to include a centered log-ratio transformation in the *KaMRaT* index module, as an alternative to the normalization step. This may improve all processing methods related to correlation, such as merging interventions. It would also be interesting to verify how the reopening of k-mer count signals could improve the performance of downstream classifiers.

The current version of *KaMRaT* merge does not optimize contig sequence representation. Presently, sequences are saved as character strings which wastes much memory (8 bits per nucleotide). A potential improvement would be to use instead a bit vector with 2 bit per nucleotide. The *BitMagic* library [`http://bitmagic.io/index.html`] may help in this task.

In the current *KaMRaT* rank module, only two machine-learning-based methods, Bayes and logistic regression, support correction of batch effect across samples. A possible improvement would be to add an extra preprocessing step specially for batch effect correction, and thereby all methods could benefit from batch effect correction.

KaMRaT mask and *KaMRaT* query currently only supports searching k-mers with exact match. It would be very interesting to add support for tolerating mismatches and indels. One possible way is to apply locality sensitive hashing for estimating sequences' edit distance [Marçais et al., 2019a]. But this may require some re-design of the indexing strategy to remain compatible with the *KaMRaT* merge/rank/filter modules.

Another interesting module for *KaMRaT* could be named "*KaMRaT* correlate". It would allow retrieving all k-mers or generic features that correlate with a given k-mer or feature. For instance, this would allow to find all host RNAs induced by the presence of a bacterial sequence in samples.

Another potentially interesting module to add to *KaMRaT* would perform k-mer clustering according to the similarity of count vectors. The current version does not integrate this module basically because of the usually large time complexity of this problem, combined to the very high number of k-mers. However, the recent article [Sun et al., 2021] solved this problem by using locality sensitive hashing, and it would be interesting to add this functionality to *KaMRaT*, building a new module on sample-count clustering.

Thanks to the modular design, adding/modifying/removing *KaMRaT*'s functionalities should not be difficult. Actually, one central idea of *KaMRaT* implementation is to allow continuous development.

Résumé en français

La bioinformatique est un domaine multidisciplinaire impliquant la biologie, les statistiques et l'informatique. Elle utilise les méthodologies computationnelles et applique des analyses statistiques pour résoudre des problèmes biologiques.

Cette thèse de doctorat concerne le développement et l'application des nouvelles technologies dans l'étude du transcriptome à la résolution du nucléotide, y compris des logiciels pour la récupération de séquences biologiques pertinentes pour le sujet de recherche, ainsi que l'indexation et la recherche de séquences arbitraires.

Introduction : Chapitres 1-2

Le génome est la base héréditaire de tous les organismes vivants, qui est une longue séquence ADN formée par 4 types de nucléotides : A, C, G, T. Les gènes s'arrangent sur le génome. Chaque gène code pour une ou plusieurs molécules d'ARN par « transcription » ; et en outre dans de nombreux cas, des polypeptides via des « traductions ». Les gènes sont séparés par des régions intergéniques. Chez les eucaryotes, un gène peut être composé d'exons et d'introns.

Des mutations existent dans tous les organismes, résultant soit d'un fonctionnement cellulaire normal, soit des interactions aléatoires avec l'environnement. Ceux-ci incluent : les mutations ponctuelles et indel des séquences courtes. Ces mutations affectent la fonction des gènes par des mécanismes d'interaction complexes.

Le transcriptome comprend l'ensemble complet des transcrits d'ARN. Il résume toutes les variations provenant d'altérations génétiques, d'initiation de la transcription et de modifications post-transcriptionnelles. Chez humain, tous les niveaux de ces variations ont des impacts potentiels sur la santé, y compris des SNVs, des fusions de gènes, et des épissages alternatifs.

NGS RNA-seq, initialement développé en 2008, est actuellement une technologie majeure pour le profilage du transcriptome. Il vise à mesurer l'ensemble des ARNm dans des échantillons donnés et considère à la résolution du nucléotide.

Une méthodologie pour analyser des données produites par RNA-seq est appelée « mapping-first », qui aligne des reads de séquence sur une référence prédéfinie.

L'alignement peut être soit sur une référence génome, soit sur une référence transcriptome. Le premier estime l'expression du gène, et il permet de trouver des nouvelles jonctions d'épissage non annotées. Le second estime l'expression du transcrit avec un algorithme espérance-maximisation, mais est limité au transcriptome connu à priori. Les méthodes basées sur l'alignement impliquent généralement énorme de complexité en temps de calcul. Les méthodes récentes de la seconde famille comme *Kallisto* et *Salmon*, pourtant, appliquent pseudo-alignement qui a largement amélioré la rapidité d'exécution.

L'autre méthodologie généralement appliquée pour l'analyse des données RNA-seq est basée sur l'assemblage des reads de séquence. Ces méthodes assemblent des reads selon leurs chevauchements pour retrouver des transcrits initiaux. Il y en a aussi deux familles de méthodes : assemblage selon un génome de référence comme *Cufflinks* et assemblage sans référence (aussi appelé *de novo*) comme *rnaSPAdes*. La première classe est plus rapide mais dépend d'une référence, alors que la seconde s'applique à tous les organismes mais demande calcul plus lourd.

Les deux types de méthodologies ont une limitation importante : l'analyse est toujours faite au niveau de gène ou de transcrit, ignorant effectivement la capacité de mesurer à la résolution de nucléotide autorisé par la technologie RNA-seq. Cela laisse des événements locaux hors de considération, par exemple des SNVs, des indels, des nouveaux sites d'épissage, etc., alors que ces événements peuvent effectivement avoir un rôle important dans le sujet de recherche. Dans un point de vue au niveau de gène ou de transcrit, tous les événements locaux d'un gène ou d'un transcrit sont agrégés à une seule valeur indiquant le niveau d'expression. De plus, parfois les événements sur-exprimés et sous-exprimés peuvent même s'annuler pendant cette agrégation, laissant les gènes ou les transcrits informatives perdre par l'analyse.

De plus, pour les méthodologies basées sur une référence, une question importante se pose de savoir si la référence permet de trouver vraiment toutes les variations dans un échantillons arbitraire de n'importe quelle condition (âge, maladie, tissue d'échantillonnage, etc.). Une autre limitation est qu'avec alignement, l'analyse n'est pas vraiment déterministe. D'abord, les résultats d'analyse avec alignement dépendent de la qualité d'alignement qui implique de multiples artefacts. Aussi, la référence elle-même évolue avec le temps.

Pour les méthodes d'assemblage *de novo*, elles ratent facilement des transcrits rares, en particulier aux positions où les profondeurs de séquençage sont faibles. De plus, leurs résultats contiennent toujours un taux de mis-assemblages, à cause de l'absence d'indications de référence. Cela potentiellement entraîne des fausses découvertes.

Des logiciels existent aussi pour rechercher des événements locaux, tels que *Kissplice*, *IRFinder*, et *LeafCutter*. Cependant, ils ne ciblent que certains types

d'événements (par exemple, les événements d'épissage et les rétentions d'intron) et ne capturent pas de manière exhaustive tous les types d'événements.

Une méthodologie émergente pour analyser des données RNA-seq au niveau des événements locaux à la résolution du nucléotide est par l'analyse de k-mers. Les k-mers sont des sous-chaînes de caractères successives de longueur k, extraites à partir des reads de séquences. Par exemple, un read $AACCGGTT$ peut être transformé en quatre 5-mers $AACCG$, $ACCGG$, $CCGGT$, et $CGGTT$. La valeur typique de k est un nombre impair inférieur ou égale à 31. Le choix de valeurs impaires empêche que certains k-mer indépendants ne soient confondues quand l'expérience RNA-seq sous mode « stranded », et le choix de $k < 32$ est lié à l'architecture informatique actuelle de 64 octets.

Pour chaque échantillon, les k-mers sont comptés par des compteurs comme *Jellyfish* ou *KMC3*, ainsi que chaque k-mer est associé à une valeur indiquant son occurrence dans cet échantillon. Une matrice de comptages de k-mers est ensuite construite, où chaque ligne est un k-mer et chaque colonne est un échantillon. L'idée est d'appliquer directement des méthodes d'analyse (e.g., analyse différentielle, algorithme d'apprentissage automatique, etc.) sur la matrice de comptage des k-mers, sans alignement et sans assemblage.

Un avantage direct de l'analyse de k-mer est que les caractéristiques permettent de capturer des événements à la résolution du nucléotide, puisque la longueur k est typiquement sélectionnée aussi petit que 31, et que les k-mers sont générés dans une façon successive. Cela complètement profite de la précision fournie par les données RNA-seq. De plus, puisque cette méthodologie ne dépend pas d'une référence connue, elle permet de détecter des nouveaux événements. Aussi, les résultats d'analyse sont déterministes, sans biais introduit par l'étape de mapping.

Un autre aspect intéressant est qu'en fait, les reads eux-mêmes sont des k-mers par leur nature. L'une des raisons pour lesquelles nous ne traitons pas directement ces reads est qu'ils sont généralement trop longs (par exemple, 101 pb). Cela rend l'espace de caractéristiques généralement grand (avec une dimension jusqu'à 4^{101}). De plus, les technologies de séquençage actuelles ne sont pas encore parfaites. Ainsi, les reads sont généralement coupés pour des raisons de qualité, ce qui rend l'espace des caractéristiques encore plus grand du fait de la variabilité de longueur de reads. Donc, les k-mers peuvent être considérés comme un moyen de raccourcir et de fixer la longueur des séquences de caractéristiques, avec une redondance largement aggravée comme le prix. S'il y avait une possibilité de séquencer des reads courts et parfaits ou quasi-parfaits à l'avenir, l'analyse du signal k-mer pourrait être appliquée directement sur les reads, qui serait une approche puissante pour l'analyse des données RNA-seq.

Des défis existent aussi pour l'analyse de k-mer. Premièrement, comme les k-mers sont récupérés par incréments de 1 nt, ils sont très interdépendants et

leur nombre explose rapidement par rapport aux gènes ou aux transcrits. Par exemple, un seul échantillon humain peut contenir jusqu'à 10^8 31-mer distincts, par contre seulement 10^4 gènes ou 10^5 transcrits. Outre la complexité de calcul considérable induite dans l'exécution, cela aggrave de plus le problème des tests multiples lors de l'estimation de la signification statistique et la malédiction de la dimensionnalité dans les tâches de classification et de clustering. Deuxièmement, les k-mers sont typiquement aussi courts que 31 nt ou moins. Les séquences courtes manquent de spécificité et rendent ainsi l'interprétation en aval difficile. Aussi, ce manque de spécificité introduit également une énorme variabilité dans les comptes, car les k-mers alignés aux plusieurs endroits au sein d'un transcrit obtiennent des comptes artificiellement plus élevés. Cette grande variabilité des comptes est une source majeure de bruit lorsque les k-mers sont utilisés comme proxy pour la quantification des transcrits.

Plusieurs méthodes existent déjà pour l'analyse de données d'expression par k-mers.

DE-kupl a été le premier logiciel à appliquer l'analyse de k-mer « directe », c'est-à-dire sans tenir compte du gène ou de l'assemblage, aux données RNA-seq. En bref, *DE-kupl* compte d'abord des k-mers échantillon par échantillon et les joint comme une matrice de comptage de k-mer. Ensuite, *DE-kupl* applique une analyse différentielle (t-test, *DESeq2*, ou *Limma-Voom*) pour extraire une liste de k-mer significatifs. Ces k-mers sont ensuite fusionnés en contigs (les séquences plus longues que k mais reste toujours au niveau local) en fonction de leurs chevauchements de séquence. Enfin, les contigs sont annotés pour l'interprétation biologique.

Gecko implémente un algorithme génétique pour sélectionner des k-mers pertinents pour les conditions des échantillons. Au stade de la préparation des données, le comptage des k-mers est suivi des étapes d'élimination des k-mers non informatives et redondantes. Ensuite, un algorithme génétique est utilisé pour chercher itérativement les k-mers qui groupent les échantillons le plus précisément.

iMOKA a été développé pour construire des classifieurs avec des k-mer. Il utilise le récent compteur de k-mer *KMC3*, suivi des deux niveaux de réduction : d'abord une combinaison de classifieur de Bayes et un filtre d'entropie adaptative ; et puis une étape pour agréger des k-mers selon leurs chevauchements de séquence, sélectionnant un k-mer représentatif par chaque groupe de k-mers chevauchés. *iMOKA* intègre également un classifieur utilisant des forêts aléatoires, ainsi qu'un prédicteur de condition d'échantillon qui prédit pour chaque échantillon nouvellement donné. Le logiciel comprend également une interface graphique conviviale pour les non-spécialistes.

Résultats : Chapitre 3-7

Chapitre 3. Développement de la boîte à outils *KaMRaT* pour l'analyse de k-mer

L'analyse directe des comptes de k-mer a montré de nombreux avantages pour la recherche transcriptomique sans référence : (i) capture exhaustive de toutes les variations de séquence sans limitation d'une référence prédéfinie ; (ii) représentation des événements et estimation de l'expression stables à travers les versions de référence ; et (iii) la prise en compte des variations de la résolution d'un seul nucléotide. À la situation actuelle, cependant, aucune méthode n'est disponible pour analyser des comptes de k-mer dans une perspective générale. Toutes les méthodes, telles que *DE-kupl* pour l'analyse différentielle de k-mer, *Gecko* et *iMOKA* pour la construction de classifieurs, traitent un problème spécifique avec leurs pipelines de travail fixés. Nous considérons que le manque de la perspective générale sur l'approche d'analyse de k-mer peut être un obstacle au développement de cette méthodologie. Cette motivation nous a conduit à proposer *KaMRaT* (k-mer Matrix Reduction Toolkit), un logiciel offrant des utilisations multifonctionnelles et des pipelines flexibles pour le traitement des comptages de k-mer.

KaMRaT prend comme entrée une matrice des comptes de k-mer et produit une matrice réduite où les caractéristiques sont moins interdépendantes et plus pertinentes pour le sujet d'étude. La matrice des comptes de k-mer est produite à partir des échantillons individuels avec des scripts compagnons à l'aide du logiciel *Jellyfish* et du programme en C *joinCounts* dans le logiciel *DE-kupl*. La rapidité et la conception modulaire de *KaMRaT* permettent à utilisateurs de mettre en œuvre et de tester différents pipelines de travail.

Le module *KaMRaT merge* permet d'étendre des k-mers pour former des contigs. Ce module hérite partiellement du programme *mergeTags* dans le logiciel *DE-kupl*, mais il intègre originalement une intervention de fusion pour contrôler le ratio de mis-extension où les k-mers indépendants sont fusionnés à cause de leur bon chevauchement simplement par coïncidence. Nous avons proposé trois méthodes pour cette intervention : MAC, Pearson, et Spearman. Selon nos évaluations, toutes ces interventions permettent de considérablement réduire les ratios de mis-extension, au prix de rendre les contigs relativement plus courts. Nous avons montré également que notre extension de k-mers ait une nature différente que l'assemblage conventionnel qui retrouve des transcrits complets. Notre méthode reste toujours au niveau local, utilisant des contigs courts de la longueur médiane à l'échelle de 100nt pour représenter précisément des événements locaux.

Le module *KaMRaT rank* intègre plusieurs possibilités de la sélection des caractéristiques en utilisant les conditions des échantillons : p-value ajustée et π-value de t-test, SNR, DIDS, classifieur Bayesien, et régression logistique. Nos évalu-

ations de ces méthodes montrent qu'elles donnent des rangs de caractéristiques divergents, et les t-tests et SNR sont plus robustes aux comptes aberrants que les autres méthodes et devraient être utiles pour exclure des signaux hétérogènes non pertinents.

En comparant avec *iMOKA* pour une tâche de classification sur les deux jeux réels, les deux approches de *KaMRaT*, *merge-rank* et *rank-merge*, se comportent aussi bien qu'*iMOKA*, mais sont relativement plus efficace en termes de temps de CPU et usage RAM de pointe. Quand la tâche de classification est plus simple (diagnostic entre des échantillons tumoraux vs normaux), les deux approches *KaMRaT* et *iMOKA* donnent tous des prédictions quasiment parfaites, alors que quand la tâche devienne plus difficile (pronostic de rechute de cancer) les deux logiciels donnent tous des prédiction remarquablement moins satisfaisantes (mais toujours au même niveau). Pour le jeu plus grand, *KaMRaT* est plus efficace en temps, et pour le jeu plus petit, *KaMRaT* consomme moins de mémoire.

En conclusion, nous pensons que les avantages de *KaMRaT* résident principalement en trois points : (i) une approche flexible et une utilisation multifonctionnelle, (ii) des méthodes de classement légères et rapides, (iii) des performances de présélection des caractéristiques au même niveau qu'un logiciel de l'état-de-l'art. Nous espérons que ce logiciel ouvrira de nouvelles possibilités pour appliquer des analyses aux signaux de k-mer et fera progresser la recherche de cancer ou d'autres maladies au niveau de sous-transcript.

