Combining machine learning and reference-free transcriptome analysis for the identification of prostate cancer signatures

Thi Ngoc Ha Nguyen

Thèse de doctorat

NNT: 2020UPASL069

université PARIS-SACLAY

Combining machine learning and reference-free transcriptome analysis for the identification of prostate cancer signatures

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.

Référent: : Faculté des sciences d'Orsay

Thèse présentée et soutenue à Orsay, le 18/12/2020, par

Thi Ngoc Ha NGUYEN

Composition du jury:

Marie-Hélène MUCCHIELLI-GIORGI Professeur, Université Evry Val d'Essonne	Présidente
William RITCHIE Docteur (HDR), CNRS, IGH Montpellier	Rapporteur & Examinateur
Duc-Hau LE Professeur, Vingroup Big Data Institute, Hanoi, Vietnam	Rapporteur & Examinateur
Laura CANTINI Docteur, CNRS, Institut de Biologie de l'Ecole Normale Supérieure	Examinatrice
Daniel GAUTHERET Professeur, I2BC, CEA, CNRS, Université Paris-Saclay	Directeur de thèse
Yann PONTY Docteur, Directeur de Recherche CNRS, LIX UMR 7161	Codirecteur de thèse
Mélina GALLOPIN Docteur, Maître de Conférence, I2BC, CEA, CNRS, Université Paris-Saclay	Co-encadrante

Acknowledgement

I guarantee that all content written in this thesis is my own and does not contain any illegal reproduction. First of all, I would like to thank the 911 Scholarship Fund from the Vietnamese Government and Campus France, which have granted me scholarships and facilitated accommodation.

I also would like to demonstrate my deepest and most sincere thanks to my thesis director Pr. Daniel Gautheret during the past four years, a dedicated and enthusiastic teacher who always guides and gives me ideas and advice when I have difficulty in research. He always encourages me and also puts pressure on me to improve every day.

For today when the thesis has been completed, I would like to extend my sincere thanks to Dr. Yann Ponty, Dr. Melina Gallopin, my dissertation co-instructors. Thanks to Yann, who always provides support and professional advice when I have problems at work. Yann was also a supporter who, with Philippe Chassignet, sought an assistant teaching job for me in Ecole Polytechnique. It was a great opportunity for me to learn, accumulate more knowledge and experience and also help me have more money to cover my daily life.

I would like to convey my deepest gratitude to Melina. She is not only my research guide, but also a wonderful friend. She always spends time answering my questions and gives me a lot of meaningful advice and solutions when I encounter mistakes in research, as well as when I am under pressure while working. Thank you very much, Melina.

I also would like to thank the members who joined my jury, Pr. Marie-Helene Mucchielli-Giorgi, Dr. Laura Cantini, Dr. William Ritchie and Pr. Duc-Hau Le. Thanks for their comments and corrections for my Ph.D. thesis.

I am truly indebted to members in my laboratory: Fabrice, Claire, Nicholas, Marc, Haoliang, YunFeng and Antoine. Fabrice and Claire who taught me the first knowl-

edge of sequencing, k-mer decomposition ... Thank you, Claire who took me to see a doctor and help me to translate because I can't speak french.

I am profoundly grateful to my family, my mum and my dad, who have always been by my side, are a great source of encouragement, a solid background for me to focus on my study. I am especially grateful to my mother and my sister-in-law for taking care of my little daughter while I study abroad. I also would like to send my most loving and happy words to my little daughter, you are always the greatest delight and motivation for me to complete this thesis.

And finally and indispensable, I would like to thank the person who has been with me most of the time living and working in France. He is the man, who always spent precious weekend cooking delicious meals and woke up much earlier than me every morning to prepare breakfast for me. Thank you much my love.

Contents

Part I

Introduction

Chapter 1

Biostatistics for transcriptome analysis

1.1 History of transcriptome and transcriptomics

1.1.1 Microarrays and RNA-seq

The transcriptome is the complete set of **R**ibo**n**ucleic **A**cid (RNA) transcripts in a given organism or subset of transcripts in a specific tissue or cell type, at a particular stage and under a certain circumstance. Thus, understanding transcriptome not only allows us to interpret the functional and structural elements of the genome, but also provides a comprehension of human biology and diseases.

The whole transcriptome study was first introduced in the early 1990s, and thanks to technological advances since late 1990, transcriptomics has become a widespread discipline in biological sciences. There are two major techniques for transcriptome analysis, including a **D**eoxyribo**n**ucleic **A**cid (DNA) microarray which express a set of predetermined sequences and a high-throughput **RNA seq**uencing (RNA-seq) captures all sequences.

General transcriptome analysis methodology undergoes two critical parts for both DNA microarray and RNA-seq technologies: data processing and application (Figure 1.1). Processing microarray data includes typically image analysis, background subtraction, normalization, and summarization, while RNA-seq undergoes preprocessing with quality control and trimming, mapping, and assembly. The expression levels of all transcripts so-called gene expression data are stored as a count matrix. In this matrix, each row corresponding to a gene and each column representing a specified condition that usually relates to environments, disease types or subtypes, and tissues. The gene expression data after the fact are determined, which are subject for **d**ifferential **e**xpression (DE) analysis, survival analysis, gene patients clustering, or disease patients classification according to biological questions.

1.1.2 Learning on transcriptomic data : problem and notations

With the rapid development of transcriptomic technologies, now it is possible to simultaneously track the expression levels of thousands of genes or transcripts (features) during critical biological processes and across collections of related samples. However, the significant number of features and the complexity of biological networks account for the challenges of understanding and interpreting the result of such massive data. In this section, we introduce the problem and notations in learning on transcriptomic data; these notations are also applied uniformly in the whole thesis.

Transcriptome sequencing data is converted into a gene expression measure (see Chapter 2) and stored into an observed expression matrix $\mathbf{x}$ with p rows and n columns where p is the number of features/variables (genes or other relevant features) and n is the number of samples/observations. The index of each feature and each sample are indicated by k and i, respectively ($k = 1, \ldots, p$; $i = 1, \ldots, n$). As a result, $\mathbf{x}_i$ is an observed vector of expression for sample i across p features (size of the vector is p), while $\mathbf{x}^{(k)}$ is an observed vector of expression for feature k across

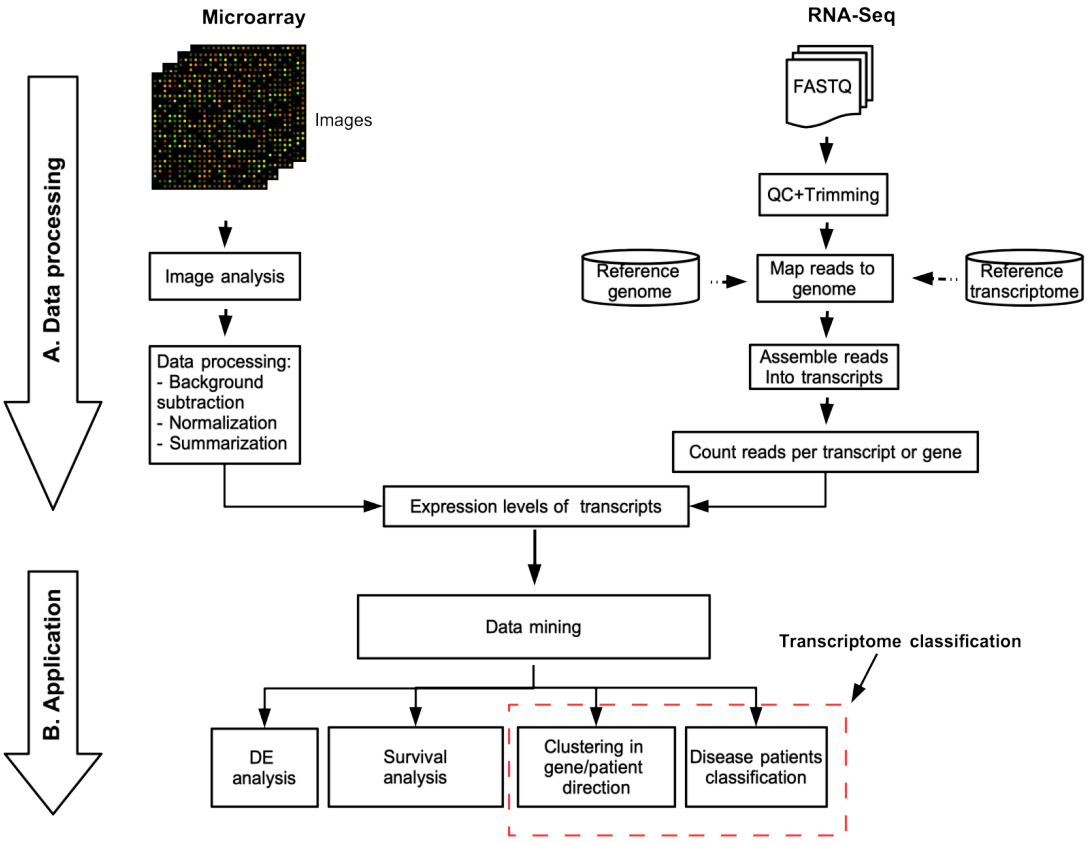


Figure 1.1: General flowchart for transcriptome analysis. Both microarray and RNA-seq technologies are undergone two parts. **A. Data processing** The raw data from microarray experiments are obtained in a bunch of images. To turn these images into probe-level values involves several steps: image analysis, background substraction, normalization and summarization. Meanwhile, RNA-seq data often stores in a list of FASTQ files, experiences processing with quality control and trimming, mapping, and assembly. **B. Application** Microarray and RNA-seq data are applied in various applications: **d**ifferential **e**xpression analysis, survival analysis and transcriptome classification.

n samples (size of the vector is n). Then, $x_i^{(k)}$ is an observed expression for feature k in sample i. A random variable modeling the expression for feature k in sample i is represented by $X_i^{(k)}$. Finally, $\mathbf{y}$ is a vector of labels for the n samples, and y_i is the label of sample i.

1.2 Differential gene expression (DGE) analysis

Identifying genes that show differences in expression level between conditions is the most popular use of transcriptome profiling. For example, in order to assess the effect of a drug, we may ask which genes are up-regulated (increased in the expression) or down-regulated (decreased in the expression) between treatment and control groups.

1.2.1 Hypothesis testing

Assume we want to detect DE genes between two conditions (1 and 2) based on their expression table. Statistical tests address this task by providing a mechanism for making quantitative decisions. To make a decision, statistical tests evaluate the evidence that the data provides against an hypothesis. This hypothesis is called null hypothesis (labeled H_0). In case of gene expresssion anaylsis, the statement of H_0 is: *The mean expression between the two conditions is equal*. The statement of alternative hypothesis (labeled H_1 or H_a) is: *The mean expression between the two conditions is different*. To test whether gene $\mathbf{x}^{(k)}$ is a diffentially expressed gene between two conditions, we apply the procedure is described below:

1. Model the expression of gene k using a random variable $X^{(k)}$, with means μ_1 and μ_2 in condition 1 and condition 2 respectively.

2. Formalize the H_0 and H_1 hypotheses:

   - H_0: *The mean expression between the two conditions is equal* ($\mu_1 = \mu_2$)

- H_1: *The mean expression between the two conditions is different* ($\mu_1 \neq \mu_2$)

3. Setup the significance level, α, defined as the probability of rejected H_0 given that H_0 is true. The significance level is used in step 5 to take the final decision.

4. Fit the model and estimate the parameters of the random variable $X^{(k)}$ for each condition.

5. Computing the value of the test statistic on the observed data $\mathbf{x}^{(k)}$, and the probability of obtaining this value or a more extreme value when H_0 is true (called **P-value**). For example, **P-value** $= 0.001$ means that the probability of seeing the experiment outcomes as extreme or more extreme than the observed data is one in 1000 when the mean expression between two conditions is equal, *i.e.*, the H_0 is true.

6. Determine to reject or not rejet H_0 based on the **P-value** and α. Finally, if P-value $> \alpha$, the evidence against H_0 is statistically significant, therefore a test statistic gives a decision for rejecting H_0. Conventionally, one often chooses the level of significance equal to 5% or 1% or 0.1%.

In a hypothesis testing, one can make two type of errors.

1. **Type I error** or false-positive: The test statistic rejects the null hypothesis while it is really true. For example, the gene k is not a differentially expressed but the test statistic states that this gene is differentially expressed. As a result, type I error introduces a **false discovery**.

2. **Type II error** or false-negative: conversely, the statistical test accepts the null hypothesis while it is really false.

Although type I and type II errors cannot be entirely avoided, test statistics control the probability of generating type I errors through the significance level α.

1.2.2 Multiple testing

Conducting a single statistical test for each gene has several limitations, the most important is that a large number of hypothesis tests are performed, potentially introducing a substantial number of falsely significant results. For instance, say we have 20 null hypotheses to test simultaneously and a given $\alpha = 0.05$, *i.e.* the probability of making a type I error is 5% for each individual test. Therefore, the chance of generating at least 1 false-positive when performing 20 tests is calculated as follows:

$$\begin{aligned} P(\text{making at least 1 error in 20 tests}) &= 1 - P(\text{not making an error in 20 tests}) \\ &= 1 - (1 - 0.05)^{20} \\ &\approx 0.64 \end{aligned}$$

Thus, if the 20 tests are independent then the chance of generating at least one incorrect rejection (so-called the family-wise error rate or **FWER**) is a round 64%, even there is no significant differences to detect. That would be a serious problem in the case of RNA-seq experiments, where we have to process tens to hundreds of thousands of tests.

Several methods have been introduced to deal with multiple testing with the aim of adjusting the α so that the probability of making at least one significant result by chance is still lower than the significance level.

Bonferroni adjustment controls FWER. Bonferroni adjustment is the most straightforward method for multiple testing correction (Dunn, 1961; Bland and Altman, 1995). This adjustment method seeks to control the FWER when multiple hypothesis tests are conducted simultaneously, as shown in Equation 1.1:

$$\text{Adjusted } \alpha = \frac{\alpha}{\text{number of hypothesis tests.}} \tag{1.1}$$

Going back to our example of testing simultaneously 20 null hypotheses with a given

$\alpha = 0.05$. Applying Bonferroni adjustment correction, we have a new adjusted α of 0.0025 ($i.e., 0.05/20$)to take into account the multiple testing.

Benjamin-Hochberg adjustment controls False Discovery Rate (FDR). This method, introduced by Benjamini and Hochberg (1995), aims to control the proportion of falsely rejected hypotheses, *i.e.* controlling FDR. **B**enjamin-**H**ochberg (BH) procedure was implemented step by step as described below.

1. Conduct all statistical tests in m hypothesis tests and extract the corresponding p-value for each test.
2. Sort these p-values in ascending order assigning a rank for each p-value, starting from 1.
3. Calculate the BH critical value for each individual p-value, as $\frac{i}{m} \times Q$, where i is the rank of p-value, while Q is the desired proportion FDR.
4. Find the largest $p-$value that is lower than its BH critical value.
5. Finally, all p-values lower than this $p-$value are considered significant.

The **B**enjamin-**H**ochberg has been designed to work for independent tests, although it works in practice on dependent tests.

Suppose we conduct 20 hypothesis tests ($m = 20$) for about 500 genes with our desired **F**alse **D**iscovery **R**ate of 0.2 ($Q = 0.2$). Table 1.1 below shows the five genes with the lowest p-value. We calculate the BH critical value for each gene as presented in column 4.

The bold p-value (gene 4) is the highest p-value that is lower than its BH critical value ($i.e., 0.036 < 0.04$). As a result, all genes that have a p-value lower than 0.036 are considered significant. Note that the p-value of gene 2 also is smaller than its BH value. However, it is not the highest value among all p-values that justify this criterion.

Table 1.1: An illustration for Benjamin-Hochberg correction

Gene	p-value	Rank	BH
1	0.015	1	0.01
2	0.018	2	0.02
3	0.032	3	0.03
4	**0.036**	**4**	**0.04**
5	0.051	5	0.05

1.2.3 Models for differential gene expression data

The first step of the statistical test is the choice of the a probabilistic model for the expression data. Microarrays have been used systematically for differential expression for over three decades, and quite a few well-established methods are developed for this purpose, such as `limma` (Smyth, 2004) based on the normal distribution. Unfortunately, because of the difference between the data obtained from microarrays and RNA-seq, these methods cannot be directly applied to RNA-seq data. The expression levels of microarray data are represented as continuous intensity hybridization signals; in contrast, these measurements in RNA-seq data are treated as discrete counts. Microarray data, as a result, commonly assumed to follow a normal distribution (see Figure 1.2), while the Poisson and the **n**egative **b**inomial (NB) distributions are two most suitable for modeling non-negative data in an RNA-seq experiment (Wang et al., 2010; Auer and Doerge, 2011; Di et al., 2011).

However, the assumption of Poisson distribution for the read counts is too tight because it does not reflect the biological variations in the data (Robinson and Smyth, 2007; Nagalakshmi et al., 2008). This disadvantage is derived from the simplicity of Poisson distribution; it assumes that the variance of the model is equal to the mean. Ignoring the biological replicates so-called over-dispersion problem the statistical analysis does, therefore, not control the false-positive rates because of the underestimation of sampling error (Anders and Huber, 2010). To deal with this problem, the NB distribution as a replacement for Poisson distribution in modeling

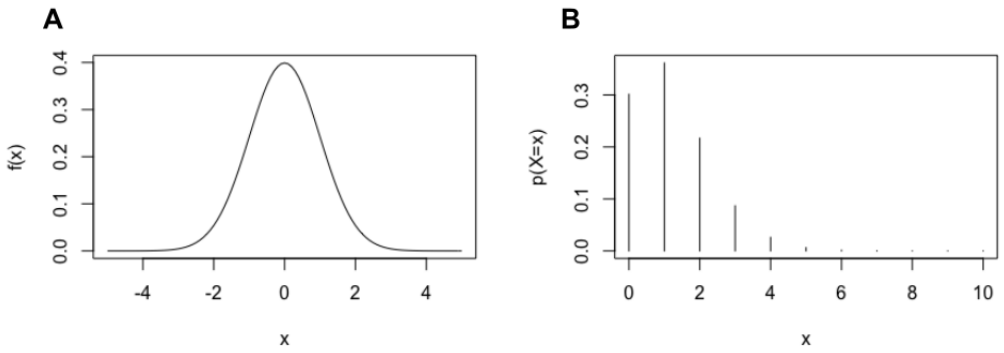


Figure 1.2: An illustration for distributions. **A**: Normal distribution with mean equals to zero and variance equals to one; **B**: Poisson distribution with mean equals to one

count data. The NB distribution is a family with two parameters, variance and mean, which the former is greater than the later (Robinson et al., 2010; Anders and Huber, 2010; Love et al., 2014). Another alternative is the transformation of the RNA-seq data using a simple logarithm, a more complex variance stabilizing transformation (Anders and Huber, 2010) or the regularized logarithm (Love et al., 2014). The *voom* and *trend* transformations, proposed in Law et al. (2014), unlock the use of models developed for microarrays to RNA-seq (Ritchie et al., 2015). Abundance of software supports statistical tests for detecting differentially expressed genes based on the distribution assumption of RNA-seq count data : `DEGseq` (Wang et al., 2010) based on Poisson distribution, `DESeq` (Anders and Huber, 2010), `DESeq2` (Love et al., 2014), `edgeR` (Robinson et al., 2010) based on NB distribution, and `limma` (Law et al., 2014) based on normal distribution. One should consider normalization before performing statistical analysis. It is an essential procedure designed to identify and correct technical biases was presented due to library preparation protocols and sequencing platforms. Normalization has a great impact on **d**ifferential **e**xpression results (Dillies et al., 2013; Bullard et al., 2010), even more than the selection of test statistic applied in hypothesis tests for DGE analysis. Some classical procedures for normalization of RNA-seq will be presented in Section 2.1.3.

1.3 Supervised learning methods

Supervised learning is an algorithmic process that learns a function mapping input to output based on example input-output pairs. Therefore, the essential goal of supervised learning is to best approximate the mapping function, and then when one has new observations, this model can predict the output variables of these data. According to the type of output data that models have to forecast, supervised learning is typically classified into classification, and regression, which are used for predicting categorical and continuous outcomes, respectively.

1.3.1 Traditional supervised learning models

We want a model f to predict y_i based on $\mathbf{x}_i$: $\hat{y}_i = f(\mathbf{x}_i)$. The value $\hat{y}_i$ is the prediction for sample i. Among numerous supervised **m**achine **l**earning (ML) algorithms, we present some classical models, such as linear regression, logistic regression, Naïve Bayes classifier.

Linear regression. One of the most basic algorithms of supervised learning predicts a real-valued output y_i, it is also known as **Linear Least Square**. The linear regression model has the form:

$$f(\mathbf{x}_i) = \beta_0 + \sum_{k=1}^{p} x_i^{(k)} \beta_k \tag{1.2}$$

where:

- β_0 is a constant, known as intercept or bias term.
- β_k is the slope of the regression line.

The goal of linear regression model is to search the values for β_0, β_k, to provide the best fit line for the data points. Least square is the most prevalent estimation

method; coefficients $\beta = (\beta_0, \beta_1 \dots \beta_p)$ are calculated to minimize the sum of squares error between $\hat{y}_i$ prediction and the actual y_i value (so-called **r**esidual **s**um of **s**quares - RSS)

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 = \sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_i^{(k)}\beta_k)^2 \tag{1.3}$$

Logistic regression. Logistic regression trains a classifier to make a binary decision about the class of a new input observation. The sigmoid function is a classifier that can help one to make this decision: it takes a real value and transforms it to the range $[0, 1]$.

- $P(y_i = 0 \mid \mathbf{x}_i)$ is the probability that the new observation $\mathbf{x}_i$ belongs to class 0.
- $P(y_i = 1 \mid \mathbf{x}_i)$ is the probability that the new observation $\mathbf{x}_i$ belongs to class 1.

Logistic regression makes the decision by learning, from a training set, the linear functions in $\mathbf{x}_i$:

$$t = \beta_0 + \sum_{k=1}^{p} x_i^{(k)}\beta_k \tag{1.4}$$

To create a probability, we use the sigmoid function: $t \rightarrow \dfrac{1}{1 + e^{-t}}$

Based on the decision boundary, one can make a decision

$$\hat{y}_i = \begin{cases} 1 & \text{if } P(y_i = 1|\mathbf{x}_i) > 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{1.5}$$

`Naïve Bayes` **classifiers.** `Naïve Bayes` classifiers are a group of algorithms based on the Bayes' theorem for classification problems. There are two fundamental assumptions in `Naïve Bayes`:

1. *Independent*: All features are assumed to be independent, meaning that for any pair of features randomly taken, the two features are not dependent.

2. *Equal*: The contribution of each feature to the model outcome is equal.

In ML classification problem, our interest is to find the best model f to predict class label y_i based on $\mathbf{x}_i$. The simple way is selecting the most probable model given by the data that we have. Bayes' theorem provides a way to compute the probability of an event y_i happening from prior knowledge.

$$P(y_i|\mathbf{x}_i) = \frac{P(\mathbf{x}_i|y_i)P(y_i)}{P(\mathbf{x}_i)} \tag{1.6}$$

where:

- P($y_i|\mathbf{x}_i$): is the probability of y_i given the data $\mathbf{x}_i$ (is-called posterior probability).
- P($\mathbf{x}_i|y_i$): is the probability of data $\mathbf{x}_i$ given the y_i was true.
- P(y_i): is the probability of y_i being true. This is called the prior probability of y_i.
- P($\mathbf{x}_i$): is the probability of the data $\mathbf{x}_i$

Other supervised algorithms. Support vector machines, discriminant analysis, k-nearest neighbor algorithm and neural networks are also widely used for supervised learning. We refer to James et al. (2013) for more details. Various popular packages were developed with different programming languages, such as the R package `caret` (Kuhn, 2008), the Python package `scikit-learn` and `TensorFlow` (Pedregosa et al., 2011; Abadi et al., 2016) which implement many supervised algorithms and provides a unified framework for performing and assessing the performance of supervised algorithms.

1.3.2 Cross-validation

Validation set method. This method is a simple strategy that can estimate the error when fitting a particular ML algorithm on a set of observations (called test error). First, the available collection of observations is randomly split into two parts, a training set and a validation set or hold-out set. Then, we fit the model on the training set, and the fitted model is used to predict the outcome of the observations in the validation set. Although simple and easy to implement, the validation set method poses two significant limitations regarding the quality and quantity observed in the training and validation set. The first drawback is that the error rate depends on which observations are placed on the training set and the validation set. Secondly, the fitted model was trained in a subset of observations, and the ML models tended to perform worse on the training set with a small number of observations. It also means that the error rate may be overestimated or higher than the error rate obtained when the model is fitted on the whole dataset.

k-fold cross-validation. To solve the two problems raising by validation set method, we present k-fold **c**ross-**v**alidation (CV) as an improvement of this approach.

In the k-fold CV, the original dataset is randomly divided into k folds of approximately equal size. The first k-1 folds are used to train the model, and the last fold is treated as the test set. In practice, one usually uses k-fold CV with $k = 5$ or $k = 10$ depending on the number of observations. This process is repeated until every k-fold serves as the test set. Then, the error rate is aggregated by averaging the error rate of each single estimation.

The validation set method may tend to overestimate the test error rate, because according to this approach, the ML classifier is trained with the training set containing only half of the observations of the whole dataset. In k-fold CV for, say, $k = 5$ or $k = 10$, each training set includes $(k - 1) \times n / k$ observations - considerably more than in the validation set approach. When several supervised learning models

are assessed, a nested k-fold **c**ross-**v**alidation is used, as detailed in Lever et al. (2016).

1.3.3 Overfitting and regularization

The bias-variance trade-off. This section presents the problem of overfitting for continuous variable prediction, also relevant for binary variable prediction. In the case of continuous variable prediction (regression), the estimated **M**ean **S**quared **E**rror (MSE)

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

on the testing dataset can be used to assess the performance of a supervised learning algorithm. Other metrics for binary variable prediction are presented in Section 1.3.6. The MSE can be written as a sum of two terms : the bias and the variance. The bias refers to the error due to a fitted model far from the true unknown model (James et al., 2013). The variance refers to the sensitivity of the the model to the training data. The relationship between the MSE, the bias and the variance is referred as the bias-variance trade-off: a model with a lot of parameters (lot of variables to predict the outcome) will have a low bias and a large variance, a model with few parameters will have a large bias and a low variance. A model with a low bias and a large variance overfits the data: the model works well in training set but fails to generalize on future observations. To minimize the MSE, we can choose to increase the bias (remove some variables from the predictor) if we greatly decrease the variance. This is the goal of the regularization or shrinkage techniques. The two best-known shrinkage methods are `ridge` regression (Hoerl and Kennard, 1970) and **L**east **A**bsolute **S**hrinkage and **S**election **O**perator (LASSO) regression (Tibshirani, 1996) and their hybridization, the `elastic net` (Zou and Hastie, 2005). The `LASSO` and `ridge` penalizations are presented below for the case of linear regression, but are also generalized to logistic regression (James et al., 2013).

Ridge regression. Ridge regression is quite similar to least squares estimation method, except the coefficients are calculated to minimize a slightly different quantity

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_i^{(k)}\beta_k)^2 + \lambda\sum_{k=1}^{p}\beta_k^2 = RSS(\beta) + \underbrace{\lambda\sum_{k=1}^{p}\beta_k^2}_{\ell_2\text{penalty}} \tag{1.7}$$

Where $\lambda \geq 0$ is a tuning parameter, which indicates the impact of the penalty term.

- $\lambda = 0$, Ridge regression produces the least squares estimate.
- $\lambda \to \infty$, the strength of shrinkage penalty grows, and the ridge regression coefficient estimates will reach zero. Each value of λ generates a different set of coefficient β_k estimates, therefore choosing a good value for λ is essential. One commonly selects λ based on the bias-variance trade-off. Here is the general trend between variance and bias when tuning λ: as λ increases, the bias increases while the variance decreases.

LASSO regression. The penalty term in Equation (1.7) shrinks all coefficients towards zero, but none of them are set to 0. This may not affect the prediction accuracy, but it raises a challenge for model interpretation in which there are a huge number of features. In this situation, LASSO is an alternative method for ridge regression, replacing the ℓ_2 penalty by ℓ_1 penalty.

$$\sum_{i=1}^{n}(y_i - \beta_0 - \sum_{k=1}^{p} x_i^{(k)}\beta_k)^2 + \lambda\sum_{k=1}^{p}|\beta_k| = RSS(\beta) + \underbrace{\lambda\sum_{k=1}^{p}|\beta_k|}_{\ell_1\text{ penalty}} \tag{1.8}$$

When one tunes parameter λ to sufficiently large, the ℓ_1 penalty forces some coefficient estimates to equal zero exactly. Hence, the LASSO performs variable selection.

1.3.4 The curse of dimensionality

When the number of variables p exceeds the number of samples n ($p \gg n$), there are infinite solutions to minimize RSS (see Equation 1.2 for linear regression), so

it is impossible to use linear regression. To deal with that, we can shrink small coefficients towards zero using regularization techniques presented above, to select a smaller number of variables of size s to predict the outcome. However, the regularization techniques are not efficient when the number of variables p is too large compared to n. In some cases, when p and s are too large compared to n, the selection of the right subset of s variables will fail (Fan and Lv, 2008).

Unfortunately, in our real-life datasets, such as gene expression data, the number of features could be more than 50,000, while the number of observations only a few hundred. In this situation, one suffers the curse of dimensionality, initially detailed in Bellman (1966): sampling a high dimensional space is hard and requires a large number of observations. In ultrahigh dimensional settings, spurious correlation can appear between totally independent variables.

To illustrate this phenomenon, we use the following toy simulated example: we simulated a sample of size n from a p-multivariate normal distribution with a diagonal covariance matrix. We compute the empirical correlation between any pair of variables and take the maximum absolute empirical correlation. We repeat the simulation 5000 times and plot the distribution of the maximum absolute empirical correlations for three settings: one low dimension setting ($n = 1500; p = 100$) and two high dimension settings ($n = 30; p = 100$ and $n = 10; p = 100$).

As seen in Figure 1.3, in low dimension, the 5000 maximum absolute empirical correlations are distributed between 0 and 0.15, which is expected given that all variables are independent and the true correlation between any pair of variables is equal to 0. However, in high dimension, the maximum absolute empirical correlations are distributed between 0.5 and 0.85 when the number of observations is $n = 30$, and between 0.8 and 1 when $n = 10$. In high dimension, we can find two variables highly correlated (empirical correlation higher that 0.9) when their true real correlation is actually 0. This simulation illustrates why we should be careful when looking for association between variables in high dimension.

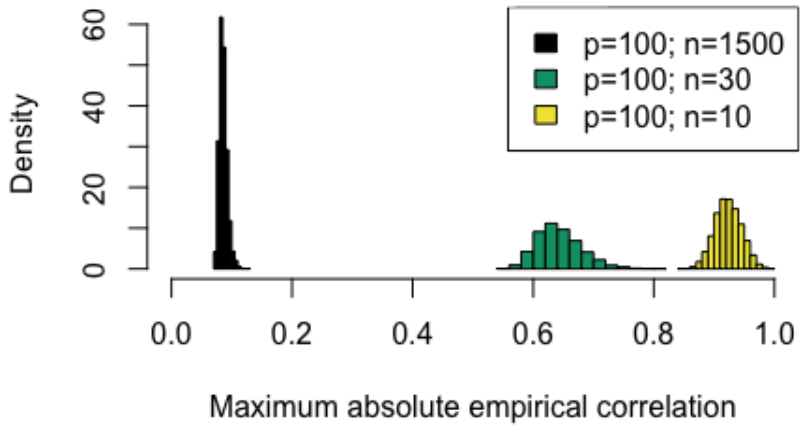


Figure 1.3: Maximum absolute empirical correlation observed in 5000 datasets generated under a multivariate normal distribution with a diagonal covariance matrix for three settings : one low dimension setting ($n = 1500; p = 100$) and two high dimension settings ($n = 30; p = 100$ and $n = 10; p = 100$). The true correlation between any pair of the p variables is 0. However, in high dimension, it is possible to find a pair of variables with high empirical correlation.

1.3.5 Feature selection for supervised learning

Feature selection is a preprocessing technique to identify a subset of features that gives a better comprehension from the input data while providing a predictor is not only faster, cost-effective, but also higher prediction performance (Guyon and Elisseeff, 2003). Moreover, feature selection is one of the approaches for dimensional reduction by selecting relevant features, eliminating irrelevant and redundant features, therefore avoiding the curse of dimensionality (Guyon et al., 2008). The goal of feature selection is also to increase the interpretability of the prediction. Depending on how one combines feature selection techniques with classifier modeling, there are three types of feature selection including *filters*, *wrappers* and *embedded methods*.

Feature selection approaches

Filter methods. Filter methods evaluate the relevance of features based on the intrinsic characteristics of data, independent of the learning method. This type of feature selection, in general, calculates a score for each feature and then ranking them. Selected features are obtained by a given threshold or a number of features that one desires to keep. Filter methods include parametric methods such as the classical t-test or non-parametric methods such as the Wilcoxon sum-rank statistics (Haury et al., 2011; Wilcoxon, 1945). We refer to Lazar et al. (2012) for an overview of filter methods.

The most outstanding advantage of filter approaches is that its computation is fast and straightforward, which is often preferred as the first processing step in a high-dimensional dataset. However, the influence of selected feature subsets on the model performance is entirely ignored in these approaches.

Wrapper methods. Conversely, wrapper and embedded methods depend on the learning method to employ feature selection. For wrapper methods, one regards each feature subset as a search problem. They iteratively perform a ML algorithm for each feature subset and then evaluate these subsets based on the model accuracy. The selected feature subset will be the optimal subset with the highest model accuracy. As a result, the major disadvantage of wrapper methods is computationally costly. But they generally tend to outperform other filter methods (Kohavi et al., 1997). `Recursive Feature Elimination (RFE)` is the gold standard for wrapper-type feature selection methods that was proposed by Guyon et al. (2002). In addition, `Genetic Algorithm (GA)` is a typical algorithm for wrapper feature selection methods that was applied in several bioinformatic projects (Petricoin III et al., 2002; Li et al., 2004; Thomas et al., 2019).

Embedded methods. Embedded techniques incorporate feature subset selection during the model is being built, thus reducing the computation time compared to

wrapper-type techniques. Several popular embedded methods are random forest or regressions or classification combined with LASSO regularization.