Chapitre 4. Classifieurs avec k-mers pour le pronostic du cancer

L'issue du cancer humain peut être prédite en partie par les profils d'expression génique. Cette prédiction est particulièrement importante dans le cancer de la prostate, où distinguer les tumeurs indolentes des tumeurs agressives permettrait d'éviter un traitement inutile et d'améliorer la qualité de vie des patients. Les prédicteurs utilisant l'expression génique prennent généralement la forme d'une signature, c'est-à-dire un ensemble de gènes ou de transcrits et de coefficients associés d'un modèle qui peuvent être utilisés pour prédire le risque ou l'issue à partir des échantillons de patient.

Ce chapitre vise à la construction de ces prédicteurs basés sur des caractéristiques de k-mers. Lorsqu'elle est appliquée à des ensembles de données médicales de RNA-seq à l'aide de méthodes statistiques, cette stratégie identifie toute sous-chaînes de caractères dont l'abondance accrue est associée à un marqueur clinique donné. Cela peut inclure de nouvelles variantes d'épissage, de lncRNAs, ou des ARN provenant de rétroéléments répétés qui sont ignorés par les protocoles conventionnels basés sur des annotations de gènes de référence.

La construction de prédicteurs basés sur des caractéristiques de k-mers est composée de plusieurs étapes : (i) application de *DE-kupl* pour construire la matrice des comptes de k-mer ; (ii) fusionner des k-mers aux contigs pour réduire l'interdépendance des séquences ; (iii) application du classifier de Bayes et de la régression LASSO pour sélectionner les contigs plus pertinents (signatures) ; (iv) construction de modèle logistique avec des contigs de signature sélectionnés ; (v) estimation des comptes des contigs de signature dans un jeu indépendant ; (vi) prédiction des conditions d'échantillons dans le jeu indépendant, et évaluation de la performance de prédiction.

En comparant notre classifieur avec un classifieur conventionnel au niveau de gène, nous avons trouvé que (i) les performances de prédiction de nos classifieurs sont comparables avec celles des classifieurs conventionnels qui utilisent les caractéristiques de gènes, pour les deux tâches pronostics de risque et de rechute ; (ii) alors que les signatures de risque montrent une reproductibilité satisfaisante, les signatures de rechute fonctionnent faiblement dans des ensembles de données indépendants ; (iii) nos classifieurs permettent de trouver des contigs de signature non-annotés.

Chapitre 5. Analyse différentielle de k-mers entre tissus tumoraux vs normaux dans des cohortes indépendantes

L'analyse du transcriptome des tissus cancéreux a joué un rôle déterminant dans la définition des sous-types de tumeurs, des signatures diagnostiques et des réseaux de régulation du cancer. Les transcriptomes cancéreux sont encore majoritairement analysés au niveau de l'expression des gènes. Peu d'études ont abordé les variations au niveau des transcrits, et la plupart d'entre elles n'ont examiné que les variantes d'épissage.

Auparavant, nous avons introduit une méthode, *DE-kupl*, qui effectue une analyse différentielle des données RNA-seq au niveau de k-mer. Comme cette méthode est sans référence et sans alignement, elle identifie tous nouveaux ARNs ou isoformes d'ARN présents dans les données à la résolution du nucléotide, y compris les transcrits mal mappés tels que les ARNs de répétition et les ARNs chimériques. Ici, nous visons à évaluer tous les événements non-annotés découverts par *DE-kupl* dans une comparaison entre les échantillons normaux vs tumoraux, sur un jeu de l'adénocarcinome pulmonaire (LUAD) comme test. Pour atténuer les événements de faux positifs inhérents à tout profil d'expression génique, nous voulons concentrer sur les événements qui ont été répliqués dans deux ensembles de données indépendants.

Nous avons identifié les contigs produits par *DE-kupl* partagés par deux jeux de données en construisant un graphique à l'aide du package Python *NetworkX*,

avec des k-mers comme des nœuds et des k-mers partagés comme des arêtes. Les contigs correspondant au même événement local devraient former un sous-graphe d'une clique entièrement connectée. Nous avons ainsi extrait toutes les cliques pour identifier les contigs partagés et puis annoté ces contigs.

En cherchant des contigs partagés entre LUADseo ∩ LUADtcga (même maladie dans différents jeux) et entre LUADtcga ∩ PRADtcga (maladies différentes), nous trouvons que les contigs sont remarquablement plus réplicable entre les jeux d'un même maladie que ceux des maladies différentes. Cela indique qu'un sous-ensemble important de signaux de contigs différentiels est réplicable dans des études indépendantes.

En regardant l'ensemble des contigs partagés par des deux jeux LUAD, nous trouvons des informations biologiques intéressantes, par exemple : (i) Une classe d'événements différentiels typique impliquait des répétitions endogènes. Les expressions des répétitions *L1* et *Alu* ont défini deux sous-groupes majeurs de tumeur. Le sous-groupe avec une expression *L1*/*Alu* plus élevée est associé aux mutations plus fréquentes dans *P53*, à une charge plus élevée de mutation et de nombre de copies, avec un infiltrat de cellules immunitaires réduit. (ii) Environ 500 contigs strictement tumoraux sont identifiés, dont 55 ont été prédit comme des sources de néoantigènes du CMH de classe I.

Chapitre 6. Association des signaux de k-mer aux gènes et aux transcripts

La recherche des événements transcriptionnels ou des ARNs spécifiques à travers des jeux de données à grande échelle est devenue essentielle en médecine de précision. Cette tâche d'interrogation et de réanalyse des ensembles de données RNA-seq pose deux défis majeurs. Le premier consiste à filtrer les jeux de données pour sélectionner un sous-ensemble de fichiers plus homogène et plus fiable à explorer dans le contexte de métadonnées incomplètes. Le deuxième consiste à effectuer la quantification des biomarqueurs d'ARN dans un temps raisonnable et avec une précision suffisante, pour extraire des informations biologiques. Les méthodes basées sur l'alignement telles que *STAR* et *CRAC* nécessitent des ressources de calcul importantes, ce qui les rend inadéquates pour interroger des biomarqueurs parmi des ensembles de données de l'ordre de 100 à 1 000 fichiers. Les outils utilisant pseudo-alignement comme *Kallisto* et *Salmon* sont beaucoup plus rapides, mais utilisent le plus souvent une référence transcriptome éloignée de la véritable diversité biologique des ARNs.

Des approches reposant sur k-mers ont émergé récemment, et sont utilisées pour l'interrogation des données transcriptomiques. Ces méthodes nécessitent moins de temps et moins de ressources de calcul et sont adaptées à diverses questions

biologiques, y compris l'analyse des événements transcriptionnels non annotés et atypiques.

Nous présentons *Kmerator Suite*, un ensemble de trois outils conçus pour (i) extraire les signatures de k-mer des transcrits, (ii) quantifier ces k-mers à partir des ensembles de données RNA-seq et (iii) visualiser les caractéristiques de grands ensembles de données RNA-seq à l'aide des signatures précalculées. Le cœur de cette suite est *Kmerator*, qui génère des signatures k-mer spécifiques aux gènes ou aux transcrits. Le deuxième outil, *countTags*, est utilisé pour quantifier les k-mers sélectionnés dans les fichiers bruts de RNA-seq. Le troisième outil, *KmerExploR*, démontre la capacité du pipeline combiné par *Kmerator* + *countTags* pour extraire des métadonnées à partir de données RNA-seq brutes avec des signatures de k-mer prédéfinies.

Une fonctionnalité principale de notre étude est de chercher les signatures de k-mers pour les gènes et les transcrits. Nous définissons trois niveaux de signatures : (i) les k-mers spécifiques au niveau de gène sont trouvés à zéro ou une fois dans la référence génome et au moins une fois dans la référence transcriptome ; (ii) les k-mers spécifiques au niveau de transcrit se trouvent zéro ou une fois dans la référence génome et une seule fois dans la référence transcriptome ; (iii) les k-mers spécifiques au niveau de la chimère ne se trouvent ni dans la référence génome ni dans la référence transcriptome. En testant sur les références génome et transcriptome humain, nous avons extrait des signatures de k-mers pour 83% des transcrits et 97% des gènes.

En appliquant *countTags* sur des signatures de k-mers identifiés par *Kmerator*, nous avons arrivé à faire : (i) estimation rapide et précise de l'expression des gènes ou des transcrits donnés ; (ii) détection de polyA vs ribo- d'extraction ; (iii) détection de strandedness ; (iv) détection de sexe ; (v) détection du biais de couverture de reads de 5' à 3' fin ; (vi) détection de contamination de HeLa, de mycoplasmes, ou de virus ; (vii) identification des espèces dans l'échantillon.

Chapitre 7. Requête de séquence arbitraire dans les données RNA-seq

Nous définissons ici une « requête de séquence arbitraire » comme la tâche consistant à trouver et quantifier une séquence d'ARN ou d'ADN arbitraire non annotée dans des fichiers RNA-seq bruts. Dans l'analyse de k-mer, cette tâche est essentielle pour la requête inter-cohorte des comptes de contigs ou de k-mers, analogue à la requête beaucoup plus simple d'expressions de gènes ou de transcrits via leurs identifiants universels. Dans une perspective biomédicale, il existe un grand besoin non satisfait d'identification des transcrits non annotés liés aux maladies à partir des vastes jeux de données RNA-seq. La difficulté majeure de cette tâche

est que les séquences en question ont des longueurs arbitraires et des ordres de nucléotides arbitraires, ce qui entraîne un espace de possibilité infini. Par conséquent, les séquences n'ont pas d'identifiants unifiés entre les cohortes, ce qui rend l'indexation et l'interrogation difficiles. *REINDEER* est un logiciel pionnier qui résout ce problème en utilisant des monotigs comme éléments d'index et de requête. Notre objectif ici est de valider l'exactitude des résultats de la requête de *REINDEER*.

En utilisant les 12 lignées cellulaires de cancer du poumon à extraites de CCLE et les 1000 gènes sélectionnés au hasard, nous avons comparé les résultats de requêtes par *REINDEER* avec les quantifications par *Kallisto-tximport*.

La première question que nous nous sommes posés est comment traitons-nous les variabilités parmi les comptes des monotigs différents retournés par *REINDEER*. En fixant une même valeur de k, nous avons testé des différentes méthodes, i.e., des valeurs moyennes, des valeurs médianes, des valeurs de mode, des valeurs minimales, des valeurs maximales, et des valeurs de somme, et comparé les résultats avec les quantifications de *Kallisto-tximport*. Les résultats montrent que par sommer des comptes de monotigs, les requêtes de *REINDEER* corrèlent mieux avec les quantifications *Kallisto-tximport*.

Ensuite, nous avons examiné l'impact d'un paramètre de *REINDEER*, -P, qui est utilisé pour tolérer des mésappariements et des lacunes entre les séquences de requête et les séquences indexées. Nous observons que, en faisant la valeur de cet argument plus strict, c'est-à-dire qui refuse plus de mésappariements et lacunes, *REINDEER* perd plus de gènes trouvés par *Kallisto*. En revanche, les corrélations entre les requêtes de *REINDEER* et les quantification de *Kallisto-tximport* restent peu impactées.

En conclusion, nous pensons qu'un bon choix de requête *REINDEER* est de calculer la somme des comptes de monotigs, en utilisant une petite valeur d'argument -P, ceci donne des résultats assez précis avec une bonne tolérance de mésappariements et lacunes.

Discussion: Chapitre 8

Cette thèse propose et discute une nouvelle méthodologie d'analyse du transcriptome basée sur l'analyse directe des comptes de k-mer. Nous avons montré que cette méthodologie est utile et efficace, avec les avantages principaux comme : (i) elle donne un aperçu au niveau de sous-transcrit à la résolution du nucléotide ; (ii) elle n'est pas liée à une étape de mapper.

KaMRaT présente encore plusieurs aspects à améliorer : (i) l'indexation et les opérations du logiciel effectuent des appels intensifs avec « seek, read, write », qui saturent les lectures et les écritures sur disque, et qui augmentent le temps

d'exécution ; (ii) l'outil ne considère pas encore le traitement des données compositionnelles ; (iii) la représentation de séquence au cours d'exécution n'est pas optimisée et peut occuper un espace mémoire non nécessaire ; (iv) l'outil ne permet pas encore un vrai traitement d'effet batch.

Bibliography

A. Abeshouse, J. Ahn, R. Akbani, A. Ally, S. Amin, C. D. Andry, M. Annala, A. Aprikian, J. Armenia, A. Arora, et al. The molecular taxonomy of primary prostate cancer. *Cell*, 163(4):1011–1025, 2015.

S. Anders and W. Huber. Differential expression analysis for sequence count data. *Nature Precedings*, pages 1–1, 2010.

S. Andrews et al. Fastqc: a quality control tool for high throughput sequence data, 2010.

J. Audoux, N. Philippe, R. Chikhi, M. Salson, M. Gallopin, M. Gabriel, J. Le Coz, E. Drouineau, T. Commes, and D. Gautheret. De-kupl: exhaustive capture of biological variation in rna-seq data through k-mer decomposition. *Genome biology*, 18(1):1–15, 2017.

Z. Avsec, V. Agarwal, D. Visentin, J. R. Ledsam, A. Grabska-Barwinska, K. R. Taylor, Y. Assael, J. Jumper, P. Kohli, and D. R. Kelley. Effective gene expression prediction from sequence by integrating long-range interactions. *bioRxiv*, 2021.

T. M. Ball, L. M. Squeglia, S. F. Tapert, and M. P. Paulus. Double dipping in machine learning: problems and solutions. *Biological psychiatry. Cognitive neuroscience and neuroimaging*, 5(3):261, 2020.

J. Barretina, G. Caponigro, N. Stransky, K. Venkatesan, A. A. Margolin, S. Kim, C. J. Wilson, J. Lehár, G. V. Kryukov, D. Sonkin, et al. The cancer cell line encyclopedia enables predictive modelling of anticancer drug sensitivity. *Nature*, 483(7391):603–607, 2012.

Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal statistical society: series B (Methodological)*, 57(1):289–300, 1995.

G. Biamonti, A. Amato, E. Belloni, A. Di Matteo, L. Infantino, D. Pradella, and C. Ghigna. Alternative splicing in alzheimer's disease. *Aging clinical and experimental research*, 33(4):747–758, 2021.

B. H. Bloom. Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426, 1970.

A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. *Bioinformatics*, 30(15):2114–2120, 2014.

P. Bradley, H. C. Den Bakker, E. P. Rocha, G. McVean, and Z. Iqbal. Ultrafast search of all deposited bacterial and viral genomic data. *Nature biotechnology*, 37(2):152–159, 2019.

N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter. Near-optimal probabilistic rna-seq quantification. *Nature biotechnology*, 34(5):525–527, 2016.

E. Bushmanova, D. Antipov, A. Lapidus, and A. D. Prjibelski. rnaspades: a de novo transcriptome assembler and its application to rna-seq data. *GigaScience*, 8(9):giz100, 2019.

S. Cascianelli, I. Molineris, C. Isella, M. Masseroli, and E. Medico. Machine learning for rna sequencing-based intrinsic subtyping of breast cancer. *Scientific reports*, 10(1):1–13, 2020.

R. Chikhi, A. Limasset, and P. Medvedev. Compacting de Bruijn graphs from sequencing data quickly and in low memory. *Bioinformatics*, 32(12):i201–i208, 06 2016. ISSN 1367-4803. doi: 10.1093/bioinformatics/btw279. URL `https://doi.org/10.1093/bioinformatics/btw279`.

T. G. Clark, M. J. Bradburn, S. B. Love, and D. G. Altman. Survival analysis part i: basic concepts and first analyses. *British journal of cancer*, 89(2):232–238, 2003.

R. Clarke, H. W. Ressom, A. Wang, J. Xuan, M. C. Liu, E. A. Gehan, and Y. Wang. The properties of high-dimensional data spaces: implications for exploring gene and protein expression data. *Nature reviews cancer*, 8(1):37–49, 2008.

M. S. Cline, B. Craft, T. Swatloski, M. Goldman, S. Ma, D. Haussler, and J. Zhu. Exploring tcga pan-cancer data at the ucsc cancer genomics browser. *Scientific reports*, 3(1):1–6, 2013.

M. Cmero, B. Schmidt, I. J. Majewski, P. G. Ekert, A. Oshlack, and N. M. Davidson. Mintie: identifying novel structural and splice variants in transcriptomes using rna-seq data. *bioRxiv*, pages 2020–06, 2021.

F. Collins, E. Lander, J. Rogers, R. Waterston, and I. Conso. Finishing the euchromatic sequence of the human genome. *Nature*, 431(7011):931–945, 2004.

E. Collisson, J. Campbell, A. Brooks, A. Berger, W. Lee, J. Chmielecki, D. Beer, L. Cope, C. Creighton, L. Danilova, et al. Comprehensive molecular profiling of lung adenocarcinoma: The cancer genome atlas research network. *Nature*, 511 (7511):543–550, 2014.

P. E. Compeau, P. A. Pevzner, and G. Tesler. How to apply de bruijn graphs to genome assembly. *Nature biotechnology*, 29(11):987–991, 2011.

T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to algorithms*. MIT press, 2009.

P. Cui, Q. Lin, F. Ding, C. Xin, W. Gong, L. Zhang, J. Geng, B. Zhang, X. Yu, J. Yang, et al. A comparison between ribo-minus rna-sequencing and polya-selected rna-sequencing. *Genomics*, 96(5):259–265, 2010.

S. Deorowicz, A. Debudaj-Grabysz, and S. Grabowski. Disk-based k-mer counting on a pc. *BMC bioinformatics*, 14(1):1–12, 2013.

X. Ding, L. Zhu, T. Ji, X. Zhang, F. Wang, S. Gan, M. Zhao, and H. Yang. Long intergenic non-coding rnas (lincrnas) identified by rna-seq in breast cancer. *PloS one*, 9(8):e103270, 2014.

A. Dobin, C. A. Davis, F. Schlesinger, J. Drenkow, C. Zaleski, S. Jha, P. Batut, M. Chaisson, and T. R. Gingeras. Star: ultrafast universal rna-seq aligner. *Bioinformatics*, 29(1):15–21, 2013.

N. Erho, A. Crisan, I. A. Vergara, A. P. Mitra, M. Ghadessi, C. Buerki, E. J. Bergstralh, T. Kollmeyer, S. Fink, Z. Haddad, et al. Discovery and validation of a prostate cancer genomic classifier that predicts early metastasis following radical prostatectomy. *PloS one*, 8(6):e66855, 2013.

P. Ewels, M. Magnusson, S. Lundin, and M. Käller. Multiqc: summarize analysis results for multiple tools and samples in a single report. *Bioinformatics*, 32(19): 3047–3048, 2016.

W. R. Francis and G. Wörheide. Similar ratios of introns to intergenic sequence across animal genomes. *Genome biology and evolution*, 9(6):1582–1598, 2017.

A. C. Frazee, A. E. Jaffe, B. Langmead, and J. T. Leek. Polyester: simulating rna-seq datasets with differential transcript expression. *Bioinformatics*, 31(17): 2778–2784, 2015.

M. Ghandi, F. W. Huang, J. Jané-Valbuena, G. V. Kryukov, C. C. Lo, E. R. McDonald, J. Barretina, E. T. Gelfand, C. M. Bielski, H. Li, et al. Next-generation characterization of the cancer cell line encyclopedia. *Nature*, 569 (7757):503–508, 2019.

A. B. Glinskii, S. Ma, J. Ma, D. Grant, C.-U. Lim, I. Guest, S. Sell, R. Buttyan, and G. V. Glinsky. Networks of intergenic long-range enhancers and snprnas drive castration-resistant phenotype of prostate cancer and contribute to pathogenesis of multiple common human disorders. *Cell Cycle*, 10(20):3571–3597, 2011.

T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, et al. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *science*, 286(5439):531–537, 1999.

R. S. Harris and P. Medvedev. Improved representation of sequence bloom trees. *Bioinformatics*, 36(3):721–727, 2020.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Wiley Online Library, 2010.

K. E. Hayer, A. Pizarro, N. F. Lahens, J. B. Hogenesch, and G. R. Grant. Benchmark analysis of algorithms for determining and quantifying full-length mrna splice forms from rna-seq data. *Bioinformatics*, 31(24):3938–3945, 2015.

Y. Heo, X.-L. Wu, D. Chen, J. Ma, and W.-M. Hwu. Bless: bloom filter-based error correction solution for high-throughput sequencing reads. *Bioinformatics*, 30(10):1354–1362, 2014.

Z. Iqbal, M. Caccamo, I. Turner, P. Flicek, and G. McVean. De novo assembly and genotyping of variants using colored de bruijn graphs. *Nature genetics*, 44 (2):226–232, 2012.

M. Jeanmougin, A. De Reynies, L. Marisa, C. Paccard, G. Nuel, and M. Guedj. Should we abandon the t-test in the analysis of gene expression microarray data: a comparison of variance modeling strategies. *PloS one*, 5(9):e12336, 2010.

W. E. Johnson, C. Li, and A. Rabinovic. Adjusting batch effects in microarray expression data using empirical bayes methods. *Biostatistics*, 8(1):118–127, 2007.

M. Kokot, M. Długosz, and S. Deorowicz. Kmc 3: counting and manipulating k-mer statistics. *Bioinformatics*, 33(17):2759–2761, 2017.

J. E. Krebs, E. S. Goldstein, and S. T. Kilpatrick. *Lewin's genes XII*. Jones & Bartlett Learning, 2017.

H.-S. Le, M. H. Schulz, B. M. McCauley, V. F. Hinman, and Z. Bar-Joseph. Probabilistic error correction for rna sequencing. *Nucleic acids research*, 41(10): e109–e109, 2013.

J. T. Leek and J. D. Storey. Capturing heterogeneity in gene expression studies by surrogate variable analysis. *PLoS genetics*, 3(9):e161, 2007.

J. T. Leek, R. B. Scharpf, H. C. Bravo, D. Simcha, B. Langmead, W. E. Johnson, D. Geman, K. Baggerly, and R. A. Irizarry. Tackling the widespread and critical impact of batch effects in high-throughput data. *Nature Reviews Genetics*, 11 (10):733–739, 2010.

T. Lemane, P. Medvedev, R. Chikhi, and P. Peterlongo. kmtricks: Efficient construction of bloom filters for large sequencing data collections. 2021.

B. Li and C. N. Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. *BMC bioinformatics*, 12(1):1–16, 2011.

H. Li. Bfc: correcting illumina sequencing errors. *Bioinformatics*, 31(17):2885–2887, 2015.

Y. I. Li, D. A. Knowles, J. Humphrey, A. N. Barbeira, S. P. Dickinson, H. K. Im, and J. K. Pritchard. Annotation-free quantification of rna splicing using leafcutter. *Nature genetics*, 50(1):151–158, 2018.

Y. Liao, G. K. Smyth, and W. Shi. featurecounts: an efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics*, 30 (7):923–930, 2014.

Liu, Yuwen, Zhou, Jie, White, Kevin, and P. Rna-seq differential expression studies: more sequence or more replication? *Bioinformatics*, 2014.

Q. Long, J. Xu, A. O. Osunkoya, S. Sannigrahi, B. A. Johnson, W. Zhou, T. Gillespie, J. Y. Park, R. K. Nam, L. Sugar, et al. Global transcriptome analysis of formalin-fixed prostate cancer specimens identifies biomarkers of disease recurrence. *Cancer research*, 74(12):3228–3237, 2014.

H. Lopez-Maestre, L. Brinza, C. Marchet, J. Kielbassa, S. Bastien, M. Boutigny, D. Monnin, A. E. Filali, C. M. Carareto, C. Vieira, et al. Snp calling from rna-seq data without a reference genome: identification, quantification, differential analysis and impact on the protein sequence. *Nucleic Acids Research*, 44(19): e148–e148, 2016.

C. Lorenzi. *Design and implementation of bioinformatic tools for RNA sequencing data analysis*. PhD thesis, University of Montpellier, 2021.

C. Lorenzi, S. Barriere, J.-P. Villemin, L. D. Bretones, A. Mancheron, and W. Ritchie. imoka: k-mer based software to analyze large collections of sequencing data. *Genome Biology*, 21(1):1–19, 2020.

M. I. Love, W. Huber, and S. Anders. Moderated estimation of fold change and dispersion for rna-seq data with deseq2. *Genome biology*, 15(12):1–21, 2014.

D. Lovell, V. Pawlowsky-Glahn, J. J. Egozcue, S. Marguerat, and J. Bähler. Proportionality: a valid alternative to correlation for relative data. *PLoS computational biology*, 11(3):e1004075, 2015.

S. C. Manekar and S. R. Sathe. A benchmark study of k-mer counting methods for high-throughput sequencing. *GigaScience*, 7(12):giy125, 2018.

G. Marçais and C. Kingsford. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics*, 27(6):764–770, 2011.

G. Marçais, D. DeBlasio, P. Pandey, and C. Kingsford. Locality-sensitive hashing for the edit distance. *Bioinformatics*, 35(14):i127–i135, 2019a.

G. Marçais, B. Solomon, R. Patro, and C. Kingsford. Sketching and sublinear data structures in genomics. *Annual Review of Biomedical Data Science*, 2:93–118, 2019b.

C. Marchet, Z. Iqbal, D. Gautheret, M. Salson, and R. Chikhi. Reindeer: efficient indexing of k-mer presence and abundance in sequencing datasets. *Bioinformatics*, 36(Supplement_1):i177–i185, 2020.

J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad. Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays. *Genome research*, 18(9):1509–1517, 2008.

J. A. Martin and Z. Wang. Next-generation transcriptome assembly. *Nature Reviews Genetics*, 12(10):671–682, 2011.

M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. *EMBnet. journal*, 17(1):10–12, 2011.

J. McClave and T. Sincich. *Statistics (Thirteenth Edition)*. Pearson Boston, 2018.

G. Mendel. Experiments in plant hybridization (1865). *Verhandlungen des naturforschenden Vereins Brünn.) English translation available online: www.mendelweb.org/Mendel.html (accessed on 1 January 2013)*, 1865.

R. Middleton, D. Gao, A. Thomas, B. Singh, A. Au, J. J. Wong, A. Bomane, B. Cosson, E. Eyras, J. E. Rasko, et al. Irfinder: assessing the impact of intron retention on mammalian gene expression. *Genome biology*, 18(1):1–11, 2017.

P. Milanez-Almeida, A. J. Martins, R. N. Germain, and J. S. Tsang. Cancer prognosis with shallow tumor rna sequencing. *Nature medicine*, 26(2):188–192, 2020.

F. Mitelman, B. Johansson, and F. Mertens. The impact of translocations and gene fusions on cancer causation. *Nature Reviews Cancer*, 7(4):233–245, 2007.

A. Morillon and D. Gautheret. Bridging the gap between reference and real transcriptomes. *Genome biology*, 20(1):1–7, 2019.

H. T. Nguyen, H. Xue, V. Firlej, Y. Ponty, M. Gallopin, and D. Gautheret. Reference-free transcriptome signatures for prostate cancer prognosis. *BMC cancer*, 21(1):1–12, 2021.

T. N. H. Nguyen. *Combining machine learning and reference-free transcriptome analysis for the identification of prostate cancer signatures*. PhD thesis, Université Paris-Saclay, 2020.

K. J. Nordström, M. C. Albani, G. V. James, C. Gutjahr, B. Hartwig, F. Turck, U. Paszkowski, G. Coupland, and K. Schneeberger. Mutation identification by direct comparison of whole-genome sequencing data from mutant and wild-type individuals using k-mers. *Nature biotechnology*, 31(4):325–330, 2013.

S. Nurk, S. Koren, A. Rhie, M. Rautiainen, A. V. Bzikadze, A. Mikheenko, M. R. Vollger, N. Altemose, L. Uralsky, A. Gershman, et al. The complete sequence of a human genome. *bioRxiv*, 2021.

V. Nygaard, E. A. Rødland, and E. Hovig. Methods that remove batch effects while retaining group differences may lead to exaggerated confidence in downstream analyses. *Biostatistics*, 17(1):29–39, 2016.

L. Pachter. Models for transcript quantification from rna-seq. *arXiv preprint arXiv:1104.3889*, 2011.

S. Paik, S. Shak, G. Tang, C. Kim, J. Baker, M. Cronin, F. L. Baehner, M. G. Walker, D. Watson, T. Park, et al. A multigene assay to predict recurrence of tamoxifen-treated, node-negative breast cancer. *New England Journal of Medicine*, 351(27):2817–2826, 2004.

P. Pandey, F. Almodaresi, M. A. Bender, M. Ferdman, R. Johnson, and R. Patro. Mantis: A fast, small, and exact large-scale sequence-search index. *Cell systems*, 7(2):201–207, 2018.

R. Patro, G. Duggal, M. I. Love, R. A. Irizarry, and C. Kingsford. Salmon provides fast and bias-aware quantification of transcript expression. *Nature methods*, 14 (4):417–419, 2017.

E. Pettersson, J. Lundeberg, and A. Ahmadian. Generations of sequencing technologies. *Genomics*, 93(2):105–111, 2009.

D. M. Powers. Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation. *arXiv preprint arXiv:2010.16061*, 2020.

T. P. Quinn. Stool studies don't pass the sniff test: A systematic review of human gut microbiome research suggests widespread misuse of machine learning. *arXiv preprint arXiv:2107.03611*, 2021.

T. P. Quinn, I. Erb, M. F. Richardson, and T. M. Crowley. Understanding sequencing data as compositions: an outlook and review. *Bioinformatics*, 34(16): 2870–2878, 2018.

A. Rahman, I. Hallgrímsdóttir, M. Eisen, and L. Pachter. Association mapping from sequencing reads using k-mers. *Elife*, 7:e32920, 2018.

J. J. M. Riethoven. Regulatory regions in dna: Promoters, enhancers, silencers, and insulators. *Methods in Molecular Biology*, 674:33, 2010.

S. Riquier, C. Bessiere, B. Guibert, A.-L. Bouge, A. Boureux, F. Ruffle, J. Audoux, N. Gilbert, H. Xue, D. Gautheret, et al. Kmerator suite: design of specific k-mer signatures and automatic metadata discovery in large rna-seq datasets. *bioRxiv*, 2021.

D. Risso, J. Ngai, T. P. Speed, and S. Dudoit. Normalization of rna-seq data using factor analysis of control genes or samples. *Nature biotechnology*, 32(9):896–902, 2014.

M. E. Ritchie, B. Phipson, D. Wu, Y. Hu, C. W. Law, W. Shi, and G. K. Smyth. limma powers differential expression analyses for rna-sequencing and microarray studies. *Nucleic acids research*, 43(7):e47–e47, 2015.

G. Rizk, D. Lavenier, and R. Chikhi. Dsk: k-mer counting with very low memory usage. *Bioinformatics*, 29(5):652–653, 2013.

N. D. Roberts, R. D. Kortschak, W. T. Parker, A. W. Schreiber, S. Branford, H. S. Scott, G. Glonek, and D. L. Adelson. A comparative analysis of algorithms for somatic snv detection in cancer. *Bioinformatics*, 29(18):2223–2230, 2013.

M. D. Robinson and A. Oshlack. A scaling normalization method for differential expression analysis of rna-seq data. *Genome biology*, 11(3):1–9, 2010.

M. D. Robinson, D. J. McCarthy, and G. K. Smyth. edger: a bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics*, 26(1):139–140, 2010.

L. Salmela and J. Schröder. Correcting errors in short reads by multiple alignments. *Bioinformatics*, 27(11):1455–1461, 2011.

F. Sanger, S. Nicklen, and A. R. Coulson. Dna sequencing with chain-terminating inhibitors. *Proceedings of the national academy of sciences*, 74(12):5463–5467, 1977.

M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary dna microarray. *Science*, 270 (5235):467–470, 1995.

J. Schröder, H. Schröder, S. J. Puglisi, R. Sinha, and B. Schmidt. Shrec: a short-read error correction method. *Bioinformatics*, 25(17):2157–2163, 2009.

N. J. Schurch, P. Schofield, M. Gierliński, C. Cole, A. Sherstnev, V. Singh, N. Wrobel, K. Gharbi, G. G. Simpson, T. Owen-Hughes, et al. How many biological replicates are needed in an rna-seq experiment and which differential expression tool should you use? *Rna*, 22(6):839–851, 2016.

J.-S. Seo, Y. S. Ju, W.-C. Lee, J.-Y. Shin, J. K. Lee, T. Bleazard, J. Lee, Y. J. Jung, J.-O. Kim, J.-Y. Shin, et al. The transcriptional landscape and mutational profile of lung adenocarcinoma. *Genome research*, 22(11):2109–2119, 2012.

J. Sheng, F. Li, and S. T. Wong. Optimal drug prediction from personal genomics profiles. *IEEE journal of Biomedical and Health Informatics*, 19(4):1264–1270, 2015.

J. Simpson, M. Cerezo, C. Churchhouse, D. Altshuler, Y. Lu, D. Kim, A. Hodgkinson, K. Walter, J. Yu, A. Roa, et al. A global reference for human genetic variation. 2015.

C. Soneson. compcoder—an r package for benchmarking differential expression methods for rna-seq data. *Bioinformatics*, 30(17):2517–2518, 2014.

C. Soneson, M. I. Love, and M. D. Robinson. Differential analyses for rna-seq: transcript-leve l estimates improve gene-level inferences [version 2; peer. 2016.

L. Song and L. Florea. Rcorrector: efficient and accurate error correction for illumina rna-seq reads. *GigaScience*, 4(1):s13742–015, 2015.

A. Srivastava, L. Malik, H. Sarkar, M. Zakeri, F. Almodaresi, C. Soneson, M. I. Love, C. Kingsford, and R. Patro. Alignment and mapping methodology influence transcript abundance estimation. *Genome biology*, 21(1):1–29, 2020.

T. Steijger, J. F. Abril, P. G. Engström, F. Kokocinski, T. J. Hubbard, R. Guigó, J. Harrow, and P. Bertone. Assessment of transcript reconstruction methods for rna-seq. *Nature methods*, 10(12):1177–1184, 2013.

Q. Sun, Y. Peng, and J. Liu. A reference-free approach for cell type classification with scrna-seq. *bioRxiv*, 2021.

A. Sveen, S. Kilpinen, A. Ruusulehto, R. Lothe, and R. Skotheim. Aberrant rna splicing in cancer; expression changes and driver mutations of splicing factor genes. *Oncogene*, 35(19):2413–2427, 2016.