Stability selection

In high dimension, the feature selection methods are known to be highly unstable : a slight change in the dataset used to select the feature can lead to a totally different set of selected features (Ein-Dor et al., 2006; Michiels et al., 2005). Meinshausen and Bühlmann (2010) proposed a method of stability selection, which improved the performance of several different feature selection algorithms including LASSO regression. Instability is a well-known drawback of the LASSO : when two variables are highly correlated, the LASSO randomly selects one out of the two. Stability selection is a generic subsampling approach that repeatedly performs a feature selection algorithm on several different subsamples. The selection results are aggregated from all repetitions, for example, counting how many times each feature ended up being selected in the important feature subset (as described in Algorithm 1.2).

Algorithm 1.2: Stability selection algorithm

Input: A dataset ($\mathbf{x}_i$, y_i), the maximum number of subsamples MAX_S,
a list of regularization parameters: Λ

Output: List probability of each feature to be selected

1: for every integer $\in \{1, \ldots,$ MAX_S $\}$
2: Create a random subsample I of $\{1, \ldots, n\}$ of size $\lfloor n/2 \rfloor$ without replacement
3: for $\lambda \in \Lambda$
4: Fit the feature selection algorithm on I using regularization λ.
5: Store the set of selected feature $\hat{S}^{\lambda}(I)$
6: end for
7: end for
8: for $k \in$ list of p features
9: for $\lambda \in \Lambda$
10: Compute the selection probability $\Pi_k^{\lambda} = P(k \in \hat{S}^{\lambda}(I))$
11: end for
12: end for
13: return Π_k^{λ}

Applying the stability selection algorithm, we have the probability of each feature

k to be selected Π_k^λ. Therefore, the final list of stable features is defined based on list of regularization parameters Λ and on a given cut-off π_{thr} $(0 < \pi_{thr} < 1)$, in practice, one often chooses $\pi_{thr} \in (0.6, 0.9)$ (Meinshausen and Bühlmann, 2010):

$$\hat{S}^{stable} = \{k : \underset{\lambda \in \Lambda}{max}(\Pi_k^\lambda) \geq \pi_{thr}\}$$

Stability selection has the benefit of controlling the FWER (Meinshausen and Bühlmann, 2010), which allows for an accurate statement about the significance of selected features.

The selection bias

When assessing the performance of feature selection combined with supervised learning model, we should be careful to avoid the selection bias described in Ambroise and McLachlan (2002). A classical mistake is to perform feature selection on the whole dataset, and work with the dataset restricted to the selected features to perform **c**ross-**v**alidation and assess the model performance. This strategy leads to overestimate the prediction error, given that the dataset used to assess the performance (testing set) was also used to select the variables. The performance of a predictive model has to be evaluated on truly unseen data.

1.3.6 Supervised learning for prediction of binary variables

When the variable to predict is not continuous, the MSE presented in Section 1.3.3 is not adapted and other performance measures must be used. The general ideas detailed in Sections 1.3.3 and 1.3.4 can be adapted to prediction for categorical variables. Below, we present the performance measures for binary supervised learning when $y_i \in \{0; 1\}$ (classification) and the problem of learning on imbalanced datasets.

Performance measures for binary supervised learning

To find out how effective is the ML model, we use different performance measures, such as Accuracy, Recall, F1-score ... The selection of measure depends on the purpose and significance of the study as well as the proportion of the number of observations in each classification group.

- **Confusion matrix**: is one of the most intuitive and most straightforward metrics used for assessing the correctness and accuracy of the model. Interestingly, confusion matrix itself is not a performance measure, but almost all of the performance metrics are derived from it. Where:

Table 1.3: Confusion matrix

		Actual	
		Positives	**Negatives**
Predicted	**Positives**	TP	FP
	Negatives	FN	TN

  - TP: the actual class of the observation was 1(True) and the predicted is also 1(True).
  - FP: the actual class of the observation was 0(False) but the predicted is 1(True).
  - FN: the actual class of the observation was 1(True) but the predicted is 0(False).
  - TN: the actual class of the observation was 0(False) and the predicted is also 0(False).

- **Accuracy**:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{1.9}$$

  Accuracy is a good measure when a dataset is well balanced between classes.

- **Precision**:

$$Precision = \frac{TP}{TP + FP} \tag{1.10}$$

Precision is a measure that shows the percentage of correctly predicted positive observations out of the total number of predicted positive observations. High precision reflects a low false positive rate.

- **Recall or sensitivity or True Positive Rate (TPR)**:

$$Recall = \frac{TP}{TP + FN} \tag{1.11}$$

This measure indicates the percentage of correctly predicted positive observations out of the total number of really positive ones. Therefore, Recall refers to an algorithm's sensitivity to precisely classify the positive observations.

- **False Positive Rate (FPR)**:

$$FPR = \frac{FP}{TN + FP} \tag{1.12}$$

FPR calculates the rate between negative observations that are misclassified and the total number of truly negative observations.

- **The Area Under the Receiver Operating Characteristics Curve**: Various researchers have adopted this measurement for assessing the classifier algorithm's performance. The **R**eceiver **O**perating **C**haracteristics (ROC) is a curve showing the relation between TPR and FPR at different thresholds. Thus, this curve describes the correlated trade-off between true and false positives. The **A**re **U**nder the ROC **C**urve (AUC) is the total two-dimensions place below ROC curve from (0,0) to (1,1) as pointed in Figure 1.4. The higher AUC, the better classification algorithm is in predicting observations. For more information on ROC analysis, we refert to Fawcett (2006).

- **F1-score**:

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1.13}$$

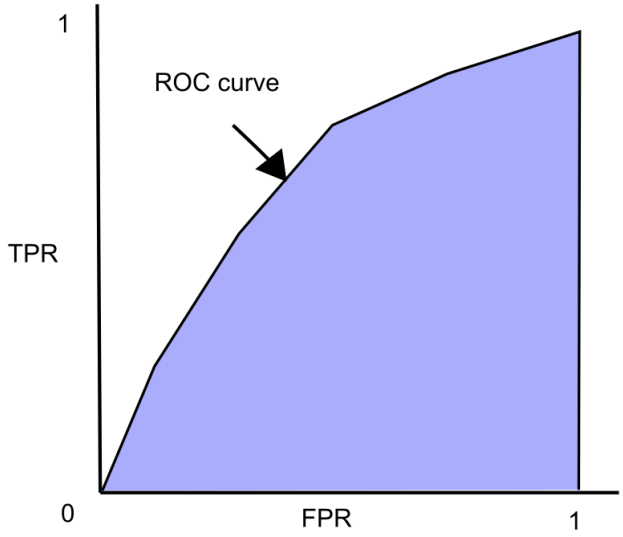


Figure 1.4: AUC - Area Under the ROC Curve

Depends on the problem to be addressed in your research, one can maximize precision or recall separately. However, if you're really interested in maximizing both then the F1-score is a suitable measure. F1-score takes both FP (so called a type I error) and FN (so called a type II error) into consideration.

Supervised learning on imbalanced datasets

Imbalanced datasets are a particular case for supervised learning where almost all observations are labeled in one class (the majority class), while fewer observations are labeled in the other class (the minority class), often more important.

When data is imbalanced, several problems can occur, such as the cost of misclassifying a minority class is typically much higher than that in a majority class. Additionally, most traditional classifiers assume an equal distribution of classes; thus, such models focus on learning characteristics of the majority class solely, ignoring the observations from the minority class that are, in fact, more meaningful and whose predictions are more relevant.

A simple technique for handling imbalanced datasets is sampling the data that

includes oversampling and undersampling.

- **Over-sampling (Up-sampling)** is the technique that adds observations to the minority class in order to reduce the skew in the class distribution (Chawla et al., 2002; Menardi and Torelli, 2014).

- **Under-sampling** removes observations from the training set that belong to majority class in an effort to better balance the class distribution (Kubat et al., 1997).

Both these techniques are implemented in `caret` and `imbalanced-learn` (Lemaître et al., 2017) packages.

Due to the development of deep learning, more recent techniques such as **G**enerative **A**dversarial **N**etworks (GAN) (Goodfellow et al., 2014) are used to efficiently deal with data imbalance. This deep-learning-based generative model includes two sub-models: a generator and discriminator models. In details, the former generates artificial samples from the existing data, while the latter classifies these generated samples as either real or fake samples. Both models are trained simultaneously. After each round, the discriminator is then updated to better distinguish real and fake samples, and according to the feedback of the discriminator, the generator is also updated.

1.3.7 Survival analysis

In logistic regression techniques, scientists are interested in studying models that predict whether a patient is cancerous or healthy. However, in some cases, they are interested in evaluating how a new treatment affects the duration of recovery or recurrence. In these cases, the standard regression is not the appropriate choice. Survival analysis is an alternative approach in which time until the event is concern. This section gives a quick overview of survival analysis techniques.

Censored observations

As discussed above, survival analysis concerns the expected duration when an event occurs (recurrence or death). Though, this event may not have been observed during the study period for some patients (so-called *censored* observations or censoring). The label of a samples y_i is not a continuous or categorical variable, such as presented in the previous sections. Here, $y_i = \{t_i, \delta_i\}$ where t_i is the survival time, time-to-event or follow-up time, and δ_i indicates if the event has been observed. If $\delta_i = 1$, the event (death, recurrence or relapse) has been observed. If $\delta_i = 0$, the observation has been censored. There are several types of censoring but mostly right-censoring. Here are some reason for right-censoring:

- A patient đid not experience the event during the study period.
- A patient dropped out before the end of study.
- A patient was lost followup time within the study duration.

Survival and hazard functions

To describe survival data, one often uses the survival and hazard functions. The survival function is the probability that a patient survives from the time start (*e.g.*, diagnosis of cancer) to a specific time in the future, also known as survival probability. While the hazard is the probability that a patient experiences an event at a certain point of time.

Various methods are used for estimating survival function or survival curve, which differ due to the assumptions of survival time distribution, such as a classical non-parametric estimator - `Kaplan-Meier` (Kaplan and Meier, 1958).

In `Kaplan-Meier` method, they assume that:

1. Censored patients have the same survival probability as patients who continue

to be followed.

2. Participation timing in the study does not affect the probability of survival.

3. The event of interest happens at a specified time.

The **Cox P**roportional **H**azards (CoxPH) model is useful to model follow-up times and their links to expression data. Under this model, the hazard function is $h(t|\mathbf{x}_i) = h_0(t)\exp(\mathbf{x}_i^T\beta)$ where $h_0(t)$ is the baseline hazard function, and β are the coefficients chosen to maximize the fitness of the model to the observed data. We refer to Witten and Tibshirani (2010a) to find methods to select features associated with survival in the model CoxPH model presented above.

1.3.8 Interpreting supervised learning models

Supervised techniques have seen widespread adoption to classify and predict patient etiology or outcome. The list of interpretable models includes several simple models, *e.g.*, linear regression, decision trees, and naive Bayes classifier. However, supervised learning models remain mostly as black-boxes. This leads to an obstacle in deploying these predictive models because humans do not understand and trust them. There are several techniques to help users understand the rationale behind the back-box model's predictions, such as **L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations (LIME) (Ribeiro et al., 2016). As a model agnostic technique, LIME can explain the prediction of any classifier model. The mechanism of LIME is that it modifies the input by creating permuted samples from the original data and evaluate the changes in the prediction.

1.3.9 Supervised learning on gene expression data

Combining supervised learning and feature selection on transcriptomic data are increasingly used in identifying transcriptome signatures associated with diseases

(Jhun et al., 2017; Yoosuf et al., 2020). A signature is a set of genes/transcripts whose behavior maximizes the prediction performance. We want to predict variables such as the tumor status (normal or tumor), the types of the cancer or the prognostic outcome. Perou et al. (2000); van 't Veer et al. (2002) have used expression data to classify samples into subclasses. However, it has been noticed that several signatures derived on different datasets for the same prediction problem poorly overlap (Michiels et al., 2005). The gold standard in supervised learning is to test the prediction model on truly unseen data. In this chapter, we have given a quick overview of the supervised learning and feature selection techniques. We have not presented all existing techniques, such as Support-Vector machines or kernel methods, k-nearest neighbors, random forest or linear discriminant analysis. Given a dataset, it would be easy to try as many supervised learning methods as possible, as many feature selection techniques as possible to select the "best" method (*i.e.* the set of choices leading to the results that we want to observe). However, it is not an appropriate approach, given that we are likely to find over-optimistic results. In this thesis, we have worked mainly with the logistic regression (and the `LASSO` logistic regression to select features) because the model is well-understood from the theoretical point of view, and the results are easy to interpret.

1.3.10 Differential analysis versus supervised learning

In section 1.2, we have presented an overview of differential analysis of gene expression data. In section 1.3, we have presented supervised learning techniques, **m**achine **l**earning-based techniques. The rationale behind each approach are quite different. Supervised techniques are used to classify and predict, usually patient etiology or outcome. Differential analysis relates to statistics where our goal is to draw inference on a population using observed data. Supervised learning techniques relates to ML where our goal is to find predictive patterns to predict the label of future, unseen observations. The most statistically significant DE genes are not necessary the most predictive genes. The most predictive genes have good

generalization performance, and are not necessary the ones with the highest mean expression difference across experimental conditions or patient status. The two approaches do not use the same diagnostic metrics : classical statistics approaches seeks the control of Type I and II errors using the whole dataset whereas supervised learning approaches use data split with training and test sets and metrics such as MSE or ROC-AUC evaluated on the test set. We refer to Bzdok (2017) for a discussion on the links and differences between classical statistical approaches and **m**achine **l**earning approaches.

1.4 Unsupervised learning methods

Unsupervised learning is an algorithmic process that models the underlying structure or distribution in the data from (only) input data without corresponding output variables, *i.e.*, there is no correct answer. The algorithms realize an automatic exploratory analysis and present interesting structures in the data. In unsupervised learning, there are various popular techniques, such as clustering, isolation forests, and variational autoencoder. In this section, we focus on clustering techniques with some classical clustering algorithms, on how to evaluate the clustering results, and we present two main clustering approaches for gene expression data.

1.4.1 Traditional clustering techniques

Clustering techniques typically include four major following categories: partitioning, hierarchy, density-based, as well as grid-based (Han et al., 2011). Each category has its own idea and also produces various typical algorithms. For example, the centers of the data points regards as the centers of the respective clusters is the idea behind partitioning clustering. Most partitioning algorithms are based on distance and given the number of partitions that need to be constructed in advance, *e.g.*, `K-means` (MacQueen et al., 1967), `K-medoids` (Park and Jun, 2009). In contrast, hierarchical clustering methods tend to group data points into hierarchical clus-

ters that are particularly beneficial for data visualization and generalization. BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) (Zhang et al., 1997) and Chameleon (Karypis et al., 1999) algorithms are famous for this type of clustering technique.

Partitioning and hierarchical clustering, as mentioned, are both data-oriented techniques, *i.e.* one partitions the set of data points to form a cluster and then accommodates to their distribution data in the space. The grid-based clustering method alternatively employs a space-oriented approach since it splits the spatial data into cells independent with data distribution. Several algorithm were implemented for this idea, *e.g.*, STING (Wang et al., 1997), CLIQUE (Agrawal et al., 1998). Among four clustering categories, density-based algorithms allow to find clusters of different shapes and sizes while remaining robust to noise in the data with several algorithms, such as DBSCAN (Ester et al., 1996), OPTICS (Ankerst et al., 1999).

In the next section, we will introduce K-means and DBSCAN, respectively that are two most classical and well-known algorithms for unsupervised clustering.

Reminder, we are given a training set including n samples: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$; where $\mathbf{x}_i$ is an expression vector of sample i across p features, *i.e.*, $\mathbf{x}_i = (x_i^{(1)}, x_i^{(2)}, \ldots, x_i^{(p)})$. Our target is to group n samples into several clusters.

K-means algorithm

This partition-based clustering is one the most popular and straightforward algorithm. The number of clusters K are given and one uses them to define clusters. A sample is assigned to a particular cluster if the Euclidean distance from it to that cluster's center is shortest.

K-means finds the best center based on two alternating steps (1) assigning samples to their clusters according to the current centers (2) updating cluster centers according to the current assignment samples. The pseudo-code of K-means clustering algorithm is as below (Algorithm 1.4):

Algorithm 1.4: Pseudo-code of K-means clustering algorithm

Input: n samples: $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$, number of cluster: k, maximum number of iterations: MAX_ITER
Output: A partition $P = [\ C_1, C_2, \ldots, C_k\ ]$

$cnt = 0$; $P = []$
Initialize μ_i $(i = 1, \ldots, k)$cluster center by choosing randomly k samples from n samples
while
    cnt +=1
    Cluster assignment: each sample $\mathbf{x}_j$ assigns to the nearest cluster
    $C_i^{(cnt)} = [\ \mathbf{x}_j$ if $distance(\mathbf{x}_j, \mu_i) \leq distance(\mathbf{x}_j, \mu_l)$ for $l = 1, \ldots, k\ ]$
    Update centers
    $\mu_i^{(cnt+1)} = \frac{1}{|C_i^{(cnt)}|} \sum_{\mathbf{x}_j \in C_i^{(cnt)}} \mathbf{x}_j$
    Update P
    $P^{(cnt)} = [\ C_1^{(cnt)}, \ldots, C_k^{(cnt)}\ ]$
    if $cnt \geq$ MAX_ITER or $P^{(cnt)} = P^{(cnt-1)}$
        return $P^{(cnt)}$
    end if
end while

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm

DBSCAN is a famous clustering algorithm, was the first introduced by Ester et al. (1996). The basic idea behind DBSCAN is how it defines a cluster as a connected dense region, therefore, samples in the high-density space tend to fall into the same cluster.

In detail, DBSCAN determines clusters by evaluating the local density at each sample using two parameters: distance radius (ϵ) and minimum number of samples ($minPts$) that are located in the neighborhood ϵ of the sample.

According to these two parameters, DBSCAN divides data samples into 3 type of samples: core, border and noise samples as illustrated in Figure 1.5.

1. *Core sample*: a single sample has at least $minPts$ samples within the neighborhood ϵ of itself.

2. *Border sample*: a single sample has at least one core sample at its neighbor-

hood ϵ.

3. *Noise sample*: a sample is neither a core nor a border sample.

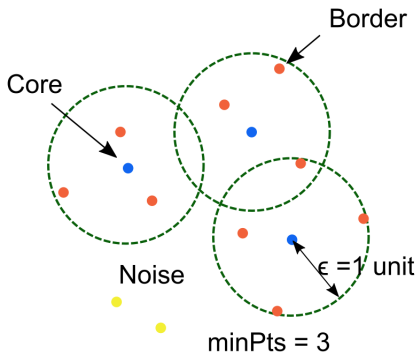


Figure 1.5: An illustration about 3 type of samples in DBSCAN algorithm with $minPts = 3$ and $\epsilon = 1$ unit. Blue, orange and yellow points are represented for the core, border and noise samples, respectively.

There are two major procedures in the implementation DBSCAN algorithm as presented in Algorithm 1.5 and Algorithm 1.6, respectively.

Algorithm 1.5: Pseudo-code of DBSCAN($setSamples, \epsilon, minPts$)

Input: setSamples =[$\mathbf{x}_1, \ldots, \mathbf{x}_n$]; distance radius: ϵ; minimum number of samples to form a cluster: $minPts$
Output: $Clusters$ stores all groups

```
C = 0; Clusters = []
for x_i ∈ setSamples and x_i is not visited
    Mark x_i as visited
    neighborsPts = findNeighbors(x_i, ε)
    if sizeof(neighborsPts) < minPts
        Mark x_i as noise sample
    else
        C = next cluster
        Clusters = Clusters ∪ extendCluster(x_i, neighborsPts, C, ε, minPts)
    end if
end for
return Clusters
```

Algorithm 1.6: Pseudo-code of extendCluster($s, C, \epsilon, minPts$)

Input: visited sample: s ; $neighborsPts$: list neighbors of s; the real cluster: C; distance radius: ϵ; minimum number of samples to form a cluster: $minPts$
Output: The real cluster C

```
Add s to cluster C
for s' ∈ neighborsPts
    if s' is not visited
        neighborsPts' = findNeighbors(s', ε)
        if sizeof(neighborsPts') >= minPts
            neighborsPts = neighborsPts ∪ neighborsPts'
        end if
    end if
    if s' is not member of any cluster
        Add s' to cluster C
    end if
end for
return C
```

DBSCAN does not specify number of clusters as partition-based K-means. This algorithm has the ability to discover clusters with arbitrary shapes, therefore DBSCAN

algorithm has better performance on the more complex structures (Ronan et al., 2016). In addition, it can deal with noise samples introduced in the dataset.

Nevertheless, this type of density-based clustering has some drawbacks. For example, to perform DBSCAN one needs two input parameters, $minPts$ and ϵ which together form the local density for each generated cluster. However, setting the right values of $minPts$ and ϵ engages a number of trials and results assessing phase. For each trial, DBSCAN algorithm runs with different values of these parameters in several times that are costly in the case of high dimensional dataset.

In terms of computational complexity, DBSCAN algorithm is greatly impacted by the number of times function `findNeighbors(...)` is invoked as shown in Algorithm 1.5 and Algorithm 1.6. As a result, if this operation is designed with optimization, the execution time of DBSCAN clustering will be significantly reduced.

1.4.2 Evaluation of unsupervised clustering

In the previous section, we have presented 4 fundamental categories of clustering techniques and various corresponding algorithms. The question is if one utilizes a clustering algorithm for their data, how do they evaluate whether the clustering result of the algorithm is good or not. In general, clustering evaluation relies on three major processes (Agarwal, 2013):

1. **Estimating clustering tendency** The purpose of this process is to discover whether the non-random structure certainly manifests in the dataset to ensure that clustering data makes sense.

2. **Specifying the number of clusters in a dataset** Few algorithms need to define the number of clusters in advance, such as K-means, K-medoids; as a result, it should be specified this value before applying a clustering technique.

3. **Computing clustering quality** based on the availability of the **ground-truth** we have two type of computing methods (i) extrinsic methods (ii) intrinsic

methods

1.4.3 Clustering approaches for gene expression data

The clustering of gene expression data has proven to be profitable in understanding gene function, gene regulation, and subtypes of cancer (Armstrong et al., 2002; Parker et al., 2009). An interesting aspect in the cluster analysis of gene expression matrices is that one can subdivide both genes and samples - the determination of applying gene-based or sample-based direction is centered on the crucial ambition of clustering tasks (Jiang et al., 2004). For example, if the task aims to infer groups of genes that are involved in the same cellular processes, gene-based clustering is an excellent choice. In such an approach, genes share similar profiles (co-expressed genes) across several treatment conditions that fall into the same clusters.

Gene-based clustering

Gene-based clustering is performed to group co-expressed genes, which may indicate co-function and co-regulation. The intrinsic characteristics of gene expression data, and the specific requirements from the biological domain take account to several great challenges for gene clustering present and is still an open problem.

First, it is impossible to have a single best algorithm for clustering (Pirim et al., 2012), and each algorithm imposes its own underlying structure biases on the data.

Second, as we discussed in Section 1.1.2, the gene expression matrix produced by microarray or RNA-seq experiments is a messy data *i.e.*, contains a tremendous amount of noise. As a result, gene-based clustering algorithms should be scale-well in order to obtain beneficial data from such a noisy environment.

Finally, a clustering algorithm does not only partition the dataset but also provides the visualization of cluster structure. It should give the biologists the multi-view about the gene clustering (*e.g.*, the relation between the clusters, the connection

between the genes within the same cluster)

Sample-based clustering

The aim of sample-based clustering is to partition samples into different groups. An aggressive sample-based clustering algorithm needs to ensure that samples within a group share similar expression patterns and vice versa; those from other groups pose noticeably different. The idea of clustering sample lead to discover the phenotype structures or sub-structures of the samples (*e.g.*, new cancer/disease subtype), and it was the first introduction in Golub et al. (1999) and Alizadeh et al. (2000). The study from Golub and his colleagues is a demonstration that samples can be distinguished by a small subset genes whose expression levels highly correlate with the class outcome (is-labeled informative genes). For other remaining genes are irrelevant to classify samples, and thus should be removed from the dataset.

There are two approaches in the selection of informative genes: supervised and unsupervised, depending on whether using actual class label information during this process. Though supervised gene selection techniques (Blum and Langley, 1997) are broadly applied, developing gene selection techniques using unsupervised learning is gaining more attention from the research community (Dy et al., 2003).

Chapter 2

Bioinformatics for RNA-seq analysis

Advances in RNA-seq has revolutionized measurement of transcriptome-wide gene expression through its ability to capture the full diversity of transcripts produced by each cell. It is a key to comprehend the functional and structural elements of the genome. However, conducting an RNA-seq study properly is a challenge due to the large number of published RNA-seq analysis protocols.

2.1 Conventional RNA-seq analysis

We present here the main bioinformatics processes used for conventional RNA-seq analysis. The selection of an analysis strategy depends on the organism studied and on research objectives. RNA-seq data analysis, in general, includes several steps as shown on the right side of Figure 1.1 part A "Data processing" - including quality control, alignment, and quantification.

2.1.1 Quality control

RNA-seq data are typically stored in FASTQ files that contain millions of raw reads. A read is a sequence obtained after the end of the sequencing process.

The limitations of each sequencing platform introduce errors, such as incorrect nucleotides that can lead to bias for the interpretation of downstream applications. As a result, sequence quality control is a crucial first step before doing any further analysis.

One of the pioneer programs for handling quality control on raw FASTQ files was FASTX-Toolkit (Hannon, 2010). This tool is capable of monitoring base quality and nucleotide distribution by a collection of Linux command-line tools. Another tool is the FastQC package developed by the Babraham Institute bioinformatics group at http://www.bioinformatics.babraham.ac.uk/projects/fastqc/. FastQC supports additional parameters to estimate sequence quality along all sequences, including quality estimated collectively across all reads within a sample, GC content distribution, and listing of over-represented sequences. As a general rule, sequence quality drops at the end of the sequences, raising the number of potentially incorrectly called nucleotides. Sequences, therefore, must be treated by trimming bases corresponding to low quality score regions. Also, if the reads are longer than the fragments sequenced, adapter sequences may be present in raw reads, which is an issue that reduces the aligned read rate (Criscuolo and Brisse, 2013). Software tools are available to perform either or both of these tasks (quality trimming and adapter removal); for example, Trimmomatic (Bolger et al., 2014) does both while Cutadapt (Martin, 2011) does the second.

2.1.2 Alignment/Mapping

Sequencing produces a collection of reads without a genomic context, so one does not know which part of the genome these reads belong to. Depending on the availability of a reference assembled genome, one can choose between two alternative options:

- **Reference-based transcriptomics**: when a reference genome is available for the organism, it is possible to infer a read's corresponding region by mapping it to the reference genome or transcriptome.

- *De novo* **assembly**: when working on an organism without a reference genome or incomplete reference genome, reads, first, can be extended into longer sequences called contigs. After that, these contigs are remapped to the assembled transcripts for the next step, quantification.

Mapping is the process of finding out the location of reads in the genome. Assessing mapping quality through checking the mapping statistics is a crucial step before continuing any downstream analyses. Mapping statistics indicate how well reads map to a reference. In case these statistics are not satisfying (*e.g.* the number of uniquely mapped reads is low), one should investigate the cause of these errors by looking into the **B**inary **A**lignment **M**ap (BAM) file that stores the read alignments.

The reads can either map to a reference genome or a reference transcriptome. In the context of alignment to genome, a reference genome and set of reads will be the input data for mapping tools. These tools employ this data to align each read against set of reads that are constitutive from a reference genome, allowing mismatches, insertion or deletion of bases. However, the precise alignment of reads that have splice junctions is the major challenge of genome alignment, specially when these junctions are not pre-annotated. Multiple bioinformatics tools are available to perform the alignment of short reads to a genome, including `Bowtie` (Langmead et al., 2009) and `STAR` (Dobin et al., 2013), `HISAT2` (Kim et al., 2019).

In case the organisms have a well-established transcriptome, the reads can directly align to a reference transcriptome which includes a set of known transcripts with most expressed isoforms. Consequently, reads align contiguously and various costly computations have been abandoned in this approach. However, transcriptome alignment produces a high rate of multi-mapping reads, resulting in computational challenges in such alignment. Several programs are designed for mapping read to a reference transcriptome, *e.g.* `Bowtie`, `Stamby` (Lunter and Goodson, 2011), `Salmon` (Patro et al., 2017).

Genome alignment or transcriptome alignment, each approach has different advan-

tages and disadvantages as described in the section above. Additionally, the former mapping allows for the discovery of novel genes or transcripts, meanwhile the latter is often faster and is the solely option when working on organisms without a reference genome or incomplete reference genome (Conesa et al., 2016).

2.1.3 Quantification

Quantification is the abundance estimation of transcripts or gene expression. Therefore, there are two levels of resolution, gene-level and transcript-level. The former is more common than the latter, but the latter has recently been recommended for all RNA-seq data analysis due to the significant improvement in gene expression quantification accuracy (Zhao et al., 2015).

Aggregation of raw counts of mapped reads, which is performed with software packages such as `featureCounts` (Liao et al., 2014) and `HTSeq-count` (Anders et al., 2015), is the simplest quantification method. This gene-level quantification takes as input multiple BAM files that are cross-referenced with a **G**ene **T**ransfer **F**ormat (GTF) file, which stores the genome coordinates of exons and genes.

Transcript quantification involves the assignment of fragments (reads or pair reads) to specific transcripts, which raises more challenges but also has several advantages. The crucial challenge here is that significantly more reads align equally well to multiple locations called multi-mapping reads. To tackle multi-mapping reads, various transcript quantification tools used **E**xpectation **M**aximization (EM) algorithm (Dempster et al., 1977), such as `Cufflink` (Roberts et al., 2011), `RSEM` (Li and Dewey, 2011), `StringTie` (Pertea et al., 2015), `kallisto` (Bray et al., 2016), `Salmon`. The EM algorithm is an iterative maximization approach for inferring the maximum likelihood estimation in problems with missing variables (so-called latent variables). In the case of multi-mapping reads, our aim is to find each transcript's abundance (latent variables) from given read data. The EM method involves two steps, first, it estimates the expected value for each latent variable, then optimizes

the model. It then repeats these two steps until it reaches a best fit for joint probability of data.

Despite rising significant multi-mapping reads, quantification of expression at the transcript level produces a straightforward interpretation due to transcripts reflects correctly what the cell expresses.

As said above standard quantification methods such as `featureCounts`, `HTSeq-count`, `Cufflinks` and `StringTie` rely on mapping to identify reads' positions in the genome based on their alignment to a reference. The gene or its isoform expression values are calculated by checking the number of overlapping reads, eventually with help of the EM algorithm. Meanwhile, software like `kallisto`, `Salmon` use pseudo-alignment, which does not specify the reads' positions in the transcripts, but instead compute their compatible transcripts based on common k-mer contents obtaining from k-mer analysis.

Raw read counts do not accurately reflect the expression level within or between samples, since these values are influenced by transcript length, total number of reads, and sequencing depth. For example, sequencing runs with more depth will have more reads aligned to each gene. Therefore, normalization is applied to identify and correct such technical biases. Some common normalization procedures are:

Reads Per Kilobase Million (RPKM). This normalization was first introduced by Mortazavi et al. (2008). To measure the raw read counts $x_i^{(k)}$ of the sample i mapped to gene k based on RPKM, there are 2 steps:

1. Normalize for read depth: calculate the "per million" factor (so-called RPM)

$$RPM(x_i^{(k)}) = \frac{x_i^{(k)}}{\frac{\sum_{k=1}^{p} x_i^{(k)}}{10^6}} \tag{2.1}$$

2. Normalize for gene length (scale per kilobase): Reads scaled for depth are further normalized for gene length.

As a result the formula for RPKM is as follows:

$$RPKM(x_i^{(k)}) = \frac{RPM(x_i^{(k)})}{\frac{geneLength^{(k)}}{10^3}} \tag{2.2}$$

where

- $x_i^{(k)}$: is the raw read counts of sample i mapped to gene k
- $geneLength^{(k)}$: is the length of gene k
- $\sum_{k=1}^{p} x_i^{(k)}$: is total raw read counts of sample i mapped to all p genes

Fragments Per Kilobase Million (FPKM). The FPKM measure (Trapnell et al., 2010) is a derivative of RPKM. However, RPKM is applied for single end RNA-seq, while FPKM is used for paired end RNA-seq. FPKM normalization ensures fragments that have two reads are not counted twice.

Transcripts Per Million (TPM). Normalization TPM (Li and Dewey, 2011) is quite similar to R/FPKM, except for the reverse order of computation.

1. Normalize for gene length:

$$RPK(x_i^{(k)}) = \frac{x_i^{(k)}}{\frac{geneLength^{(k)}}{10^3}} \tag{2.3}$$

2. Normalize for read depth: Reads scaled for gene length are further scaled for depth.

The formula of TPM is as follows:

$$TPM(x_i^{(k)}) = \frac{RPK(x_i^{(k)})}{\frac{\sum_{k=1}^{p} RPK(x_i^{(k)})}{10^6}} \tag{2.4}$$

where

- $x_i^{(k)}$: is the raw read counts of sample i mapped to gene k.
- $geneLength^{(k)}$: is the length of gene k
- $\sum_{k=1}^{p} RPK(x_i^{(k)})$: is total normalized raw read counts based on gene length of sample i mapped to all p genes.

Recently, TPM has largely replaced R/FPKM, as this method is more consistent across libraries (Soneson et al., 2015).

Other common normalization methods. Normalization of counts is crucial when performing differential analysis. There are several approaches to compute normalization scaling factors for differential analysis: DEseq scaling factor (Anders and Huber, 2010), trimmed mean of M values (TMM) (Robinson and Oshlack, 2010) or quantile normalization (Bolstad et al., 2003). We refer to Dillies et al. (2013) for more details.

2.2 k-mer strategies

k-mer analysis is a paradigm in next-generation sequencing data analysis that involves converting sequence files into k-mers (Pevzner, 1989), *i.e.*, all possible subsequences of length k obtained from reads produced by DNA or RNA sequencing as illustrated in Figure 2.1.

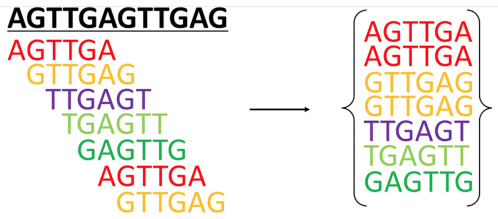


Figure 2.1: An illustration of how a read can be broken down into k-mers, in this case $k = 6$; Source: Hua and Zhang (2019)

Recently, k-mer-based methods have been utilized in several novel tools for alignment-

free transcript quantification, *e.g.*, kallisto, Salmon. The most noticeably is that the alignment-free quantification pipelines are significantly faster in computation than the alignment-based quantification pipelines. Since the former operates by breaking down reads into k-mers, then a fast matching process is conducted against a pre-indexed transcript databases.