D. Takai and P. A. Jones. Origins of bidirectional promoters: computational analyses of intergenic distance in the human genome. *Molecular biology and evolution*, 21(3):463–467, 2004.

A. Thomas, S. Barriere, L. Broseus, J. Brooke, C. Lorenzi, J.-P. Villemin, G. Beurier, R. Sabatier, C. Reynes, A. Mancheron, et al. Gecko is a genetic algorithm to classify and explore high throughput sequencing data. *Communications biology*, 2(1):1–8, 2019.

C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. Van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter. Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation. *Nature biotechnology*, 28(5):511–515, 2010.

K. Van den Berge, K. M. Hembach, C. Soneson, S. Tiberi, L. Clement, M. I. Love, R. Patro, and M. D. Robinson. Rna sequencing data: Hitchhiker's guide

to expression analysis. *Annual Review of Biomedical Data Science*, 2:139–173, 2019.

E. L. van Dijk, Y. Jaszczyszyn, D. Naquin, and C. Thermes. The third revolution in sequencing technology. *Trends in Genetics*, 34(9):666–681, 2018.

L. J. Van't Veer, H. Dai, M. J. Van De Vijver, Y. D. He, A. A. Hart, M. Mao, H. L. Peterse, K. Van Der Kooy, M. J. Marton, A. T. Witteveen, et al. Gene expression profiling predicts clinical outcome of breast cancer. *nature*, 415(6871): 530–536, 2002.

Y. Wang, H. Xue, C. Pourcel, Y. Du, and D. Gautheret. 2-kupl: mapping-free variant detection from dna-seq data of matched samples. *BMC bioinformatics*, 22(1):1–22, 2021.

K. A. Wetterstrand. DNA sequencing costs: Data, 2020. URL `https://www.genome.gov/about-genomics/fact-sheets/DNA-Sequencing-Costs-Data`.

Q. Xu, G. Georgiou, S. Frölich, M. van der Sande, G. J. C. Veenstra, H. Zhou, and S. J. van Heeringen. Ananse: An enhancer network-based computational approach for predicting key transcription factors in cell fate determination. *bioRxiv*, pages 2020–06, 2021.

L. N. Yaddanapudi. The american statistical association statement on p-values explained. *Journal of anaesthesiology, clinical pharmacology*, 32(4):421, 2016.

Y. Yu, J. Liu, X. Liu, Y. Zhang, E. Magner, E. Lehnert, C. Qian, and J. Liu. Seqothello: querying rna-seq experiments at scale. *Genome biology*, 19(1):1–13, 2018.

H. Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.

Y. Zhang, G. Parmigiani, and W. E. Johnson. Combat-seq: batch effect adjustment for rna-seq count data. *NAR genomics and bioinformatics*, 2(3):lqaa078, 2020.

Acronyms

A adenine. 9, 43

ANOVA Analysis of Variance. 23, 24, 39

C cytosine. 9, 43

CCLE Cancer Cell Line Encyclopedia. 16, 129, 130

cDBG colored de Bruijn Graph. 31

cDNA complementary DNA. 13, 14

DBG de Bruijn Graph. 31, 32, 38, 44, 128, 138

DNA deoxyribonucleic acid. 9, 10, 12–15, 31, 127, 167

dNTP deoxynucleoside triphosphate. 14

FN False Negative. 27

FP False Positive. 27

FPKM Fragments Per Kilobase Million. 39

G guanine. 9, 43

indel insertion or deletion. 11, 15

lincRNA long intergenic non-coding RNA. 10, 141

mRNA messenger RNA. 10, 13, 14

ncRNA non-coding RNA. 13, 14

NGS Next-Generation Sequencing. 12, 13, 22, 36, 43–45, 127

PCA Principal Component Analysis. 25, 46, 87

PCR Polymerase Chain Reaction. 12, 14

polyA+ polyadenylated. 14, 141

PR AUC Area Under the Precision-Recall Curve. 28

ribo- ribosomal RNA-depleted. 14, 141

RNA ribonucleic acid. 9–11, 13–15, 36, 39, 41, 73, 127, 138–140, 142, 144

ROC AUC Area Under the ROC Curve. 29

ROC curve Receiver Operating Characteristic curve. 29

RPKM Reads Per Kilobase Million. 39

rRNA ribosomal RNA. 14

SBT Sequence Bloom Tree. 48

SMRT Single-Molecule Real-Time. 12

SNP Single Nucleotide Polymorphism. 11, 45

SNV Single Nucleotide Variant. 10, 11, 15, 139

SVM Support Vector Machine. 25, 26

T thymine. 9, 43

TCGA The Cancer Genome Atlas. 15

TN True Negative. 27

TP True Positive. 27, 28

TPM Transcripts Per Million. 39, 130, 131

tRNA transfer RNA. 14

Annex 1: Application of k-mer Approach on DNA-seq Data

My Contribution

My work in the project:

- Participation in the idea of matching counterparts of cs-$k$mers (Figure 9).

The Article

BMC Bioinformatics

SOFTWARE

Open Access

2-kupl: mapping-free variant detection from DNA-seq data of matched samples

Yunfeng Wang[1,3], Haoliang Xue[1], Christine Pourcel[1], Yang Du[3] and Daniel Gautheret[1,2*]

*Correspondence: daniel.gautheret@universite-paris-saclay.fr
[1] Institute of Integrative Cell Biology (I2BC), Université Paris-Saclay, CNRS, CEA, 1 avenue de la Terrasse, 91190 Gif-sur-Yvette, France
Full list of author information is available at the end of the article

Abstract

Background: The detection of genome variants, including point mutations, indels and structural variants, is a fundamental and challenging computational problem. We address here the problem of variant detection between two deep-sequencing (DNA-seq) samples, such as two human samples from an individual patient, or two samples from distinct bacterial strains. The preferred strategy in such a case is to align each sample to a common reference genome, collect all variants and compare these variants between samples. Such mapping-based protocols have several limitations. DNA sequences with large indels, aggregated mutations and structural variants are hard to map to the reference. Furthermore, DNA sequences cannot be mapped reliably to genomic low complexity regions and repeats.

Results: We introduce 2-kupl, a k-mer based, mapping-free protocol to detect variants between two DNA-seq samples. On simulated and actual data, 2-kupl achieves higher accuracy than other mapping-free protocols. Applying 2-kupl to prostate cancer whole exome sequencing data, we identify a number of candidate variants in hard-to-map regions and propose potential novel recurrent variants in this disease.

Conclusions: We developed a mapping-free protocol for variant calling between matched DNA-seq samples. Our protocol is suitable for variant detection in unmappable genome regions or in the absence of a reference genome.

Keywords: DNAseq, WGS, WES, k-mers, Contigs, Recurrent variants, PRAD, Mapping-free

Background

Searching for genomic variants is a fundamental aspect of medical research, whether in the study of Mendelian diseases or of somatic, cancer-related alterations [1]. While certain variants result in gene dysfunction and disease [2], others are largely asymptomatic but give rise to neoantigens relevant to immune escape and therapeutic efficacy or treatment [3]. Genome variants are also of interest in microbiology to analyze the differences between microbial strains [4] and reveal mechanisms underlying phenotypes. In this study, we address the problem of finding genomic differences between a matching pair of high throughput DNA sequencing (DNA-seq) datasets from the same individual (human somatic variation) or from two bacterial strains.

Genomic variants include mutations, indels and structural variants (SV). Mutations and indels can alter genes by disrupting the genetic code, while SVs, by pulling distant regions together or splitting one region into segments, can create chimeric genes or have a broader impact on whole chromosomal regions [5]. Variants are typically detected by whole-genome (WGS) or whole-exome (WES) sequencing through comparison with reference sequences. Aligners such as BWA [6] are first applied to map reads to the reference sequences. The variant calling step then detects differences between mapped reads and the reference. Popular variant callers include MuTect2 [7], VarScan [8], somaticsniper [9] and MuSE [10]. Based on variants observed between two sequence samples and a common reference genome, these programs can then infer differences between the two samples (e.g., in MuTect2's somatic mode).

Reference-based variant calling has well-known limitations. Aligners may encounter difficulties while handling reads with low mapping qualities [11], originating from repeat regions, low complexity regions or complex variants. These reads of low mapping quality are usually discarded. Furthermore, some species have no reliable reference, which is common in microbes [12].

Alternative approaches to variant calling involve mapping-free protocols [13]. These methods do not rely on a reference genome and can directly predict variants from the raw fastq file. A typical strategy is to use a de Bruijn graph (DBG) [14]. A DBG is constructed using k-mers (subsequences of fixed size k) decomposed from the sequence reads. The occurrence of k-mers harboring a mutant allele and a wild type allele generates a bubble structure in the DBG. Variant callers developed based on DBGs include DiscoSNP++ [15] and Lancet [16]. DBG-based methods also introduce new issues. First, complex genomic variants and repeats may result in complicated graphs that are difficult to parse [17]. Second, short contigs may be discarded at the post-processing step, where branch pruning may cause many false negatives. Furthermore, sequences assembled by k-mers without variants have little contribution if the purpose is detecting variants. Only reconstructing the active regions spanning the variants is more efficient than considering all k-mers [13]. Although it is possible to extend DBG-based methods to SV detection, the lack of sensitivity to local events makes these approaches less suitable for finding variants in ambiguous regions, such as repeats [18]. This motivates the need for a method to detect variants in arbitrary genome regions directly from DNA-seq data.

We present 2-kupl, a k-mer-based bioinformatics pipeline that compares matched case and control samples to discover case-specific variants. 2-kupl identifies sequence fragments (contigs) specific to the mutant dataset and their wild-type counterpart in the control dataset. This operation is done without relying on a reference genome. We compare the accuracy and CPU-requirements of 2-kupl with that of other variant calling software using both simulated and real DNA-seq datasets. We analyze the nature of novel variants detected by 2-kupl and potential reasons for their absence in conventional protocols. We also use 2-kupl to detect recurrent variants in prostate adenocarcinoma (PRAD) WES samples from the TCGA project [19]. Finally, we evaluate 2-kupl precision in bacterial WGS data. Overall, we demonstrate that 2-kupl is a practical and powerful alternative for the discovery of genomic variants in hard-to map regions or species with no reliable reference.

Results

A novel algorithm for detecting variants between two DNAseq samples

We developed 2-kupl to predict variants between pairs of matched DNAseq libraries. Input libraries consist of a "case" and a "control" sample such as a pair of tumor and normal tissues from one patient or a pair of mutant and wild-type bacterial strains. Data can be either WGS or WES. 2-kupl extracts case-specific k-mers (cs-kmers) and matching control k-mers (ct-kmers) corresponding to a putative mutant and reference sequences and merges them into contigs. As 2-kupl begins with a shortlist of cs-kmers, the number of k-mers considered from unaltered regions and non-specific variants is drastically reduced compared with DBG-based methods (see Methods). If a reference genome is provided, 2-kupl can also align contigs to the reference and generate genomic coordinates just like with mapping-based methods.

Performance on simulated WES data

We first applied 2-kupl to the detection of somatic mutations in a simulated human cancer WES dataset containing a known number of spliked-in mutations and indels. We compared 2-kupl with three other software, including two mapping-free methods (DiscoSNP++ and Lancet) and the leading mapping-based pipeline GATK-MuTect2. Results are summarized in the first column of Table 1. The number of cs-kmers to process is reduced by nearly 20% after data cleaning by 2-kupl.

88.6% of cs-kmers were matched to ct-kmer, corresponding to predicted point mutations or indels. We evaluated mutations and indel calls by 2-kupl and concurrent methods (Table 2). For mutation calling, 2-kupl performed better than the other mapping-free methods in terms of F1 score (Table 2). Lancet and GATK achieved better recall than 2-kupl, but Lancet also introduced more false positives. 2-kupl had a higher recall for calling indels than DiscoSNP++ and Lancet but was outperformed by DiscoSNP++ in FDR and precision (Table 3). Expectedly, GATK-MuTect2 outperformed all mapping-free approaches regardless of variant types. DiscoSNP++ did not perform as well as others in terms of recall ratio due to the different usage. DiscoSNP++ first pooled together two samples and screened case-specific variants afterwards. This procedure contributes to eliminate many false positives but also leads to ignoring some low frequency variants exclusively present in the case sample. Lancet performed well in terms of recall but at a high cost of false positives. As expected, most false positives had few reads containing

Table 1 Number of k-mers and contigs after applying 2-kupl on two matched libraries

	Simulated WES	TCGA-ZG-A9ND WES
All k-mers (tumor/normal)	465,718,268/465,610,133	184,233,006/177,517,776
Raw cs-kmers	23599	393525
Cleaned cs-kmers	18439	291350
Matched cs-kmers	16914	240360
All contigs	1245	106426
Mutations	1026	9901
Indels	112	1105
Unmapped	0	58
Low confidence	107	312

Table 2 Comparison of four approaches on mutations using simulated WES data

Mutations	2-kupl	DiscoSNP++	Lancet	GATK-MuTect2
True positive	581	373	604	689
False positive	45	3	126	2
False negative	241	530	218	133
Recall	0.71	0.41	0.73	0.84
FDR	0.07	0.01	0.17	0.003
Precision	0.93	0.99	0.83	0.997
F1 score	0.80	0.58	0.78	0.91

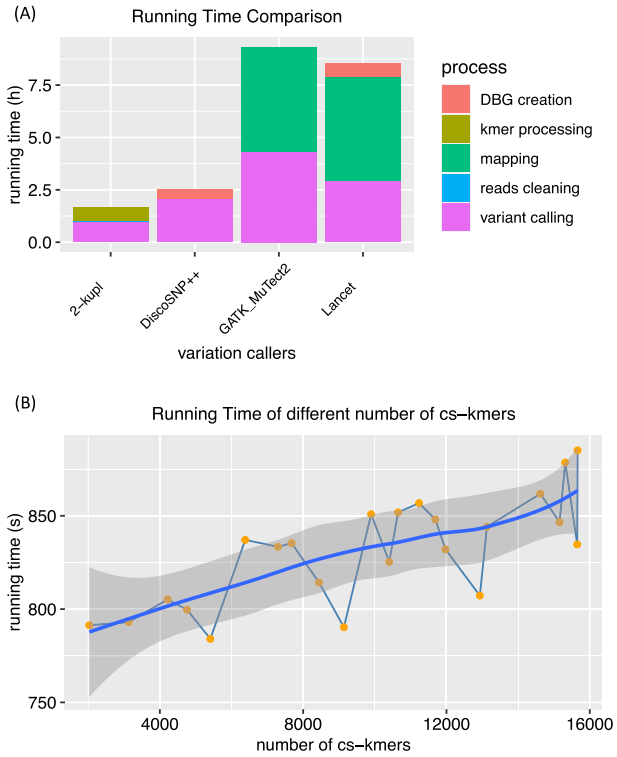


Fig. 1 Running time and performance with different types of variants. **a** Overall running times of four software. The time consumed by each process in four protocols is marked in different colors. **b** Running times of 2-kupl for different numbers of cs-kmers. The line with dots represents the exact running time corresponding to certain number of cs-kmers. The solid line is the fitted line, and the shaded background is the confidence interval

the alternative allele, which is frequent with Lancet. The high recall and high rate of false positives produced by Lancet are consistent with the conclusions of Meng and Chen [20]. The GATK-MuTect2 pipeline outperformed all mapping-free approaches when calling mutations. The use of a reference sequence and the Haplotype Caller algorithm gives GATK-MuTect2 a clear advantage. Even though 2-kupl got a relatively lower recall than GATK-MuTect2, it had better control of the false positives and got a higher precision when calling indels (Table 3).

Another advantage of 2-kupl is the short running time (Fig. 1a). 2-kupl took 1.6 h to analyze the simulated WES data with default parameters. DiscoSNP++ took

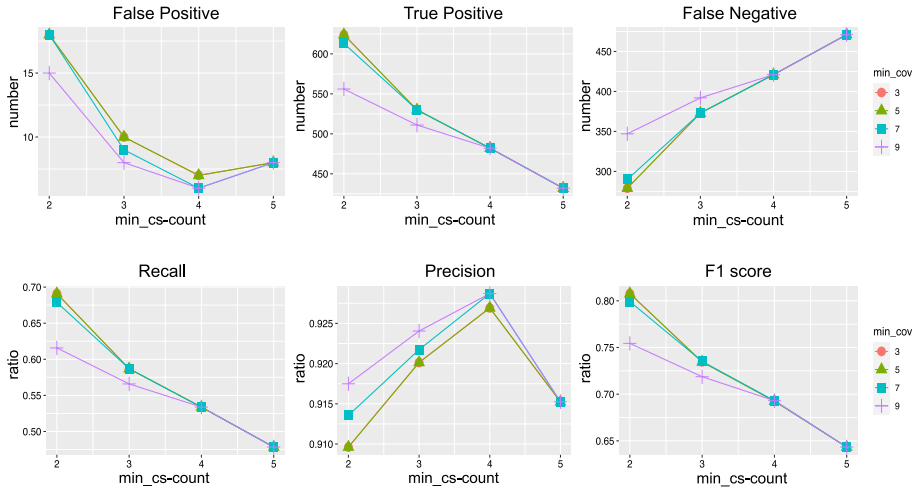


Fig. 2 Robustness of 2-kupl using different parameters. The x-axis indicates the min_cs-count parameter and the y-axis represents the corresponding ratio or number. The thresholds of coverage and cs-count are denoted as min_cov and min_cs-count, respectively. The trend lines under different min_cov parameters are represented by four colors

Table 3 Comparison of four approaches on indels using simulated WES data

indels	2-kupl	DiscoSNP++	Lancet	GATK-MuTect2
True positive	42	29	40	49
False positive	16	1	44	26
False negative	39	52	41	32
Recall	0.52	0.36	0.49	0.60
FDR	0.27	0.03	0.52	0.35
Precision	0.72	0.97	0.47	0.65
F1 score	0.60	0.52	0.48	0.63

2.54 h to call variants from both case and control samples. Both Lancet and GATK-MuTect2 require prior mapping of reads to the human genome (which takes 3.17 h), explaining in part their longer runtimes.