Besides, k-mer approaches are used on genomic and metagenomic data. Wood and Salzberg (2014) have developed a program to assign taxonomic labels to metagenomic data using k-mers. Ounit et al. (2015) have classified metagenomic and genomic sequences using k-mers. Drouin et al. (2016) have used k-mers and **m**achine **l**earning to discover biomakers on genetic data.

2.3 Reference-free approaches for RNA-seq analysis

The reliance on a reference genome or transcriptome is a source of bias in conventional analysis of RNA-seq data since it leads to ignore numerous RNAs produced in disease tissues, notably through deficient RNA processing and genome alterations. Furthermore, there are still many species for which no reference genome or transcriptome is available. Hence, RNA-seq study independently of alignment or transcript assembly in a reference-free manner is an interesting alternative.

Throughout this thesis, we will use terms "reference-free" or "gene-free" to describe methods that do not rely on a reference genome or transcriptome, and "reference-based" or "gene-based" to describe conventional methods.

Below we describe three RNA-seq analysis methods with distinct aims, but that all work in a reference-free manner. DE-kupl performs differential expression analysis, GECKO performs predictive modelling and MINTIE finds RNAs which are specific to a sample by contrast to a set of control samples.

2.3.1 DE-kupl: exhaustive capture of biological variation using k-mers

Our laboratory contributed to the development of DE-kupl (Audoux et al., 2017), a pipeline that aims at capturing the full diversity of transcripts produced by each sample, *e.g.*, novel splice variants, **l**ong **n**on-**c**oding **RNA** (lncRNA), repetitive RNAs, through k-mer decomposition. This computational protocol has four principal parts (Figure 2.2):

- **Count k-mers**: Raw sequences stored in FASTQ files are indexed by the `jellyfish count` command of Jellyfish (Marçais and Kingsford, 2011) tools. Next, the `jellyfish dump` command is used to count k-mers (k=31) occurence in each library.

- **Filter and mask**: k-mers with counts lower than a preset threshold are removed. Also, k-mers present in reference transcripts can be removed to focus on "novel" k-mers.

- **Differential abundance analysis**: k-mers with significantly distinct abundances between the conditions under study are selected. In the thesis, we use the term "differential expression analysis" for k-mers, although it is a slight misuse of language given that k-mers are not "expressed" in the same sense than genes are expressed.

- **Extending and annotating**: k-mers are extended into contigs using the `dekupl-mergeTags` procedure; then, these contigs are annotated based on their sequence alignment.

The DE-kupl pipeline has several parameters to build and filter the k-mer counts matrix. Parameter `ctg_length` sets the length of the k-mers to extract (usually 31). Parameter `lib_type` sets the type of library (stranded or unstranded). Parameter `min_recurrence` sets the minimal number of samples and parameter

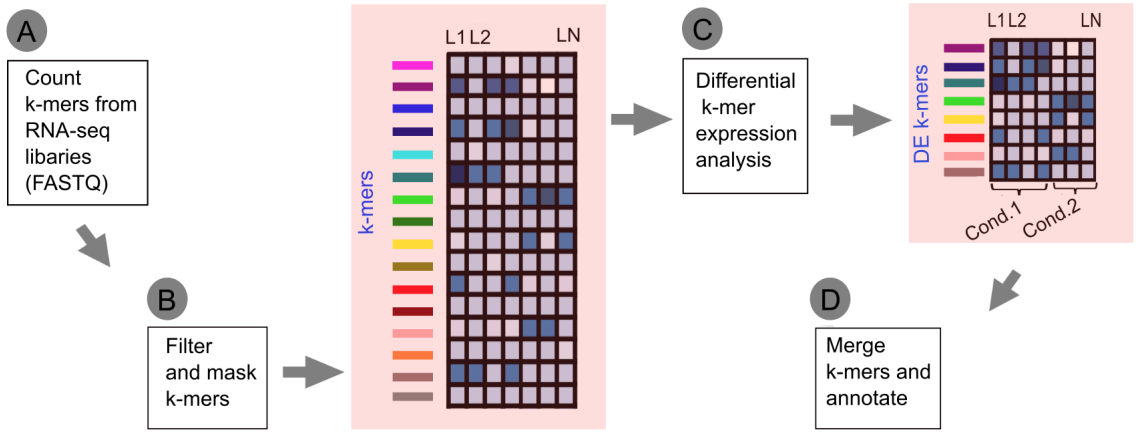


Figure 2.2: The DE-kupl pipeline includes 4 major steps. **A.** Jellyfish tool is used to create index and count k-mers present in all libraries. **B.** k-mer counts are joined into a count matrix, then, solely k-mers are high recurrence and absent in the reference transcriptome are retained. **C.** Remaining k-mer counts continually normalize before applying **d**ifferential **e**xpression testing. **D.** DE k-mers are extended into contigs based on their overlapping. These contigs are annotated to different biological events.

`min_recurrence_abundance` sets a threshold: any k-mer with more than `min_recurrence` samples below `min_recurrence_abundance` counts is removed from the analysis. When the masking option is used, the user provides reference transcripts to the pipeline and all k-mers found in the reference transcripts are removed from the matrix.

Parameter `diff_method` sets the model used for differential analysis: DESeq2 or t.test. The DESeq2 option implements a testing procedure using a **n**egative **b**inomial model, which is adapted to very long data matrices using a chunk-based strategy. The t-test option implements a testing procedure based on normal distribution of log-transformed counts. The P-values are subsequently corrected using the **B**enjamin-**H**ochberg procedure described in 1.2.2. k-mers with adjusted P-values below a given threshold are retained for further analysis.

The `dekupl-mergeTags` procedure in DE-kupl was designed to merge k-mers into

contigs based on their overlapping sequence. Contigs overlapping by (k-1) to (k-15) nucleotides are repeatedly merged into longer contigs until one of the following conditions is fulfilled : (i) there is no more overlapping contig, or (ii) ambiguity is introduced, *i.e.*, occurring competing extension paths.

To understand how the procedure works, we give here an example. Consider the set of 4 k-mers {ATG, TGA, TGC, CAT} shown in Figure 2.3. In this case, k=3, so `dekupl-mergeTags` starts working with overlaps of size $k-1=2$. k-mer 1 {ATG} overlaps with k-mer 4 {CAT} in AT, and with k-mer 2 {TGA} and k-mer 3{TGC} in TG. Only k-mers overlapping with a single other k-mer are taken into account for merging. Therefore, k-mer 1 and k-mer 4 are merged together to create contig 4+1 CATG, but k-mer 2 and k-mer 3 are not merged with k-mer 1. The vector of counts of the contig 4+1 CATG is set to be equal to the vector of counts of the k-mer with the lowest P-value from the differential abundance test performed in the second step of DE-kupl. In Figure 2.3, k-mer 1 has the smallest P-value among k-mer 1 and k-mer 4. The count vector of contig 4+1 is equal to the count vector of k-mer 1. For a given contig, the constitutive k-mer with the smallest P-value is called the representative k-mer of the contig.

After merging differentially abundant k-mers into contigs, DE-kupl assigns them to putative biological events. To do so, DE-kupl defines 11 classes corresponding to 11 potential biological events, including splicing, polyadenylation, **l**ong **i**ntergenic **n**on-**c**oding **RNA** (lincRNA) .... The assignment rules for differentially abundant contigs are described in Table 3 from Audoux et al. (2017). Two additional files are generated by the DE-kupl annotation procedure which are a per locus table and a contig locations file helpful for visualization.

By moving alignment to the final stage of the procedure, DE-kupl ensures to capture the whole variation in the input sequences at the initial stage. Even unmappable repetitive k-mers, low complexity regions that would not be retained by the conventional reference-based approach, are also obtained until the final stage.

(A)

	k-mer	Condition 1			Condition 2			P-value
k-mer 1	ATG	5	7	3	625	467	565	0.001
k-mer 2	TGA	23	43	42	34	65	29	0.030
k-mer 3	TGC	56	12	40	12	10	5	0.035
k-mer 4	CAT	24	31	56	210	100	320	0.025

(B)

	Contig	k-mer	Condition 1			Condition 2			P-value
contig 4+1	CATG	ATG	5	7	3	625	467	565	0.001
k-mer 2	TGA	TGA	23	43	42	34	65	29	0.030
k-mer 3	TGC	TGC	56	12	40	12	10	5	0.035

Figure 2.3: The `dekupl-mergeTags` procedure for a set of 4 DE k-mers **A.** k-mer count table and corresponding P-values from the differential abundance test performed in `DE-kupl`. **B.** There are three overlapping pairs: (k-mer 1, k-mer 2), (k-mer 1, k-mer 3) and (k-mer 4, k-mer 1). However, the two first pairs are ambiguities: we don't know if we need to merge k-mers 1 and 2, or k-mers 1 and 3. Only one contig 4+1 is created by merging k-mer 4 and k-mer 1. The resulting contigs are k-mer 2, k-mer 3 and the merged contig from k-mer 4 and k-mer 1. The count of contig 4+1 is represented by k-mer with the lowest P-value: here, the count of k-mer 1.

2.3.2 `GECKO`: a genetic algorithm to classify samples using k-mers

Another program that adopts a reference-free approach to classify and explore RNA-seq data is `GECKO` (Thomas et al., 2019). `GECKO` uses k-mer optimization with an adaptive genetic algorithm for the classification of various biological conditions, based on any type of sequencing data, such as microRNA, **m**essenger **RNA** (mRNA) .... Thomas and colleagues designed `GECKO` with two major stages:

1. **Data preparation**: First, a k-mer count matrix is created by applying Jellyfish from input FASTQ or BAM files. GECKO then removes k-mers below a noise threshold, k-mers that are consistent across all samples and k-mers that are redundant.

2. **Adaptive Genetic Algorithm**: this steps discovers groups of k-mers that can correctly classify input samples via wrapper-type feature selection **G**enetic **A**lgorithm (GA). From the filtered k-mer matrix, GA creates a population with thousands of randomly selected k-mers (called individuals). Then, individuals within this population replace one of their k-mers with another k-mer through the mutation stage. This is followed by a crossing-over phase where a portion of the k-mers in the individuals is exchanged, and selectively, those that have not classified the input sample well enough are eliminated from the population and altered.

2.3.3 MINTIE: reference-free approach combining *de novo* assembly and differential analysis

Cmero et al. (2020) recently designed MINTIE as an integrated RNA-seq pipeline that combines *de novo* assembly with **d**ifferential **e**xpression to identify unique variations in a case sample. Contrary to the previous software packages, this pipeline runs in "a single case versus N controls" concept. The novelty of MINTIE (common to DE-kupl and GECKO) is that this approach discovers novel sequence without mapping to a reference genome. Instead, it uses DE analysis on equivalence classes that are unique to the assembly, instead of doing the test on k-mers.

To do that, firstly, *de novo* assembly is performed on the case sample. All assembled transcripts in this case and a set of controls are quantified based on a standard reference transcripts (CHESS v2.2 (O'Leary et al., 2016) by default). As DE-kupl, MINTIE solely retains the sequences absent in reference transcripts. These novel sequences are then compared by performing **d**ifferential **e**xpression testing between

1 case and N controls. Finally, significant over-expressed transcripts in the case samples are annotated based on their alignment to the human genome.

2.4 Conclusion

In this chapter we have presented two major strategies for RNA-seq data analysis: referenced-based and reference-free. There are three major steps in reference-based RNA-seq data analysis which are quality control, read alignment as well as gene and transcript levels quantification. We have also introduced some software tools related to each step.

The reference free software presented above differ by their aims: DE-kupl and MINTIE identify novel transcript forms, *e.g.*, splice variants, fusion transcripts, ... that are specific to or overrepresented in a set of samples, while GECKO creates predictive classifiers composed of non-reference transcript fragments. Results presented in the corresponding publications suggest that the non-referential approach is feasible both for the discovery of transcripts and for predictive modelling. Besides, these results would not be limited by the alignment of mapped reads to the reference genome or transcriptome.

However, there are still several limitations in current approaches, for instance, DE-kupl was initially proposed to discover un-refererenced biological events in RNA-seq, and was not design to perform prediction of samples status, while the set of k-mers discovered by GECKO was validated without an independent dataset.

As a result, my thesis aim was to employ reference-free RNA-seq analysis with **m**achine **l**earning-based approach for the discovery of **p**rostate **ca**ncer (PCa) signatures which were validated in independent datasets. Towards that goal, I will first explore in the next chapter clinical information on PCa and the different predictive signatures that were found in previously published studies.

Chapter 3

Transcriptomics of Prostate Cancer

3.1 General introduction to Prostate Cancer

In France, **p**rostate **ca**ncer (PCa) is one of the most prevalent malignancies amongst men, accounting for 25% of the diagnosed cancer and 8.5% of cancer-related deaths in males in 2018 as shown in Figure 3.1. However, PCa is indolent or grows slowly in a large proportion of men, such that it may not become clinically significant during the patient's lifetime. Nearly all **p**rostate **ca**ncers are adenocarcinoma that starts in glandular cells.

For PCa, the most popular screening method is the **p**rostate-**s**pecific **a**ntigen (PSA) blood test. This is a useful test, especially in men with many risk factors through early detection of cancer, giving patients a better chance of getting treatment or surgery sooner before cancer develops and spreads. Nevertheless, the PSA test has been somewhat controversial due to over-diagnosis and over-treatment in men with no symptoms of the disease (Schröder et al., 2009; Andriole et al., 2009).

Once a biopsy confirms that a man has PCa, he will have to undergo the process of determining the stage of the tumor and its grading: staging figures out where the tumor is located, whether it has spread, and how far if so, while grading measures

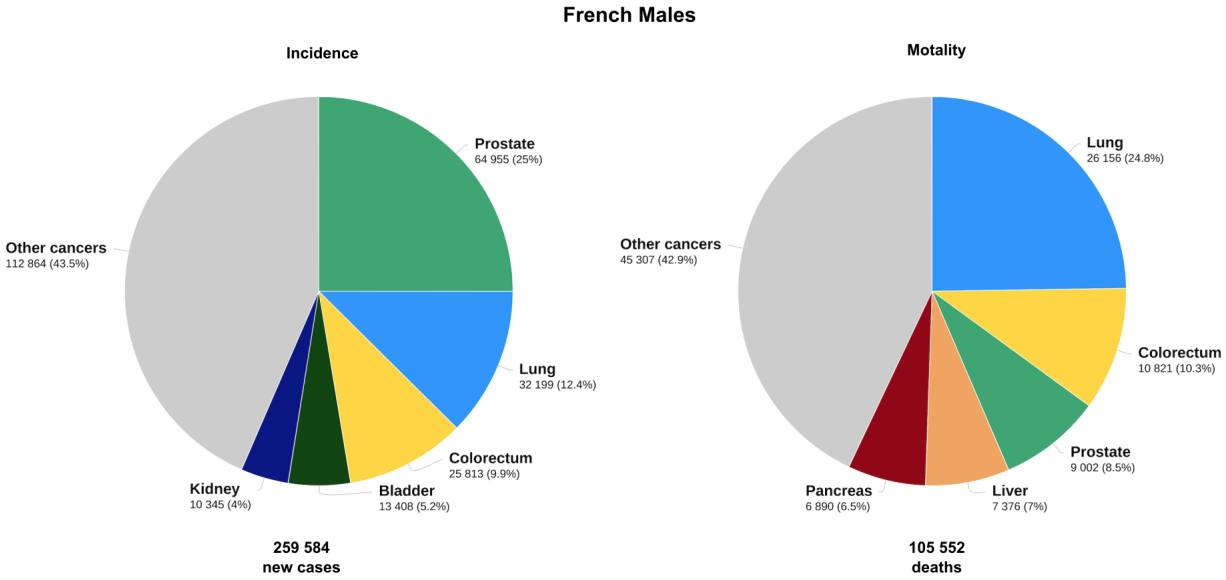


Figure 3.1: The distribution of new cases and deaths for the top 5 most popular cancers in 2018 for french males. Source: `https://gco.iarc.fr`, GLOBOCAN 2018, (Bray et al., 2018; Ferlay et al., 2019).

how quickly the cancer will grow and spread. These are vital and essential processes to assist doctors in making decisions about the right treatment for each case and disease severity.

Stages of prostate cancer. A standard staging system is the **T**umour, **N**ode, **M**etastasis (TNM) was designed by the American Joint Committee on Cancer. The TNM system relies on 3 main factors:

- Tumor (T): the extent of the primary tumor.
- Node (N): whether the tumor spread to adjacent lymph nodes.
- Metastasis (M): whether the cancer metastasized to other parts of the body.

In brief, there are 4 primary stages in PCa from I(1) to IV(4) (some of them are divided further, *e.g.*, A, B ...), of which the lower the stage, the less the tumor has spread.

Gleason score for grading prostate cancer. The Gleason scoring system is the most common PCa grading system. The pathologist looks at tumor cells from a biopsy under a microscope and assigns scores to the two most frequent cell types observed. Cell types scores range from 1 (normal) to 5 (aggressive). The sum of these two scores gives an overall **G**leason **s**core (GS) that ranges between 6 and 10.

Prostate cancer risk groups. In 1998, based on the clinical TNM stage, biopsy **G**leason **s**core, and pretreatment PSA level, D'Amico et al. (1998) proposed a stratification model of PCa patients into low, intermediate, or high risk of **B**io**c**hemical **R**ecurrence following surgery. Risk is calculated by a combination of the above factors:

- Low risk: TNM stage T1c, T2a and PSA level $\leq$ 10 ng/mL and GS $\leq$ 6
- Intermediate risk: TNM stage T2b or GS of 7 or PSA level $>$ 10 and $\leq$ 20 ng/mL
- High risk: TNM stage T2c or PSA level $>$ 20 ng/mL or GS $\geq$ 8

Recurrent prostate cancer. Prostate cancer cells may still survive following the initial treatment, such as surgery, radiation therapy, and/or hormone therapy, an event referred to as recurrent or relapsed cancer. Recurrent PCa can be detected through an increase in PSA levels to a certain threshold at any time after treatment. This is known technically as a **B**io**c**hemical **R**ecurrence (BCR). Alternatively, recurrence can be observed clinically through the observation of metastasis.

3.2 Diagnostic and Prognostic of Prostate Cancer

PCa diagnostic rates have increased over the decades due to an aging population, increasing awareness, and PSA blood test application for screening and diagnosis. However, prostate tumors have varying degrees of aggressiveness, and it is crucial

to distinguish indolent tumors from aggressive ones to avoid unnecessary treatment. Finding prognostic gene signatures would improve cancer diagnosis, especially if it can help identifying cancer at an early stage disease when more treatment options are available.

Since **G**leason **s**core is one of the best predictors of PCa prognosis (Humphrey, 2004), various studies used gene expression data to derive signatures predicting **G**leason **s**core (Bibikova et al., 2007; Penney et al., 2011; Sinnott et al., 2017; Jhun et al., 2017). However, in principle prediction of actual clinical progression, recurrence or metastasis would be more desirable. Several publications have used information on disease progression obtained after several years of followup to provide such predictors. In these models, disease progression was assessed indirectly by monitoring PSA levels (Latil et al., 2003; Long et al., 2014; Ren et al., 2018; Sinha et al., 2019) or through direct clinical observations (Klein et al., 2015; Shahabi et al., 2016).

Some studies used the microarray technology which detects a predefined list of genes (Singh et al., 2002; Erho et al., 2013; Klein et al., 2015) while more recent studies used RNA-seq (Jhun et al., 2017; Stelloo et al., 2018; Sinha et al., 2019), but applied a reference-based pipeline in which reads are aligned to the genome and assigned to annotated genes before gene expression quantification.

3.3 Supervised learning on reference-based approach for PCa

Supervised learning methods have been developed for various applications in **p**rostate **ca**ncer study, such as the identification of transcriptome signatures for PCa diagnosis or prognosis (Singh et al., 2002; Shahabi et al., 2016; Jhun et al., 2017). Several researchers have employed different supervised learning models combining with feature selection on transcriptomic data to infer such RNAs signature. For instance, in the article of Singh et al. (2002), they have used gene ranking based on the signal to noise statistic combined with k-nearest neighbors algorithm to detect gene

signatures of **G**leason **s**core. Another article revealed mRNA signature to predict the lethality among men with GS of 7 (Sinnott et al., 2017). In this study, from the panel of 157-gene signature that was developed in their previous published (Penney et al., 2011), Sinnott and his colleagues have adopted a nearest shrunken centroids PAM classifier for building a model prediction. Bibikova et al. (2007), Long et al. (2014) and Klein et al. (2014) have applied survival models, *e.g.* Kaplan-Meier analysis and CoxPH regression model for inferring signatures that correlated with PCa recurrence. In addition, other well-established supervised learning methods, the LASSO and Elastic net regression models have been used regularly in the recent studies about PCa signatures (Shahabi et al., 2016; Jhun et al., 2017).

3.4 Conclusion

In this chapter, we have introduced **p**rostate **ca**ncer with the PSA screening, several clinical information about staging, grading . . . in this cancer as well as various publications using ML-based approaches for the identification of PCa signatures. However, none of the published signature mentioned above has explored the possibility of finding new genes or transcript isoforms associated to risk or relapse. However, a new generation of predictors using reference-free transcriptomic approaches, as described in Section 2.3, could potentially identify new transcript present in a sample, *e.g.*, novel splice variants, lncRNAs or RNAs from repeated retroelements (Audoux et al., 2017). My thesis aims to apply this type of predictors for the identification of PCa signatures and gain some contributions which will be presented in the next chapter.

Chapter 4

Challenges and contributions

4.1 Adapting tools to the dimensionality of datasets generated by gene-free approaches

k-mer analysis creates a very high number of features to consider, typically tens to hundreds of millions k-mers per RNA-seq library. Many of those may result from errors and/or technological artifacts (such as adapter contamination) (Section 2.1.1), or may be highly correlated in their expression, leading to poorly informative or redundant features. Very large numbers of redundant and irrelevant features can result in over-fitting, low accuracy, and long training times. As a result, matrix reduction is an essential step before applying ML techniques for downstream analysis, such as DE analysis, survival analysis or transcriptome classification. Part of my PhD thesis was devoted to the development of different strategies for dimension reduction based on k-mer counts.

In Chapter 5, I have studied a range of filtering and clustering strategies based on k-mer counts and tested them with real datasets. Among filtering approaches, the supervised signal-to-noise method produced the fastest and most effective method in reducing low-expressed or irrelevant k-mers prior to differential analysis. However,

this filtering approach is not an independent filter and cannot be used safely prior to differential expression. Among filters that did not use sample label information, normalized entropy proved to be the most efficient. Details of the filtering criteria of each method, as well as their effects, are elaborated in Section 5.2.1.

In Section 5.3, I have explored the potential use of unsupervised clustering techniques to cluster k-mers based on the similarity of their counts. The results of this section have led to propose another approach to reduce the size of the k-mer matrix: merging k-mers into contigs based on the similarity of their counts and on the overlap of the k-mer sequences. This reduction process allows to reduce the k-mer count matrix to a smaller contig count matrix with less correlated and redundant features than in the initial matrix. This procedure, which is briefly summarized in the discussion of Chapter 5, has been developed by Haoliang Xue in our lab. It was applied to real data in Chapter 7.

4.2 Combining k-mer based reference-free approach and predictive models

A major methodological aim of this thesis was to advance reference-free k-mer methods one step further by applying k-mer decomposition to predictive models with results assessed in independent datasets.

In Chapter 6, I have used results produced by `DE-kupl` in a **p**rostate **ca**ncer (PCa) dataset provided by collaborators to perform prediction of sample status. Since `DE-kupl` was developed as a statistical pipeline, which captures all k-mers showing significantly different abundances among conditions; it was not designed for predictive modeling. I have applied a procedure to compute and test a predictive signature from the k-mer contigs produced by `DE-kupl`, and to evaluate this signature in independent datasets. In an independent validation cohort, this model reached an AUC score of more than 90% for PCa diagnosis. Chapter 6 is a first demonstration of the feasibility of combining a k-mer based reference-free approach

and predictive models.

In Chapter 7, my goal was to compare k-mer based classifiers to conventional gene-based classifiers for risk and relapse prediction of PCa. Using a large public domain RNA-seq dataset, the number of original features present in the k-mer expression matrix vastly exceeded that in the gene-based matrix, *i.e.* about 94 million compared to 60 thousand. To compare gene-based and gene-free model performances, I needed a common pipeline. To this aim, my goal was to first reduce drastically the k-mer matrix using tools such as presented in Chapter 5. However, as these tools were either designed for filtering prior to differential analysis or otherwise not satisfying, we opted to use the above mentioned reduction process based on k-mer sequence overlap, combined with a drastic screening procedure. The screening step was designed to single out important features and reduce feature space from a ultrahigh dimension to a lower dimension. This screening was applied both on the gene count matrix and on the contig count matrix, as both problems suffer from the curse of dimensionality. The resulting reduced feature matrices (reduced contig matrix and reduced gene expression matrix) are subsequently submitted to the same feature selection process (lasso logistic regression in my case). This allowed us to compare gene-based and gene-free signatures on a fair basis. The rationale behind my approach is detailed in the discussion of Chapter 6. The comparison between the k-mer based and gene-based approaches is detailed in Chapter 7. Classifiers built from either strategy had similarly high performances for risk prediction and a noticeably lower performance for recurrence prediction.

4.3 Demonstrating the ability of gene-free approaches to discover unreferenced RNA subsequences

In Chapter 2 (Section 2.3), we have shown that k-mer analysis can, in principle, capture the full transcript variation present in a RNA-seq sample. This variation can be afterward assigned to biological events such as lncRNAs, splice and

polyadenylation variants, introns, repeats (Audoux et al., 2017), which are ignored by standard protocols based on reference gene annotations. Evidence suggests that non-reference RNA is regularly present in diseased tissues and can form clinically useful biomarkers (Morillon and Gautheret, 2019).

The clinical signature we found in Chapter 6 consisted of only nine unreferenced lncRNAs and was more effective than the commercial **p**rostate **ca**ncer biomarker PCA3 in detecting high-risk tumors. Meanwhile, the reference-free signatures in Chapter 7 contained a set of RNA sequences containing unannotated RNAs and novel variant of annotated RNAs that were not part of gene-based signatures.

4.4 Measuring reference-free signatures across independent RNA-seq datasets

When inferring a prognostic classifier for PCa, contig signatures are derived from an initial discovery set. How can we evaluate the robustness and generalization of this signature? To do that, one needs to transfer contig information to a different clinical cohort and obtain a comparable quantitative expression measure. However, this task poses a real challenge as this requires a mechanism that allows an exact matching of each nucleotide to ensure contigs from the signature are correctly identified in the new dataset.

I have proposed two solutions for the measurement of signature contigs in the new dataset, introduced in Chapter 6, Section 6.3 and Chapter 7, Section 7.2.8. These two solutions were designed to suit different study contexts. In Chapter 6, the set of candidates for contig signatures was a given panel extracted by expert knowledge. As a result, contigs in the signature derived from this set were highly expressed, and the task of finding them in other datasets was relatively easy. Conversely, the contig signatures inferred in Chapter 7 were automatically identified from approximately 94 million k-mers generated from more than 400 libraries (risk prediction model) and had highly variable expression levels. This required a careful matching procedure to

ensure as many contigs as possible could be quantified in the independent dataset.

Part II

Results

Chapter 5

Methods for dimension reduction in k-mer analysis

5.1 Introduction

As described in Chapter 4 of this thesis (Section 4.1), the selection of a subset of informative k-mers, which can be assimilated to a feature reduction, is an essential task. Each k-mer is represented by its counts across samples and its sequence. In this chapter, we investigate methods to reduce the dimension based on the counts of each k-mers.

We explore two ideas in order to reduce the number of k-mers based on their counts: filtering k-mers and clustering k-mers. The two strategies are presented respectively in Sections 5.2 and 5.3 of this chapter.

5.2 Filtering k-mers based on their counts

In this section, we describe and evaluate the performances of a set of filtering techniques in the context of **d**ifferential **e**xpression analysis. The goal of the filtering

techniques is to remove low expressed or irrelevant features prior to differential analysis. Bourgon et al. (2010) have proposed independent filtering to increase detection power of differential expression analysis. In a nutshell, a filtering strategy is independent if the filter statistic and the statistic used to perform the tests are independent under the null hypothesis, which implies that the filter statistics should not take into account the labels of the samples. We refer to Bourgon et al. (2010) for a more comprehensive overview of filtering strategies prior DE analysis.

5.2.1 Filtering strategies

All below filtering methods are uni-variate filters that evaluate each k-mer individually, and in particular do not take into account their interactions. Those methods calculate a score for all k-mers, and either select the top m k-mers with the highest scores, or all k-mers whose score exceeds a given threshold τ, for $m \in \mathbb{N}$ and $\tau \in \mathbb{R}$ some pre-defined values.

In the following, we consider a dataset with n samples and p k-mers $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(p)}$. For a given k-mer k, the vector of occurrence is $\mathbf{x}^{(k)} = (x_1^{(k)}, \ldots, x_n^{(k)})$. The vector class $\mathbf{y} = (y_1, y_2, \ldots, y_n)$ indicates the labels (conditions) for the n samples.

Here is an example, say, we have a dataset including 14 samples and 2 k-mers. Our samples divide into two groups, *e.g.*, normal patients and cancer patients. The expression of each k-mer in this illustration as Table 5.1 below:

Table 5.1: An illustration of using notations in filtering strategies

		Samples													
		Normal patients (is-labeled 0)								**Cancer patients (is-labeled 1)**					
		y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}	y_{12}	y_{13}	y_{14}
Kmers	$\mathbf{x}^{(1)}$	30	0	0	0	8	0	0	0	93	311	161	168	228	669
	$\mathbf{x}^{(2)}$	14	12	14	12	14	14	12	12	12	12	12	12	14	12

As a result:

- $n = 14$; $p = 2$
- $\mathbf{x}^{(1)}$ = (30, 0, 0, 0, 8, 0, 0, 0, 93, 311, 161, 168, 228, 669)
- $\mathbf{x}^{(2)}$ = (14, 12, 14, 12, 14, 14, 12, 12, 12, 12, 12, 12, 14, 12)
- $\mathbf{y}$ = (0, 0, 0, 0, 0, 0, 0, 0, 1, 1, 1, 1, 1, 1)

We present four strategies based of diverse metrics: variance, entropy, **M**edian **A**bsolute **D**eviation (MAD) and signal-to-noise ratio. The first three strategies are independent filters since the score does not depend on the labels of the samples. While the signal-to-noise strategies incorporates this information. It is permissible to use the labels in order to filter k-mers in my case study since we only used the signal-to-noise ratio as a base-line for comparison with other unsupervised filters.

In this section, we focus on filtering strategies prior to differential expression to increase the power of the analysis. However, the filtering strategies presented here can also be applied prior supervised learning. In practice, special consideration should be given to applying supervised filters to construct the input data for a classification model. In such a project, in order to correctly evaluate the classifier, the entire process containing the supervised filter and model training must be evaluated in an independent dataset, *i.e.* test set must be independent with supervised filtering. Otherwise, the prediction performance of the classifier will be considered as overly optimistic (Ambroise and McLachlan, 2002).

Variance

The variance filter uses the variance of each k-mer across samples as its score. Its rationale is that low variance k-mers, whose counts are very similar across samples, would not be helpful to discriminate the different types of outputs, and should therefore be removed. Formally, the variance score of a k-mer $\mathbf{x}^{(k)}$ is defined as:

$$J_{variance}(\mathbf{x}^{(k)}) = \frac{1}{n}\sum_{i=1}^{n}\left(x_i^{(k)} - \mu^{(k)}\right)^2 \tag{5.1}$$

where $\mu^{(k)}$ is the sample mean of k-mer $\mathbf{x}^{(k)}$, defined as $\mu^{(k)} = \frac{1}{n}\sum_{i=1}^{n} x_i^{(k)}$.

Entropy

The idea of the entropy filter is to measure the uncertainty of a random k-mer according to a series of samples. To that purpose, one computes the following metrics:

$$J_{entropy}(\mathbf{x}^{(k)}) = \frac{-\sum_{i=1}^{n} f_i^{(k)} \times \log_2 f_i^{(k)}}{\log_2 n} \tag{5.2}$$

where $f_i^{(k)}$ is the frequency of k-mer $\mathbf{x}^{(k)}$ within the i-th sample, such that $f_i^{(k)} = \frac{x_i^{(k)}}{\sum_{i'=1}^{n} x_{i'}^{(k)}}$.

Note that the entropy is maximized by the poorly informative uniform distribution. Indeed, when all possible counts of $x_i^{(k)}$ occur with the same probability, its entropy is maximized, and the k-mer should be removed from of the set of k-mers to submit to differential analysis.

Median Absolute Deviation (MAD)

The MAD filter ranks all k-mers according to their median distance of their count to the median count across samples. The underlying idea of this score is to keep k-mers having higher variability in a dataset. The difference between variance and MAD methods resides in the way of ranking a k-mer. The former relies on the mean value of that k-mer across all samples while the latter takes into account the median. The median is less sensitive to outliers than the mean. Therefore, the MAD filter removes k-mer with outliers that would have been kept using the Variance filter.

$$J_{mad}(\mathbf{x}^{(k)}) = median_i\left(\left|x_i^{(k)} - median(\mathbf{x}^{(k)})\right|\right) \tag{5.3}$$

Signal-to-noise ratio

This filtering method measures the correlation between each k-mer and the class distinction (*e.g.*, normal patients vs cancer patients) and materializes a "signal-to-noise" ratio. This method was introduced by Golub et al. (1999).

$$J_{signal.noise}(\mathbf{x}^{(k)}) = \frac{\mu_1^{(k)} - \mu_2^{(k)}}{\sigma_1^{(k)} + \sigma_2^{(k)}} \tag{5.4}$$

where:

- $\mu_1^{(k)}$, $\mu_2^{(k)}$ denote the means of the log expression levels of k-mer $\mathbf{x}^{(k)}$, for samples in condition 1 and condition 2 respectively;
- $\sigma_1^{(k)}$, $\sigma_2^{(k)}$ denote the standard deviation of the log expression levels of k-mer $\mathbf{x}^{(k)}$, for samples in condition 1 and condition 2 respectively.

Large values of $|J_{signal.noise}(\mathbf{x}^{(k)})|$ indicate a strong correlation between the k-mer $\mathbf{x}^{(k)}$ and the condition distinction. Its sign corresponds to which condition $\mathbf{x}^{(k)}$ has highest expression, *e.g.*, the positive sign indicates this k-mer has highest expression in the first condition.

5.2.2 Metrics to evaluate filtering performance

To evaluate the performance of the filtering strategies on a real dataset, we run `DE-kupl` using the `DEseq2` option for `diff_method` (Love et al., 2014) on the dataset. The list of differentially expressed k-mers obtained is subsequently regarded as our **ground-truth**, *i.e.* the list of k-mers we would like to retain in the filtering approach.

Our main rationale for using the outcome of `DE-kupl` with `DEseq2` as the **ground-truth** is that `DESeq2` is one of the most used and powerful method to identify differentially expressed genes on RNA-seq data. In the case of k-mer analysis, the set of differentially abundant k-mers obtaining from `DE-kupl` with option `DEseq2` are

the set of k-mers adjusted P-values under 0.05 under **B**enjamin-**H**ochberg multiple testing correction. These tests are performed on a large list of redundant and correlated k-mers (several millions). As a result, this list of differentially abundant k-mers contains false discoveries; this leads to the following goals for filtering: first, to considerably reduce the number of k-mers to be further analyzed; and, second, to retain in this reduced set only those that are significant to a downstream DE analysis and improve power detection. Any independent filter can be safely run prior to `DE-kupl`, greatly speeding up its performances while retaining its significant differentially abundant k-mers.

For each filtering strategies, we consider as predicted the top-ranking m ($m = 100, 1000, \ldots$) k-mers. For any given threshold m, we create a confusion table that compares the top-ranking k-mers, found using a given filtering strategy, and the k-mers detected as differentially-expressed by its obtained by `DE-kupl` with option `DESeq2`, detailed in Table 5.2.

These table entries have the following meaning in the context of our study. **TP** is the number of common k-mers between the k-mers detected as DE by `DE-kupl` with option `DESeq2` and the k-mers retained by the filtering strategy. While **FP** is the number of k-mers detected as non-DE by `DE-kupl` but retained by the filtering strategy. The number of k-mers detected as DE by `DE-kupl` but filtered by the filtering strategy was defined as **FN**. Finally, **TN** indicates to number of k-mers detected as non-DE by `DE-kupl` and also filtered by the filtering strategy.

Table 5.2: Confusion matrix created by `DE-kupl` with option `DESeq2` and filtering strategy.

		`DE-kupl` **with option** `DESeq2`	
		DE k-mers	**non-DE k-mers**
Filtering strategy	**Retained k-mers**	TP	FP
	Filtered k-mers	FN	TN

From the confusion matrix we can define:

$$TPR = \frac{TP}{TP + FN} \tag{5.5}$$

and

$$FPR = \frac{FP}{FP + TN}. \tag{5.6}$$

TPR represents the proportion of DESeq2-predicted k-mers that are retained by a given filtering. Meanwhile, **FPR** is the proportion of k-mers that are predicted as non-informative by DESeq2, but nevertheless retained by the filtering, and are thus detrimental to the performances.

For each strategy, we draw a single graph - ROC curve which is created by plotting TPR against FPR at different top-ranking m k-mers. Then, the best filtering method is the method that has the biggest area under the ROC curve, *i.e.*, has the highest AUC score (as described in "Performance measures for binary supervised learning" in Section 1.3.6).

5.2.3 Experiments and Results

Dataset

To evaluate the performance of the 4 filtering methods, we used a RNA-seq data from a **D**iffuse **I**ntrinsic **P**ontine **G**lioma (DIPG) study. This unpublished dataset was communicated by Marie-Anne Debily (Institut Gustave Roussy). DIPG is a pediatric cancer deriving from the brain stems that control most human abilities, such as talking, walking, and hearing. Several studies (Castel et al., 2015; Nagaraja et al., 2017) have pointed that up to 90% of DIPG patients have mutations in two of the ten genes for histone H3. Histone is a basic protein that associates with DNA to form chromatin. The two affected genes are Hist1H3B and H3F3A, hereafter named H3.1, H3.3, respectively. In this project, we had access to 14 RNA-seq libraries from tumors, linked to their clinical information. The samples were grouped according to the mutation in histone H3. Six samples were mutated in H3.1, and eight in H3.3.

Setup

Methods were compared using AUC scores, as described in detail in Section 5.2.2.

The process to obtain our **ground-truth** was the following. We ran `DE-kupl` with option `DESeq2` on the DIPG dataset. Parameters were `ctg_length` 31, `min_recurrence` 4, `min_recurrence_abundance` 3 , `pvalue_threshold` 0.05, `lib_type` stranded, `diff_method` DESeq2. The recurrence filters kept around 190 million k-mers, and the reference Gencode V24 filter reduced the number of k-mers to 127,444,888. After differential analysis, the number of remaining k-mers was 199,311 k-mers. All experiments were performed in `Python` on a Dell desktop, Intel Xeon(R) processor, CPU E5-1680 v4 @ 3.40GHz $\times$ 16 and 32G memory.

Results

The different filtering strategies were compared using the metrics defined in Section 5.2.2, and the results were presented in Figure 5.1. Signal-to-noise ratio was the method with the highest AUC score, followed by Entropy and MAD filtering methods with AUC of 0.86 and 0.81, respectively. Variance was the method with the lowest AUC score (AUC = 0.75).

We also noticed that the most efficient strategy was the signal-to-noise filtering, which was not surprising given that this method incorporated sample labels. Entropy and MAD methods, although they did not require knowledge of the class labels, still obtained good AUCs. Interestingly, the unsupervised entropy filter appeared to be a highly promising filtering method. Indeed, as shown in the ROC curve of Figure 5.1, using this filter with a well-chosen cutoff was able to accurately filter 45% of the total differential expression k-mers, of which only 3% were associated with false discoveries. Moreover, a more liberal cutoff allows to recover 90% of the `DEseq2` k-mers while keeping the false positive rate at about 20%. As a result, an aggressive filtering could be implemented using this filter.

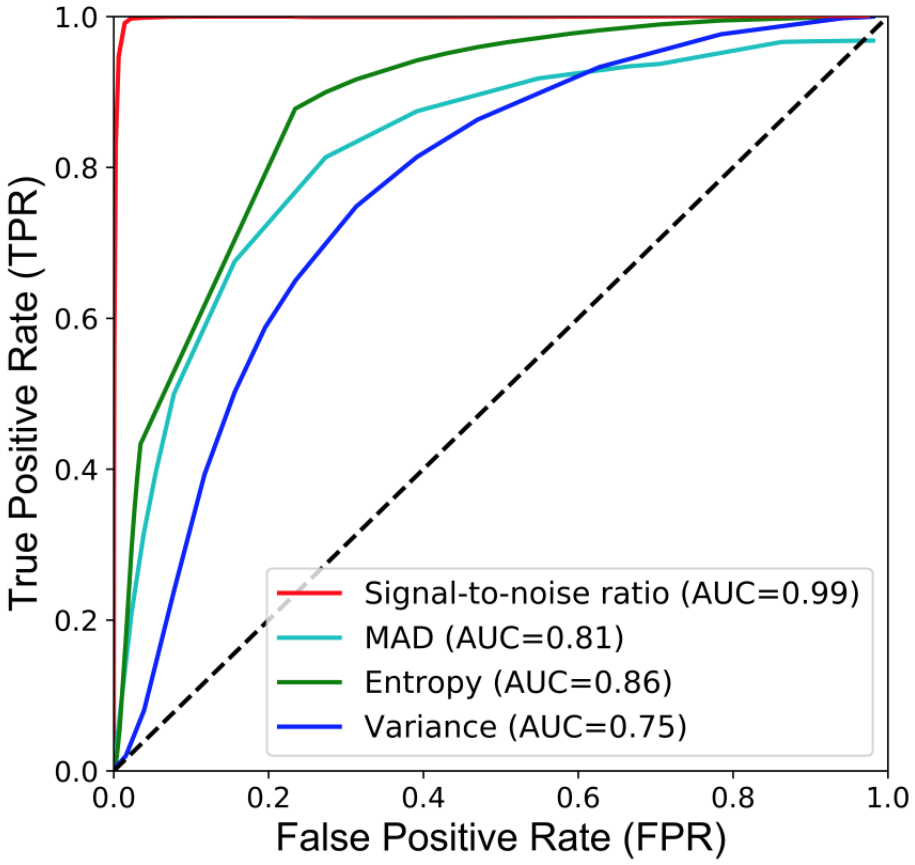


Figure 5.1: Filtering performance of the four filtering methods presented in Section 5.2.1 evaluated as presented in Section 5.2.2 on the DIPG dataset.

Table 5.3: Running time for each filtering method and `DE-kupl` with `DEseq2` option

Method	**CPU time (h:m:s)**
Variance	0:33:35
Signal to noise ratio	0:36:38
Entropy	0:44:16
MAD	0:43:38
`DE-kupl` with `DEseq2` option	**8:17:30**

In terms of running time, variance was the fastest method (as shown in Table 5.5); it took about 33 minutes to complete filtering of 127 million k-mers. Signal-to-noise ratio was three minutes slower, which was still 21-fold faster than our benchmark - `DE-kupl` with `DEseq2` option. Entropy and MAD methods showed similar running

times of around 43 minutes.

5.2.4 Conclusion on count-based filtering

This work demonstrates the ability to use statistical filtering to, first, reduce the number of k-mers considerably to be further analyzed, and, second, retain in this reduced set only those that are relevant to a downstream DE analysis.

Signal-to-noise ratio was the fastest and most effective method out of 4 filtering strategies. However, since the signal-to-noise ratio takes the class information into account, it is not a independent filter as defined in Bourgon et al. (2010) and cannot be used safely prior differential analysis. This filter plays a role of an upper bound. For filtering prior supervised learning, the signal-to-noise ratio is only suitable for filtering features in training process of a supervised classification problem. Meanwhile, entropy is a satisfactory independent and unsupervised filter that could be used as a preliminary step before further analysis. However, I did not propose a solution to chose the right cutoff of the filter: how many k-mers should be filtered? Additional work is required to answer this question.

5.3 Clustering strategies

In this section, we test another strategy to reduce the dimension of the k-mers table using unsupervised learning. The idea is to replace individual k-mers with smaller number of "exemplary" k-mers, each representative of a cluster of similarly-behaving k-mers. Among various clustering algorithms as outlined in Section 1.4, we selected DBSCAN for clustering k-mers based on their counts. We chose this method because in density-based clustering DBSCAN represents one of the most broadly used algorithms (Xu and Tian, 2015) and takes as input attribute data, *i.e.*, each sample is presented by a vector of numerical.

In order to avoid the brute-force computation of pairwise distances, and thus execute DBSCAN in reasonable time, we use methods tailored to detect **A**pproximate **N**earest **N**eighbors (ANN), which we remind in the first Section 5.3.1.

In the next section (Section 5.3.2), we present experiments performed to investigate two questions:

1. **Efficacy of the approximated neighbors algorithms**: In order to speed up the initial computation of approximate neighbors, many methods exist, including ANNOY and LSHF. We ask here which algorithm is most suitable for searching nearest neighbors k-mers.
2. **Evaluation of clustering feasibility based on DBSCAN**: Is the clustering performed by DBSCAN useful?

5.3.1 Strategies to pre-compute distances for DBSCAN clustering

In this section, we consider strategies to precompute approximate neighbors based on a clustering strategy whose aim is to group similar samples into the same cluster. The term sample will refer in the following to a k-mer, and our main task is to group p samples: $\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(p)}$, each sample includes n feature.

As discussed in Section 1.4.1, the time complexity of the DBSCAN algorithm is essentially dominated by the computation of pairwise distances, implemented in the in function `findNeighbors`$(\mathbf{x}^{(i)}, \epsilon)$. In the original DBSCAN, this function returns the list of nearest neighbors of the sample $\mathbf{x}^{(i)}$ within a given distance ϵ, each call having complexity in $O(p \times n)$, where p is the number of samples and n is the number of features representing in each sample. It follows that the execution of the function on all p samples will induce a time complexity in $O(p^2 \times n)$, quickly becoming intractable for larger numbers of samples. Here, instead of exactly computing the list of nearest neighbors, we aim to get the **A**pproximate **N**earest **N**eighbors (ANN), avoiding a costly pairwise distance calculation in case of a large numbers of samples. Towards that goal, we consider two algorithms: **A**pproximate **N**earest **N**eighbors **O**h **Y**eah (ANNOY) (Bernhardsson, 2015) - one of the most robust ANN algorithms based on the benchmark of Aumüller et al. (2017) and the **L**ocal **S**ensitive **H**ashing **F**orest (LSHF) (Bawa et al., 2005) algorithm.

To adapt **A**pproximate **N**earest **N**eighbors to our task, we are given a set S of p samples in a n-dimensional space. The goal is, given a query sample q and a distance ϵ, to return a set L of l samples, $L \subseteq S$, that satisfies distance $\underset{l \in L}{d(l, q)} \leq c\epsilon$, where an approximation ratio $c > 1$.

Approximate neighbors using tree-based ANNOY Random Projections

The idea of ANNOY is to perform spatial partitioning by using random projections and building up a tree for searching l nearest samples from a query sample q.

Random projection is a special strategy to solve dimensionality reduction. By using a random matrix, this technique projects high-dimensional vectors to a lower-dimensional subspace. The main idea in random projection is based on the Johnson-Lindenstrauss lemma, which was first introduced by William B.Johnson and Joram Lin-denstrauss in 1984 (Johnson and Lindenstrauss, 1984). The lemma states that any point in an n-dimensional space can be mapped to a low d-dimensional space ($d \ll n$), while the Euclidean distance between any two points remains roughly the

same.

In our own trials, the random projections were adopted as an intermediate step to finding a list of nearest samples by a given sample in such a smaller space. To do that, a tree-based ANNOY algorithm was designed with two main phases:

Building a tree-like data structure. First, from the set S of p samples, the algorithm randomly selects two samples, and computes the hyper-plane that is equidistant from them. Then, this hyper-plane is used to split this set of samples into two parts. Each part is split into two, and so on until each part has at least t predefined samples. Therefore, this process ends up with a binary tree with each node defines an equidistant hyper-plane, while each leaf is a partition containing a lower number of samples than t. The lower the t-value the higher the tree height, in practice, Bernhardsson (2015) recommends using $t \approx 100$. An interesting aspect is that samples are close to each other in the space have a higher probability to be close to each other in the tree.

In order to improve the precision and performance of searching nearest neighbors, we can build a collection of trees, *a.k.a.* a forest. It is presented by the `n_trees` parameter in the ANNOY algorithm. This parameter affects the build time and the tree size: The higher the value, the more accurate results, but also the more extensive trees.

Searching the data structure. To search the set L of l nearest samples from the query sample q in space, one uses tree traversal from root of the tree. Based on the side of hyper-plane presented at each node, one can chose to go down to the right or left. The search finishes with a leaf containing candidate neighbors and takes only $O(log(p))$. However, going down by one side of the binary tree poses two questions: (1) what if the number of nearest samples results from this side lower than desired l (2) what if several actual nearest samples are outside of this leaf. Therefore, the tree-based ANNOY allows traversal by both sides of the node. Additionally, in case

we have several trees, each tree covers all samples, so when searching down those trees simultaneously, some samples will be present in multiple trees. Samples that are the union of the leaf nodes are the candidate neighbors of the query sample q. The next step is to compute all distances, rank the samples, and return the set L of l nearest neighbors. ANNOY scales well to datasets with up to 1,000 dimensions (Bernhardsson, 2015).

One can specify the number of nodes to observe throughout searching in ANNOY by setting up parameter search_k. This option affects search performance. The higher the value the more accurate results, but also the longer time for searching.

Approximate neighbors using LSHF

Local **S**ensitive **H**ashing **F**orest is an indexing scheme proposed by Bawa et al. (2005). This index is based on the well-known technique of local sensitive hashing (Gionis et al., 1999), improved through: (a) removing the need for various data dependent tuning parameters, and (b) improved accuracy of returned results.

The key idea of basic local sensitive hashing is to hash the samples in the dataset (in our case, the count values of the samples) using several hashing functions. For each function, ensure that the collision probability of two similar samples is higher than that of two non-similar samples. In this index, each sample q is placed into a bucket with label $g(q) = (h_1(q), h_2(q)..., h_m(q))$, where $h_1, h_2, \ldots, h_m$ are selected randomly with replacement from some family of local sensitive hashing functions. As a result, any given sample q is assigned a fixed-length label, including m-digit. Instead of fixing a label for every sample, the lsh-based LSHF introduces a variable length of $g(q)$. The label length of each sample depends on the number of hash functions needed to make sure this label is unique. After that, Bawa et al. (2005) construct a (logical) prefix tree from all these labels (so-called a LSH tree), with each leaf defining one sample. An illustration of a prefix LSH tree as shown in Figure 5.2 below.

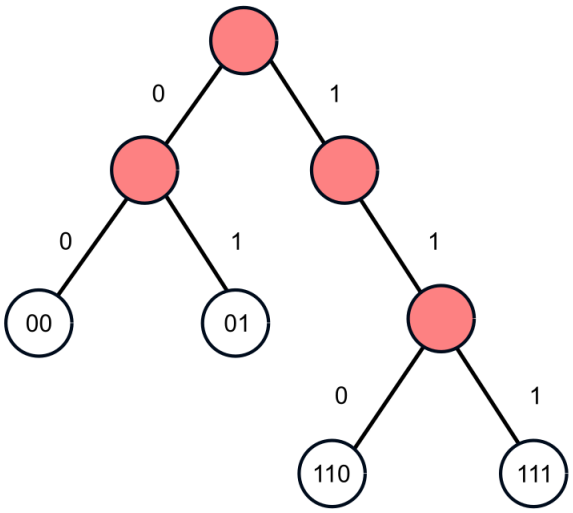


Figure 5.2: A prefix tree created from a set of 4 LSH labels, with each hash function returning one bit as output. The tree leaves represent the 4 samples and their labels. The internal nodes are shown with red circles, some of them have two children but the root's right child has only one child. Source Bawa et al. (2005).

To improve the accuracy of query results, one can construct a set of LSH trees to create a LSH Forest as implied in the name of this algorithm. This option is defined in `n_estimators` parameter in `LSHF`.

To search l **A**pproximate **N**earest **N**eighbors of a given query sample q in a forest including u number of LSH trees, one, first, needs to set parameters `n_neighbors` and `n_estimators` to l and u, respectively. Then, the lsh-based algorithm is performed in two phases.

The top-down phase. For each tree T_i ($i = \{1, \ldots, u\}$) in the forest, one must find the leaf b_i with the maximum prefix that matches the query label by traversing the descending tree.

The bottom-up phase. Let $b = \underset{i=1,\ldots,u}{max}\{b_i\}$ be the deepest level in the set of leaf nodes obtained from the top-down phase. Then, from the deepest level b towards

the root node, samples are collected synchronously across overall prefix trees until reaching the root node, or `n_candidates` number of candidates is collected.

5.3.2 Experiments and Results

Dataset

For this analysis of k-mers clustering, we used the TCGA prostate adenocarcinoma (TCGA-PRAD) RNA-seq dataset as a discovery set (Abeshouse et al., 2015) which totals 558 samples (which we considered as unlabeled).

Accuracy and efficiency of the approximated neighbors algorithms

Benchmark description. We first processed raw sequences (FASTQ files) in TCGA-PRAD with the `jellyfish count` command of the `Jellyfish` software, which produced a Jellyfish index. Next, using the `jellyfish dump` command we created a raw-counts text file, including two columns, each line containing a k-mer and its frequency of occurrence. Finally, k-mers count were merged into a single matrix, and filtered for low-recurrence by running the `dekupl-jointCounts` binary with `min_recurrence` and `min_recurrence_abundance` set to 3 and 10 respectively. The resulting matrix had around 15 millions k-mers. Due to the enormous size of the matrix, we did not attempt to perform clustering directly on this table. Instead, we extracted 1% of the k-mers (150,000 k-mers) to investigate the potential performance of clustering, prior to any test on the complete matrix.

`LSHF` and brute force search methods were both implemented in the `sklearn` `Python` package, while `ANNOY` was built as a `C++` library with `Python` bindings. All experiments were implemented in `Python` on a Dell desktop, Intel Xeon(R) processor, CPU E5-1680 v4 @ 3.40GHz $\times$ 16 and 32G memory.

Using the dataset reduced to the selected 150,000 k-mers from the TCGA-PRAD

dataset, we evaluated the performance of tree-based ANNOY and lsh-based LSHF using the following procedure:

- **Query selection**: We randomly selected 100 k-mers from total k-mers for query selection set, and randomly picked up one k-mer from this set for one query k-mer;
- **Building tree**: The remaining 149,900 k-mers were utilized in the preprocessing step of the chosen indexing method.

These results were computed by running the same procedure 100 times.

Results To answer our first question on the respective impact of the two indexing methods, we compared the performance of ANNOY and LSHF in term of accuracy and speed. A brute force exhaustive search was used to provide a reference, iterating over all possible items and computing the distance between them and our query point.

For example, given a query k-mer K, one attempts to find 10 nearest neighbors to K. First, we run a brute force search to obtain the list of the 10 exact nearest neighbors (K_1, K_2 ... K_{10}). Then, ANNOY and LSHF are executed with the same requirement, producing approximate lists over which we measure:

- **Speedup**, the responding time ratio between LSHF or ANNOY versus brute force method when searching 10 nearest k-mers;
- **Accuracy**, the percentage of exact results (K_1, K_2 ... K_{10}) present in the set of candidates retrieved from the ANNOY or LSHF query.

The results of this experiment are shown in Table 5.4 and Table 5.5. The ANNOY algorithm largely outperformed the LSHF algorithm in accuracy score (96% $vs.$ 63%). In terms of speed, the tree-based method was faster than the lsh-based method. For example, given a sample K, it took 0.137 seconds for finding 20 ANN to K from

150,000 samples using LSHF, while it took only 0.008 seconds with ANNOY, *i.e.* about 17-fold faster.

As a result, ANNOY was chosen in combination with DBSCAN for clustering k-mers.

Table 5.4: Speed and accuracy of LSHF when querying a randomly given k-mer K within a dataset of about 150,000 k-mers (558 dimensions).

Brute force time (seconds)	n_estimator	n_candidates	n_neighbors	Query time (seconds)	Speedup (times)	Accuracy
0.414	10	5,000	12	0.141	3.10	0.63 +/- 0.23
0.407	10	5,000	20	0.137	3.10	0.63 +/-0.2
0.414	10	5,000	50	0.149	2.80	0.59 +/-0.2
0.412	10	5,000	100	0.126	3.40	0.65 +/-0.19

Table 5.5: Speedup and accuracy of ANNOY when querying a random k-mer K within a dataset of about 150,000 k-mers (558 dimensions)

Bruce force time (seconds)	n_trees	Build time (H:M:S)	n_neighbors	Search_k	Query time (seconds)	Speedup (times)	Accuracy
0.414	200	0:02:42	12	35,000	0.008	56.6	0.95 +/-0.08
0.407			20		0.008	56.0	0.95 +/- 0.06
0.414			50		0.008	57.2	0.94 +/- 0.07
0.412			100		0.008	56.0	0.92 +/-0.07
0.416			12	45,000	0.009	47.8	0.96 +/-0.05
0.417			20		0.009	47.7	0.97 +/-0.04
0.417			50		0.009	47.6	0.96 +/-0.05
0.415			100		0.009	47.8	0.95 +/-0.05
0.410	250	0:03:24	12	35,000	0.008	55.6	0.96 +/-0.07
0.412			20		0.008	55.5	0.94 +/- 0.07
0.412			50		0.008	55.6	0.93 +/-0.08
0.414			100		0.008	55.7	0.92 +/-0.08
0.408			12	45,000	0.009	46.9	0.97 +/-0.06
0.412			20		0.009	46.9	0.96+/-0.06
0.412			50		0.010	46.7	0.96+/-0.06
0.408			100		0.009	46.8	0.95 +/-0.06

Evaluation of clustering using DBSCAN with approximate neighbors

As discussed in the previous section, ANNOY beats LSHF in both speed and accuracy. In this section, our aim is to test the feasibility of combining the density-based DBSCAN with ANNOY in clustering k-mers.

Benchmark description for clustering assessment using DBSCAN and ANNOY. We used for this task the TCGA-PRAD dataset generated in the previous experiment (Section 5.3.2), *i.e* with about 150,000 k-mers .

ANNOY was used as in previous experiment, *i.e*, C++ libray with Python binding, while DBSCAN was implemented in Python programming with two pseudo-code algorithms (as shown in Algorithm 1.5 and Algorithm 1.6). Noted that instead of calling the `findNeighbors` function we called the `get_nns_by_item` function which is supported by ANNOY to return $minPts$ closet samples within distance ϵ.

The approximate neighbors ANNOY algorithm was used with parameters `n_trees` = 200, `search_k` = 35,000 and `n_neighbors` = 20 (called selected ANNOY). The density-based clustering DBSCAN, meanwhile, was tuned with different values for both ϵ and $minPts$ parameters. As discussed in Section 1.4.1, selection the right values of these parameters was an expensive process due to DBSCAN algorithm run with a number of experiments and results exploration. To adapt the density-based algorithm DBSCAN to our study, we set the initial value of ϵ to 0.6, and $minPts$ with 3 different values of $\{20, 15, 12\}$.

The experiments were done on the same hardware as above. We ran DBSCAN with parameter ϵ=0.6 and $minPts$=20. As previously, each time we found $minPts$ nearest k-mer within ϵ distance, we invoked the `get_nns_by_item` procedure from selected ANNOY. The experimental results collected included elapsed time, number of clusters and number of k-mers belonging to each cluster. This process was repeated with two different values of $minPts$, ($minPts = 15$ and $minPts = 12$). Results are shown in the Table 5.6.

Description of results. As shown in Table 5.6, the number of clusters and the number of k-mers considered noise showed opposite trends. For example, when DBSCAN parameter ϵ was kept at 0.6, while $minPts$ was decreased from 20 to 12. The number of clusters was increased by about 1,000 clusters (from 23,476 to 24,482 clusters); meanwhile, the number of noise k-mers underwent a significant reduction from 46,106 to 39,359 k-mers. Runtime did not vary significantly at about 49 minutes to complete the grouping of about 150,000 k-mer.

Table 5.6: Result of clustering 150,000 k-mers when combining DBSCAN and selected ANNOY with different values of ϵ and $minPts$

DBSCAN	Selected ANNOY	Clustering time (H:M:S)	# Clusters	# k-mers in "noise" group
ϵ: 0.6 ; $minPts$: 20		0:47:46	23,476	46,106
ϵ: 0.6 ; $minPts$: 15	n_trees: 200 ; $search\_k$: 35,000 ; $n\_neighbors$: 20	*0:50:51*	*24,053*	*42,182*
ϵ: 0.6 ; $minPts$: 12		0:48:07	24,482	39,359
ϵ: 0.4 ; $minPts$: 12		1:04:27	3,006	114,635
ϵ: 0.3 ; $minPts$: 12		1:02:36	307	138,751

In order to evaluate the efficiency of the clustering task, we explored the result of selected ANNOY in combination with DBSCAN with $\epsilon = 0.6$ and $minPts = 20$. This produced 24,053 clusters, including a cluster of "noise" k-mers.

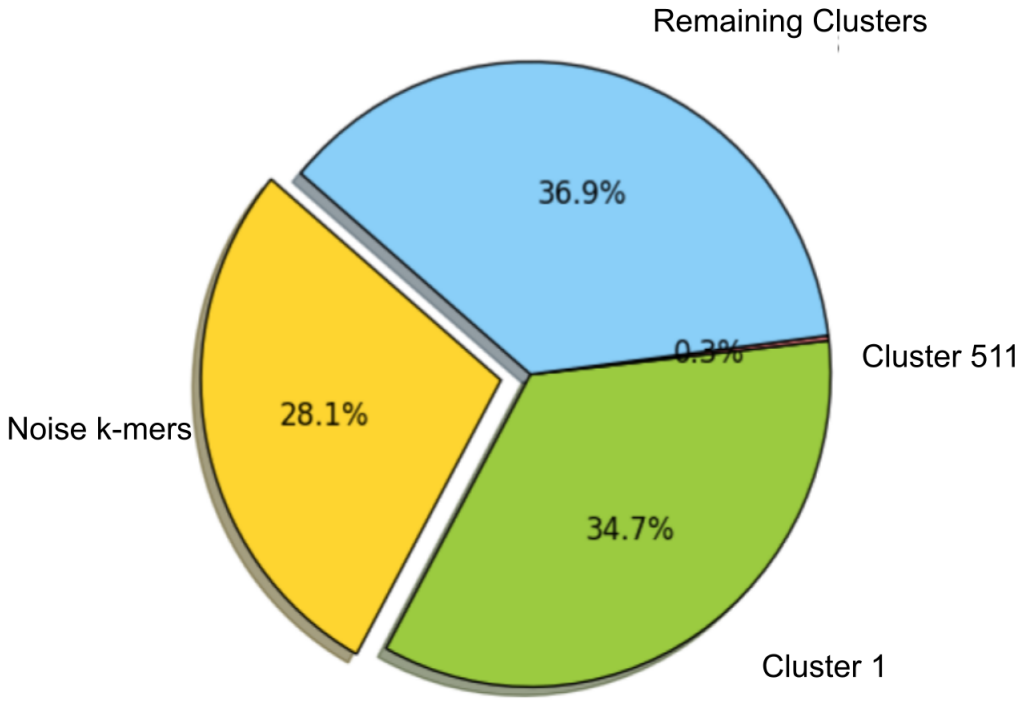


Figure 5.3: Pie chart - Distribution of clusters with more than 300 kmers, obtained by running DBSCAN with $\epsilon = 0.6$, $minPts$ =15 and selected ANNOY in a 150,000 k-mers dataset

The clustering was strongly imbalanced, as shown in a pie chart representation Figure 5.3. Cluster 1 was the biggest cluster (52,075 k-mers), accounting for 34.7% of the total k-mers, and was about 135-fold larger than the second biggest cluster - Cluster 511 (385 k-mers) (Figure 5.3). The number of kmers considered to be noise was approximately 28% of the total k-mers.

Among the three clusters inspected for evaluating the performance of the DBSCAN and ANNOY combination in the clustering task, Cluster 511 (the second largest) was the most satisfactory (Figure 5.6). Indeed, k-mers in Cluster 551 shared similar profiles across all samples, as could be expected from a correct clustering.

The expression of k-mers within the other clusters showed tendencies that we did not anticipate. For instance, we expected k-mers in Cluster 1 to be relatively homogeneous in expression across samples, but an expression heatmap of 100 random k-mers in this cluster (Figure 5.4) revealed substantially heterogeneous abundance

levels.

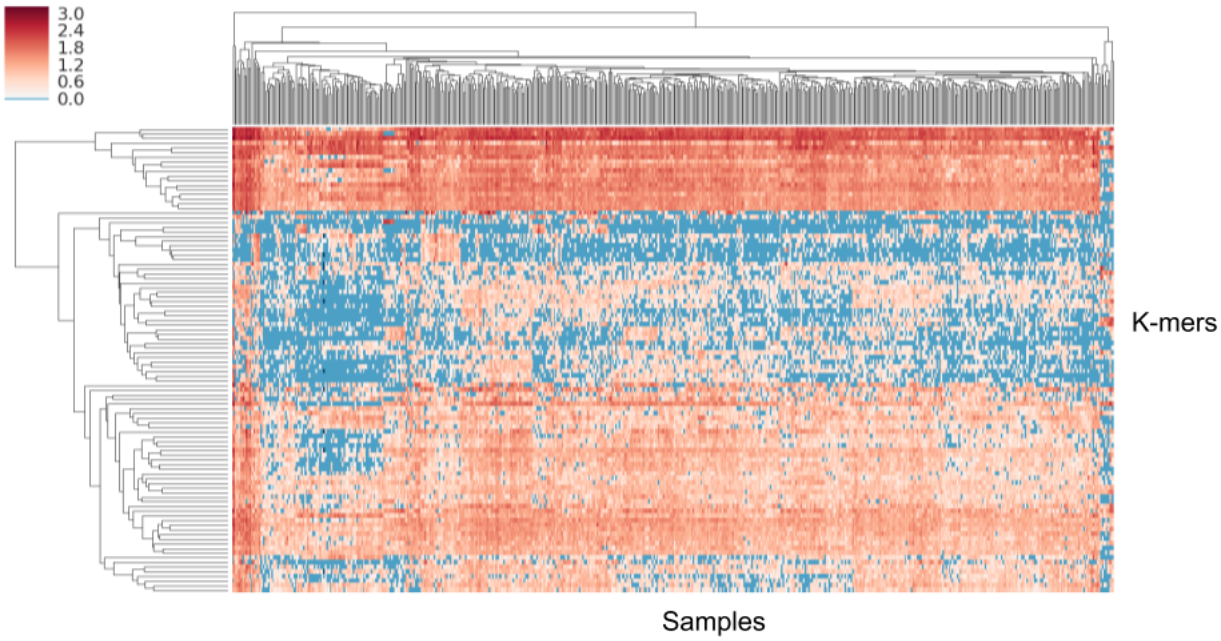


Figure 5.4: Heatmap of log10 counts for 100 randomly selected k-mers among 52,075 k-mers in DBSCAN cluster 1.

Conversely, the expectation that k-mers categorized as noise should be noisy, *i.e.*, appear with random count fluctuations was not really supported by the heatmap representation (Figure 5.5). At first sight, noise k-mers appeared as they could be grouped into a single real cluster due to their similar count profiles.

We hypothesize that this erratic behavior is induced by the agglomerating strategy implemented by the DBSCAN algorithm. Indeed, the capacity of this method to detect clusters of arbitrary geometry also allows it to repeatedly enlarge a cluster in a given direction. As a result, two samples from the same cluster may be substantially different, as long as there exists a sequence of regions of sufficiently high density connecting them. This difficulty unfortunately appears to be shared by most algorithms that are based on local (approximate) neighborhoods, and we could not identify a way to overcome it.