To evaluate 2-kupl run time dependency on the number of cs-kmers, we ran 2-kupl on datasets with different numbers of cs-kmers (Fig. 1b). Running time increased linearly with the number of cs-kmers. Each additional 10,000 cs-kmers increased the running time by nearly 50 s.

We estimated the performance of 2-kupl under different parameter combinations. Coverage and cs-count thresholds ('mim_cov' and 'min_cs-count', respectively) were varied from 3 to 9. Results are shown in Fig. 2. The min_cs-count parameter was negatively related to recall and positively related to false negatives. The min_cov parameter was inversely related to F1 score, recall, FDR, and true positives. Precision reached an inflection point when min_cs-count was set to 4.

Performance on simulated WGS data

We further benchmarked 2-kupl on a simulated WGS dataset with an average read depth of 50X (vs. 230 in WES). For mutation calls, 2-kupl and GATK-MuTect2 achieved the same recall ratio of 0.86 (Table 4). The precision of 2-kupl was slightly lower than GATK-MuTect2 but still above 0.9. For indels, the recall of 2-kupl dropped to 0.82 (Table 5). The false positive call rates of 2-kupl increased with WGS data relative to WES data due to the lower coverage of WGS. A limitation of 2-kupl is that false signals can not be ruled out by allele frequency in low coverage regions. Also, k-mers may be incorrectly considered as cs-kmers when there is not enough reads covering the locus in the control sample.

The simulated WGS dataset contained 157 SVs (deletions, duplications, and translocations longer than 50bp). Expectedly, GATK-MuTect failed to detect the majority of SVs (Table 6). We thus compared 2-kupl with Delly, a software that finds structural variants based on aligned reads [21]. Overall 2-kupl had a slightly lower precision and recall than Delly (Table 6). We investigated 22 SVs missed by Delly and captured by 2-kupl. We found these reads were left unmapped by BWA due to multiple hits in the genome and thus could not be assessed by Delly (Additional file 6: Table S5). An advantage of 2-kupl here is that all k-mers covering SV junctions are kept and assembled regardless of mapping status. Furthermore, 2-kupl is capable of detecting small variants in the same run.

Assessing 2-kupl on a real normal-tumor WES dataset

To assess 2-kupl results on actual WES data, we applied 2-kupl on one WES dataset of matched tumor and normal tissues from the TCGA-PRAD dataset. We first compared 2-kupl and GDC portal somatic variant calls (see Methods) on the TCGA patient with the highest tumor mutational burden. The numbers of k-mers, contigs and variants obtained by 2-kupl are shown in the second column of Table 1. Mutation calls by 2-kupl and GDC portal variants are shown in Table 7. Although total call numbers were similar, only 327 calls (9%) were shared by the two approaches, including 319 mutations and 8 indels. Among the variants detected by 2-kupl, 193 (5.13%) mapped to noncoding regions and 101 (2.7%) were annotated as repeats by RepeatMasker [22]. 2-kupl also captured 57 (1.5%) unmapped variants. 173 2-kupl variants (4.6%) were mapped to low mappability "blacklist" regions [23]. In spite of the small general overlap of 2-kupl and GDC portal variants, the two methods have a much stronger agreement on high scoring 2-kupl calls (Additional file 1: Fig. S1A). Of note, mutation calls obtained on the same sample by four different mapping-based protocols also show poor consistency (Additional file 1: Fig. S1B).

We further analyzed mutations specific to 2-kupl. These calls may have been rejected in GDC portal variants for a number of valid reasons, including low mapping quality, location in short tandem repeats or presence in normal samples. A real "miss" by the reference-based pipeline should be recorded only when reads could not possibly be aligned to the genome while they indeed contained a valid mutation.

Figure 3a shows a case of false positives introduced due to artifactual cs-kmers. Generally, k-mers harboring a mutation present in both tumor and normal tissues are supposed to be ruled out. However, erroneous tumor-specific "cs-kmers" can escape the

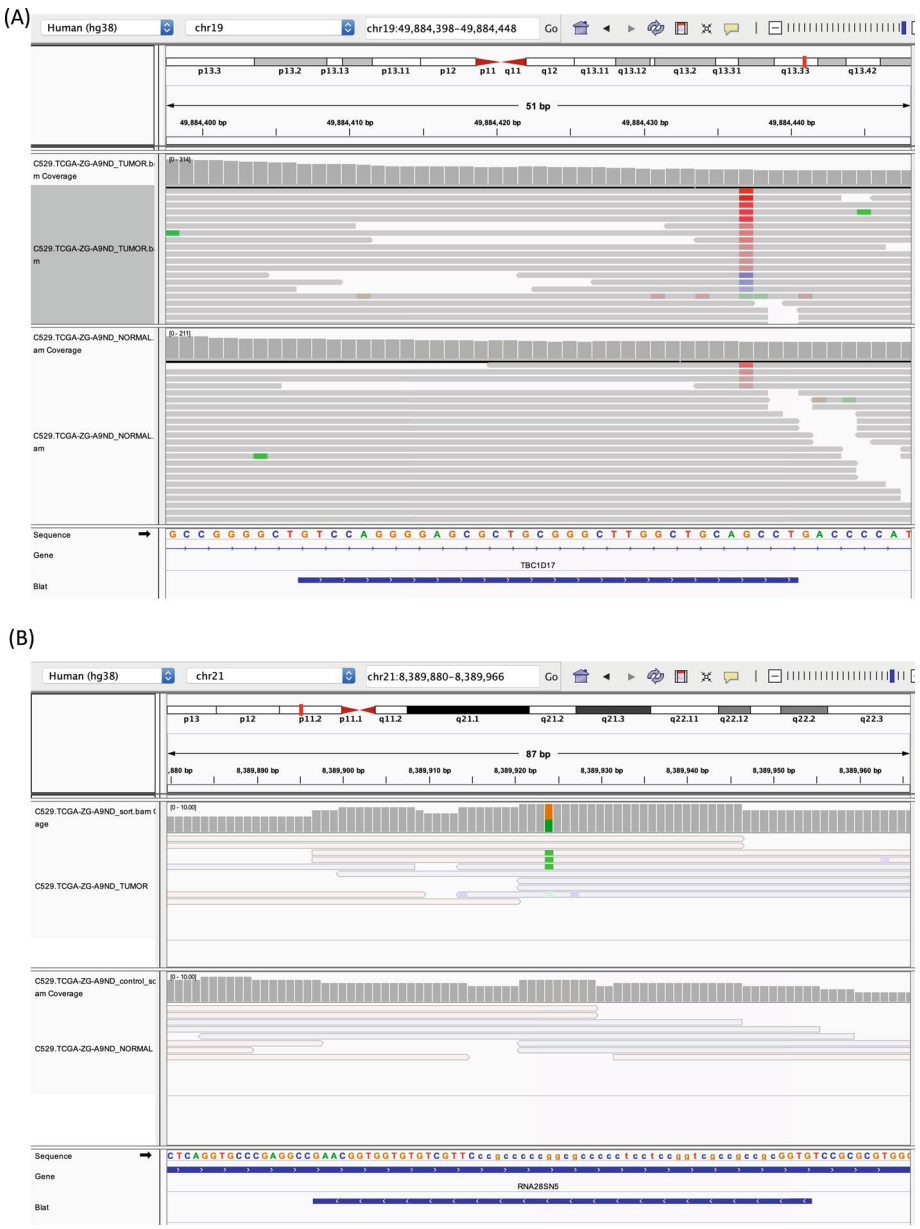


Fig. 3 IGV views of variant calls in TCGA-PRAD WES dataset. The two central tracks show aligned reads from the tumor (top) and normal (bottom) WES library. The lower track shows gene annotation and 2-kupl contigs. **a** A likely false-positive call by 2-kupl at a position of low mapping quality, **b** A likely true positive within a repeat region. Reads in transparent color have low MAPQ (mapping quality) values (<10)

filtering process if the same k-mer in the normal tissue happens to be low quality and is discarded.

Certain 2-kupl specific mutations are possibly true positives discarded by mapping-based protocols due to their location within a repeat region. Figure 3b shows such a potential somatic mutation. The mutation is located within a ribosomal RNA gene that is repeated multiple times in the genome and further contains a C-rich repeat (represented in lower cases). Reads generated from these repetitive regions are given low

Table 4 Comparison of 2-kupl and GATK-MuTect2 on mutations using simulated WGS data

mutations	2-kupl	GATK-MuTect2
True positive	13835	13920
False positive	1248	30
False negative	2220	2135
Recall	0.86	0.86
FDR	0.08	0.002
Precision	0.91	0.99
F1 score	0.89	0.93

Table 5 Comparison of 2-kupl and GATK-MuTect2 on indels using simulated WGS data

indels	2-kupl	GATK-MuTect2
True positive	3315	3620
False positive	504	108
False negative	750	445
Recall	0.82	0.89
FDR	0.13	0.02
Precision	0.84	0.96
F1 score	0.84	0.92

Table 6 Comparison of 2-kupl, GATK-MuTect2 and Delly on structural variants using simulated WGS data

mutations	2-kupl	GATK-MuTect2	Delly
True positive	133	49	135
False positive	27	0	16
False negative	24	108	22
Recall	0.85	0.3	0.86
FDR	0.17	0	0.11
Precision	0.83	1	0.89
F1 score	0.84	0.47	0.88

Table 7 Number of mutations and indels detected by 2-kupl and GDC portal variants

	2-kupl	GDC portal variants	overlap
Mutation	3607	3093	319
Indel	151	823	8
Total	3758	3916	327

MAPQ values by mappers and variants in these regions are then discarded by variant callers.

Among unmapped 2-kupl calls, only one has a Phred score in the top 5% (Additional file 1: Fig. S2). The mutant sequence and its inferred reference are shown in Additional file 1: Fig. S3. The mutant contig is covered by 0 and 47 reads in the Normal and Tumor

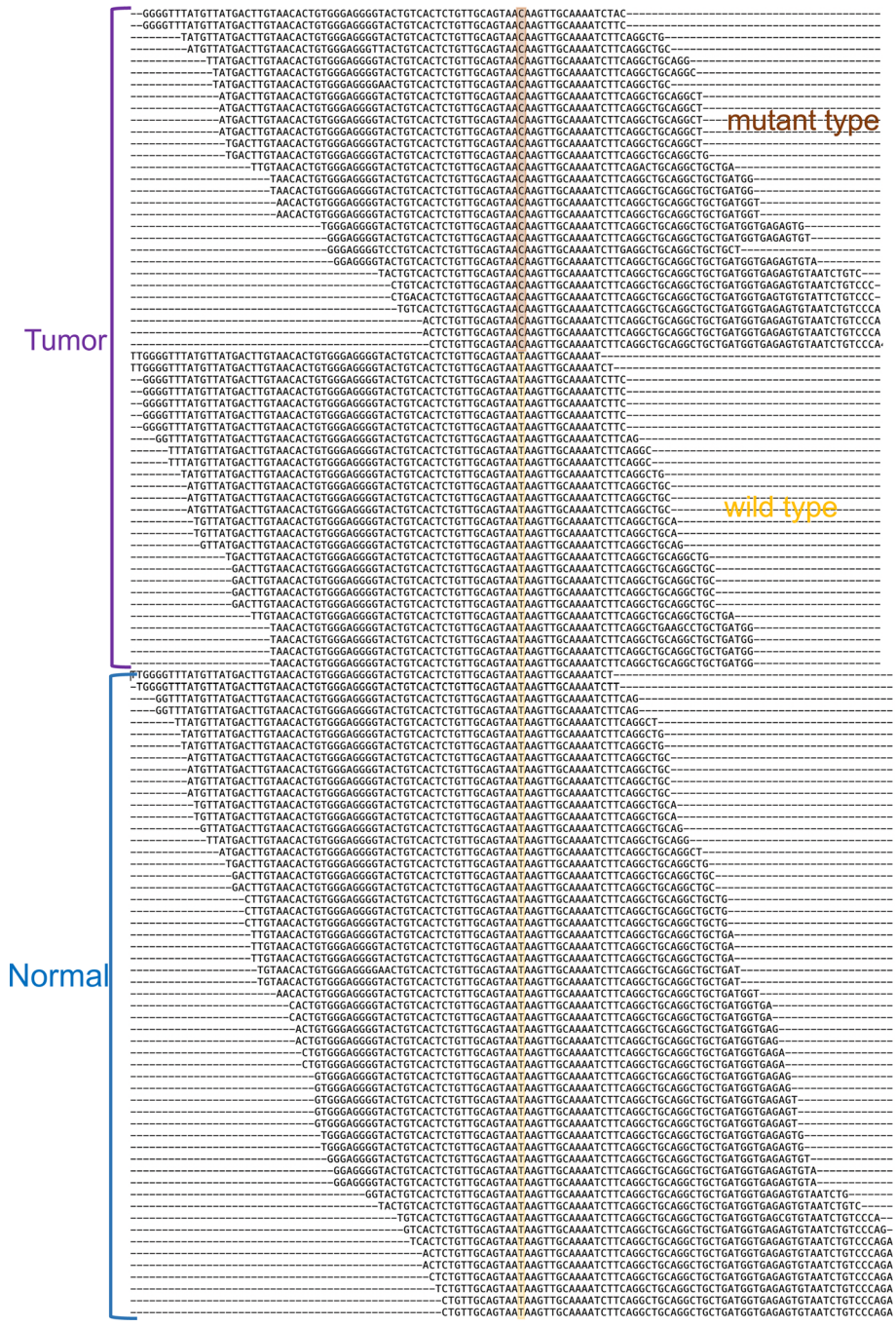


Fig. 4 An unmapped somatic variant from a TCGA PRAD patient. Only reads matching the central k-mer of the tumor-specific variant or its inferred counterpart are shown. Reads from the tumor and normal samples are distinguished. The position of variation is highlighted

sample, respectively while the reference is covered by 88 and 65 reads in the Normal and Tumor sample, respectively (Fig. 4). The sequence maps to a centromeric repeat of Chr22, with three mismatches. The mapping procedure would thus miss this highly significant variant.

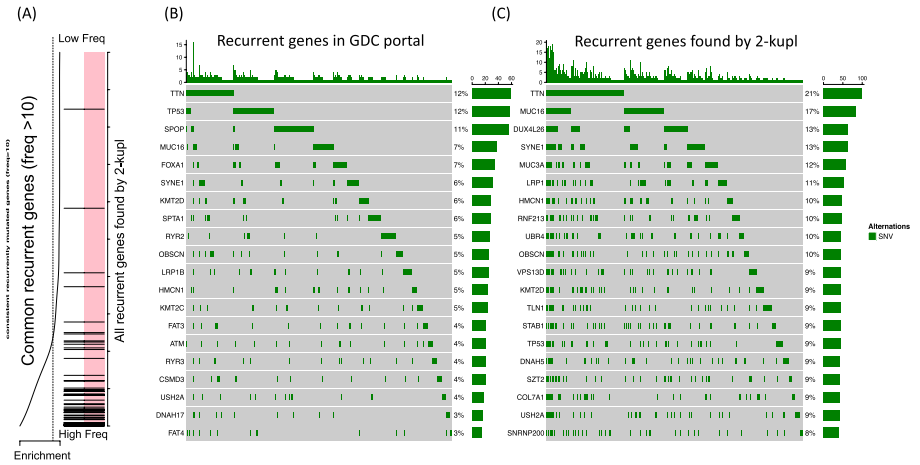


Fig. 5 Recurrently mutated genes in the TCGA-PRAD WES dataset. **a** Enrichment analysis of recurrent genes. The vertical bars are the common recurrently mutated genes (altered in at least ten patients) between GDC portal and 2-kupl. The x axis represents the recurrent genes found by 2-kupl sorted by frequency. The smooth curve reflects the degree to which the common genes are overrepresented in the whole 2-kupl recurrent genes. **b** The 20 genes with the highest mutational frequency detected in GDC portal variants. **c** The top 20 recurrent genes with the highest mutational frequency detected by 2-kupl

Recurrent mutations in TCGA-PRAD

Recurrence across patients is a powerful criterion for distinguishing drivers from passenger mutations [24–26] and has been used to discover drivers and define molecular subtypes of prostate cancer [27]. We applied 2-kupl to each pair of Normal/Tumor samples in the complete PRAD WES dataset (N=498) and identified 3211 recurrent variants (Additional file 2: Table S1). For comparison we retrieved from the GDC portal recurrent variants predicted for the same dataset (GATK-MuTect2 pipeline, see Methods). Among 3734 recurrent variants in the GDC portal, 854 were shared with 2-kupl recurrent variants (Additional file 2: Table S1). We further compared the recurrent variants to a comprehensive dataset of recurrent prostate cancer mutations from Fraser et al. [28] based on 200 whole-genome and 277 whole-exome sequences from multiple sources. Comparisons were restricted to exonic regions. Within the 48 recurrent mutations in exonic regions from Fraser et al, a similar number was shared with 2-kupl or the GDC-portal (22 and 21, respectively) (Additional file 3: Table S2). Among recurrent mutations specific to 2-kupl, we note the one found at chr14:37592023 within an exon of FOXA1, a putative prostate cancer driver [29], in three TCGA-PRAD patients.