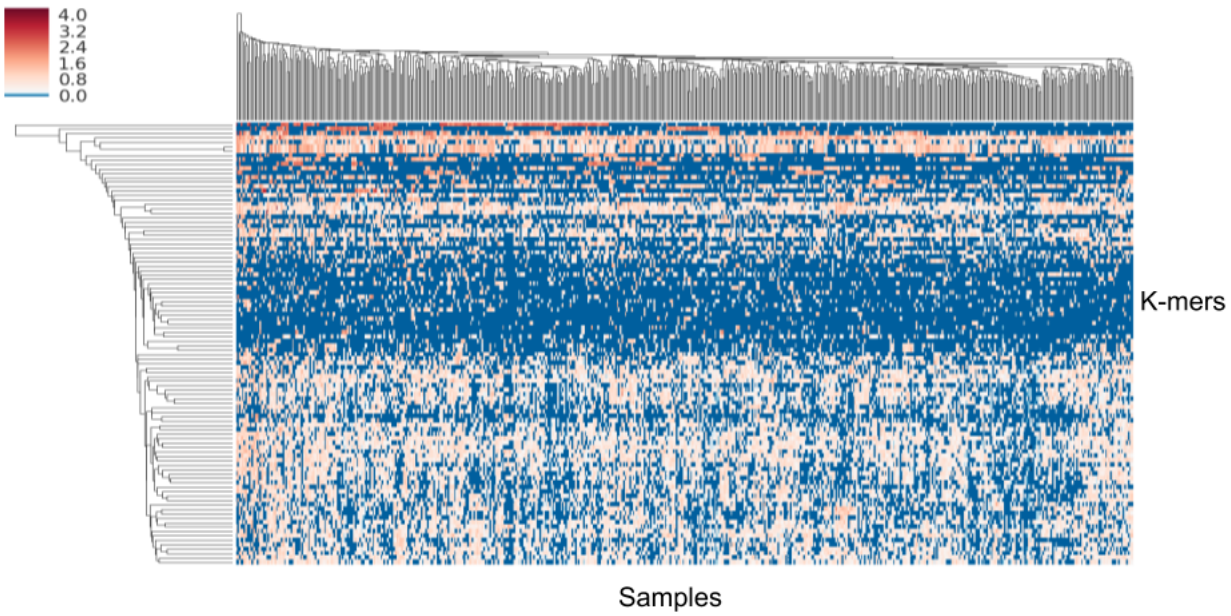


Figure 5.5: Heatmap of log10 counts for 100 randomly selected k-mers among 42,182 k-mers considering as noise in DBSCAN.

5.3.3 Conclusion of the clustering analysis

The lack of homogeneity of k-mer clusters was a severe limitation in the outcome of DBSCAN-based k-mer clustering in combination with ANNOY, which unfortunately, could not be overcome during this thesis.

Indeed, let us remind that our goal was to reduce k-mers using unsupervised learning. In our expectation, k-mers sharing similar expression across all samples would be grouped into the same cluster, then, among them, one k-mer would be selected as a representative of this cluster. This selected k-mer would represent all other k-mers in its cluster. We did obtain a smaller set of selected k-mers, but we were not convinced at all by the quality of the clustering.

In addition, the selection of initial values of the two tuning parameters, ϵ and $minPts$ notably affected the results obtaining by DBSCAN. For instance, setting ϵ to 0.6 and $minPts$ to values ranging from 20 to 12, we observed an increase of

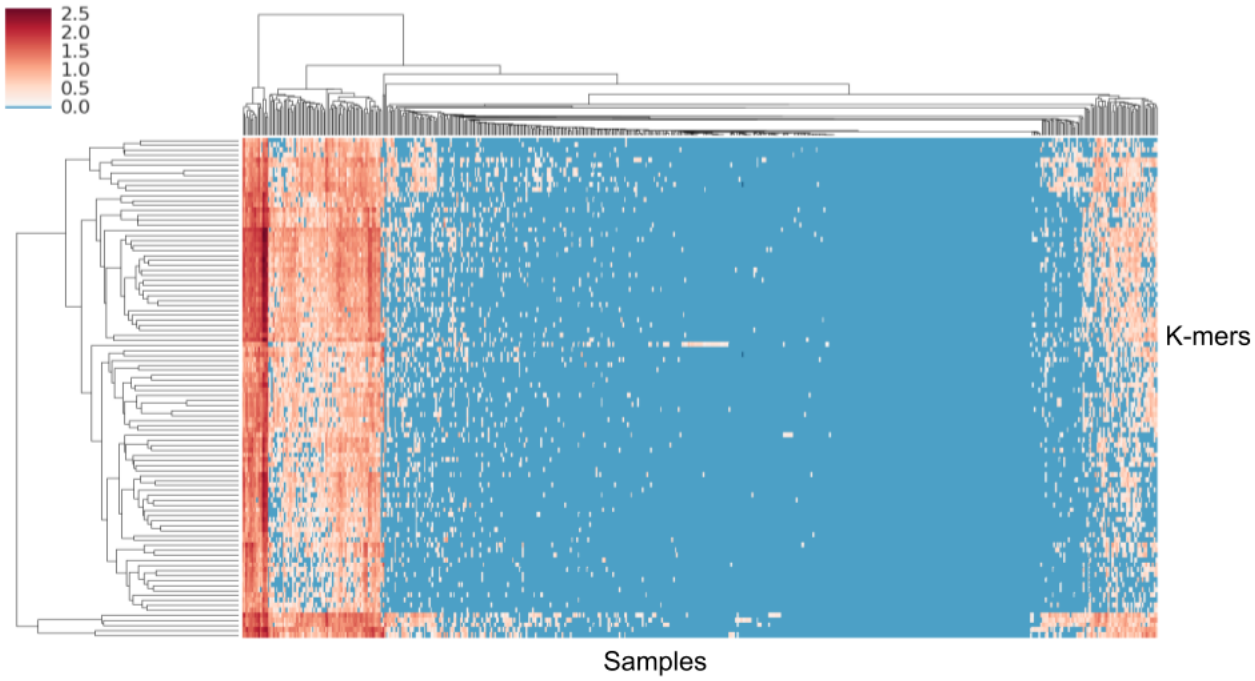


Figure 5.6: Heatmap of log10 counts for 100 randomly selected k-mers among 385 k-mers in DBSCAN cluster 511.

the number of clusters, while the number of noise k-mers was shown to noticeably decrease. Conversely, keeping $minPts$ at 12 and ϵ to values ranging from 0.6 to 0.3, we saw a significant decrease of the number of clusters, and the opposite trend was right for the number of noise k-mers. These two input parameters must be chosen with a consideration between the running time, number of clusters, number of noise k-mers, and also the quality of clustering results. Resulting in DBSCAN had to be run several times, followed by an assessment of the results. This process turned out to be computationally costly, especially in our task with about 150,000 k-mers in a 558-dimensions space. Moreover, we remind that all this analysis was done with only 1% of the actual number of k-mers.

5.4 Discussion

In this chapter, we explored different solutions to reduce the dimension of the k-mers matrices using their counts. The filtering strategies have shown their effectiveness in significantly reducing number of low-expression k-mers prior to **d**ifferential **e**xpression analysis. The idea of k-mer clustering has shown undesirable results, with k-mers in the biggest cluster showing heterogeneity across samples while the noise k-mers should be grouped in the a real cluster. However, our clustering results was based on one density-based clustering algorithm. We did not test other clustering algorithms, *i.e.*, from other density-based methods or from other categories, such as hierarchy, grid-based ... algorithms. These results indicated that `DBSCAN` was not a suitable clustering method for our dataset which was able to have varying densities. The varying densities lead to the inefficient of `DBSCAN`; it probably gathers "wrong" k-mers, *i.e.*, dissimilarity expression when expanding cluster due to `DBSCAN` only uses one global density threshold ϵ (Ertöz et al., 2003).

However, each k-mer is characterized not only by its counts, but also by its sequence. Therefore another idea for k-mer reduction is to merge k-mers based on their overlapping sequences. In the `DE-kupl` pipeline, Audoux et al. (2017) developed an iterative procedure `dekupl-mergeTags` that merges k-mers into contigs (described in Section 2.3.1). This procedure was not entirely satisfying as it did not take into account the expression profiles of k-mers, which could lead to merging unrelated k-mers or contigs. We considered upgrading the procedure by taking into account the compatibility between merged k-mers or contigs. This procedure was implemented by Haoliang Xue as part of his thesis, therefore I will just summarize it here.

The compatibility between two contigs is measured by the **M**ean **A**bsolute **C**ontrast (MAC) value between the counts of the two contigs across all samples.

$$MAC(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}) = mean_{i \in samples} \left(\left| \frac{x_i^{(1)} - x_i^{(2)}}{x_i^{(1)} + x_i^{(2)}} \right| \right) \tag{5.7}$$

where

- $\mathbf{x}^{(1)}$ and $\mathbf{x}^{(2)}$ are count vectors of two contigs to be merged
- $x_i^{(1)}$ and $x_i^{(2)}$ are counts in sample i from the corresponding count vectors.

Contig extension is rejected if $MAC > 0.25$. By this intervention, all contigs are guaranteed to have member k-mers with consistent sample count vectors. Finally, the new contig's sample count vector is set to the mean of composite k-mer's sample count vectors. This upgraded version of contig extension has been used in the study presented in Chapter 7, and has contributed to finding transcriptome signatures for **p**rostate **ca**ncer prognosis.

Chapter 6

Reference-free transcriptome exploration reveals novel RNAs for Prostate cancer diagnosis

Our article, published in 2019, aims to demonstrate the ability to find new RNA biomarkers capable of predicting **p**rostate **ca**ncer using DE-kupl, the reference-free computational pipeline presented in Section 2.2. Below, I present a summary of the article with an emphasis on my own contributions.

The biomarker discovery workflow consists of three main steps:

- Discovery of a set of candidate contigs on the PAIR cohort (*discovery set :* 8 normal and 16 tumor specimens from total stranded RNA-seq) using DE-kupl and manual selection of contigs. This step leads to 23 candidate contigs.

- Selection of a smaller set of contigs among the set of 23 candidate contigs using the NanoString assay of nine normal and 135 tumor specimens (*selection set :* expression of the 23 contigs, 6 housekeeping genes and PCA3, a known PCa-associated lncRNA, measured using the NanoString technology on an extended PAIR cohort with one additional normal specimen and 119

additional specimens for tumor tissues). This step leads to a signature, *i.e.* a set of contigs selected for their ability to predict a sample status.

- Validation of the predictive performance of the signature on the TCGA-PRAD cohort (***validation set:*** a poly(A)-selected and unstranded RNA-seq dataset with 52 samples in normal tissues and 505 samples in tumor tissues). The TCGA-PRAD cohort is independent from the PAIR and extended PAIR cohort used for discovery and selection. The performance of the signature is evaluated on truly unseen data.

My contributions in this article includes 3 main tasks that are implemented in R programming:

- Perform features selection on the selection dataset (NanoString assays on the extended PAIR cohort) using `LASSO` logistic regression to select the best predictive contigs (normal versus tumor) and assess the performance of the predictive signatures on this dataset.
- Evaluate contig expression measurements in TCGA-PRAD dataset and assess the performance of the signature on this dataset.
- Compare the gene-free classifier to a classifier inferred using conventional gene expression.

6.1 Discovery of `DE-kupl` contigs associated to Prostate cancer

In this section, I describe the discovery of contigs associated to PCa using `DE-kupl`. This step has been performed by other co-authors of the paper, before the benchmark of filtering strategies I exposed in Chapter 5, Section 5.2.1. For this reason, the entropy filter has not been used.

Discovery on the PAIR cohort. The DE-kupl was applied on the PAIR cohort (discovery set) to identify tumor-specific transcripts. DE-kupl, described in Section 2.3.1, was executed with parameters ctg_length 31, lib_type stranded, min_recurrence 6, min_recurrence_abundance 5, pvalue_threshold 0.05, diff_method DESeq2. k-mer masking was performed using the Gencode v24 reference transcriptome. The DE-kupl pipeline identified 1,179 tumor up-regulated contigs in the noncoding regions, longer than 200 nucleotides and showing an adjusted P-value below 0.01 from the differential abundance test. The 1,179 tumor up-regulated contigs are subsequently manually selected to retain a list of 23 PCa RNA contigs embedded into putative lncRNAs. The Integrated Genome Viewer (IGV) (Robinson et al., 2011) was used to visualize contigs expression and performed the manual selection using the following criteria. When several contigs are located in the same genomic location (5 kb window), only the contig with the lowest adjusted P-value is retained. Contigs which are contigs antisense to expressed exons, bidirectional or positioned in close vicinity to other transcribed protein-coding genes are also filtered out.

6.2 Selection of predictive DE-kupl contigs

Selection on the NanoString dataset. The NanoString technology was used to measure the expression of the selected 23 DE-kupl contigs, 6 housekeeping genes and PCA3 in the extended PAIR. The expression level of all DE-kupl contigs were lower than PCA3, expect for two contigs that are subsequently removed from the analysis. The NanoString dataset was used to select, among the set of 21 DE-kupl contigs and PCA3, a smaller subset of non-correlated features able to predict the status normal versus tumor. This selection was performed using the LASSO logistic regression, with regularization parameters λ chosen by cross validation, as implemented in the glmnet R package (Friedman et al., 2010). Given that the NanoString is also highly unbalanced, with more than 10 times tumor samples than normal samples, the dataset was upsampled as explained in Section 1.3.6. The selection was performed

on 2 different set of variables : the set of 21 DE-kupl contigs and PCA3 (*mixed signature*), and a restricted set of 15 contigs (selected among the DE-kupl contigs) that were assigned to putative novel lncRNAs (*new-lnc signature*). The performance of the two signatures are measured on the NanoString dataset using the AUC under the ROC obtained with boosted logistic regression, on 100 datasets sampled from the initial upsampled dataset, with 70% observations from training the model and 30% for testing the model. Given that the NanoString measurement was performed on the same samples used in the PAIR cohort (plus additional samples), the dataset obtained using the NanoString technology cannot be regarded as an independent dataset from the PAIR cohort. Therefore, the performance of the selected signature on the NanoString dataset is overly optimistic: a validation on an external dataset, with no overlap with the discovery cohort, is necessary to conclude.

6.3 Measuring DE-kupl contigs in an independent cohort

Contig expression measurements in TCGA-PRAD dataset. The two signatures (*mixed signature*, *new-lnc signature*) are subsequently used to predict tumor status on the independant dataset, the TCGA-PRAD cohort.

Computing the contig expression measurements of the selected contigs in an external cohort is not an easy task because there is no common reference across the two datasets, as in reference-based approaches. DE-kupl contigs were produced by the `dekupl-mergeTags` procedure based on sequence overlaps. The count vector of each differentially abundant contig corresponds to the count vector of its representative k-mer (*i.e.* the constitutive k-mer with the lowest P-value in the differential abundance test performed in DE-kupl, as detailed in Section 2.3.1). To obtain the abundance of the DE-kupl in the TCGA-PRAD cohort, we use the representative k-mer for each contig and measure the abundance of this k-mer in the TCGA-PRAD cohort. First, the TCGA-PRAD FASTQ files were converted to k-mer counts using `jellyfish`

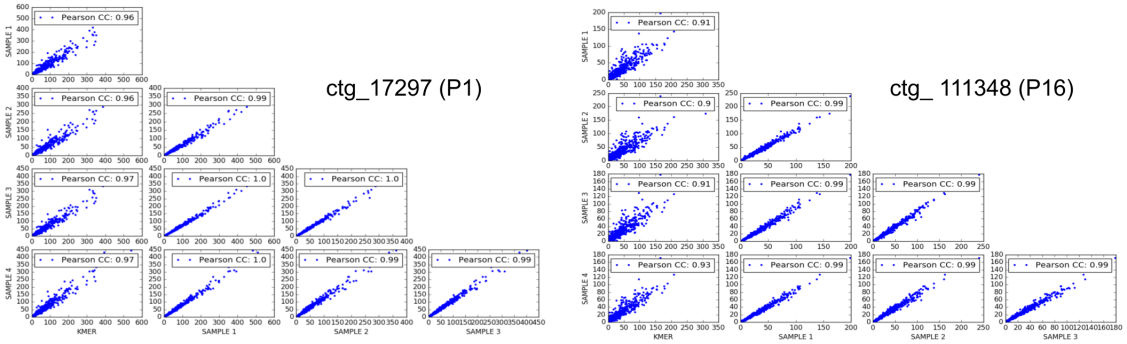
`count` and representative k-mers were counted in each Jellyfish count file using the `jellyfish query` command. Then, we normalized these counts using total number of reads in corresponding libraries.

To determine whether this method was a secure method for assessing contig expression, we compared the vector of counts obtained with the average vector of counts obtained when sampling k-mers along each contig (instead of taking the representative k-mer). The counts for all k-mers were obtained using the `jellyfish dump` files created for each TCGA-PRAD library. We performed the sampling of k-mers along each contig using the procedure described below:

1. Extract the list of all k-mers from the contig.
2. Sample 10 k-mers regularly spaced from this list, *i.e.* beginning with the first 10% and stopping at the last 10% of this list.
3. The step 2 is repeated 4 times. Each list of 10 sampled k-mers (noted SAMPLE 1 to 4) was obtained by shifting the beginning position of the first sampled k-mer.
4. For each SAMPLE, we average the counts of 10 sampled k-mers corresponding to each library to obtain an average count per library (named average sampled k-mer for SAMPLE 1 to 4).

To compare the contig count vector obtained using the representative k-mer and the contig count vector obtained by sampling along the contig, we compute the Pearson correlation between each representative k-mer and the 4 average sampled k-mers. Figure 6.1 shows the relationship between the counts of the representative k-mers and the 4 average sampled k-mers for contigs P1 and P16. We see that the counts are highly correlated, which supports the use of the representative k-mers to measure the abundance of `DE-kupl` contigs across independent datasets.

A



B

		(I) representative k-mer *vs.* 4 samples		(II) sample *vs.* sample	
ID	contig	mean	STD	mean	STD
P1	ctg_17297	0.96	0.005	0.99	0.005
P16	ctg_111348	0.91	0.011	0.99	1.11e-16

Figure 6.1: Evaluation of contig expression measurements in TCGA-PRAD dataset. **A:** Counts of the 4 average sampled k-mer versus the counts of the representative k-mer for contigs P1 (ctg_111348) and P16 (ctg_172917). **B:** Pearson correlations between counts of representative k-mers and the counts of the 4 average sampled k-mers from contigs P1 and P16, respectively. For each contig, we computed the mean and standard deviation of the correlations between (1) the representative k-mer and the 4 average sampled k-mers (2) any pairs of the 4 average sampled k-mers.

6.4 Performance of DE-kupl predictive contigs in an independent cohort

Performance of the signatures on the TCGA-PRAD dataset. To evalute the performance of the set of selected contigs, we retrain the model on the validation dataset, given that the two datasets (NanoString and TCGA-PRAD) are not obtained using the same technonology (NanoString versus poly(A)+ unstranded

RNA-seq). The performances of the two signatures are evaluated using the same method as on the Selection set. Both markedly out-performed PCA3 for tumor detection with AUC of 0.92 for mixed and of 0.91 for new-lncRNA signatures against AUC of 0.73 for PCA3. Notably, the new-lncRNA signature included only unannotated lncRNA sub-sequences that predicted tumor status with performance similar to that of the mixed signature. PCA3 was not retained within the mixed signature set, instead contigs embedded into the well characterized PCAT1 lncRNA and into two already annotated lncRNAs, LOC283117 and LINC01006. Note that we also assessed the predictive performance of the signatures to predict risk prognosis and tumor recurrence status, with success. For more details, we refer to the published article included at the end of the chapter.

6.5 Comparing the gene-free classifier vs conventional gene-based classifier

We also compared predictive performances of signatures retrieved by the gene-free classifier to the one inferred using conventional gene expression counting. First, on the Discovery Set, `DE-kupl` was used to produce a gene count matrix with all the same parameters set as a gene-free classifier. The original gene expression matrix includes around 56,000 genes compares with 24 observations ($p \gg n$). As a result, `DE-kupl` with `DESeq2` option was used to filter genes prior to building a classifier. Only up-regulated genes with adjusted P-value lower than 0.05 and Log2FC higher than 2 are kept. 520 genes are retained. To select a set of predictive genes among the list of 520 genes, feature selection was performed on the gene count matrix of the discovery set using `LASSO` penalized logistic regression combined with stability selection as detailed in Section 1.2. Retained gene signatures were the 5 genes with a probability of being selected above 0.5 on 2000 resamples from the initial dataset. This signature was used to build ROC curves and compute the mean and standard deviation of the AUC on the Validation Set as described above for the `DE-kupl`-derived contigs, with a mean AUC of 0.91. The performance of the gene signature

was similar to the performance of the signature derived using DE-kupl. Among the 5 genes, 4 correspond to noncoding trasncripts. PCA3 was not included in the list of 5 genes.

6.6 Discussion

The DE-kupl pipeline, combined with visualization of expression contigs and manual selection was able to retrieve RNA subsequences as powerful as the signature derived from GENCODE-annotated genes. This demonstrates the ability of reference-free approach to retrieve interesting unreferenced contigs.

However, the two approaches (gene-free and gene-based) are not fairly compared in this work: the gene-free approach involves manual selection of contigs, based on expert knowledge. The selection of contigs in the gene-free approach involves the use of the NanoString dataset, with no equivalent in the gene-based approach. We would like to compare the two approaches using a pipeline as similar as possible. The reason why we want to compare the two approaches (gene-free and conventional RNA-seq) is that the gene-free approaches highly increase the number of features in the dataset (from 50 000 genes in conventional RNA-seq to millions of k-mers or contigs in the gene-free approaches). Therefore, we may suspect that the k-mer approach is more prone to overfitting. To address this question, we propose to compare a gene-free and a gene-based classifier.

Besides, in both approaches (gene-free and gene-based), the preliminary step prior to feature selection and supervised learning was performed using differential analysis (finding differentially expressed k-mers or genes). One could think to pre-select features based on their predictive performance instead of the output of the differential analysis.

For the two reasons mentioned above, the goal of the next chapter is two-fold: to propose a pipeline to perform prediction using k-mers and to compare this pipeline to a conventional RNA-seq pipeline to discover gene signature using RNA-seq data.

As presented in Section 1.3, there are many feature selection and supervised learning techniques. We decided to use the lasso logistic regression combined with LASSO penalty because they have been used in recent papers to discover PCa signatures using conventional RNA-seq (Shahabi et al., 2016; Jhun et al., 2017). In both cases, gene-based and gene-free, the number of features is too large to directly apply LASSO logistic regression on the count matrix (see Section 1.3.4). For this reason, we adopted a preliminary drastic screening step designed to reduce the number of features to a lower number and avoid to run the LASSO logistic regression in an ultrahigh dimensional setting. To single out important features, we use univariate ranking of features based on their ability to predict new data using a Naïve Bayes rule and a F1-score computed by **c**ross-**v**alidation. Given the large number of k-mers to rank, we choose the Naïve Bayes rule as suggested by authors from Thomas et al. (2019) because the C++ implementation of the Naïve Bayes was the fastest to run among the set of available tools. Other solutions are possible, such as the use of other algorithms than Naïve Bayes to rank the features, and other feature selection techniques and multivariate supervised learning than the LASSO logistic regression. However, to perform a fair comparison between the gene-based and the gene-free approaches, we selected the tools prior to running the comparison, independently of external considerations: we did not try to optimize the set of tools used to bias the comparison towards one approach or the other. Another issue is the choice of tools for inferring the k-mer signature: Thomas et al. (2019) have proposed to use a **G**enetic **A**lgorithm and we have proposed the pipeline summarized above and described in more detail in section 7.2.2. In this thesis, we have not addressed this question. Instead we focused primarily on the comparison of gene-based and gene-free approaches. In the proposed pipeline, we used a matrix reduction technique based on k-mers extension into contigs proposed in section 5.4 to avoid working directly on a matrix of k-mers full of correlated and redundant k-mers. We have not use the filtering strategies prior univariate screening step for the following two reasons. First, the filtering strategies exposed in Chapter 5 are primarily designed to remove low-abundant k-mers prior to differential expression analysis. Second, the screening step leads to a drastic reduction of the feature space (from thousands

or millions to a few hundreds). In this context, filtering prior screening would only slightly decrease the running time of the screening step, and would not change the final set of features retained for subsequent feature selection and supervised learning.

Chapter 7 corresponds to a preprint currently under review for publication.

Published Online: 15 November, 2019 | Supp Info: http://doi.org/10.26508/lsa.201900449

Methods

TRANSPARENT PROCESS OPEN ACCESS Life Science Alliance

Reference-free transcriptome exploration reveals novel RNAs for prostate cancer diagnosis

Marina Pinskaya[1,*], Zohra Saci[1,*], Mélina Gallopin[2], Marc Gabriel[1], Ha TN Nguyen[2,3], Virginie Firlej[4,5], Marc Descrimes[1], Audrey Rapinat[6], David Gentien[6], Alexandre de la Taille[4,5,7], Arturo Londoño-Vallejo[8], Yves Allory[9], Daniel Gautheret[2], Antonin Morillon[1]

The use of RNA-sequencing technologies held a promise of improved diagnostic tools based on comprehensive transcript sets. However, mining human transcriptome data for disease biomarkers in clinical specimens are restricted by the limited power of conventional reference-based protocols relying on unique and annotated transcripts. Here, we implemented a blind reference-free computational protocol, DE-kupl, to infer yet unreferenced RNA variations from total stranded RNA-sequencing datasets of tissue origin. As a bench test, this protocol was powered for detection of RNA subsequences embedded into putative long noncoding (lnc)RNAs expressed in prostate cancer. Through filtering of 1,179 candidates, we defined 21 lncRNAs that were further validated by NanoString for robust tumor-specific expression in 144 tissue specimens. Predictive modeling yielded a restricted probe panel enabling more than 90% of true-positive detections of cancer in an independent The Cancer Genome Atlas cohort. Remarkably, this clinical signature made of only nine unannotated lncRNAs largely outperformed PCA3, the only used prostate cancer lncRNA biomarker, in detection of high-risk tumors. This modular workflow is highly sensitive and can be applied to any pathology or clinical application.

DOI 10.26508/lsa.201900449 | Received 4 June 2019 | Revised 5 November 2019 | Accepted 5 November 2019 | Published online 15 November 2019

Introduction

RNA sequencing (RNA-seq) has revolutionized our knowledge of human transcriptome and has been implemented as a pivot technique in clinical applications for discovery of RNA-based biomarkers allowing disease diagnosis, prognosis and therapy follow-up. However, most biomarker discovery pipelines are blind to uncharacterized RNA molecules because they rely on the alignment of uniquely mapped reads to annotated references of the human transcriptome, which are far from complete (Deveson et al, 2018; Uszczynska-Ratajczak et al, 2018; Morillon & Gautheret, 2019). Indeed, unspliced variants, rare mRNA isoforms, RNA hybrids originating from *trans*-splicing or genome rearrangements, unannotated intergenic or antisense noncoding RNAs, mobile elements, or viral genome insertions would be systematically missed. A recent approach to RNA-seq data analysis, DE-kupl, combines k-mer decomposition and differential expression analysis to discover transcript variations yet unreferenced in the human transcriptome (Audoux et al, 2017). Applied to poly(A)+ RNA-seq datasets of in vitro cell system, DE-kupl unveiled a large number of RNA subsequences embedded into novel long noncoding (lnc)RNAs. These transcripts of more than 200 nucleotides in length transcribed by RNA polymerase II from intergenic, intronic, or antisense noncoding genomic locations constitute a prevalent class of human genes. Some lncRNAs are now recognized as precisely regulated stand-alone molecules participating in the control of fundamental cellular processes (Quinn & Chang, 2015; Jarroux et al, 2017). They show aberrant and specific expression in various cancers and other diseases promoting them as biomarkers, therapeutic molecules and drug targets (Van Grembergen et al, 2016; Leucci, 2018). Importantly, some lncRNAs can be robustly detected in biological fluids (blood and urine) as circulating molecules or encapsulated into extracellular vesicles, hence, raising an attractive possibility of their usage as biomarkers in non-invasive clinical tests (Wang et al, 2014; Silva et al, 2015; Deng et al, 2017; Wang et al, 2018; Zhao et al, 2018). The only example of a lncRNA-based biomarker so far introduced in clinical practice of prostate cancer (PCa) is the PCA3 lncRNA (de Kok et al, 2002). PCA3 is transcribed antisense to the tumor suppressor *PRUNE2* gene and

[1]ncRNA, Epigenetic and Genome Fluidity, Université Paris Sciences & Lettres (PSL), Sorbonne Université, Centre National de la Recherche Scientifique (CNRS), Institut Curie, Research Center, Paris, France [2]Institute for Integrative Biology of the Cell, Commissariat à l'Energie Atomique, CNRS, Université Paris-Sud, Université Paris-Saclay, Gif sur Yvette, France [3]Thuyloi University, Hanoi, Vietnam [4]Université Paris-Est Créteil, Créteil, France [5]Institut National de la Santé et de la Recherche Médicale, U955, Equipe 7, Créteil, France [6]Translational Research Department, Genomics Platform, Institut Curie, Université PSL, Paris, France [7]Assistance Publique – Hôpitaux de Paris, Hôpital Henri Mondor, Département d'Urologie, Créteil, France [8]Telomeres and Cancer, Université PSL, Sorbonne Université, CNRS, Institut Curie, Research Center, Paris, France [9]Compartimentation et Dynamique Cellulaire, Université PSL, Sorbonne Université, CNRS, Institut Curie, Research Center, Paris, France

Correspondence: antonin.morillon@curie.fr; daniel.gautheret@u-psud.fr
Zohra Saci's present address is CHU Sainte-Justine Research Centre, University of Montreal, Montreal, Quebec, Canada
*Marina Pinskaya and Zohra Saci contributed equally to this work

promotes its pre-mRNA editing and degradation (Salameh et al, 2015). Being overexpressed in 95% of PCa cases, PCA3 is detected in urine and helps diagnosis providing, in addition to other clinical tests, more accurate metrics regarding repeated biopsies (Groskopf et al, 2006; Galasso et al, 2010). However, it remains inaccurate in discrimination between low- and high-risk tumors because its expression may dramatically decrease in aggressive PCa cases tempering its systematic usage (Loeb & Partin, 2011; Alshalalfa et al, 2017).

Since PCA3 discovery and the development of RNA-seq technologies, the PCa transcriptome has been extensively explored by The Cancer Genome Atlas (TCGA) consortium and others to identify numerous PCa-associated lncRNAs (PCAT family) such as PCAT1, PCAT7, or PCAT114/SChLAP1 (Prensner et al, 2014; Iyer et al, 2015). However, none of them has been yet introduced into clinical practice because of the variable expression incidence, as for SChLAP1 detected in 25% of PCa cases presenting metastatic traits (Prensner et al, 2013), or low specificity, as PCAT1 or PCAT7, thus infringing their clinical value. Additional efforts are required for more accurate and exhaustive RNA identification, as well as more rigorous validations of clinical potency through independent RNA measurement technologies and clinical cohorts. Regardless a large number of transcriptomic studies and variety of clinical samples analyzed, discovery of RNA-based biomarkers from publicly available RNA-seq datasets is still limited at two levels: (i) most experimental setups are based on poly(A) selected, unstranded cDNA sequencing, and (ii) computational analyses are generally focused on annotated genes and full-length RNA assemblies. This impedes the detection of low and poorly polyadenylated RNAs but also partially degraded RNAs from formalin-fixed paraffin-embedded tissues or other clinical samples (Zhao et al, 2014; Zhao et al, 2018). In addition, non-stranded RNA-seq reads counting is less accurate at 5′ RNA ends or even impossible for co-expressed paired sense/antisense transcripts and for yet unannotated RNAs among non-coding, fusion, repeat-derived transcripts (Davila et al, 2016; Audoux et al, 2017).

Here, we propose a conceptually novel exploratory framework combining the total stranded RNA-seq of clinical samples and the reference-free DE-kupl algorithm for discovery of novel tumor-specific transcript variations. As a proof-of-concept, we focused on the least explored, noncoding portion of the genome devoid of annotated protein-coding sequences to build an exhaustive catalog of PCa associated subsequences (contigs) embedded into lncRNA genes. The catalog was further refined through minimal filtering to isolate the subset of contigs with best differential expression features and validate 21 of them by a custom NanoString assay in the extended cohort of 144 prostate specimens. From this, a predictive modeling derived a panel of nine yet unannotated lncRNAs validated for robust expression in an independent TCGA cohort. Importantly, its clinical performance surpassed the PCA3 lncRNA specifically in discrimination of high-risk tumors. The proposed probe-set can be further used for development of a PCa diagnostic test. Moving beyond this point, the proposed computational and experimental platform may serve as a tool for biomarkers discovery of any disease and any clinical task aiming at improved medical care and development of precision medicine approaches.

Results

Identification of PCa-specific RNA variants in the *Discovery Set* by DE-kupl

The biomarker discovery workflow included three major phases: discovery, selection, and validation (Fig 1). First, for discovery, we performed a deep total stranded RNA-seq of ribosomal RNA-depleted RNA samples isolated from prostate tissues after radical prostatectomy (*Discovery Set*, PAIR cohort, Table S1). This *Discovery Set* was processed by DE-kupl to identify tumor-specific transcripts. DE-kupl directly queries FASTQ files for subsequences (k-mers) with differential counts/expression (DE) between two conditions (Fig 2A) (Audoux et al, 2017). Overlapping k-mers are then assembled into contigs and, in a final step, mapped to the human genome for annotation. In the aim to focus exclusively on novel, yet unannotated RNA elements, k-mers exactly matching GENCODE-annotated transcripts were masked. We eventually retained contigs within the noncoding regions (antisense to protein-coding or noncoding genes, intergenic) longer than 200 nucleotides and showing adjusted *P*-values below 0.01 to capture the most significant expression changes linked either to new transcriptional or processing events within known or putative lncRNA loci.

With these criteria, we identified 1,179 tumor up-regulated contigs assigned to four main categories according to their mapping features: contiguous (uniquely mapped) contigs (N = 935), splice variants (N = 54), repeats (N = 167), and unmapped contigs (N = 23) (Figs 2B and S1, and Table S2). Among them, 586 contigs were

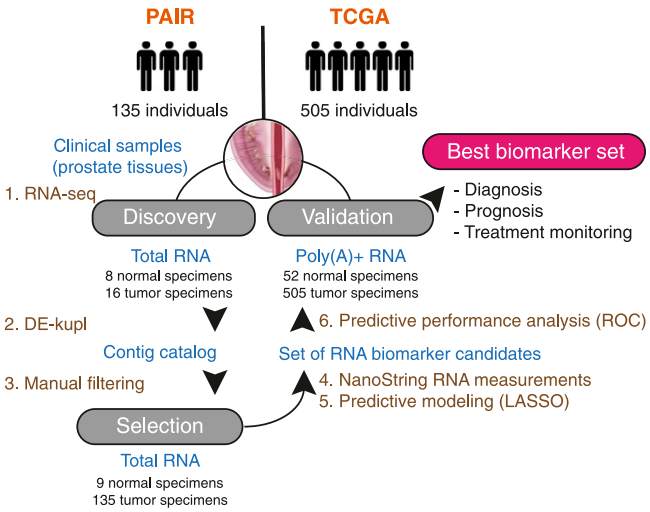
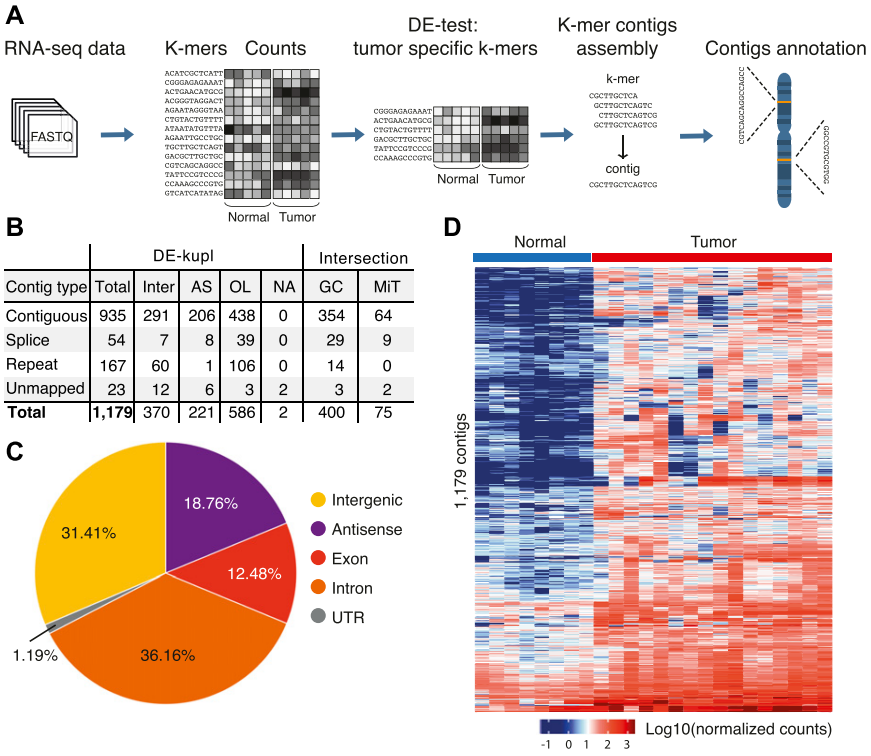


Figure 1. Experimental and computational workflow for discovery and validation of RNA-based clinical biomarkers.
Raw total stranded RNA-seq data of a small clinical cohort is processed by DE-kupl to allow comparison of 8 normal against 16 tumor specimens (in this case, formaldehyde-fixed paraffin-embedded tissues from radical prostatectomy) and cataloguing of all differentially expressed RNA variations (contigs). The whole set is filtered according to desired criteria and the top ranked contigs are selected for an independent experimental validation by NanoString in the extended clinical cohort. Finally, predictive modeling infers the best panel of candidate RNAs for validation of its clinical potency in an independent cohort (in this case TCGA).



	DE-kupl					Intersection	
Contig type	Total	Inter	AS	OL	NA	GC	MiT
Contiguous	935	291	206	438	0	354	64
Splice	54	7	8	39	0	29	9
Repeat	167	60	1	106	0	14	0
Unmapped	23	12	6	3	2	3	2
Total	**1,179**	370	221	586	2	400	75

Figure 2. K-mer decomposition protocol for detection of differentially expressed RNA variants in PCa.
(A) DE-kupl workflow with principle steps of contigs counting, DE-test and filtering, assembly and annotation. **(B)** Catalog of DE-kupl contigs of different subgroups: contiguous—contigs mapped as unique fragments; spliced—contigs mapped as spliced fragments; repeat—multiply mapped contigs; Inter—contigs mapping into intergenic regions; OL—at least one nucleotide overlapping of GENCODE lncRNA annotations; and AS—antisense to a protein-coding or a noncoding gene. Contigs of each subgroup showing 50% sequence overlap with GENCODE v27 (GC)– and MiTranscriptome v2 (MiT)–annotated genes are counted. **(C)** Pie chart of 1,179 contigs distribution across GENCODE-annotated features. **(D)** Unsupervised hierarchical cluster heat map of Log10(normalized counts) of 1,179 contigs assessed in 8 normal and 16 tumor specimens by total stranded RNA-seq of the *Discovery Set*. NA stands for non-annotated in human genome.

embedded into already referenced GENCODE lncRNA genes, but represented new sequence variations or RNA processing events, as PCAT7 (ctg_111158, P6) or CTBP1-AS (ctg_25348, P10). The rest mapped to intergenic noncoding locations (370 contigs) or antisense to referenced protein-coding or noncoding genes (221 contigs) (Fig 2C). Intersection with existing annotations revealed 50% sequence overlap of contigs with 400 (33.93%) GENCODE and 75 (6.36%) MiTranscriptome lncRNA genes (Fig 2B). An unsupervised clustering of prostate specimens based on contigs expression counts allowed proper discrimination of tumor from normal tissues of the *Discovery Set* (Fig 2D).

In conclusion, DE-kupl identified a thousand of PCa-associated RNA variants for the majority embedded into yet unreferenced transcripts which may represent putative novel lncRNAs. This depository was further explored for clinical relevance.

Naïve assembly of transcription units identifies novel prostate cancer associated lncRNAs

To complement the reference-free protocol, we applied a reference-based protocol to build a catalog of lncRNAs from the same *Discovery Set*. Total RNA-seq produces much more intronic and exon–exon junction reads than poly(A)–selected RNA-seq. This complexity renders laborious in time and machine memory the data analysis by splice graph–based assemblers such as Cufflinks (Hayer et al, 2015; Kukurba & Montgomery, 2015). To bypass this difficulty, we developed a more straightforward lncRNA annotation pipeline, HoLdUp, which identifies transcription units (TUs) based on coverage analysis (Fig 3A). In this workflow, uniquely mapped reads were assembled into TUs and mapped to the GENCODE annotation to extract intergenic and antisense lncRNAs (see the Materials and Methods section for details). They were further ranked according to their expression level, presence of splice junctions, and existence of matched ESTs. In total, we retained 168,163 TUs with above-threshold expression of 0.2 quartile of mRNA expression (Class 2) and, within this group, the most robust 2,972 TUs with at least one splice junction and one EST (Class 1) (Fig 3B). Globally, newly detected transcripts were as much expressed as GENCODE-annotated lncRNAs but lower than mRNAs (Fig S2A). Only 0.33% of Class 1 lncRNAs were present with at least 50% nucleotide sequence overlap in the recent GENCODE v27 catalog and 43.37% of TUs in the MiTranscriptome lncRNA repertoire; the rest represented putative novel lncRNA genes (Figs 3B and S2B). Of 2,972 TUs, DE analysis retrieved 127 of Class 1 TUs significantly up-regulated in tumor specimens (adjusted *P*-value below 0.01, DESeq), including multiple intergenic transcripts and transcripts antisense to protein-coding genes, such as *HDAC9*, *TPO*, and *FBXL7* (Table S3 and Fig S2B).

Intersection of DE-kupl contigs with PCa up-regulated HoLdUp TUs (N = 127) and the recent GENCODE lncRNA annotation (N = 206) showed that 687 DE-kupl contigs of 1,179 make part of the stand-alone transcripts. Moreover, up to 85.5% and 96.8% DE-kupl contigs embedded into GENCODE and HoLdUp Class 1 lncRNA genes, respectively, were also detected by DESeq as significantly up-regulated transcripts in the same dataset, when the RNA-seq reads were counted within the entire TU (Figs 3C and S2C). One such example is the contig ctg_23999 (P22) embedded into a novel HoLdUp assembled Class 1 TU antisense to the protein-coding *FBXL7* gene (Fig 3D).

In conclusion, the reference-based assembly protocol HoLdUp is complementary to DE-kupl and allows attributing short RNA

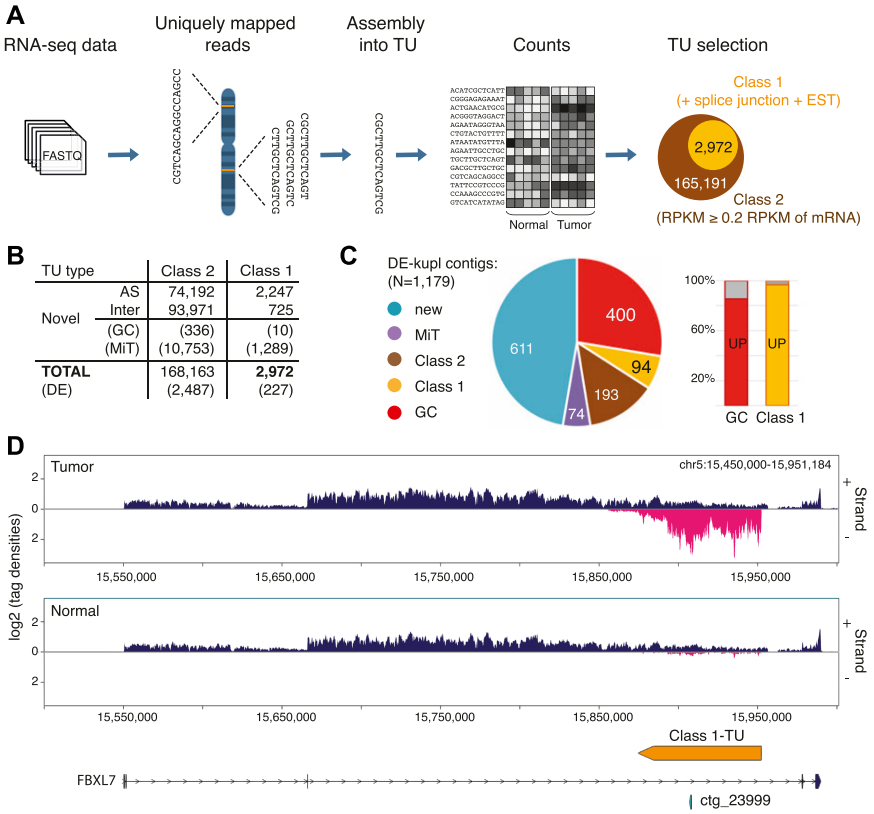


Figure 3. Reference-based lncRNA discovery from total stranded RNA-seq.
(A) HoLdUp protocol for the ab initio assembly of TUs constituting putative lncRNA genes and their classification into Class 2 and Class 1 TUs according to robustness of detection. **(B)** HoLdUp catalog and TUs overlap with GENCODE v27– (GC) and MiTranscriptome (MiT)–annotated lncRNAs. DE stands for differentially expressed transcripts (DESeq adj. *P*-value < 0.01). **(C)** Pie chart representation of non-exclusive distribution of DE-kupl contigs across different lncRNA annotations: MiTranscriptome (violet), Class 1 (yellow), Class 2 (brown), GENCODE (red), and novel (blue); number of contigs is marked in each section. Proportion of DE-kupl contigs embedded into up-regulated (UP) GENCODE (red) and Class 1 (yellow) lncRNAs is expressed as a histogram. **(D)** VING-generated RNA-seq profiling along plus (+) and minus (−) strands of chr5:15,500,295-15,939,910 in tumor and normal prostate specimens: the GENCODE-annotated protein-coding gene *FBXL7* (blue), antisense DE-kupl contig ctg_23999 (P22), and antisense HoLdUp Class 1-TU (orange). Arrow-lines and rectangles represent introns and exons, respectively. DE, differentially expressed; RPKM, reads per kilo base per million mapped reads; and TU, transcription unit.

subsequences to whole TUs. Nevertheless, DE-kupl was more powerful illuminating much more transcriptomic variations not only within the annotated genes but also within putative new noncoding regions in highly complex and heterogeneous total RNA-seq datasets of clinical origin.

Selection of a restricted set of 23 PCa RNA contigs showing the highest differential expression

We further leveraged the DE-kupl contig catalog to define a robust PCa signature among putative new lncRNAs using several filters (Fig S3A). Hereafter, we will use the term *signature* to describe the set of contigs or genes selected for their ability to predict a sample status. First, contigs were sorted according to their adjusted *P*-value and, second, were visually selected using the Integrative Genomic Viewer applying the following criteria: (i) when several contigs were present within the same genomic region (5 kb window) the contig with the lowest adjusted *P*-value was retained, (ii) contigs antisense to expressed exons, bidirectional or positioned in close vicinity to other transcribed protein-coding genes were filtered out. We retained several contigs embedded into already annotated PCa associated lncRNA genes, such as *CTBP1-AS* (ctg_25348, P10), *PCAT7* (ctg_111158, P6), and *PCAT1* (ctg_105149, P18), or lncRNAs referenced elsewhere as ctg_104447 (P11) mapped into *LOC283177*, ctg_123090 (P5) into *AC004066.3*, and ctg_73782 (P8) into *LINC01006*. It should be noted that the GENCODE referenced genes enclosing these new subsequences also showed differential expression when counting on the whole gene annotation (Fig S3B). However, in contrast to DE-kupl ranking, they were not among the strongest hits in the DESeq analysis with exception of PCAT7 (Table S4). This observation points to the fact that through expression counting within the small subsequences, DE-kupl is more resolutive and hence sensitive in the discovery of DE sequences. Visualization of RNA-seq reads and junctions of a region embedding *FBP2* and its antisense *PCAT7* genes revealed a new contig ctg_28650 (P2) downstream of the *PCAT7* annotation and antisense to *FBP2*. The continuous coverage and absence of splice junctions in reads profiling suggest that P2 is enclosed into an extension of the last *PCAT7* exon (Fig S3C and D). This contig was retained in the restricted list as the strongest candidate antisense to *FBP2*, overcoming ctg_111158 (P6) assigned to the *PCAT7* gene itself. Still, additional experiments are required to validate this lncRNA variant, yet absent from the existing PCAT7 annotation.

In total, 23 candidates belonging to contiguous (N = 21), spliced (N = 1), or repeat (N = 1) subgroups of contigs were selected for further validation, all being expressed at least six times more in tumor tissues than in normal prostate (Fig S3E and Tables S2 and S5). Among them, 12 candidates mapped antisense to annotated protein-coding or lncRNA genes and 11 located to intergenic regions. To facilitate further reading, contigs' identity are replaced by probes' identity from P1 to P23 according to increasing *P*-values of DE of the *Discovery Set* (Table S5).

After the manual filtering, we aimed to validate the expression of selected 23 contigs in the extended PAIR cohort of nine normal and

135 tumor specimens (*Selection Set*) (Table S6). This cohort contained one additional specimen for normal tissue and 119 additional tumor specimens. To measure contigs expression, an alternative RNA quantification procedure based on the NanoString nCounter platform for direct enzyme-free multiplex digital RNA measurements was carried out (Fig 4A). In addition to DE-kupl contigs, a probe for PCA3 was used as a benchmark lncRNA. We also measured the expression of six housekeeping genes and selected three lowly expressed mRNAs (GPATCH3, ZNF2, and ZNF346) as custom internal controls for relative quantifications (Table S7 and Fig S4).

The NanoString assay revealed that all DE-kupl contigs were expressed at a lower level than PCA3, but still 21 of 23 contigs were significantly overexpressed (Wilcoxon *P*-value < 0.01) in tumor specimens (Fig 4A and Table S8). Two contigs, intergenic P22 (ctg_119680) and repeat P17 (ctg_36195), did not show significant difference in expression between normal and tumor specimens. Ranking according to *P*-values revealed 12 contigs better than PCA3. Among the top DE contigs were those embedded into *PCAT1* (ctg_105149, P18), *CTBP1-AS* (ctg_25348, P10), and *PCAT7* (ctg_111158, P6) genes, whereas the rest were assigned to novel lncRNAs. Notably, apart from P17 (ctg_36195) and P22 (ctg_119680), expression measurements were consistent between the two technologies, total stranded RNA-seq and NanoString, although the *P*-value ordering was different (Fig S5 and Table S9).

Thus, 21 of 23 contigs were validated in the extended set of RNA specimens using the independent single-molecule measurement technology.

Validation of contig-based RNA candidates in an independent clinical cohort

Independent validation of DE-kupl contigs was performed using the biggest PCa clinical resource of 557 poly(A)+ RNA-seq datasets, including 52 normal and 505 tumor tissues from radical prostatectomy (TCGA-prostate adenocarcinoma [PRAD] cohort, *Validation Set*) (Fig 1 and Table S10).

The occurrence of sequences representing 23 DE-kupl contigs was measured and compared with PCA3. In total, 16 of 23 DE-kupl contigs had significant support for overexpression in tumor specimens in the TCGA-PRAD cohort (Wilcoxon *P*-value < 0.01, Fold Change [FC] > 2) (Fig 4B and Table S11). Among the best scored candidates, the two novel DE-kupl contigs, P16 (ctg_111348) antisense to *DLX1* and intergenic P1 (ctg_17297), surpassed PCA3 that ranked third. However, important discrepancies were observed between expression counts in poly(A)+ RNA-seq TCGA datasets and NanoString or total RNA-seq PAIR datasets. First, P22 (ctg_119680) was detected as DE in TCGA-PRAD but failed the DE test when measured by NanoString (Figs 4 and S5). Second, the expression of nine DE-kupl contigs were near the base line in the TCGA dataset, including those showing relatively high expression and low *P*-values in the PAIR cohort, such as P14 (ctg_61528) antisense to *TPO* or the intergenic P9 (ctg_9446). Detection of these contigs in TCGA-PRAD was compromised independently of their genomic location (intergenic or antisense) or of the expression level of a sense-paired gene. We hypothesized that it is most likely due to a relatively low RNA-seq coverage and/or to a loss of poorly or non-polyadenylated transcripts during cDNA library preparation in the TCGA experimental setup. Finally, ranking of contigs according to increasing *P*-values was very different between *Selection* and *Validation Sets* highlighting discrepancies between technologies, clinical origins, and cohort sizes.

Regardless all experimental biases, 16 of 23 DE-kupl contigs were validated in the independent clinical cohort as significantly overexpressed in tumors. This cohort was further used for validation of clinical potency of contigs.

Expression of DE-kupl contigs is independent of tumor risk and recurrence metrics

Several clinical studies have revealed high heterogeneity of expression and low efficiency of the PCA3 biomarker in detection of high-risk tumors, questioning its robustness and reliability in PCa

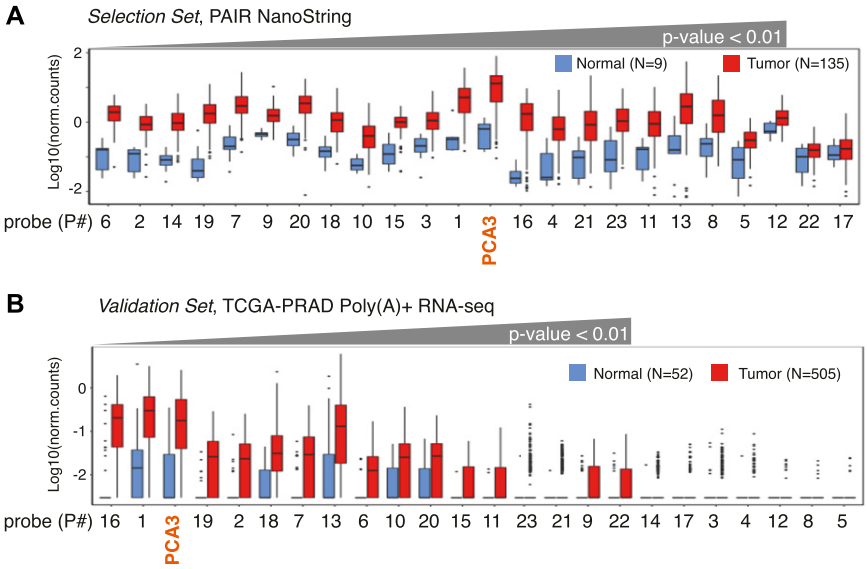


Figure 4. Expression of lncRNA subsequences in PAIR and TCGA-PRAD cohorts.
(A) Box-plot of Log10(norm.counts) of PCA3 and 23 DE-kupl contigs in 144 PAIR specimens of the *Selection Set* by NanoString. **(B)** Box-plot of Log10(norm.counts) of PCA3 and 23 DE-kupl contigs in 557 TCGA-PRAD specimens of the *Validation Set* by poly(A)+ unstranded RNA-seq. Normal tissues: in blue, tumor tissues: in red.

diagnostics (Alshalalfa et al, 2017; Fenstermaker et al, 2017). We assessed contig expression in tumors of different clinical metrics. For risk prognosis, the most common metric is a three-group risk stratification system established by D'Amico et al (1998), which takes into account preoperative PSA level, biopsy Gleason Score, and clinical TNM stage. As mentioned above, this scheme is highly debated because of disagreements on the PSA score in relation to PCa over-diagnosis (Carlsson et al, 2012; Loeb et al, 2014). To define a molecular signature independent of PSA, we excluded this criterion and categorized tumor specimens into low-, intermediate-, and high-risk groups uniquely on the basis of Gleason and TNM features, below referred to as naïve indexing (Fig S6A and B). In addition to risk assessment, we also separated specimens in two subgroups depending on the tumor recurrence status (Fig S6B). Then, expression of PCA3 and the 23 DE-kupl contigs were compared for each subgroup of the *Selection Set*.

To evaluate the robustness of contig expression, we ranked probes by decreasing FC for high-risk against low-risk tumors and positive against negative recurrence status (Fig 5). Most contigs showed robust expression independently of the tumor classification. In contrast, the PCA3 level was more disperse with the lower median and mean expression and higher *P*-values in high-risk and recurrence positive specimens (Table S12). While considering only 21 significantly overexpressed contigs, 17 of them outperformed PCA3 in both contrasts (Table S12). Notably, among the best performe were contigs P6 (ctg_111158) and P2 (ctg_28650) both antisense to *FBP2*, P10 (ctg_25348) embedded into CTBP1-AS, as well as the novel P16 (ctg_111348) antisense to *DLX1* and the intergenic P1 (ctg_17297).

In conclusion, most DE-kupl contigs showed robust expression independent of tumor metrics. Hence, even if used alone, they may offer a better clinical potency for PCa diagnosis than PCA3.

Inferring a multiplex RNA-probe panel and evaluation of its performance in PCa diagnosis

To extract parsimonious probe signature predicting the tumor status, we applied Least Absolute Shrinkage and Selection Operator (LASSO) logistic regression on the *Selection Set* of 144 PAIR specimens (Ghosh & Chinnaiyan, 2005). First, the initial 21 DE-kupl contigs and PCA3 validated for expression by NanoString were submitted to LASSO to define the best mixed signature comprised of already known and yet unannotated lncRNA probes for discrimination of tumor from normal tissues (Fig S7A). Then, LASSO was performed with the probe subset composed uniquely of contigs assigned to putative novel lncRNAs (N = 15) to infer the best new-lncRNA signature. It resulted in two panels of nine mixed and nine new-lncRNA candidates (Figs 6A and S7B). Retrieved signatures were then used to predict a tumor status in the *Validation Set* of the TCGA-PRAD cohort using a leave-one-out cross-validated boosted logistic regression. To assess the sensitivity of DE-kupl contigs in PCa diagnosis, a predictive accuracy index, area under curve (AUC) of the receiver-operating characteristic (ROC), was calculated for each signature and PCA3 alone in the PAIR (*Selection Set*) and TCGA-PRAD (*Validation Set*) datasets (Figs 6B and S7B). Remarkably, all signatures still hold their predictive capacity in the independent TCGA-PRAD cohort in spite of the important differences in experimental setups between the two studies. Both markedly outperformed PCA3 for tumor detection with AUC of 0.92 for mixed and of 0.91 for new-lncRNA signatures against AUC of 0.73 for PCA3 (Fig 6B and C). In addition, these signatures were much better in predicting high-risk tumors where PCA3 is particularly inaccurate (Fig 6C). Remarkably, the new-lncRNA signature composed uniquely of yet unannotated lncRNA subsequences predicted the tumor status with the same performance as the mixed signature. Logistic regression did not retain PCA3 within the mixed signature set, instead contigs embedded into the well characterized PCAT1 lncRNA and into two already annotated but yet functionally uncharacterized lncRNAs LOC283177 and LINC01006 were present.

We also compared predictive performances of signatures retrieved by the k-mer–based classifier to the one inferred using conventional gene expression counting. Differential expression analysis for GENCODE-annotated genes of the *Discovery Set* retrieved 520 up-regulated genes, protein-coding and noncoding, with adjusted *P*-values lower than 0.05 and a logFC higher than 2

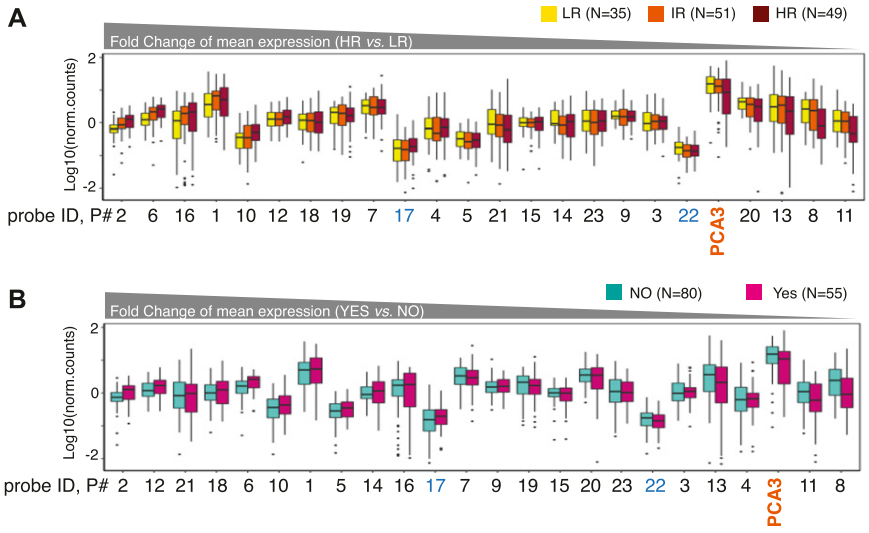


Figure 5. Expression of lncRNA subsequences in prostate specimens of different clinical metrics in the PAIR cohort (*Selection Set*).
(A, B) Box-plot of Log10(norm.counts) of PCA3 and 23 DE-kupl contigs depending on tumor risk (A) and recurrence status (B) assessed by NanoString. PCA3 is marked in orange, and the contigs showing insignificant expression change between normal and tumor specimens are in blue. Contigs are ordered by the decreasing FC of mean expression in high-risk versus low-risk specimens in the (A) panel and in Yes versus NO recurrence specimens in the (B) panel. HR, high-risk; IR, intermediate-risk; LR, low-risk.

A

Probe	Signature: mixed	Signature: new-lnc	contig origin
P8	ctg_73782		LINC01006
P18	ctg_105149		PCAT1
P11	ctg_104447		LOC283177
P1	ctg_17297	ctg_17297	intergenic
P2	ctg_28650	ctg_28650	AS to *FBP2*
P7	ctg_117356	ctg_117356	AS to *snoU13*
P15	ctg_512	ctg_512	AS to *PXDN*
P20	ctg_44030	ctg_44030	integenic
P23	ctg_29077	ctg_29077	AS to *AC011523.2*
P3		ctg_57223	intergenic
P12		ctg_2815	intergenic
P14		ctg_61528	AS to TPO

C

	PCA3	mixed	new-lnc
Normal *vs.* Tumor	0.73±0.05	0.92±0.03	0.91±0.03
Normal *vs.* HR	0.69±0.05	0.91±0.03	0.91±0.03
Normal *vs.* IR	0.78±0.05	0.90±0.05	0.90±0.04
Normal *vs.* LR	0.78±0.11	0.92±0.04	0.91±0.03

B

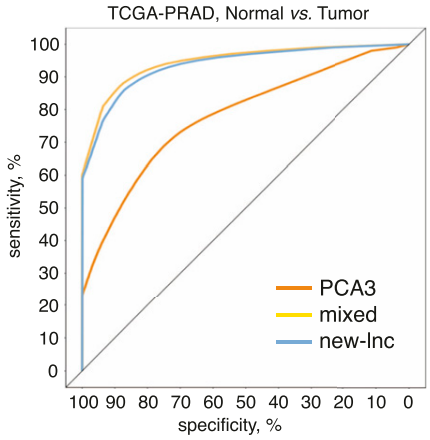


Figure 6. Predictive performance of PCA3 and multiplex mixed and new-lncRNA signatures inferred from the LASSO penalized logistic regression.
(A) Multiplex biomarker signatures composed of either known and unannotated RNAs (mixed) or of only unannotated RNAs (new-lnc). **(B)** ROC for the PCa prediction in the TCGA dataset (*Validation Set*) using two signatures and PCA3 alone. **(C)** Mean and SD of AUC computed over 100 samplings of the *Validation Set* for PCA3 and two signatures to classify samples according to their tumor status. AS, antisense; AUC, area under the curve; HR, high-risk; IR, intermediate-risk; LR, low-risk tumors.

(Table S4). These genes were then selected on the *Discovery Set* using LASSO penalized logistic regression to extract a GENCODE whole gene counting based (WGC) signature for further validation (Fig S7B). Given the high dimensional setting (more variables than observations available), we performed the stability selection (Meinshausen & Bühlmann, 2010) and kept the five genes that had a probability of being selected higher than 0.5 on 2,000 samplings of the original dataset. Remarkably, the retrieved subset was composed majorly of noncoding transcripts (four of five), although PCA3 did not pass the selection. Of them, the protein-coding HPN mRNA, the PCAT7 lncRNA, and the GLYATL1P4 pseudogene have been already associated with PCa in other studies (Willard & Koochekpour, 2012; Du et al, 2013; Kim et al, 2019). Notably, the GLYATL1P4 transcript makes part, together with 21 other RNAs, of the Decipher test proposed in clinics to guide timing of radiation therapy after radical prostatectomy in men with high-risk cancer (Alford et al, 2017). The predictive performance in discrimination between normal and tumor specimens of the WGC signature was tested by the ROC analysis on the *Validation Set* and resulted in the mean AUC of 0.91 (Fig S7B). Hence, k-mer based signature discovery method retrieving yet unreferenced RNA subsequence was as powerful as the signature derived from GENCODE-annotated genes. Although the predictive modeling enabled to reach the same performance only from 23 contig probes instead of 520 DE-genes and, remarkably, this was achieved in TCGA-PRAD datasets where contigs expression counting is most likely disfavored considering all aforementioned drawbacks of poly(A)–selected datasets of low coverage.

Discovery of novel RNA signatures with high tumor predictive potential also highlights both the incompleteness of current cancer transcriptome datasets and the biological value of transcript information that can be extracted through different experimental (total stranded RNA-seq and NanoString quantification) and computational (DE-kupl) tools. De-kupl–derived novel signature demonstrated a sensitivity and robustness towards tumor risk prediction surpassing the state of the art for discrimination of prostate cancer. Furthermore, established nine-probe RNA signature was performed not only independently of tumor origin and its clinicopathological characteristics but also of the technology used for RNA measurements.

Discussion

Molecular biomarker assays are invaluable tools in cancer diagnosis, prognosis and treatment follow-up. Within this scope, sequencing technologies unveiled the pervasiveness and diversity of the human transcriptome, promoting lncRNAs as important cancer signatures (Schmitt & Chang, 2016). These molecules are highly dynamic and reflect cellular states in a sensitive and specific way because of their involvement in genetic and regulatory flows of information. However, the variety of RNA species and high heterogeneity of expression present a challenge for their detection and proper quantification in clinical samples. Predominant microarray and unstranded poly(A)+ RNA-seq–based approaches allowed identification of numerous lncRNAs with tumorigenic function. However, their clinical performance as biomarkers stays rather poor because of the aforementioned RNA features hindering RNA detection, quantification, and clinical validation under conventional experimental setups. Here, we presented an innovative experimental and computational platform that permits discovery of RNA biomarkers of high clinical potency from total stranded RNA-seq datasets of clinical origin.

As a proof-of-concept, we focused on PCa as the only type of cancer using, so far, a lncRNA-based diagnostic test (Progensa). The *Discovery Set* based on comparison of 8 normal with 16 tumor specimens from total RNA-seq datasets was processed by DE-kupl to extract the most significant differentially expressed subsequences in the form of k-mer contigs. Further filtering based on contig length, genomic position, and expression levels powered the pipeline towards the discovery of putative lncRNAs, for the majority, yet unreferenced in the human transcriptome. Then, the catalog of contigs was manually refined and tested for expression using the NanoString single-molecule RNA counting technology in the extended cohort of 144 specimens. Contig expression was next assessed in the independent, publicly available TCGA-PRAD dataset generated by the poly(A)+ unstranded RNA-seq technology. The expression of contigs was systematically compared with that of the benchmark biomarker lncRNA, PCA3. In total, 16 of 23 contigs were validated in both setups but with important differences. Primarily,

RNA measurements were consistent between two different technologies: NanoString and total stranded RNA-seq. In contrast, the TCGA poly(A)+ unstranded datasets revealed weakness and high heterogeneity of contig counts over the selected regions, resulting in unexpectedly low signals even for PCA3, considered as a highly expressed lncRNA. Hence, our results promote the total stranded RNA-seq as a first-choice strategy for discovery of RNA biomarkers from clinical samples and when searching for transcripts others than highly abundant mRNAs. It reflects far more precisely the transcriptomic landscape of clinical samples and, hence, is more advantageous as a *Discovery Set* for development of clinical tests. At the same time, full-length transcript assembly from short-read sequencing is inaccurate, time and computer memory consuming, and this is aggravated by the added complexity of total (ribo-depleted) RNA-seq libraries (Hayer et al, 2015). DE-kupl bypasses this issue by directly extracting from raw data RNA subsequences significantly overexpressed in a defined condition. In PCa tissues, this allowed identification of 1,179 lncRNA-hosted candidates. Further analysis isolated a restrained set of nine contigs either within putative new lncRNAs or mixed annotated and novel lncRNAs allowing PCa diagnosis independently of tumor risk classifications with higher accuracy than the actual PCA3. Remarkably, the best performing mixed signature did not include PCA3, consistent with the low potency of this biomarker in detection of aggressive tumors. Instead, both mixed and new-lncRNA signatures contained contigs embedded into putative novel lncRNA genes. We strongly believe that these signatures can complement the existing clinical tests as lncRNA-based PCA3 (Progensa) or mostly mRNA-based Decipher to improve the accuracy of tumor stratification and clinical decisions for better patient care (Alford et al, 2017). Still, in this study, to compute signature coefficients, sample information (normal or tumor) was used because the extended *Selection* and independent *Validation Sets* used two different technologies for RNA measurements. This precluded us from calculating an objective signature performance. An additional cohort using the same NanoString technology as the *Selection Set* should now be tested to explore the clinical potential of the obtained signature.