We further compared 2-kupl calls to GDC portal variants at the level of genes (Detailed in Method section). The GDC portal reported 6944 genes mutated in two or more patients, versus 14137 recurrent genes by 2-kupl. Enrichment analysis shows a good convergence of the most frequently mutated genes by the two methods (Fig. 5). Figure 5b, c show oncoplot views of the top 20 genes according to the GDC portal and 2-kupl, respectively, showing eight shared genes. Both gene lists are contaminated by long (TTN) or highly polymorphic genes (Mucins) whose recurrence is an artifact due to higher mutation counts. Although many software are available to account for those effects [30], we purposely analyze the uncorrected list of genes here. Among the top 20 mutated genes by 2-kupl and GDC portal, 7 and 9 genes, respectively, are known

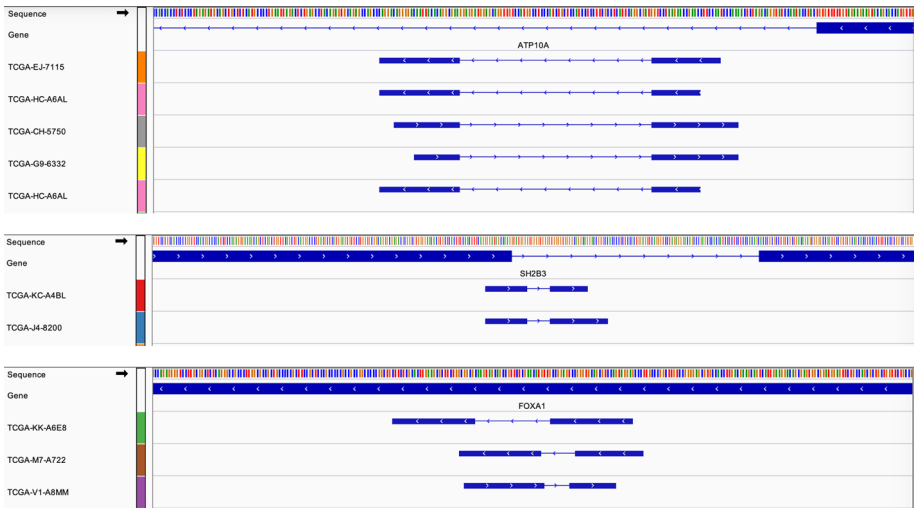


Fig. 6 Recurrent structural variants mapping to three prostate cancer genes. In each track, lines represent the genome sequence (top), annotated genes, and variant contigs identified in different patients

prostate cancer-related genes. Among those, UBR4, DNAH5 and LRP1 were only detected by 2-kupl. When considering the top 50 recurrently mutated genes according to 2-kupl and GDC portal, 19 and 23, respectively, are cancer-related. Among those, HSPG2, DNAH3, UBR4, COL6A3, CABIN1, IGF2R, PTPRF, DNAH5, HTT and TRRAP were only detected by 2-kupl.

UBR4 contains 48 2-kupl mutations, more than any other gene. Additional file 1: Figure S4 shows read alignment at this gene for patient TCGA-EJ-7125 who carries the most UBR4 mutations (8/48 mutations). While seven of these mutations are absent in GDC portal variants, all can be visually validated as tumor-specific mutations as per the IGV display (Additional file 1: Fig. S4 A-G).

Besides recurrent mutations and indels, we found 20 genes with 43 recurrent structural variants predicted in at least two patients (Additional file 2: Table S1). All these predicted variants can be supported by at least one read from the tumor library. Three recurrent structural variants map to prostate cancer genes SH2B3, ATP10A and FOXA1 (Fig. 6). Variants in gene ATP10A and SH2B3 have exactly the same junctions in at least two patients. As the three variants in gene FOXA1 impact on the same exon, we grouped them as one same recurrent event despite not representing the exact same variation. All these recurrent structural variants are longer than 10bp. State-of-the-art procedures usually miss such variants at the mapping stage.

Performance on bacterial WGS data

2-kupl can be applied to pairwise comparisons of DNA-seq datasets in any species. We present here an application to bacterial whole genome sequences. A frequent problem in bacterial genetics is identifying mutations in strains for which no reliable reference genome is available. We investigated the performance of 2-kupl on 21 DNA-seq datasets

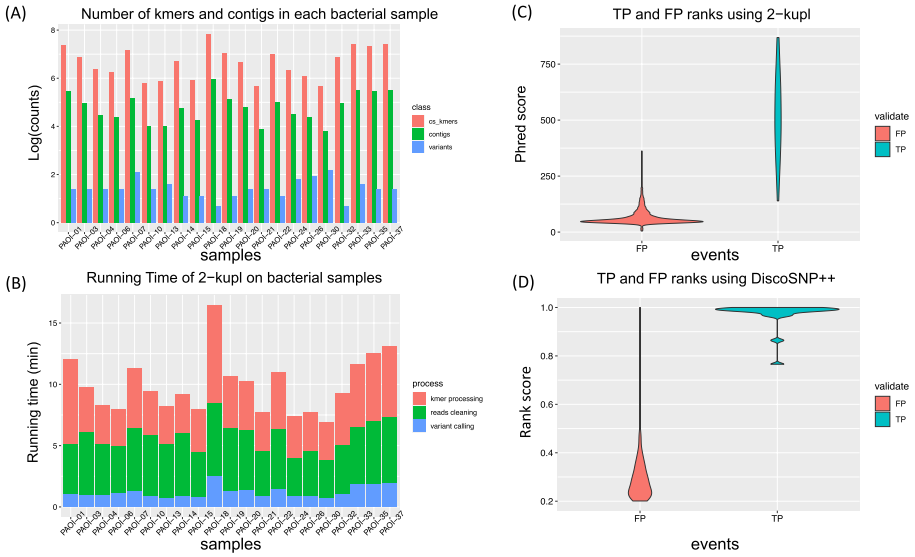


Fig. 7 Performance of 2-kupl on bacterial DNA-seq datasets. **a** Number of cs-kmers, contigs and variants are shown for each bacterial sample. **b** Running time of 2-kupl on each sample is shown for different steps. **c** Distribution of Phred scores computed by 2-kupl in TP and FP events. **d** Distribution of DiscoSNP++ score ranks in TP and FP events

Table 8 comparison between 2-kupl and DiscoSNP++ on the bacteria DNA-seq data

	2-kupl	**DiscoSNP++**
True positive	26	23
False positive	45	129
False negative	0	3
Recall	1	0.88
FDR	0.64	0.85
Precision	0.36	0.15
F1 score	0.52	0.26

from a *Pseudomonas aeruginosa* strain, in which 26 variants had been previously identified and confirmed by geneticists (see Methods).

About 141 variant contigs were predicted on average for each pair of WT/mutant strains, with an average running time of 10 minutes (Fig. 7a, b). Score ranking by 2-kupl and DiscoSNP++ allowed a clear separation of TP from FP (Fig. 7c, d). True positive calls were ranked first in 19 out of 19 mutant samples by 2-kupl and in 16 out of 16 samples by DiscoSNP++. Compared with Phred scores used in 2-kupl, DiscoSNP++ scales the rank scores from zero to one and thus the true positive variants are more concentrated.

2-kupl could recall all true positive variants, including SNVs and large deletions longer than 100 bp, while DiscoSNP++ missed three large deletions (555 bp, 213 bp and 109 bp, Additional file 5: Table S4). Meanwhile, DiscoSNP++ obtained 129 false positives versus 45 for 2-kupl (Table 8). Therefore 2-kupl had the best recall and precision on this dataset, especially for large indels.

Discussion

Most variant detection protocols rely on reference genomes. However, even for species with a high-quality reference genome such as humans, depending on a reference is subject to limitations. Genomes contain large numbers of highly variable, repetitive or otherwise unmappable regions, which are unsolvable by short-read sequencing techniques. Hundreds of unsolved regions remain in telomeres and centromeres, also known as 'dark matter' [31]. The X chromosome is the only complete human chromosome as of today [32]. Pathogenic variants within these unannotated regions are easily missed by mapping-based approaches due to low mapping quality, especially with low depth in whole-genome sequencing. Furthermore, the human genome varies across individuals and populations and a single reference genome does not account for this diversity [33].

2-kupl is able to detect variants, including mutations, indels and structural variants, without relying on a reference genome. Based on matched DNA-seq data, 2-kupl captures case-specific k-mers and counterpart k-mers (i.e. without the variation) into the same bucket. Sequence contigs harboring a local variation and its putative reference are inferred through the assembly of k-mers in each bucket.

To control artifacts induced by sequencing errors, 2-kupl takes both base quality and coverage into account. The general sequencing error rate in short-read NGS data is larger than 0.1% [34]. It is worth consuming computing resources and running time to remove these 0.1% artifacts because these sequencing errors result in large numbers of artifactual cs-kmers. To reduce the impact from low-quality bases, we combine Cutadapt and an 'OverrideN' function that flags low quality bases in the mid part of reads. This significantly reduces the number of cs-kmers and speeds up the computing procedure.

We compared the performance of 2-kupl with that of three competing methods in terms of running time, recall and precision. 2-kupl outperformed mapping-free methods DiscoSNP++ and Lancet in terms of recall or precision but did not reach the performance of the state-of-the-art alignment-based GATK-MuTect2 on human data.

DiscoSNP++ suffers from limitations of DBG data structures in regions with sequencing errors, genomic variants and repeats [18]. Efficient solutions searching for bubbles from such complicated structures are still under development. Furthermore, short contigs may be discarded within the post-process, cutting branches, for instance [35]. In our bacterial DNA-seq analysis, DiscoSNP++ missed three validated large deletions.

Lancet has a higher recall ratio than 2-kupl but also introduces more false positives. Furthermore, Lancet missed variants from repetitive regions and is not able to detect fusions from distant regions.

2-kupl has a higher F1 score than DiscoSNP++ and Lancet and performs better in terms of recall ratio or precision than either of them. Expectedly, 2-kupl did not outperform GATK-MuTect2 on WES data. First, GATK-MuTect2 uses a sophisticated Bayesian model to estimate a genotype's likelihood given the observed sequence reads that cover the locus. When GATK-MuTect2 encounters a region showing signs of variation, it discards the existing mapping information and completely reassembles the reads in that region. This allows GATK-MuTect2 to be more accurate when calling regions that are traditionally difficult to call. Despite slightly fewer true positives, 2-kupl also detects fewer false positives than GATK-MuTect2. It is worth mentioning that 2-kupl has the lowest time complexity among the four methods.

By applying 2-kupl to the TCGA-PRAD patients, we were able to detect recurrent mutations and indels missed by the GDC portal's GATK-MuTect2 pipeline. Reads in these regions have either low mapping qualities or multiple hits and were discarded in the GDC portal pipeline. Mapping-based methods all suffer from this issue and are powerless when faced with low complexity regions. 2-kupl identified recurrent mutations and recurrently mutated genes in high agreement with GATK-MuTect2. Mutated genes were enriched in PRAD-related genes, some of which specific to 2-kupl. As an example, we visually confirmed multiple 2-kupl-specific mutations in UBR4. Recurrent variants detected from the unmappable regions by 2-kupl provide insights into potential novel somatic variants even though the locus of origin of the contig sometimes cannot be determined.

Standard variant calling pipelines may miss mutations for multiple reasons: low allele frequencies, tumor contamination, ambiguities in short read alignment, inadequate sequencing depth, high GC content, sequencing errors and ambiguities in short read alignment. Different programs are affected by these factors to varying degrees. As a consequence, the mutations called by different pipelines are not consistent [36]. 2-kupl is not affected by some of these sources (GC content, alignment artifacts and mappability) and can detect a number of recurrent mutations (ie. potential driver events) that are not found by standard pipelines.

Several natural directions exist for extending 2-kupl. First, 2-kupl lacks sensitivity in detecting structural variants. All cs-kmers covering the junction are retained and extended to contigs. Unfortunately, neither the ct-kmers nor the reads are easily obtained when considering a hamming distance of one. A structural variation can be detected only if enough supporting reads are covering at least one side of the variation. Focusing on the cs-kmers regardless of ct-kmers could address this problem but at the cost of more false positives. A second limitation occurs when control samples are contaminated with tumor cells, which is relatively frequent in tissue biopsies. To address this problem, 2-kupl includes a parameter representing a k-mer count threshold in the control sample. However, a fixed contamination threshold may introduce unwanted non-specific variants. Future works should evaluate probabilistic approaches to address this issue.

Conclusions

In conclusion, the identification of different kinds of variants, using DNA-seq data, remains challenging. The leading protocols developed for DNA-seq highly rely on the reference. In general, the methods that align sequencing data to the reference (mapping-based methods), perform better than do the mapping-free methods. However, 2-kupl can capture events falling into the difficult-to-map regions, and can perform better than other mapping-free protocols. 2-kupl is the fastest tool in the comparison with other methods because the mapping procedure is not included. The high agreement in top ranking variants by 2-kupl and GDC portal variants indicates the capacity of using 2-kupl as an extension and supplementation of the mapping-based methods.

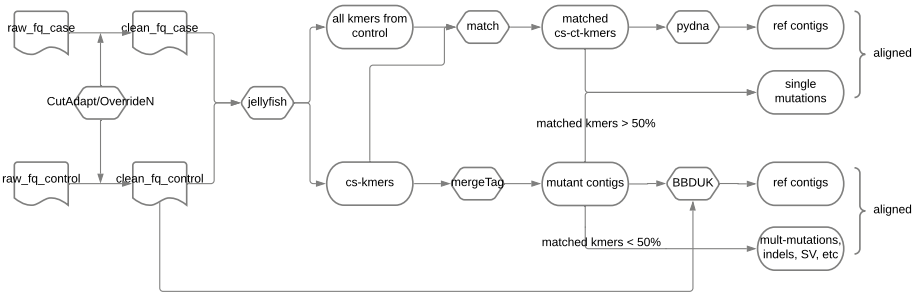


Fig. 8 Overall workflow of 2-kupl. This flowchart describes the analysis process of 2-kupl, including the input and output file format and function of each module

Methods

Outline of 2-kupl pipeline

The general pipeline is presented in Fig. 8. The input is composed of DNA-seq data from two matched samples. Samples typically correspond to control/normal/wild-type and a case/tumor/mutant-type. For cancer data, we strongly recommend using as a control of a distant tissue such as white blood cells rather than adjacent normal tissues, as the later can be contaminated by tumor cells and 2-kupl only considers variant sequences that are absent in the control dataset. Sequence types can be either single-end or paired-end sequencing reads. 2-kupl then identifies pairs of case-specific k-mers (cs-kmers) and counterpart k-mers (ct-kmers). 2-kupl returns predicted variants exclusive to the case sample, including mutations, indels and structural variations. Variant statistics including cs-count, coverage, allele frequency and variant P-value are computed. A variant file and an alignment file are produced. 2-kupl accepts multiple threads and uses 10 threads by default.

2-kupl is developed purely in Python. The main dependencies include Jellyfish [37] and GSNAP [38]. Other dependent python libraries and instructions can be found from the Github repository https://github.com/yunfengwang0317/2-kupl

Data cleaning

Low quality sequences are trimmed with Cutadapt [39] (parameter '–quality-cutoff' = 10). As Cutadapt does not remove low-quality bases within the central part of reads, we implemented an overriding function that replaces each low-quality base (Phred score<10) with N. This procedure is applied to both case and control libraries.

k-mer indexing and counting

Jellyfish is used to index and quantify k-mers from both case and control with options k=31 and -C (canonical k-mers). As Jellyfish removes k-mers containing Ns, none of the low-quality bases is present in the k-mer list. The generated k-mers subsequently undergo two filtering steps. First, k-mers with counts below a user-specified cut-off (default=3) are removed. These low abundance k-mers are assumed to result from sequencing errors or off-target regions in the case of WES data. Second, k-mer lists from case and control are compared and only case-specific k-mers (cs-kmers) are retained.

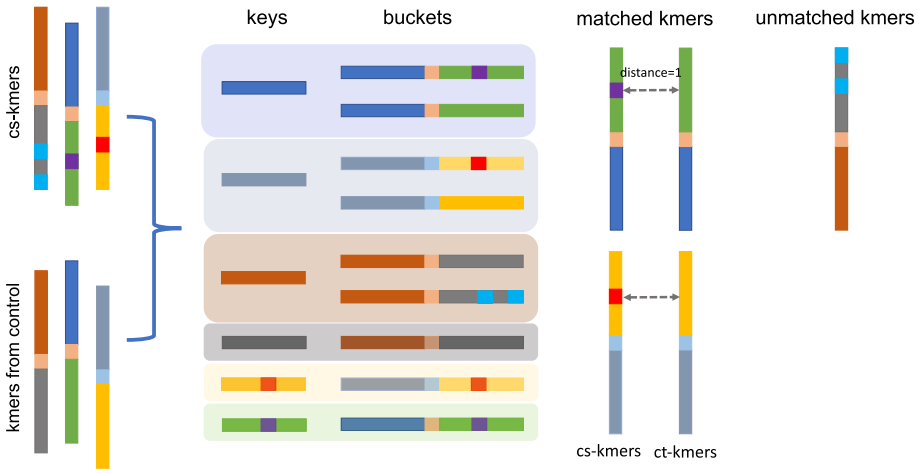


Fig. 9 Procedure for matching cs-kmers to ct-kmers. Long rectangles represent one 31-mer. Short rectangles (keys) represent the head or tail 15 bp of a cs-kmer. Color changes indicate sequence differences

Matching counterparts of cs-kmers

For each cs-kmer harboring a point mutation, there should exist a counterpart k-mer (ct-kmer) from the control dataset with only one base substitution (Hamming distance =1), which can be considered as a product of the wild type sequence. Note that Hamming distance=1 only considers substitutions. Hence single nucleotide insertions and deletions are rejected at this step and will be treated later with unmatched k-mers. Finding the matched ct-kmer for each cs-kmer should allow us to infer the variation without reference sequences. We initially build a hash table where the keys are the continuous 15 bases from each side of cs-kmers. For each 15-bases key, we create a bucket of all k-mers starting or ending with the key. Then we survey the buckets and seek all k-mer pairs with a hamming distance of one in the same bucket. We thus generate all k-mer pairs (ki, kj) with a hamming distance of one. For any pair of k-mers with a Hamming distance of one, if one k-mer comes from the cs-kmer list and the other comes from the control, this pair of k-mers is considered to be matched. Otherwise, we allocate the cs-kmers to the "unmatched k-mers" group. These unmatched k-mers either contain variants of more than one nucleotide (multiple mutations, indels and structural variants) or come from low coverage regions. The schematic workflow is shown in Fig. 9.

Assembly of cs-kmers into mutant contigs

cs-kmers are assembled into mutant contigs that correspond to variants and their local context. The assembly process is done using the "mergeTag" function from DEkupl [40] (https://github.com/Transipedia/dekupl). Two k-mers overlapping by k-i bases are merged iteratively with i ranging from 30 to 25 (min_overlap parameter is set to 25 by default). The merging process is interrupted when no k-mers can be added or ambiguity occurs (two different overlapping k-mers are encountered).

Inferring reference contigs

We use two distinct procedures for reference sequence determination, depending on whether or not sufficient ct-kmers are available to build a reference contig.

For each mutant contig, if more than half of its component k-mers are matched, all the ct-kmers are merged by the python package pydna [41]. The resulting mutant contigs correspond to isolated mutations. Merged contigs produced by ct-kmers can be regarded as putative references. For each pair of mutant and reference contig, we then define two values representing counts of supporting k-mers for the mutant allele (cs-count) and supporting k-mers for both mutant and reference alleles (coverage). The cs-count is computed from the median k-mer count of cs-kmers and coverage is calculated from the sum of the median count of cs-kmers and ct-kmers. Herein, we select the median count instead of the mean count because mean values are more sensitive to high-count k-mers from repeats or copy number amplification regions.

For mutant contigs in which less than half of the k-mers are paired, we consider that a reference cannot be assembled from paired-kmers. A procedure was implemented to retrieve the reference from the original reads. Reads with at most one mismatch to any k-mer from the mutant contig are retrieved from the control fastq file using BBDUK [42]. These reads are then assembled by CAP3 [43]. In this way, we can infer the putative reference for each contig and evaluate coverage based on the number of reads retrieved by BBDUK. The cs-kmers in these contigs have no matching ct-kmers and contigs are thus considered to contain multiple mutations, indels and structural variants (Additional file 6: Table S5).

Filtering low-quality variants

The cs-count and coverage substantially impact the reliability of events called by 2-kupl. For instance, a sequencing error could be repeatedly generated in a region of high coverage. Besides, sequencing errors may, by chance, be detected as mutations with high allele frequency in low coverage regions. Thus, false positives are introduced due to either high cs-count in high coverage regions or high allele frequency in low coverage regions. However, coverage varies between whole-genome sequencing (WGS) and whole-exome sequencing (WES) data. WGS does not use an upfront enrichment step so it generates a more uniform coverage of the genome. On the other hand, the enrichment steps involved in WES lead to non-uniform coverage, generating coverage 'hot' and 'cold' spots [44]. 2-kupl provides several criteria for users to evaluate call reliability. A Fisher's exact test P-value is calculated based on the cs-count and coverage in case and matched control libraries for each variation. A Phred quality score is subsequently computed as $-10log_{10}$P. Users can specify cutoffs for cs-count, coverage, allele frequency and Phred to filter false positives. Default cutoffs for cs-count, coverage, allele frequency and Phred are set to 3, 10, 0.05 and 5, respectively.

VCF format export

Events identified by 2-kupl are exported as a variant call format (VCF) file [45]. 2-kupl outputs the contig harboring the variation and the corresponding putative reference without the variation for each event. If users provide an available reference, the mutant

contig is mapped to this reference using GSNAP [38]. After the mapping process, actual chromosome and position information are provided in the VCF file. Besides the VCF file, 2-kupl also exports an alignment of each contig and its putative reference obtained using the pairwise2 python package [46]. Contigs corresponding to indels and structural variants are further mapped to reference by BLAST [47] (default parameters) which we found better suited to fragmented alignments.

Comparison with other software

DiscoSNP++ [15] is designed for detecting SNVs and small indels from fastq files without using reference. DiscoSNP++ first generates a DBG of two matched samples pooled together [48] and detects variants based on searching bubbles in the graph. The context contigs can be extracted from DBG bubbles that correspond to local variants. As DiscoSNP++ calls variants in each sample rather than specific to one sample, we applied cutoffs to DiscoSNP++ allele frequencies (AF) to extract case-specific calls as found by 2-kupl. After testing multiple combinations, DiscoSNP++ achieved the best performance when AF cutoffs for both case and control samples were set to 0.05. Lancet [16] relies on localized colored DBG to detect somatic variants in paired samples. K-mers shared by two matched samples or specific to either of them are marked in different colors in the DBG. In this way, Lancet is able to detect case-specific events. It is worth mentioning that Lancet uses bam format files as input so it also leverages the reference before variant detection. We also compared 2-kupl with the leading reference-based GATK-MuTect2 pipeline [7]. GATK-MuTect2 takes mapped sequence files as input, detects variants based on the reference and compares the variants of two matched samples to identify case-specific variants (somatic mode). Version hg38 of the human genome was used in all reference-based procedures. To make runtime comparisons fair, we took the mapping procedure into account in Lancet and GATK-MuTect2. Alignment was performed using BWA with default parameters. Thus all four protocols started with fastq files. To evaluate the dependency of 2-kupl running time on the number of k-mers, we ignored the part up to k-mer counting. Mapped reads were visualized with the Integrative Genomics Viewer (IGV) [49] 2.6.2 on hg38. For structural variant detection in simulated WGS data, we also compared 2-kupl with Delly [21] a structural variant discovery software. Delly uses BAM alignment files as input and infers structural variants at single nucleotide breakpoint resolution using both insert size and split reads information.

Simulated WES analysis

We downloaded simulated WES data from Meng and Chen [20]. This dataset was developed based on the NA12878 pilot genome [50] (reference data set of 5.4 million phased human variants validated by genetic inheritance from sequencing a three-generation 17-member pedigree). The authors used BAM-Surgeon [51] to select genomic loci and introduce random SNV and indel spike-ins, and generated 2x100nt reads WES files at 230X coverage. For our benchmark, we used a tumor sample described by authors as one of the most complicated, NA12878_79_snv_indel_sorted.bam (with four sub-populations, expected variant allele frequency (VAFs) of 0.5, 0.35, 0.2 and 0.1). Picard was

used to convert bam files to fastq format files with default parameters. 2-kupl was run using default parameters on pairs of simulated normal-tumor fastq files.

Simulated WGS analysis

A simulated WGS dataset containing two matched samples was generated by DWGSM (https://github.com/nh13/DWGSIM), with a mean coverage of 50X across available positions. The rates of mutations in case and control group samples were set as 0.0001 and 0, respectively. The fraction of indels in all variants was restricted to 20%. The expected VAF ranged from 0.1 to 0.5. All other parameters were set as default values. Besides the mutations and indels, the simulated WGS dataset also included structural variants including deletions, duplications and translocations longer than 50 bp. DWGSM generates fastq format files that are directly used as input for 2-kupl.

TCGA-PRAD data analysis

Matched normal-tumor WES data of 498 patients from TCGA-PRAD (Prostate Adenocarcinoma) [52] were retrieved with permission from dbGAP [53]. BAM files were converted to paired-ends fastq files using Picard tools with default parameters. 2-kupl somatic variant calls were obtained for each normal/tumor pair using default parameters. Detailed analysis of variant calling was performed on the TCGA-PRAD sample with the highest tumor mutational burden (barcode TCGA-ZG-A9ND).

2-kupl results on the TCGA-PRAD dataset were compared to variant calls downloaded from the GDC portal. Briefly, the GDC portal workflow uses BWA to map reads to the human genome and determines variants with five state of the art variant callers, as described here: https://docs.gdc.cancer.gov/Data/Bioinformatics_Pipelines/. We used the maftools R package [54] to retrieve variants predicted using the GATK-MuTect2 pipeline and filtered against a "panel of normals". This mutation dataset is hereafter referred to as the "GDC portal" dataset.

To remove putative germline variants from 2-kupl results, we built a boolean matrix representing the presence of each k-mer in each normal sample. Any k-mer present in at least two normal samples was excluded. Retained recurrent variants were considered as tumor-specific (Additional file 2: Table S1). Mutations detected by 2-kupl and absent in the GDC portal variants were considered as 2-kupl specific. To verify whether calls absent in GDC portal variants were not discarded at earlier stages of the GDC portal pipeline, we also retrieved the protected MAF file containing all unfiltered variants called by the MuTect2 workflow.

The oncoplot graph for GDC portal variants (Fig. 5a) was drawn using maftools. To obtain recurrently mutated genes by 2-kupl, we aggregated variants belonging to the same gene in 2-kupl results and constructed a gene-level occurrence matrix that was fed to maftools (Fig. 5b). Recurrent variants from 2-kupl and the GDC Portal were also compared with a comprehensive prostate cancer dataset from 200 whole-genome sequences and 277 whole-exome sequences from localized prostate tumours [28] (Additional file 3: Table S2)

Recurrently mutated genes were annotated using a collection of 1404 PRAD-related genes collected from CLINVAR [55], COSMIC [56], DISEASE [57], KEGG [58],

OMIM [59], PheGenl [60] and driver predictions by Martincorena et al. and Armenia et al. [29, 61] (Additional file 4: Table S3).

Bacterial genome analysis

We obtained WGS fastq files from the *Pseudomonas aeruginosa* PAO1Or wild-type strain and 24 phage-tolerant mutants [62]. Mutations in the phage-tolerant variants were previously validated by mapping of the WGS raw sequences to the PAO1Or genome (Genbank accession LN871187) and confirmed by PCR amplification and Sanger sequencing. We used one control WGS file and 21 mutant WGS files corresponding to 26 validated variants. Detailed variants (Additional file 5: Table S4) include seven mutations, 13 small indels and six large deletions longer than 100 bp. 2-kupl was run using default parameters on every mutant WGS file compared to the control WGS file.

Abbreviations

WES: Whole-exome sequencing; WGS: Whole-genome sequencing; TCGA: The Cancer Genome Atlas; PRAD: Prostate Adenocarcinoma; GDC: Genomic Data Commons; DBG: De Bruijn graph; MAPQ: Mapping quality; SV: Structural variant.

Supplementary Information

The online version contains supplementary material available at https://doi.org/10.1186/s12859-021-04185-6.

Additional file 1: Fig. S1. The distribution of shared SNVs in 2kupl and consistency of four mapping-based protocols. **Figure S2**. Phred score distribution. **Figure S3**. Alignment of the mutant contig and inferred reference from one unmapped event. **Figure S4**. IGV views of UBR4 mutations occurred on patient of TCGA-EJ-7125

Additional file 2: Table S1. This supplementary table includes recurrent SNVs, SVs and mutated genes identified by 2-kupl.

Additional file 3: Table S2. Comparison with the Fraser et al's recurrent PRAD mutations.

Additional file 4: Table S3. Prostate cancer related genes collected from various resources.

Additional file 5: Table S4. True positive variants in the bacterial WGS data.

Additional file 6: Table S5. 2-kupl detected structural variants that are missed by Delly.

Acknowledgements

Not applicable.

Authors' contributions

YW and DG designed the software and benchmarking procedures, YW developed the software, ran all analyzes and analyzed results, HX contributed to the software, YD reviewed and commented on the draft paper, CP provided bacterial genome data and analyzed results, YW and DG wrote the manuscript. All authors read and approved the final manuscript.

Funding

This work was funded in part by Agence Nationale de la Recherche grant ANR-18-CE45-0020 and by a PhD studentship to YW by Annoroad Technology, Beijing.

Availability of data and materials

2-kupl is open source under MIT license and available at GitHub https://github.com/yunfengwang0317/2-kupl.

Declarations

Ethics approval and consent to participate

Not applicable.

Consent for publication

Not applicable.

Competing interests

DG is an associate editor at BMC Bioinformatics. The authors declare that they have no competing interests.

Author details
[1]Institute of Integrative Cell Biology (I2BC), Université Paris-Saclay, CNRS, CEA, 1 avenue de la Terrasse, 91190 Gif-sur-Yvette, France. [2]IHU PRISM, Gustave Roussy, 114 rue Edouard Vaillant, 94800 Villejuif, France. [3]Annoroad Gene Technology Co., Ltd, Beijing 100176, China.