In addition to the clinical value, functions of the newly discovered lncRNA variants embedding DE-kupl contigs will be important to explore. Foremost, proper assignment of contigs to stand-alone transcripts is required, and this task can be accomplished computationally through ab initio discovery and assembly of novel transcripts as demonstrated here by HoLdUp or other assemblers, and then through experimental validation at the transcript-specific or transcriptomic level. In the latter case, high-throughput RACE (rapid amplification of cDNA ends) or long-read RNA-seq approaches can be useful. Among others, detailed examination of newly discovered contigs revealed a genomic locus on chromosome 19 transcribed in PCa specimens in both directions into the GENCODE-annotated AC011523.2 lncRNA and a novel, antisense transcript embedding the P23 contig (ctg_29077). Located between *KLK15* and the PSA encoding *KLK3* genes, this region makes part of a super-enhancer annotated in several PCa cell lines (Jiang et al, 2019). Moreover, bidirectionally produced enhancer RNAs from this locus have been shown to regulate the expression of neighboring *KLK3* and *KLK2* genes through Med1-dependent chromatin looping in several PCa cell lines (Hsieh et al, 2014). Presence of the P23 contig within the mixed and new-lncRNA signatures supports, in addition to the clinical potency, possible regulatory functions of the RNA contigs inferred by DE-kupl. More globally, most DE-kupl contigs within co-transcribed sense–antisense pairs were annotated as super-enhancers in prostate tissues and cell lines or other biosamples, for example, P15 (ctg_512), P7 (ctg_117356), and P4 (ctg_63866) (Jiang et al, 2019). In most cases, their function in gene expression regulation and chromatin configuration has not yet been investigated and experimentally validated, but it is tempting to speculate that defined sense-antisense transcripts may influence a super-enhancer activity and, consequently, may fine-tune the expression of neighboring genes.

In this work, we propose DE-kupl as a tool for discovery of novel disease-associated transcriptomic variations, which can be further explored for biological and clinical relevance. As a pilot project, we oriented the pipeline towards the discovery of novel lncRNAs, but using proper masking and filtering criteria defined by the investigator, other variant transcripts, including single nucleotide variations, novel splice events, gene fusions, circular RNAs, or exogenous viral RNAs, could be probed. The workflow can be applied to any RNA-seq datasets of any clinical origin (tissue, blood, and urine) to generate a probe panel that may be implemented as a multiplex platform for simultaneous detection of RNAs in clinical samples. Moreover, different experimental contrasts (normal versus pathology, low- versus high-risk grade, chemoresistant versus sensitive, etc.) will define the biomarker usage in diagnosis, prognosis, or other clinical applications, hence providing clinicians and researchers with a simple and highly sensitive tool for genomic and personalized medicine.

Materials and Methods

Tissue samples

Tumor and normal biopsy specimens were retrospectively collected from prostate cancer patients who provided informed consent and were approved for distribution by the Henri Mondor institutional board (PAIR cohort). Tumor classification in low-, intermediate-, and high-risk prognosis was performed according to Gleason and TNM scores and regardless PSA values (Table S1 and Fig S6B).

RNA extraction, quantification, and cDNA library production

Total RNA was extracted using the TRizol reagent (Thermo Fisher Scientific), according to the manufacturer's procedure, quantified, and quality-controlled using a 2100 Bioanalyzer (Agilent). RNA samples with RNA Integrity Number (RIN) above six were depleted for ribosomal RNA and converted into cDNA library using a TruSeq Stranded Total Library Preparation kit (Illumina). cDNA libraries were normalized using an Illumina duplex-specific Nuclease protocol before a paired-end sequencing on HiSeq 2500 (Illumina). At least 20× coverage per sample was considered as minimum of unique sequences for further data analysis.

RNA-seq data

Raw paired-end strand-specific RNA-seq data were generated by our laboratory from ribo-depleted total RNA samples of prostate tissues (8 normal and 16 tumor specimens, Table S1) and can be retrieved from the gene omnibus portal, accession number GSE115414. TCGA prostate cancer poly(A)–selected RNA-seq and corresponding clinical data were obtained from publicly available TCGA dataset (http://cancergenome.nih.gov), 557 inputs in total (52 normal and 505 tumors of high- [N = 240], intermediate- [N = 128], and low-risk [N = 132] groups). Among them, 369 patients showed no tumor recurrence, 108 presented a new tumor event (Table S10).

Computational workflow for k-mer contigs discovery from total stranded RNA-seq dataset

DE-kupl run was performed from (June 2017) with parameters ctg_length 31, min_recurrence 6, min_recurrence_abundance 5, pvalue_threshold 0.05, lib_type stranded, diff_method DESeq2. K-mer masking was performed against the GENCODE v24 annotation. DE-kupl analysis of the 8 against 16 PAIR RNA-seq prostate libraries yielded 124,809 DE contigs, in total. Contigs were annotated by alignment on the hg19 human genome assembly using the DE-kupl *annotate* procedure. We further selected contigs of size above 200 nucleotides and classified them into four categories (contiguous, repeat, spliced, and unmapped) based on their location and mapping features (Table S2).

Computational workflow for reference-based ab initio transcripts assembly from total stranded RNA-seq dataset (HoLdUP)

The human genome version hg19 and the GENCODE v14 annotation were used in this study. First, we performed a quality control of all sequencing data by FastQC Babraham Bioinformatics software. Reads were mapped using TopHat 2.0.4, allowing three mismatches and requesting uniquely mapped reads, which were further assembled using the BedTools suite. Overlapping contigs from all libraries were merged, and only contigs supported by at least 10 reads in either library were further assembled in segments if mapped in the same strand and separated by less than 100 nucleotides. We compared the segments with the GENCODE v14 annotation to extract antisense and intergenic TUs longer than 200 nucleotides. To classify lncRNAs, we applied the following criteria: (i) an expression level above 0.2 quartile of mRNA expression in at least one condition per tissue (Class 2); (ii) within this class, all TUs containing at least one TopHat-identified exon–exon junction and at least one spliced EST from UCSC mapped contigs were assigned to Class 1. The whole catalog, the R code, and Data Tables can be downloaded from https://github.com/MorillonLab/HoLDuP_pipeline.

Overlap between GENCODE, MiTranscriptome, DE-kupl, and HoLdUp catalogues

Intersection between transcripts was counted only in the case of 50% overlap of nucleotide sequence between genomic coordinates of each fragment.

Differential expression analysis

Read counting was performed on the compiled annotation (GENCODE v27, HoLdUp Class 1 and Class 2) for each sample, using *featureCounts* 1.6.0 with the following parameters: -F "SAF" -p -s 2 -O and the *DESeq R* package (Liao et al, 2014; Love et al, 2014). Only RNAs with adjusted *P*-value below 0.01 were retained as differentially expressed to constitute the prostate tumor signature (Tables S3 and S4). Gene expression counts were normalized using the DESeq2 median of ratio (Anders & Huber, 2010). Scripts are available at https://github.com/MorillonLab/Prostate_additional_scripts.

NanoString nCounter expression assay

100 ng of total RNA was used for direct digital detection of 29 target transcripts: six housekeeping genes (*RPL11*, *GAPDH*, *NOL7*, *GPATCH3*, *ZNF2*, and *ZNF346*), 23 contigs and the one known PCa-associated lncRNA, PCA3. Each target gene of interest was detected in RNA samples of 144 specimens (9 normal and 135 tumor) of the PAIR cohort (Table S6) on NanoString nCounter V2 using reporter and capture probes of 35- to 50-nucleotide targeting sequences listed in Table S4. Data was normalized through the use of NanoString's intrinsic negative and positive controls according to the normalization approach of the nSolver analysis software (https://www.nanostring.com/products/analysis-software/nsolver) and then contig expression was calculated relative to the average signal of three housekeeping genes (*GPATCH3*, *ZNF2*, and *ZNF346*). Raw and normalized data for each specimen, and mean and fold change expression in normal against tumor samples are presented in Tables S7 and S8.

Contig expression measurements in TCGA-PRAD datasets

DE-kupl provides representative k-mers for each differentially expressed contig. We converted the TCGA-PRAD FASTQ files to k-mer counts using *Jellyfish count* and counted representative k-mers in each Jellyfish count file using the *Jellyfish query* command (Marçais & Kingsford, 2011). Counts were normalized by total number of reads in corresponding libraries. To determine whether counts of DE-kupl derived representative k-mer were a reliable proxy for evaluating contig expression, we compared representative k-mer counts to average counts from k-mers sampled along each contig. All individual counts were obtained using *Jellyfish Dump* files produced for each TCGA-PRAD library. Sampling was performed as follows: (i) we extracted all k-mers from the contig that were unique in the Ensembl human v91 transcript reference, and (ii) from this list, we sampled 10 regularly spaced k-mers, starting from the first 10% and ending in the last 10% of the list. This sampling procedure was repeated four times for each contig. For the whole TCGA library and each contig, the 10 k-mer counts obtained by Jellyfish were averaged, yielding one average count per sample per library Table S13. Pearson correlation analysis for two DE-kupl contigs P1 and P16 are shown in Fig S8A and B. Jellyfish commands can be retrieved from https://github.com/MorillonLab/Prostate-kmer-signatures.

RNA-seq data visualization

RNA-seq reads profiling along a locus of interest was performed using our in-house R script VING using one "normal" and one

"tumor" RNA-seq subsets build by random sampling of 10% of reads from each raw data sample (Descrimes et al, 2015). The normal samples were assigned to the group "controls" and the tumor specimens–to the group "cases," with the assumption that the "cases" should have higher values than "controls."

Unsupervised clustering of prostate specimens

Specimens were ranked based on the Log10(norm.counts) levels of contigs assessed by the NanoString nCounter assay using a ComplexHeatmap R-package (Gu et al, 2016). Scripts are available from GitHub: https://github.com/MorillonLab/Prostate_additional_scripts.

Variable selection using the LASSO penalized logistic regression and external validation of signatures

Signature inference was performed in R using the normalized *Selection Set* (23 probes in 144 observations) as a variable selection dataset and contigs counts table of the *Validation Set* (23 probes in 557 observations) as an external validation dataset (R Core Team). First, we performed penalized logistic regression using the *glmnet* R package to select probes predicting the tumor status on the *Selection Set* upsampled to correct the imbalance class distribution (9 normal versus 135 tumor specimens) (Friedman et al, 2010). Selection was performed using all probes (signature_mixed including PCA3) or using only new-lncRNA contigs only (signature_new-lnc) (Fig S7). Second, we built predictors using the boosted logistic regression from the *caTools* and *caret* packages (Kuhn, 2008; Tuszynski, 2008). Note that the final gene subsets (signatures) do not have coefficients computed on the *Selection Set* over the *Validation Set* because in contrast to NanoString, the TCGA-PRAD RNA-seq datasets are poly(A)–selected and unstranded. To build the ROC curves, we sampled 100 datasets in two, for training (70%) and testing (30%) preserving the relative ratio of labels in each. We used boosted logistic regression with upsampling, setting the number of boosting iterations to 100 and using leave-one-out cross validation scheme on the training set. After training, we evaluated the predictor on the testing set and repeated the procedure for each one of the 100 training and testing sets described above to obtain an average ROC curve, mean and SD for AUC scores. Contig expression counts in the *Validation Set* (TCGA-PRAD) were obtained as described above using the DE-kupl derived representative k-mer for each contig. Quantifications based on 10 randomly sampled k-mers per contig did not alter predictive performance (Fig S8C). To build a classifier based on the conventional WGC procedure, we used DESeq2 across the GENCODE annotation on the *Discovery Set* and kept only up-regulated genes with adjusted *P*-value lower than 0.05 and Log2FC higher than 2. To perform gene selection on the *Discovery Set*, we used LASSO penalized logistic regression combined with stability selection. Only genes with probability above 0.5 on 2,000 up-regulated samples from the initial dataset were retained. The remaining genes were then used to build ROC curves and compute the mean and SD of the AUC on the *Validation Set* as described above for the DE-kupl-derived representative k-mers. The results file, R codes, and data tables are provided through the GitHub repository: https://github.com/MorillonLab/Prostate-kmer-signatures.

Data access

Raw paired-end strand-specific RNA-seq data can be retrieved from the gene omnibus portal, accession number GSE115414. TCGA prostate cancer poly(A)–selected RNA-seq and corresponding clinical data can be obtained from TCGA portal (https://www.cancer.gov/tcga).

Supplementary Information

Supplementary Information is available at https://doi.org/10.26508/lsa.201900449.

Acknowledgements

We deeply thank Dominika Foretek, Maxime Wery, and Alexandre Serero for editorial suggestions; Camille Gautier, Claire Bertrand, and Anna Almeida (Morillon lab, Institut Curie) and Sylvain Baulande for RNA-seq (Next Generation Sequencing platform, Institut Curie); and Cedric Saule and Jeremy Le Coz for the DE-kupl run (Gautheret lab, I2BC). The Cancer Genome Atlas RNA-seq data for prostate adenocarcinoma (PRAD) were downloaded from the dbGaP Web site under authorization granted to D Gautheret (project #13359). Funding: Constitution of the prostate cancer cohort was performed with the financial support from the INCa-Ligue-ARC PAIR program to Y Allory and A Londoño-Vallejo. The Genomics platform and NanoString technology of Institut Curie were set up with the support of Agence Nationale de la Recherche (LabEx and EquipEx: ANR-10-IDEX-0001-02 PSL, ANR-11-LBX-0044), Insitut National du Cancer (INCa-DGOS-4654, SIRIC11-002). RNA-seq efforts were supported by a grant from the ICGex program at Institut Curie to A Londoño-Vallejo and A Morillon and benefited from the facilities and expertise of the Next Generation Sequencing platform of Institut Curie, supported by Agence Nationale de la Recherche (ANR-10-EQPX-03, ANR10-INBS-09-08) and Cancerop̂le Ile-de-France. M Descrimes, M Gabriel, A Morillon, M Pinskaya, and Z Saci were supported by Agence Nationale de la Recherche (DNA-Life) and the European Research Council (ERC-consolidator DARK-616180-ERC-2014) attributed to A Morillon; D Gautheret and HTN Nguyen were supported by ITMO Cancer–Systems Biology (bio2014-04) and Agence Nationale de la Recherche "France Génomique" (ANR-10-INBS-0009) attributed to D Gautheret.

Authors Contributions

M Pinskaya: supervision, validation, investigation, methodology, project administration, and writing—original draft, review, and editing.
Z Saci: software and formal analysis.
M Gallopin: formal analysis, supervision, and writing—original draft, review, and editing.
M Gabriel: software and formal analysis.
HTN Nguyen: formal analysis.
V Firlej: resources and data curation.
M Descrimes: formal analysis.
A Rapinat: investigation.
D Gentien: investigation.
A De la Taille: resources and data curation.
A Londoño-Vallejo: data curation.
Y Allory: data curation.
D Gautheret: conceptualization, formal analysis, supervision, funding acquisition, project administration, and writing—original draft, review, and editing.

A Morillon: conceptualization, supervision, funding acquisition, project administration, and writing—original draft, review, and editing.

Conflict of Interest Statement

The authors declare that they have no conflict of interest.

References

Alford AV, Brito JM, Yadav KK, Yadav SS, Tewari AK, Renzulli J (2017) The use of biomarkers in prostate cancer screening and treatment. *Rev Urol* 19: 221–234. doi:10.3909/riu0772

Alshalalfa M, Verhaegh GW, Gibb EA, Santiago-Jiménez M, Erho N, Jordan J, Yousefi K, Lam LLC, Kolisnik T, Chelissery J, et al (2017) Low PCA3 expression is a marker of poor differentiation in localized prostate tumors: Exploratory analysis from 12,076 patients. *Oncotarget* 8: 50804–50813. doi:10.18632/oncotarget.15133

Anders S, Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol* 11: R106. doi:10.1186/gb-2010-11-10-r106

Audoux J, Philippe N, Chikhi R, Salson M, Gallopin M, Gabriel M, Le Coz J, Drouineau E, Commes T, Gautheret D (2017) DE-kupl: Exhaustive capture of biological variation in RNA-seq data through k-mer decomposition. *Genome Biol* 18: 243. doi:10.1186/s13059-017-1372-2

Carlsson S, Vickers AJ, Roobol M, Eastham J, Scardino P, Lilja H, Hugosson J (2012) Prostate cancer screening: Facts, statistics, and interpretation in response to the US preventive services task force review. *J Clin Oncol* 30: 2581–2584. doi:10.1200/jco.2011.40.4327

D'Amico AV, Whittington R, Kaplan I, Beard C, Schultz D, Malkowicz SB, Wein A, Tomaszewski JE, Coleman CN (1998) Calculated prostate carcinoma volume: The optimal predictor of 3-year prostate specific antigen (PSA) failure free survival after surgery or radiation therapy of patients with pretreatment PSA levels of 4-20 nanograms per milliliter. *Cancer* 82: 334–341. doi:10.1002/(sici)1097-0142(19980115)82:2<342::aid-cncr14>3.0.co;2-z

Davila JI, Fadra NM, Wang X, McDonald AM, Nair AA, Crusan BR, Wu X, Blommel JH, Jen J, Rumilla KM, et al (2016) Impact of RNA degradation on fusion detection by RNA-seq. *BMC Genomics* 17: 814. doi:10.1186/s12864-016-3161-9

de Kok JB, Verhaegh GW, Roelofs RW, Hessels D, Kiemeney LA, Aalders TW, Swinkels DW, Schalken JA (2002) DD3(PCA3), a very sensitive and specific marker to detect prostate tumors. *Cancer Res* 62: 2695–2698.

Deng J, Tang J, Wang G, Zhu Y-S (2017) Long non-coding RNA as potential biomarker for prostate cancer: Is it making a difference? *Int J Environ Res Public Health* 14: E270. doi:10.3390/ijerph14030270

Descrimes M, Ben Zouari Y, Wery M, Legendre R, Gautheret D, Morillon A (2015) VING: A software for visualization of deep sequencing signals. *BMC Res Notes* 8: 419. doi:10.1186/s13104-015-1404-5

Deveson IW, Brunck ME, Blackburn J, Tseng E, Hon T, Clark TA, Clark MB, Crawford J, Dinger ME, Nielsen LK, et al (2018) Universal alternative splicing of noncoding exons. *Cell Syst* 6: 245–255.e5. doi:10.1016/j.cels.2017.12.005

Du Z, Fei T, Verhaak RGW, Su Z, Zhang Y, Brown M, Chen Y, Liu XS (2013) Integrative genomic analyses reveal clinically relevant long noncoding RNAs in human cancer. *Nat Struct Mol Biol* 20: 908–913. doi:10.1038/nsmb.2591

Fenstermaker M, Mendhiratta N, Bjurlin MA, Meng X, Rosenkrantz AB, Huang R, Deng F-M, Zhou M, Huang WC, Lepor H, et al (2017) Risk stratification by urinary prostate cancer gene 3 testing before magnetic resonance imaging-ultrasound fusion-targeted prostate biopsy among men with no history of biopsy. *Urology* 99: 174–179. doi:10.1016/j.urology.2016.08.022

Friedman J, Hastie T, Tibshirani R (2010) Regularization paths for generalized linear models via coordinate descent. *J Stat Softw* 33: 1–22. doi:10.18637/jss.v033.i01

Galasso F, Giannella R, Bruni P, Giulivo R, Barbini VR, Disanto V, Leonardi R, Pansadoro V, Sepe G (2010) PCA3: A new tool to diagnose prostate cancer (PCa) and a guidance in biopsy decisions. Preliminary report of the UrOP study. *Arch Ital Urol Androl* 82: 5–9.

Ghosh D, Chinnaiyan AM (2005) Classification and selection of biomarkers in genomic data using LASSO. *J Biomed Biotechnol* 2005: 147–154. doi:10.1155/jbb.2005.147

Groskopf J, Aubin SMJ, Deras IL, Blase A, Bodrug S, Clark C, Brentano S, Mathis J, Pham J, Meyer T, et al (2006) APTIMA PCA3 molecular urine test: Development of a method to aid in the diagnosis of prostate cancer. *Clin Chem* 52: 1089–1095. doi:10.1373/clinchem.2005.063289

Gu Z, Eils R, Schlesner M (2016) Complex heatmaps reveal patterns and correlations in multidimensional genomic data. *Bioinformatics* 32: 2847–2849. doi:10.1093/bioinformatics/btw313

Hayer KE, Pizarro A, Lahens NF, Hogenesch JB, Grant GR (2015) Benchmark analysis of algorithms for determining and quantifying full-length mRNA splice forms from RNA-seq data. *Bioinformatics* 31: 3938–3945. doi:10.1093/bioinformatics/btv488

Hsieh C-L, Fei T, Chen Y, Li T, Gao Y, Wang X, Sun T, Sweeney CJ, Lee G-SM, Chen S, et al (2014) Enhancer RNAs participate in androgen receptor-driven looping that selectively enhances gene activation. *Proc Natl Acad Sci U S A* 111: 7319–7324. doi:10.1073/pnas.1324151111

Iyer MK, Niknafs YS, Malik R, Singhal U, Sahu A, Hosono Y, Barrette TR, Prensner JR, Evans JR, Zhao S, et al (2015) The landscape of long noncoding RNAs in the human transcriptome. *Nat Genet* 47: 199–208. doi:10.1038/ng.3192

Jarroux J, Morillon A, Pinskaya M (2017) History, discovery, and classification of lncRNAs. *Adv Exp Med Biol* 1008: 1–46. doi:10.1007/978-981-10-5203-3_1

Jiang Y, Qian F, Bai X, Liu Y, Wang Q, Ai B, Han X, Shi S, Zhang J, Li X, et al (2019) SEdb: A comprehensive human super-enhancer database. *Nucleic Acids Res* 47: D235–D243. doi:10.1093/nar/gky1025

Kim HL, Li P, Huang H-C, Deheshi S, Marti T, Knudsen B, Abou-Ouf H, Alam R, Lotan TL, Lam LLC, et al (2019) Validation of the Decipher Test for predicting adverse pathology in candidates for prostate cancer active surveillance. *Prostate Cancer Prostatic Dis* 22: 399–405. doi:10.1038/s41391-018-0101-6

Kuhn M (2008) Building predictive models in R using the caret package. *J Stat Softw* 28. http://www.jstatsoft.org/v28/i05/

Kukurba KR, Montgomery SB (2015) RNA sequencing and analysis. *Cold Spring Harb Protoc* 2015: 951–969. doi:10.1101/pdb.top084970

Leucci E (2018) Cancer development and therapy resistance: Spotlights on the dark side of the genome. *Pharmacol Ther* 189: 22–30. doi:10.1016/j.pharmthera.2018.04.001

Liao Y, Smyth GK, Shi W (2014) featureCounts: An efficient general purpose program for assigning sequence reads to genomic features. *Bioinformatics* 30: 923–930. doi:10.1093/bioinformatics/btt656

Loeb S, Bjurlin MA, Nicholson J, Tammela TL, Penson DF, Carter HB, Carroll P, Etzioni R (2014) Overdiagnosis and overtreatment of prostate cancer. *Eur Urol* 65: 1046–1055. doi:10.1016/j.eururo.2013.12.062

Loeb S, Partin AW (2011) Review of the literature: PCA3 for prostate cancer risk assessment and prognostication. *Rev Urol* 13: e191–e195.

Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. *Genome Biol* 15: 550. doi:10.1186/s13059-014-0550-8

Marçais G, Kingsford C (2011) A fast, lock-free approach for efficient parallel counting of occurrences of k-mers. *Bioinformatics* 27: 764–770. doi:10.1093/bioinformatics/btr011

Meinshausen N, Bühlmann P (2010) Stability selection: Stability selection. *J R Stat Soc Ser B Stat Methodol* 72: 417–473. doi:10.1111/j.1467-9868.2010.00740.x

Morillon A, Gautheret D (2019) Bridging the gap between reference and real transcriptomes. *Genome Biol* 20: 112. doi:10.1186/s13059-019-1710-7

Prensner JR, Chen W, Iyer MK, Cao Q, Ma T, Han S, Sahu A, Malik R, Wilder-Romans K, Navone N, et al (2014) PCAT-1, a long noncoding RNA, regulates BRCA2 and controls homologous recombination in cancer. *Cancer Res* 74: 1651–1660. doi:10.1158/0008-5472.can-13-3159

Prensner JR, Iyer MK, Sahu A, Asangani IA, Cao Q, Patel L, Vergara IA, Davicioni E, Erho N, Ghadessi M, et al (2013) The long noncoding RNA SChLAP1 promotes aggressive prostate cancer and antagonizes the SWI/SNF complex. *Nat Genet* 45: 1392–1398. doi:10.1038/ng.2771

Quinn JJ, Chang HY (2015) Unique features of long non-coding RNA biogenesis and function. *Nat Rev Genet* 17: 47–62. doi:10.1038/nrg.2015.10

R Core Team (n.d.) *R: A language and environment for statistical computing*. R Foundation for Statistical Computing. http://www.R-project.org/

Salameh A, Lee AK, Cardó-Vila M, Nunes DN, Efstathiou E, Staquicini FI, Dobroff AS, Marchiò S, Navone NM, Hosoya H, et al (2015) PRUNE2 is a human prostate cancer suppressor regulated by the intronic long noncoding RNA PCA3. *Proc Natl Acad Sci U S A* 112: 8403–8408. doi:10.1073/pnas.1507882112

Schmitt AM, Chang HY (2016) Long noncoding RNAs in cancer pathways. *Cancer Cell* 29: 452–463. doi:10.1016/j.ccell.2016.03.010

Silva A, Bullock M, Calin G (2015) The clinical relevance of long non-coding RNAs in cancer. *Cancers* 7: 2169–2182. doi:10.3390/cancers7040884

Tuszynski J (2008) *caTools: Tools: moving window statistics, GIF, Base64, ROC AUC, etc.* https://CRAN.R-project.org/package=caTools

Uszczynska-Ratajczak B, Lagarde J, Frankish A, Guigó R, Johnson R (2018) Towards a complete map of the human long non-coding RNA transcriptome. *Nat Rev Genet* 19: 535–548. doi:10.1038/s41576-018-0017-y

Van Grembergen O, Bizet M, de Bony EJ, Calonne E, Putmans P, Brohée S, Olsen C, Guo M, Bontempi G, Sotiriou C, et al (2016) Portraying breast cancers with long noncoding RNAs. *Sci Adv* 2: e1600220. doi:10.1126/sciadv.1600220

Wang F, Ren S, Chen R, Lu J, Shi X, Zhu Y, Zhang W, Jing T, Zhang C, Shen J, et al (2014) Development and prospective multicenter evaluation of the long noncoding RNA MALAT-1 as a diagnostic urinary biomarker for prostate cancer. *Oncotarget* 5: 11091–11102. doi:10.18632/oncotarget.2691

Wang Y-H, Ji J, Wang B-C, Chen H, Yang Z-H, Wang K, Luo C-L, Zhang W-W, Wang F-B, Zhang X-L (2018) Tumor-derived exosomal long noncoding RNAs as promising diagnostic biomarkers for prostate cancer. *Cell Physiol Biochem* 46: 532–545. doi:10.1159/000488620

Willard SS, Koochekpour S (2012) Regulators of gene expression as biomarkers for prostate cancer. *Am J Cancer Res* 2: 620–657.

Zhao S, Zhang Y, Gamini R, Zhang B, von Schack D (2018) Evaluation of two main RNA-seq approaches for gene quantification in clinical RNA sequencing: polyA+ selection versus rRNA depletion. *Sci Rep* 8: 4781. doi:10.1038/s41598-018-23226-4

Zhao W, He X, Hoadley KA, Parker JS, Hayes D, Perou CM (2014) Comparison of RNA-Seq by poly (A) capture, ribosomal RNA depletion, and DNA microarray for expression profiling. *BMC Genomics* 15: 419. doi:10.1186/1471-2164-15-419

Chapter 7

A Comparative Analysis of Reference-Free and Conventional Transcriptome Signatures for Prostate Cancer Prognosis

Ha TN Nguyen[1], Haoliang Xue[1], Virginie Firlej[2], Yann Ponty[3], Mélina Gallopin[1], Daniel Gautheret[1*]

* Correspondence: daniel.gautheret@universite-paris-saclay.fr

[1]Institute for Integrative Biology of the Cell, CEA, CNRS, Université Paris-Saclay, Gif-Sur-Yvette, France.

[2]Université Paris Est Creteil, TRePCa, Creteil, France.

[3]LIX UMR 7161, Ecole Polytechnique, Institut Polytechnique de Paris, France.

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 **l**ong **n**on-**c**oding **RNA**s 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

7.1 Introduction

The outcome of human cancer can be predicted in part through gene expression profiling (Perou et al., 2000; Singh et al., 2002; van 't Veer et al., 2002). Outcome prediction is particularly important in **p**rostate **ca**ncer (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 the **T**umour, **N**ode, **M**etastasis (TNM) grade that evaluates tumor extent and propagation. Gleason, TNM and PSA levels can be combined into a low, medium or high risk status (D'Amico et al., 1998). Several studies used gene expression profiles to derive predictors of Gleason score or risk (Bibikova et al., 2007; Penney et al., 2011; Sinnott et al., 2017; Jhun et al., 2017). 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 (**B**io**c**hemical **R**ecurrence (BCR)) (Latil et al., 2003; Long et al., 2014; Ren et al., 2018; Sinha et al., 2019) or upon direct clinical observation (Erho et al., 2013; Karnes et al., 2013; Klein et al., 2015; Shahabi et al., 2016). 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.

Gene expression profiling of prostate biopsies is performed either using DNA microarrays (Erho et al., 2013; Karnes et al., 2013; Klein et al., 2015; Shahabi et al., 2016) or high throughput **RNA seq**uencing (RNA-seq) (Bibikova et al., 2007; Penney et al., 2011; Sinnott et al., 2017; Jhun et al., 2017). 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 (Morillon and Gautheret, 2019). Therefore one may expect that prognostic models incorporating non-reference RNAs may carry substantial benefits.

Our group (Audoux et al., 2017; Pinskaya et al., 2019) and others (Thomas et al., 2019) 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 sub-sequence whose increased abundance is associated to a given clinical label. This may include novel splice variants, **l**ong **n**on-**c**oding **RNA**s or RNAs from repeated retroelements (Audoux et al., 2017; Pinskaya et al., 2019) 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 (Michiels et al., 2005; Ein-Dor et al., 2006; Michiels et al., 2007; Venet et al., 2011) are expected to worsen with k-mer count matrices. The second challenge is related to the transfer of a k-mer signatures 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.

7.2 Materials and Methods

7.2.1 Data acquisition and outcome labelling

We used tumor samples from TCGA-PRAD (Abeshouse et al. (2015), N=505) for signature discovery and from ICGC-PRAD (Fraser et al. (2017), N=284) and Stelloo et al (Stelloo et al. (2018), N=91) for independent validation. All three datasets 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 7.1).

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 Stelloo et al. (2018) RNA-seq files ("Porto" cohort) were retrieved from GEO, under accession GSE120741. Clinical information was retrieved from Liu et al. (2018a) for TCGA-PRAD, from Fraser et al. (2017) for ICGC-PRAD and from sample metadata of GEO accession GSE120741 for Stelloo et al. (2018).

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.

Table 7.1: 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	284	40	23	49	7
STELLOO	Poly(A)+ stranded 1x65nt	20M	91			43	48

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 (Pinskaya et al., 2019). Tumors with Gleason score 6/7 (3+4) and TNM grade pT1/2 were classified as low risk. Tumors with Gleason score 8/9 and/or TNM grade pT3b/4 were defined as high-risk. 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.

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 7.2). BCR information was obtained from table S1 of Liu et al. (2018a) for TCGA-PRAD and from table S1 (PFS field) of Fraser et al. (2017) for ICGC-PRAD. Precomputed relapse data for Stelloo et al. was taken from SRA accession PRJNA494345.

Table 7.2: 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

7.2.2 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 (Figure 7.1). 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 (Ambroise and McLachlan, 2002). 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 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.

7.2.3 Gene and k-mer count matrices

`DEkupl-run` (Audoux et al., 2017) was used to produce gene and k-mer count matrices for each dataset. `DEkupl-run` converts FASTQ files to k-mer counts using `Jellyfish` (Marçais and Kingsford, 2011), 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 Supplementary Table S1. K-mer size was set to

31 as commonly adopted for human transcriptome applications (Bray et al., 2016; Audoux et al., 2017). 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 (Bray et al., 2016) 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.