Received: 3 February 2021 Accepted: 11 May 2021
Published online: 05 June 2021

References

1. Li MM, Datto M, Duncavage EJ, Kulkarni S, Lindeman NI, Roy S, Tsimberidou AM, Vnencak-Jones CL, Wolff DJ, Younes A, et al. Standards and guidelines for the interpretation and reporting of sequence variants in cancer: a joint consensus recommendation of the association for molecular pathology, american society of clinical oncology, and college of american pathologists. J Mol Diagn. 2017;19(1):4–23.
2. MacArthur D, Manolio T, Dimmock D, Rehm H, Shendure J, Abecasis G, Adams D, Altman R, Antonarakis S, Ashley E, et al. Guidelines for investigating causality of sequence variants in human disease. Nature. 2014;508(7497):469–76.
3. Jiang T, Shi T, Zhang H, Hu J, Song Y, Wei J, Ren S, Zhou C. Tumor neoantigens: from basic research to clinical applications. J Hematol Oncol. 2019;12(1):1–13.
4. Shiloach J, Reshamwala S, Noronha SB, Negrete A. Analyzing metabolic variations in different bacterial strains, historical perspectives and current trends-example e. coli. Curr Opin Biotechnol. 2010;21(1):21–6.
5. Hurles ME, Dermitzakis ET, Tyler-Smith C. The functional impact of structural variation in humans. Trends Genet. 2008;24(5):238–45.
6. Li H, Durbin R. Fast and accurate short read alignment with burrows-wheeler transform. Bioinformatics. 2009;25(14):1754–60.
7. Benjamin D, Sato T, Cibulskis K, Getz G, Stewart C, Lichtenstein L. Calling somatic SNVs and indels with mutect2. BioRxiv. 2019;861054.
8. Koboldt DC, Zhang Q, Larson DE, Shen D, McLellan MD, Lin L, Miller CA, Mardis ER, Ding L, Wilson RK. Varscan 2: somatic mutation and copy number alteration discovery in cancer by exome sequencing. Genome Res. 2012;22(3):568–76.
9. Larson DE, Harris CC, Chen K, Koboldt DC, Abbott TE, Dooling DJ, Ley TJ, Mardis ER, Wilson RK, Ding L. Somaticsniper: identification of somatic point mutations in whole genome sequencing data. Bioinformatics. 2012;28(3):311–7.
10. Fan Y, Xi L, Hughes DS, Zhang J, Zhang J, Futreal PA, Wheeler DA, Wang W. Muse: accounting for tumor heterogeneity using a sample-specific error model improves sensitivity and specificity in mutation calling from sequencing data. Genome Biol. 2016;17(1):1–11.
11. Li H, Ruan J, Durbin R. Mapping short DNA sequencing reads and calling variants using mapping quality scores. Genome Res. 2008;18(11):1851–8.
12. Loeffler C, Karlsberg A, Martin LS, Eskin E, Koslicki D, Mangul S. Improving the usability and comprehensiveness of microbial databases. BMC Biol. 2020;18:1–6.
13. Audano PA, Ravishankar S, Vannberg FO. Mapping-free variant calling using haplotype reconstruction from k-mer frequencies. Bioinformatics. 2018;34(10):1659–65.
14. Compeau PE, Pevzner PA, Tesler G. How to apply de Bruijn graphs to genome assembly. Nat Biotechnol. 2011;29(11):987–91.
15. Uricaru R, Rizk G, Lacroix V, Quillery E, Plantard O, Chikhi R, Lemaitre C, Peterlongo P. Reference-free detection of isolated SNPs. Nucleic Acids Res. 2015;43(2):11.
16. Narzisi G, Corvelo A, Arora K, Bergmann EA, Shah M, Musunuri R, Emde A-K, Robine N, Vacic V, Zody MC. Lancet: genome-wide somatic variant calling using localized colored debruijn graphs. bioRxiv. 2017;196311.
17. Iqbal Z, Caccamo M, Turner I, Flicek P, McVean G. De novo assembly and genotyping of variants using colored de bruijn graphs. Nat Genet. 2012;44(2):226–32.
18. Heydari M, Miclotte G, Van de Peer Y, Fostier J. Illumina error correction near highly repetitive DNA regions improves de novo genome assembly. BMC Bioinform. 2019;20(1):1–13.
19. Tomczak K, Czerwińska P, Wiznerowicz M. The cancer genome atlas (TCGA): an immeasurable source of knowledge. Contemp Oncol. 2015;19(1A):68.
20. Meng J, Chen Y-PP. A database of simulated tumor genomes towards accurate detection of somatic small variants in cancer. PLoS ONE. 2018;13(8):0202982.
21. Rausch T, Zichner T, Schlattl A, Stütz AM, Benes V, Korbel JO. Delly: structural variant discovery by integrated paired-end and split-read analysis. Bioinformatics. 2012;28(18):333–9.
22. de Koning AJ, Gu W, Castoe TA, Batzer MA, Pollock DD. Repetitive elements may comprise over two-thirds of the human genome. PLoS Genet. 2011;7(12):1002384.
23. Amemiya HM, Kundaje A, Boyle AP. The encode blacklist: identification of problematic regions of the genome. Sci Rep. 2019;9(1):1–5.
24. Pon JR, Marra MA. Driver and passenger mutations in cancer. Annu Rev Pathol. 2015;10:25–50.
25. Greenman C, Stephens P, Smith R, Dalgliesh GL, Hunter C, Bignell G, Davies H, Teague J, Butler A, Stevens C, et al. Patterns of somatic mutation in human cancer genomes. Nature. 2007;446(7132):153–8.
26. Goncearenco A, Rager SL, Li M, Sang Q-X, Rogozin IB, Panchenko AR. Exploring background mutational processes to decipher cancer genetic heterogeneity. Nucleic Acids Res. 2017;45(W1):514–22.
27. Barbieri CE, Baca SC, Lawrence MS, Demichelis F, Blattner M, Theurillat J-P, White TA, Stojanov P, Van Allen E, Stransky N, et al. Exome sequencing identifies recurrent spop, foxa1 and med12 mutations in prostate cancer. Nat Genet. 2012;44(6):685–9.
28. Fraser M, Sabelnykova VY, Yamaguchi TN, Heisler LE, Livingstone J, Huang V, Shiah Y-J, Yousif F, Lin X, Masella AP, et al. Genomic hallmarks of localized, non-indolent prostate cancer. Nature. 2017;541(7637):359–64.

29. Martincorena I, Raine KM, Gerstung M, Dawson KJ, Haase K, Van Loo P, Davies H, Stratton MR, Campbell PJ. Universal patterns of selection in cancer and somatic tissues. Cell. 2017;171(5):1029–41.
30. Li J, Drubay D, Michiels S, Gautheret D. Mining the coding and non-coding genome for cancer drivers. Cancer Lett. 2015;369(2):307–15.
31. Blaxter M. Revealing the dark matter of the genome. Science. 2010;330(6012):1758–9.
32. Miga KH, Koren S, Rhie A, Vollger MR, Gershman A, Bzikadze A, Brooks S, Howe E, Porubsky D, Logsdon GA, et al. Telomere-to-telomere assembly of a complete human x chromosome. Nature. 2020;585(7823):79–84.
33. Sherman RM, Forman J, Antonescu V, Puiu D, Daya M, Rafaels N, Boorgula MP, Chavan S, Vergara C, Ortega VE, et al. Assembly of a pan-genome from deep sequencing of 910 humans of African descent. Nat Genet. 2019;51(1):30–5.
34. Ma X, Shao Y, Tian L, Flasch DA, Mulder HL, Edmonson MN, Liu Y, Chen X, Newman S, Nakitandwe J, et al. Analysis of error profiles in deep next-generation sequencing data. Genome Biol. 2019;20(1):1–15.
35. Medvedev P, Pham S, Chaisson M, Tesler G, Pevzner P. Paired de bruijn graphs: a novel approach for incorporating mate pair information into genome assemblers. J Comput Biol. 2011;18(11):1625–34.
36. Hwang S, Kim E, Lee I, Marcotte EM. Systematic comparison of variant calling pipelines using gold standard personal exome variants. Sci Rep. 2015;5(1):1–8.
37. Marçais G, Kingsford C. A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. Bioinformatics. 2011;27(6):764–70.
38. Wu TD, Reeder J, Lawrence M, Becker G, Brauer MJ. Gmap and gsnap for genomic sequence alignment: enhancements to speed, accuracy, and functionality. In: Statistical genomics. Springer; 2016. p. 283–334.
39. Chen C, Khaleel SS, Huang H, Wu CH. Software for pre-processing illumina next-generation sequencing short read sequences. Source Code Biol Med. 2014;9(1):1–11.
40. Audoux J, Philippe N, Chikhi R, Salson M, Gallopin M, Gabriel M, Le Coz J, Commes T, Gautheret D. Exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. BioRxiv. 2017;122937.
41. Pereira F, Azevedo F, Carvalho Â, Ribeiro GF, Budde MW, Johansson B. Pydna: a simulation and documentation tool for DNA assembly strategies using python. BMC Bioinform. 2015;16(1):1–10.
42. Bushnell B. BBMap. https://sourceforge.net/projects/bbmap. 2018.
43. Huang X, Madan A. Cap3: A DNA sequence assembly program. Genome Res. 1999;9(9):868–77.
44. Wang Q, Shashikant CS, Jensen M, Altman NS, Girirajan S. Novel metrics to measure coverage in whole exome sequencing datasets reveal local and global non-uniformity. Sci Rep. 2017;7(1):1–11.
45. Danecek P, Auton A, Abecasis G, Albers CA, Banks E, DePristo MA, Handsaker RE, Lunter G, Marth GT, Sherry ST, et al. The variant call format and vcftools. Bioinformatics. 2011;27(15):2156–8.
46. Cock PJ, Antao T, Chang JT, Chapman BA, Cox CJ, Dalke A, Friedberg I, Hamelryck T, Kauff F, Wilczynski B, et al. Biopython: freely available python tools for computational molecular biology and bioinformatics. Bioinformatics. 2009;25(11):1422–3.
47. McGinnis S, Madden TL. Blast: at the core of a powerful and diverse set of sequence analysis tools. Nucleic Acids Res. 2004;32(suppl_2):20–5.
48. Li Z, Chen Y, Mu D, Yuan J, Shi Y, Zhang H, Gan J, Li N, Hu X, Liu B, et al. Comparison of the two major classes of assembly algorithms: overlap-layout-consensus and de-Bruijn-graph. Brief Funct Genom. 2012;11(1):25–37.
49. Robinson JT, Thorvaldsdóttir H, Winckler W, Guttman M, Lander ES, Getz G, Mesirov JP. Integrative genomics viewer. Nat Biotechnol. 2011;29(1):24–6.
50. Zook JM, Catoe D, McDaniel J, Vang L, Spies N, Sidow A, Weng Z, Liu Y, Mason CE, Alexander N, et al. Extensive sequencing of seven human genomes to characterize benchmark reference materials. Sci Data. 2016;3(1):1–26.
51. Ewing AD, Houlahan KE, Hu Y, Ellrott K, Caloian C, Yamaguchi TN, Bare JC, P'ng C, Waggott D, Sabelnykova VY, et al. Combining tumor genome simulation with crowdsourcing to benchmark somatic single-nucleotide-variant detection. Nat Methods. 2015;12(7):623–30.
52. Abeshouse A, Ahn J, Akbani R, Ally A, Amin S, Andry CD, Annala M, Aprikian A, Armenia J, Arora A, et al. The molecular taxonomy of primary prostate cancer. Cell. 2015;163(4):1011–25.
53. Tryka KA, Hao L, Sturcke A, Jin Y, Wang ZY, Ziyabari L, Lee M, Popova N, Sharopova N, Kimura M, et al. Ncbi's database of genotypes and phenotypes: dbgap. Nucleic Acids Res. 2014;42(D1):975–9.
54. Mayakonda A, Lin D-C, Assenov Y, Plass C, Koeffler HP. Maftools: efficient and comprehensive analysis of somatic variants in cancer. Genome Res. 2018;28(11):1747–56.
55. Landrum MJ, Lee JM, Benson M, Brown GR, Chao C, Chitipiralla S, Gu B, Hart J, Hoffman D, Jang W, et al. Clinvar: improving access to variant interpretations and supporting evidence. Nucleic Acids Res. 2018;46(D1):1062–7.
56. Bamford S, Dawson E, Forbes S, Clements J, Pettett R, Dogan A, Flanagan A, Teague J, Futreal PA, Stratton MR, et al. The cosmic (catalogue of somatic mutations in cancer) database and website. Br J Cancer. 2004;91(2):355–8.
57. Pletscher-Frankild S, Pallejà A, Tsafou K, Binder JX, Jensen LJ. Diseases: text mining and data integration of disease-gene associations. Methods. 2015;74:83–9.
58. Kanehisa M, Furumichi M, Tanabe M, Sato Y, Morishima K. Kegg: new perspectives on genomes, pathways, diseases and drugs. Nucleic Acids Res. 2017;45(D1):353–61.
59. Hamosh A, Scott AF, Amberger JS, Bocchini CA, McKusick VA. Online mendelian inheritance in man (omim), a knowledge-base of human genes and genetic disorders. Nucleic Acids Res. 2005;33(suppl_1):514–7.
60. Ramos EM, Hoffman D, Junkins HA, Maglott D, Phan L, Sherry ST, Feolo M, Hindorff LA. Phenotype-genotype integrator (phegeni): synthesizing genome-wide association study (gwas) data with existing genomic resources. Eur J Hum Genet. 2014;22(1):144–7.
61. Armenia J, Wankowicz SA, Liu D, Gao J, Kundra R, Reznik E, Chatila WK, Chakravarty D, Han GC, Coleman I, et al. The long tail of oncogenic drivers in prostate cancer. Nat Genet. 2018;50(5):645–51.
62. Latino L. Pseudolysogeny and sequential mutations build multiresistance to virulent bacteriophages in pseudomonas aeruginosa. PhD thesis, Université Paris-Saclay; 2016.

Publisher's note

université PARIS-SACLAY

ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Titre: Analyse de k-mers pour la transcriptomique du cancer à la résolution du nucléotide
Mots clés: classifieurs, apprentissage automatique, NGS, médecine de précision, ARN, transcriptomique

Résumé: Le transcriptome intègre des variations d'ARN produites par deux processus principaux : les altérations génétiques (mutations, fusions de gènes, etc.) et les modifications post-transcriptionnelles (épissages alternatifs, etc.). C'est un objet de recherche idéal pour étudier l'association génotype-phénotype. Les techniques de next-generation sequencing (NGS, séquençage de nouvelle génération) permettent une mesure du transcriptome à la résolution du nucléotide, de manière à la fois rapide et économique. Les analyses conventionnelles du transcriptome basée sur la quantification des gènes ou des transcrits n'utilisent pas la pleine précision de ces données NGS, mais héritent d'une perspective plus ancienne issue des puces à ADN (microarrays) et qui considère le gène ou le transcrit comme la caractéristique élémentaire pour l'analyse statistique ou pour l'apprentissage automatique. Dans cette thèse, nous discutons et développons une nouvelle perspective d'analyse du transcriptome, basée sur les signaux de k-mers (sous-chaînes de caractères de longueur fixe comme k, avec typiquement k = 31). Ainsi, au lieu de quantifier des gènes ou des transcrits prédéfinis, nous comptons des k-mers courts et arbitraires, et les prenons directement comme caractéristiques élémentaires. Cela permet de représenter des événements au cours de l'expression du gène à la résolution nucléotide, et d'entrer au-dessous du niveau des transcrits pour examiner les événements locaux. En outre, cette approche évite aussi que les signaux informatifs s'annulent à l'étape de la quantification de gène ou de transcrit.

La thèse comporte différents aspects : (i) Le logiciel KaMRaT (chapitre 3), développé au cours de la thèse, prend en charge diverses méthodes pour réduire la dimensionnalité de k-mers, et pour améliorer leur spécificité. Il intègre : un module pour classer-sélectionner des k-mers en évaluant l'association entre les comptages des k-mers et le phénotype des échantillons; un module d'extension pour fusionner des k-mers chevauchants ; un module de filtrage tenant compte de leurs niveaux d'expression ; un module de masquage pour extraire les k-mers avec une liste de séquences donnée. Les résultats montrent que KaMRaT est un logiciel à la fois économe en ressource de calcul, flexible et facile à utiliser (ii) La comparaison entre les classifieurs utilisant k-mers ou gènes (chapitre 4) montre qu'un classifieur basé sur des caractéristiques de type k-mer fonctionne aussi bien que celui basé sur les caractéristiques de type gène, dans le problème du pronostic du cancer de la prostate, le premier offrant de plus de la possibilité de découvrir de nouveaux événements non-annotés. (iii) L'analyse de la réplicabilité des signaux k-mers informatifs dans une recherche inter-cohorte (chapitre 5) montre que les signaux k-mers sont réplicables entre jeux de données comparables mais indépendantes, et la recherche inter-cohorte de k-mers permet de trouver des signaux informatifs stables. (iv) Le logiciel Kmerator (chapitre 6) permet l'utilisation de signaux de type k-mer, donc sans référence, comme un proxy pour la mesure d'expression génique. (v) Enfin, l'application des logiciels REINDEER et Kmerator (chapitre 7) permet d'utiliser de grandes collections d'échantillons d'ARN-seq pour y rechercher des séquences d'ARN arbitraires.

université PARIS-SACLAY

ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Title: k-mer based analysis for cancer transcriptomics at nucleotide resolution
Keywords: classifiers, machine learning, NGS, precision medicine, RNA, transcriptomics

Abstract: The transcriptome integrates RNA variations produced by two main processes: DNA alteration (e.g., mutations, gene-fusions, etc.) and post-transcriptional modifications (e.g., alternative splicing, etc.). It is an ideal research object for genotype-phenotype association. Next-generation sequencing (NGS) techniques allow a measurement of transcriptome at single-nucleotide resolution, both rapidly and at a relatively low cost. Conventional transcriptome analyses based on gene/transcript quantification do not make use of the full precision of this NGS data. Instead, they inherit the perspective of microarray measurements that consider gene or transcript as the elementary features for statistical analysis or machine learning. In this thesis, we discuss and develop a novel perspective of transcriptome analysis based on k-mers (substrings with fixed length k, typically k = 31) signals. Rather than quantifying predefined genes/transcripts, we count short and arbitrary k-mers and use them directly as elementary features. This allows representing gene expression events at the single-nucleotide resolution, and thereby driving insights into local events occuring at sub-transcript level. Also, this approach prevents the informative signals from cancelling each other at gene/transcript quantification stage.

This thesis presents different aspects of this endeavor: (i) The KaMRaT software (chapter 3), developed during the thesis, supports various methods for reducing k-mer dimensionality and improving their specificity. This includes: a ranking-selecting module by evaluating association between sample counts and conditions; an extension module merging overlapped k-mers; a filter module considering their expression level; and a masking module to extract k-mers with a given sequence list. Results show that KaMRaT is an effective and efficient software, with highly flexible and easy-to-use characteristics. (ii) A comparison between classifiers obtained using k-mer or conventional gene features (chapter 4) shows that k-mer-feature-based classifier performs as well as classical gene-feature-based one, in the prostate prognostic problem, with the former further supporting finding novel unannotated events. (iii) The analysis of the replicability of informative k-mer signals in an inter-cohort research (chapter 5) shows that k-mer signals are replicable across independent data sets, and the k-mer-based inter-cohort research allows finding stable informative signals. (iv) The Kmerator software (chapter 6) allows utilization of reference-free k-mer signals as a proxy to reference-based gene expression measures. (v) Application of the REINDEER and Kmerator software (chapter 7) allows for arbitrary sequence indexing and abundance query across RNA-seq samples.