7.2.4 Reduction of k-mer matrix via contig extension

k-mer occurence tables were converted into contig occurence tables using an extension procedure similar to that described in Audoux et al. (2017). 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. (2017) by considering sample count compatibility between contigs, as shown in Figure 7.2. Sample count compatibility is measured by the **M**ean **A**bsolute **C**ontrast (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. The algorithm is implemented in C++ to be published (`https://github.com/i2bc/PCa-gene-based_vs_gene-free/tree/master/KaMRaT`)

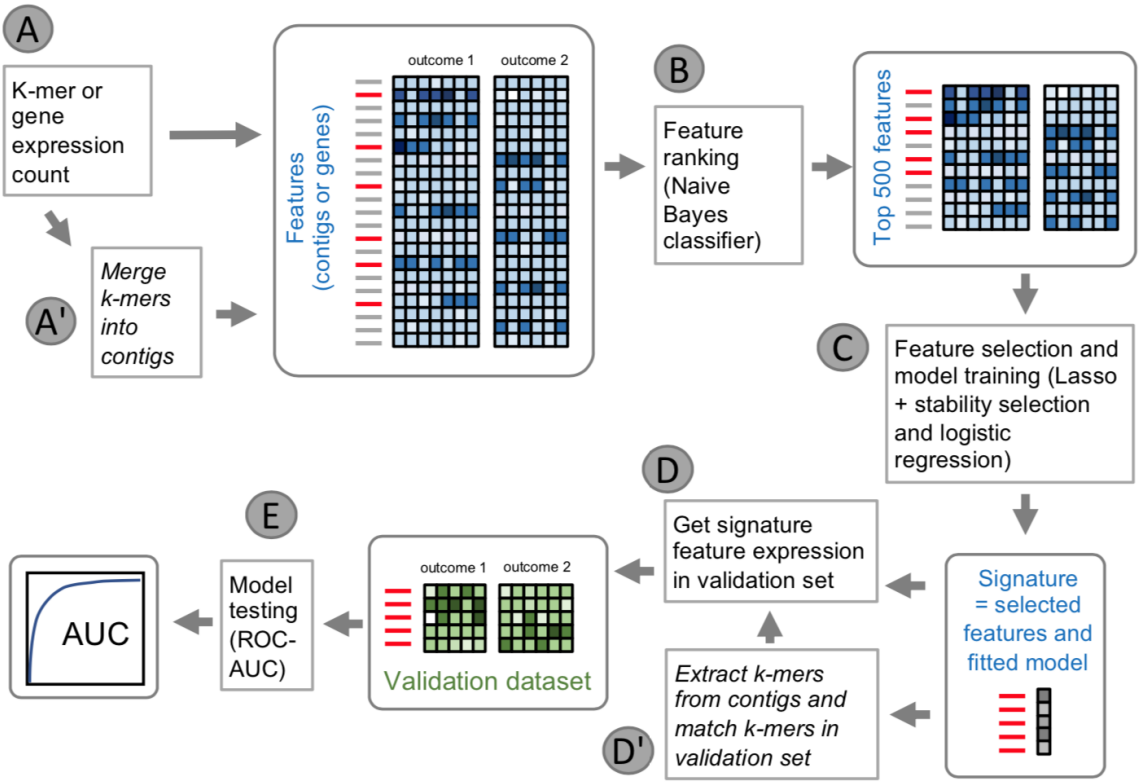


Figure 7.1: Uniform procedure for signature inference based on k-mer or gene expression. **A**. The discovery matrix is built from normalized k-mer counts or gene expression counts. Samples are labelled by their outcome (risk or relapse) status. Normalization is performed as count per billion for k-mers or count per million for genes. **B**. Features are ranked according to their F1-score computed by **c**ross-**v**alidation using a `Naïve Bayes` classifier. 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 **A**re **U**nder the ROC **C**urve (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.

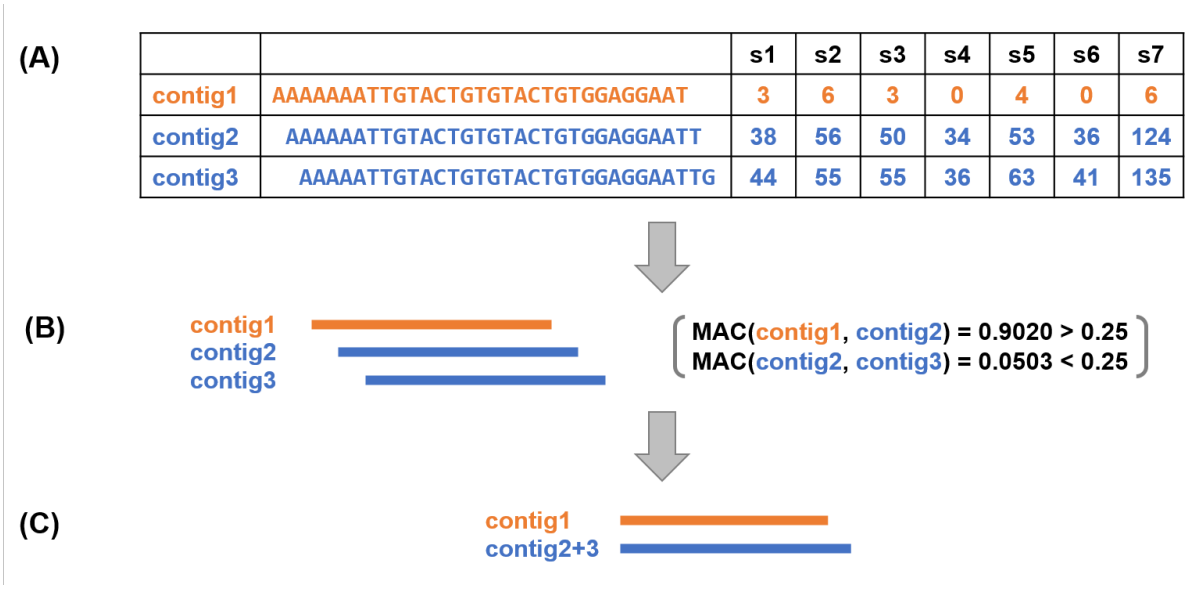


Figure 7.2: Merging procedure of 3 example contigs: **A**. Count table of contigs in samples. Both pairs $(contig1, contig2)$ and $(contig2, contig3)$ have good overlaps shifting by only one nucleotide, but the sample count vectors of $contig1$ and $contig2$ are not compatible. **B**. Merging intervention considering sample count compatibility between contigs. The **M**ean **A**bsolute **C**ontrast (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.

7.2.5 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.

7.2.6 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 performed prediction of status (risk/relapse) using a `Bayesian` 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 **c**ross-**v**alidation (CV) ($F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$, 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 CV returned an undefined value, F_1 score was set to 0 (the worst). The average F_1 score was used to rank features. The `Bayesian` classifier implementation was taken from the `MLPack` library (Curtin et al., 2018). The C++ code to perform feature ranking is available at `https://github.com/i2bc/PCa-gene-based_vs_gene-free/tree/master/KaMRaT`.

7.2.7 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 (Friedman et al., 2010). We implemented stability selection as described in Meinshausen and Bühlmann (2010): 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 using

a 10-fold **c**ross-**v**alidation scheme, repeated 20 times, as implemented in the `caret` package. 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 (Rainer, 2017). R scripts to perform the feature selection, model fitting and evaluation on the discovery and validation sets are available at: `https://github.com/i2bc/PCa-gene-based_vs_gene-free`.

7.2.8 Matching signature contigs in the validation cohort

To measure contig expression in the validation cohort we implemented the procedure schematized in Figure 7.3. 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. Step 1 is implemented in `C++` at: `https://github.com/i2bc/PCa-gene-based_vs_gene-free/tree/master/kmerFilter`, step 2 is implemented in `R` at: `https://github.com/i2bc/PCa-gene-based_vs_gene-free/blob/master/infer_gene-free_risk_signature.R`.

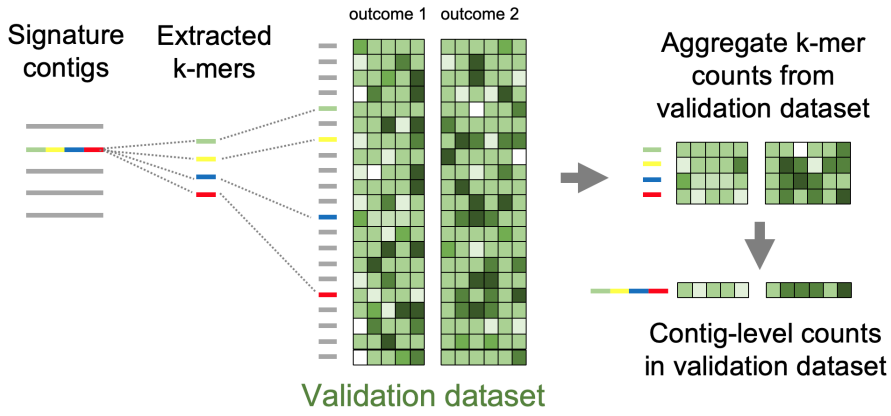


Figure 7.3: Procedure for inferring signature contig expression in an independent validation dataset. The colored contig from the signature is quantified in the validation cohort by extracting all its constituent k-mers and retrieving the corresponding k-mer counts from validation k-mer count matrix. 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.

7.3 Results

7.3.1 A reference-free risk signature for prostate cancer

We first applied the gene-free and gene-based signature discovery procedures detailed in Section 7.2.2 to infer PCa risk signatures. The k-mer table for 374 TCGA-PRAD risk-labelled samples 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 (Table 7.3). Contig sizes (mean=49nt, median=34nt, Table 7.4) were small relatively to a typical human RNA, which is characteristic of the adopted contig extension procedure (Audoux et al., 2017) (see Section 7.2.4).

Table 7.3: Result of filtering procedure on the k-mer and gene matrices for risk analysis

	Initial matrix	Low expression filter	k-mer merging	Naive Bayes ranking	Feature Selection by Lasso LR	Validation
k-mers or contigs	(not generated)	94,539,338 k-mers	5,234,940 contigs	500 contigs	26 contigs (1,444 k-mers)	21 contigs (1,404 k-mers)
genes	60,554	38,382	NA	500	14	14

Table 7.4: Contig sizes (Risk model)

	After k-mer merging	After Naive Bayes ranking
mean contig size (nt)	49.1	189
median contig size (nt)	34	61

The 5.2M contig matrix and the 38k gene expression matrix were submitted to screening using univariate `Naïve Bayes` classification and the top scoring 500 features were retained for feature selection (Section 7.2.6). Interestingly, the 500 top scoring contigs were significantly longer than prior to selection (median 61nt vs. 34nt, Table 7.4), 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 (Table 7.3, Figure 7.4, Suppl. Figure S5). Ten-fold **c**ross-**v**alidation performances of both signatures were very high on the discovery dataset (0.90 and 0.93 for genes and k-mers, respectively) (Table 7.5), which is an over-estimated performance since features here were tested on the same dataset used to select features (Ambroise and McLachlan, 2002).

Figure 7.4. A 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 (Supplementary file 1). Eleven of the 21 genes were also found in a list 180 genes compiled from published PCa outcome signatures (Supplementary 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 (Supplementary file 2, Figure S6.A, Figure 7.4.A). 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 (Liu et al., 2016), 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 (Wang et al., 2014), COL1A1 (Liu et al., 2018b), GTSE1 (Wu et al., 2017) and PTPRN2 (Chen et al., 2013), 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) (Robinson et al., 2011) in Supplementary 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 (Supplementary 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 (Xie et al., 2020) (Mo et al., 2020). 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 (Supplementary 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 Methods). 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 Table 7.3 and Figure 7.4B). 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, Table 7.5), although the later did not sustain any loss when transferred to the independent cohort (Table 7.3). High validation AUCs indicate a strong replicability of both the reference-free and reference-based risk signatures.

Table 7.5: Signature performances for risk prediction

	AUC - risk prediction	
	TCGA Cross-validation	**ICGC Independent dataset**
Reference-free	0.93 +/- 0.04	0.85
Reference-based	0.90 +/- 0.05	0.86

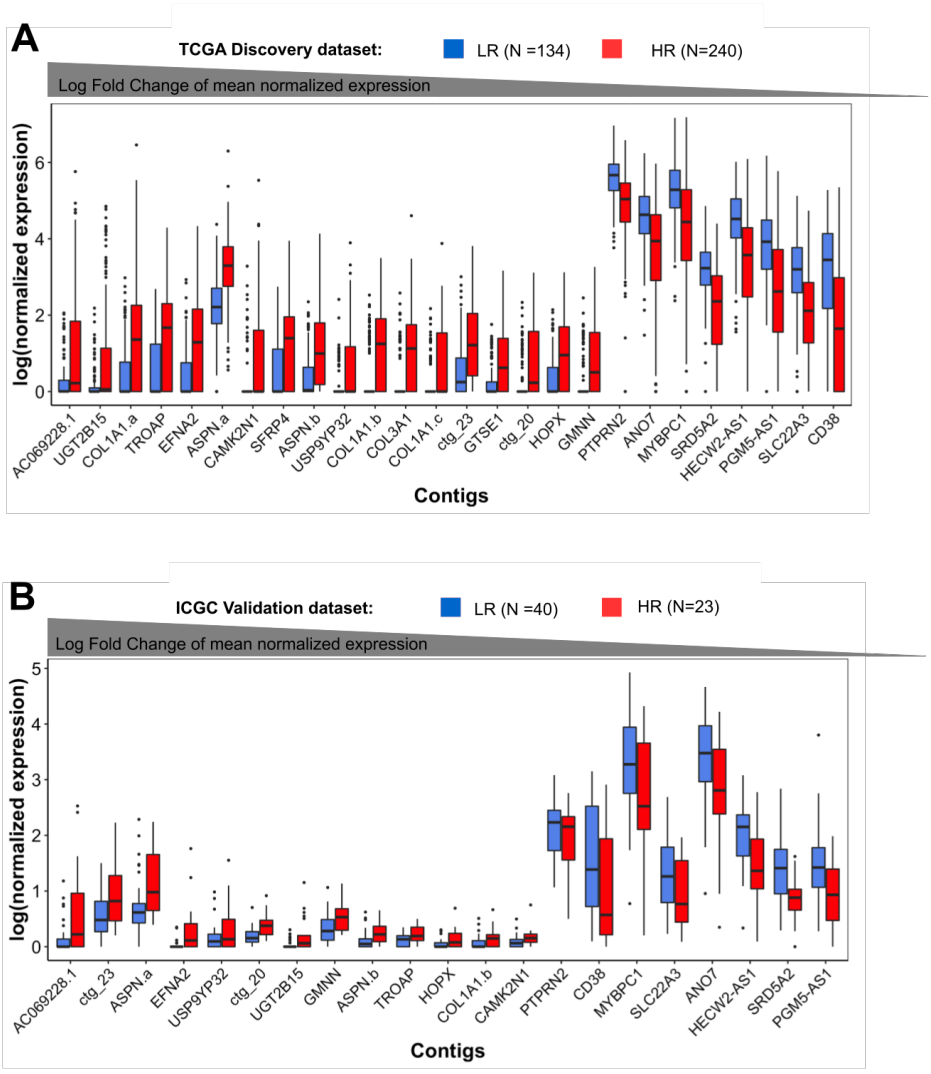


Figure 7.4: Expression of risk signature contigs in LR and HR samples. A: TCGA-PRAD discovery cohort. B: ICGC-PRAD validation cohort

7.3.2 Relapse signatures contain key PCa drivers

Application of the gene-free and gene-based signature discovery procedures (Section 7.2.2) to relapse analysis produced a 14-contig reference-free signature and a 10-gene reference-based signature (Supplementary File 2, Figure 7.5A, Supplementary Figure S6 A). 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 (Chen et al., 2004), the androgen receptor (AR) and two of its main targets, KLK2 and KLK3, the later encoding the PSA protein (Figure 7.5A). Another contig matched SPDEF, a gene whose loss is associated to PCa metastasis (Chen et al., 2017).

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 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 (Figure 7.5A). 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 (Leyten et al., 2015). 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 (Bussemakers et al., 1999) and 5 others (DDC, RRM2, FEV, TSPAN1, HMGCS2) are involved in PCa etiology (Koutalellis et al., 2012; Mazzu et al., 2019; Zhong et al., 2019; Munkley et al., 2017; Wan et al., 2019). Therefore both gene-based and gene-free relapse signatures were significant in terms of PCa related functions of their component genes or contigs.

7.3.3 Relapse signatures do not accurately classify independent cohorts

Table 7.6: Signatures performances for relapse prediction

Method	AUC - relapse prediction		
	TCGA Cross-validation	ICGC Independent dataset	STELLOO Independent dataset
Reference-free	0.93 +/- 0.1	0.51	0.62
Reference-based	0.84 +/- 0.11	0.66	0.59

Contrary to the risk signatures, relapse signatures showed little overlap with each other and with published PCa signatures (Supplementary File 2). Only PCA3 and KLK2 were found in prior signatures (Shahabi et al., 2016; Klein et al., 2014) 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 **c**ross-**v**alidation performance on the discovery set (AUC of 0.84 and 0.93 respectively, Table 7.6). 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. The reference-based model performed slightly better over ICGC-PRAD, and the reference-free model was slightly better over the Stelloo dataset (Table 7.6). Furthermore, several genes and contigs in the discovery signatures had inconsistent expression variations in the validation datasets (Figure 7.5B,C, Supplementary Figure S6B,C, Supplementary 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 non-relapse (Table 7.1) and the Stello dataset had very low coverage (Table 7.1) 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 (Table 7.7, Figure 7.5.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, Table 7.6). Other limitations of the relapse model are addressed in the discussion.

Table 7.7: Result of filtering procedure on the k-mer and gene matrices for relapse analysis

	Initial matrix	Low expression filter	k-mer merging	Naive Bayes ranking	Feature Selection by Lasso LR	Validation in ICGC	Validation in Stelloo
k-mers or contigs	(not generated)	97,731,857	6,184,108 contigs	500 contigs	14 contigs (219 k-mers)	12 contigs (215 k-mers)	3 contigs (71 k-mers)
genes	60,554	36,006	NA	500	10	10	10

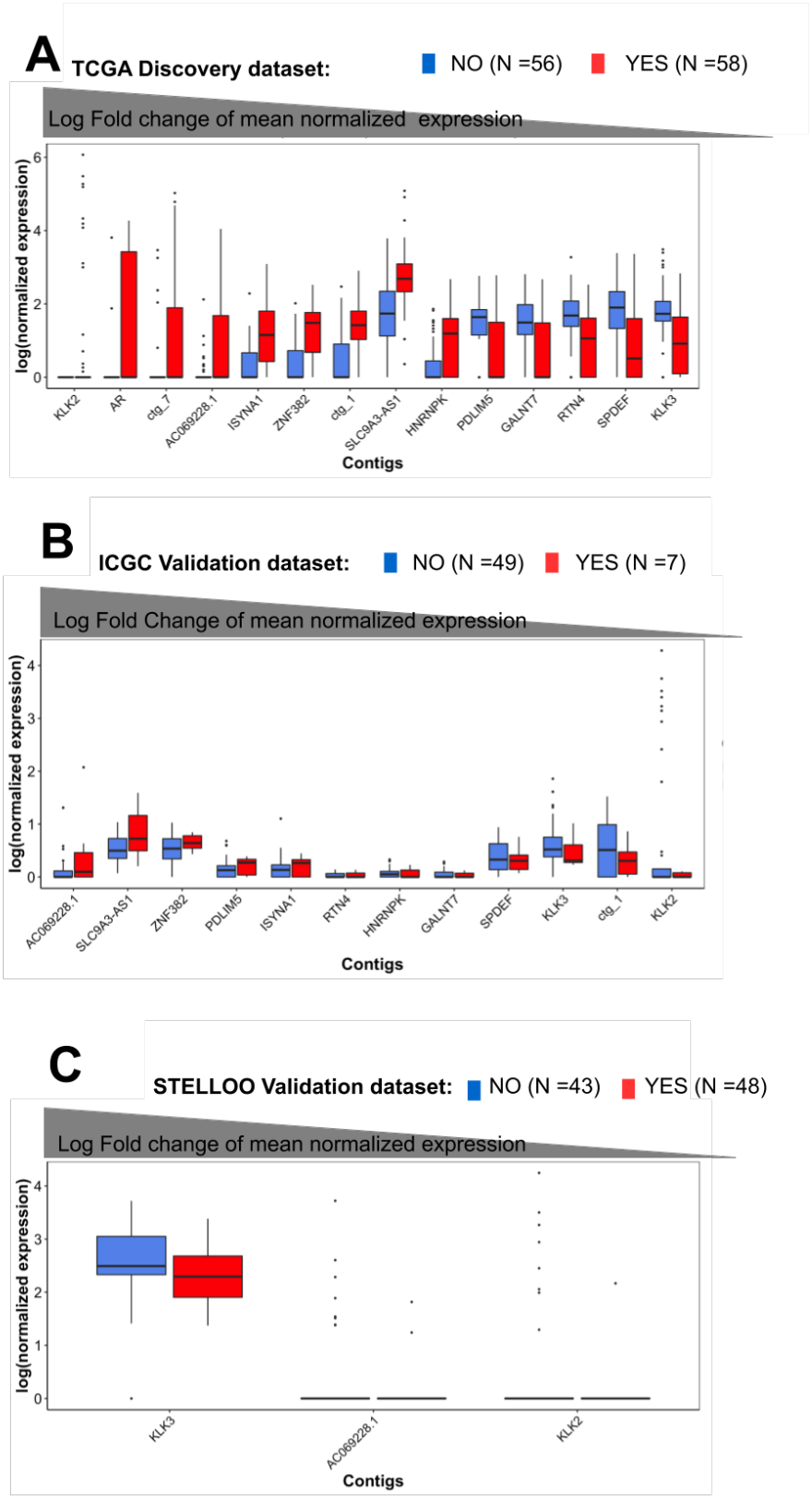


Figure 7.5: Expression of relapse signature contigs in relapse/non relapse samples. A: TCGA-PRAD discovery cohort. B: ICGC-PRAD validation cohort. C: Stelloo validation cohort.

7.4 Discussion

7.4.1 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 (Morillon and Gautheret, 2019). 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 *denovo* 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` (Thomas et al., 2019). `GECKO` uses **m**achine **l**earning (**G**enetic **A**lgorithm) directly on the k-mer count matrix while we first reduce the matrix by grouping k-mers into contigs, before classification and **m**achine **l**earning. 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.

7.4.2 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 (Latil et al., 2003; Erho et al., 2013; Klein et al., 2015; Shahabi et al., 2016). 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 (Witten and Tibshirani, 2010b; Klein et al., 2014; Long et al., 2014; Karnes et al., 2013; Sinha et al., 2019). 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 would be 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 (de Ronde et al., 2013; Campos-Laborie et al., 2019). Adaptations of these solutions to large k-mers matrices will also be considered in the future.

7.5 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.

7.6 Acknowledgements

This work was funded in part by Agence Nationale de la Recherche grant ANR-18-CE45-0020.

Supplementary figures and tables

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

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

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

Table S1. Filtering parameters for count tables

	Analysis	min_recurrence	min_recurrence_abundance
TCGA-PRAD	Risk	3	10
ICGC-PRAD		5	5
TCGA-PRAD	Relapse	3	5
ICGC-PRAD		4	2
STELLOO		3	5

Figure S1. IGV view of RNA-seq reads from the TCGA-PRAD discovery set aligned at the genomic location of risk signature contig ctg_23 (red box). This contig is located in an intron of LDLRAD4. Frames LR and HR show reads sampled from all samples in the LR and HR subsets, respectively, at identical depth for each. Blue boxes and lines in the bottom frame correspond to Gencode annotations of LDLRAD4 transcript isoforms (thick lines: exons, thin lines with arrows: introns).

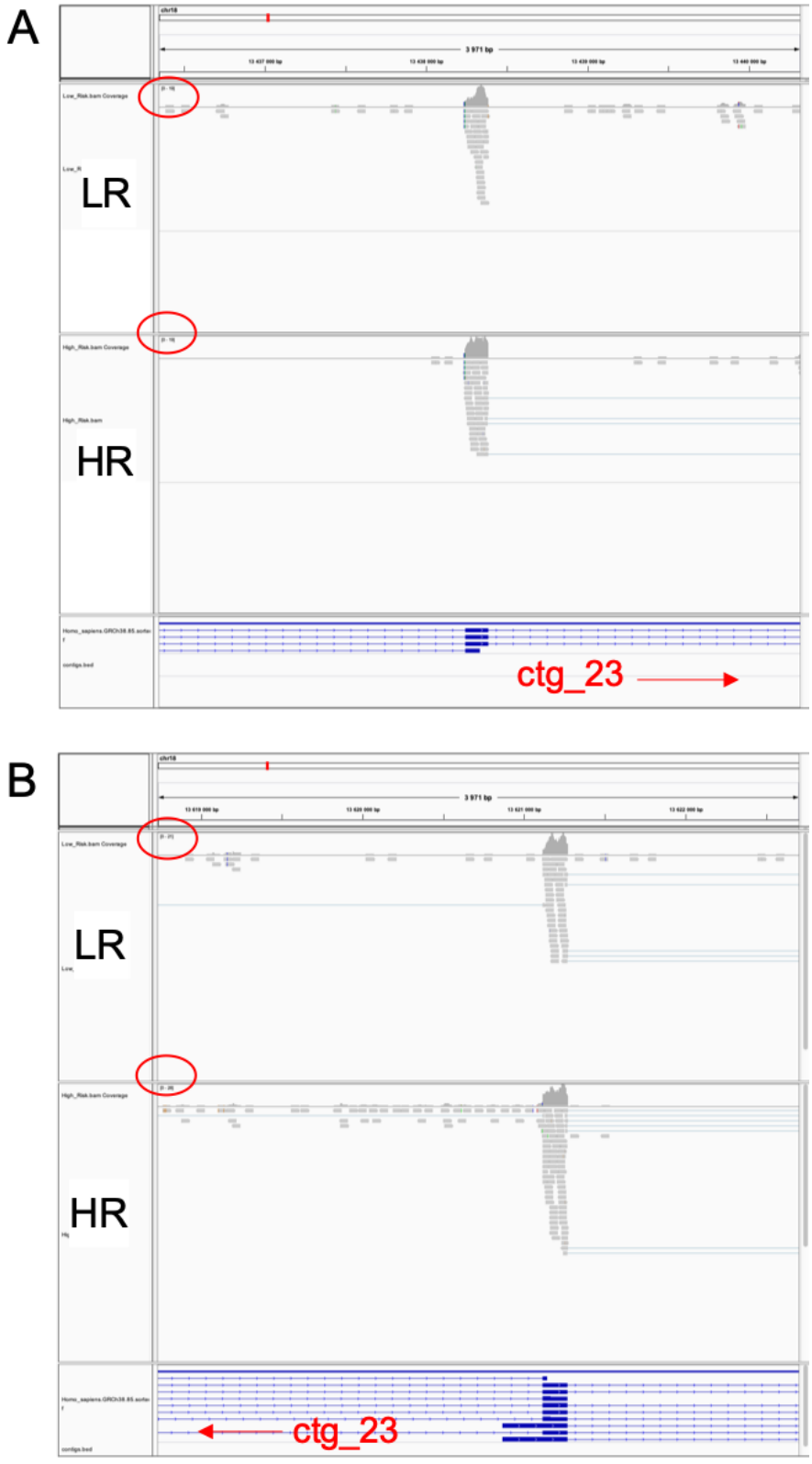


Figure S2. IGV view of RNA-seq reads aligned on the LDLRAD4 exons flanking signature contig ctg_23 on the left (A) and right (B) side of the genomic location of the contig. HR and LR frames are as described above. Note the coverage depth about 6 times lower than ctg_23 coverage in HR condition (red circles) and its lack of variation between LR and HR conditions.

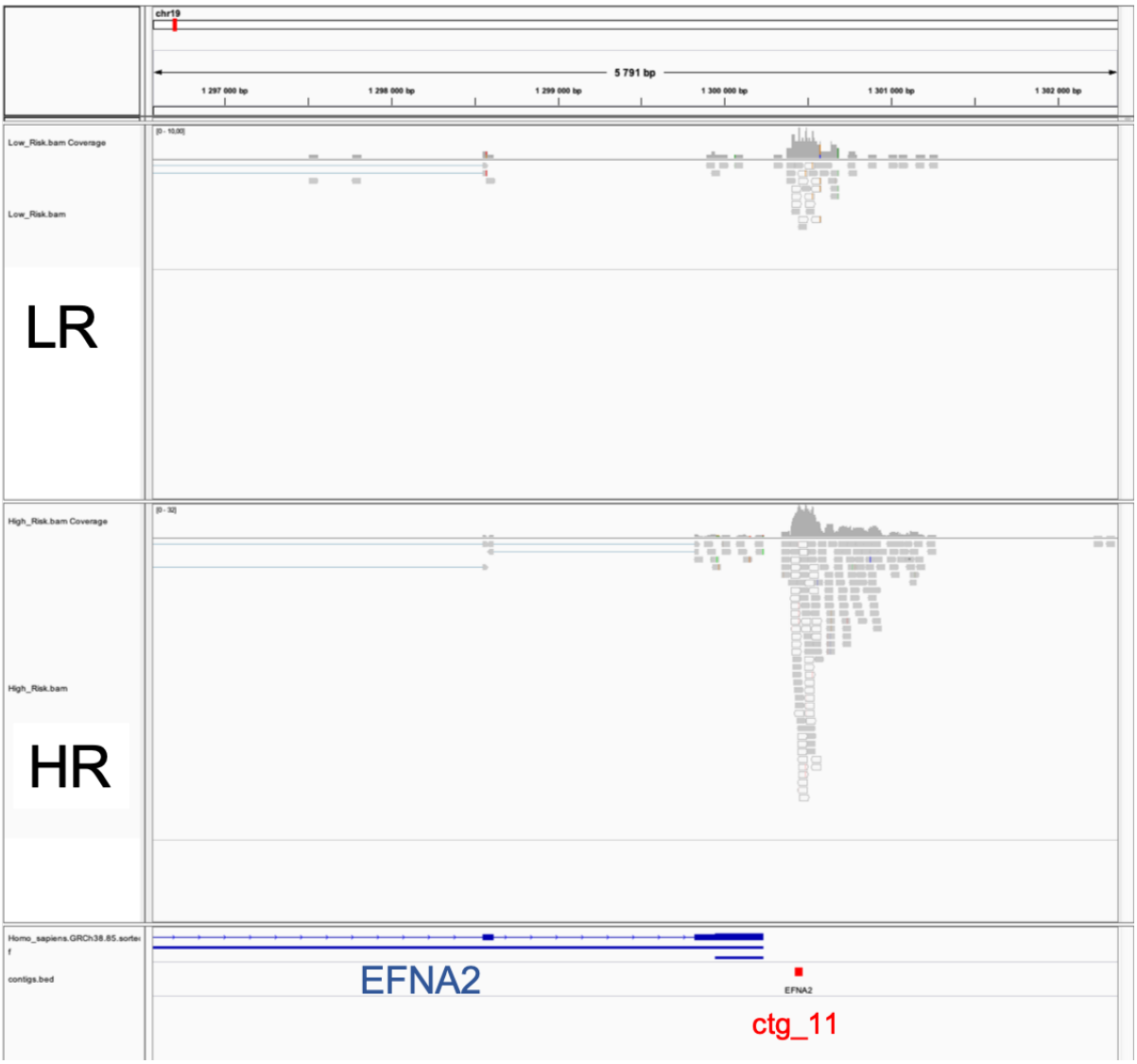


Figure S3. IGV view of RNA-seq reads aligned at risk signature contig ctg_11. Figure legend is as above. ctg_11 was assigned to EFNA2 based on an 3' extended isoform (not shown), but it appears it is more likely an independent transcript.

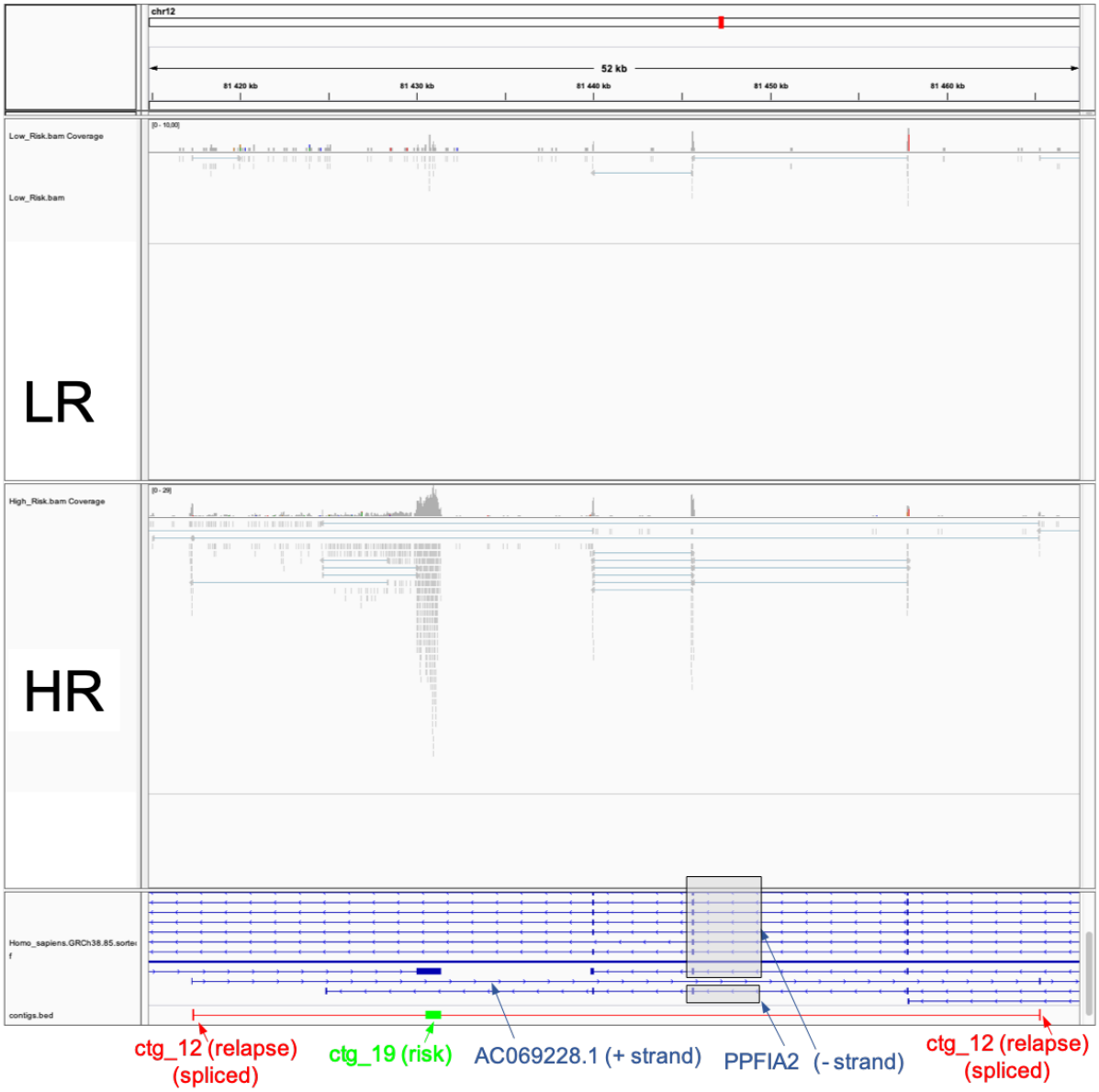


Figure. S4. IGV view of RNA-seq reads aligned at locus AC069228.1, where two signature contigs (ctg_19 from the risk model and ctg_12 from the relapse model) are aligned. Figure legend is as above. Contigs match two different transcripts of the AC069228.1 lncRNA gene, located antisense of gene PPFIA2 (boxed transcripts). In spite of the unstranded nature of aligned reads, mapping to AC069228.1 is unambiguous as only this gene has annotated exons at the corresponding locations.

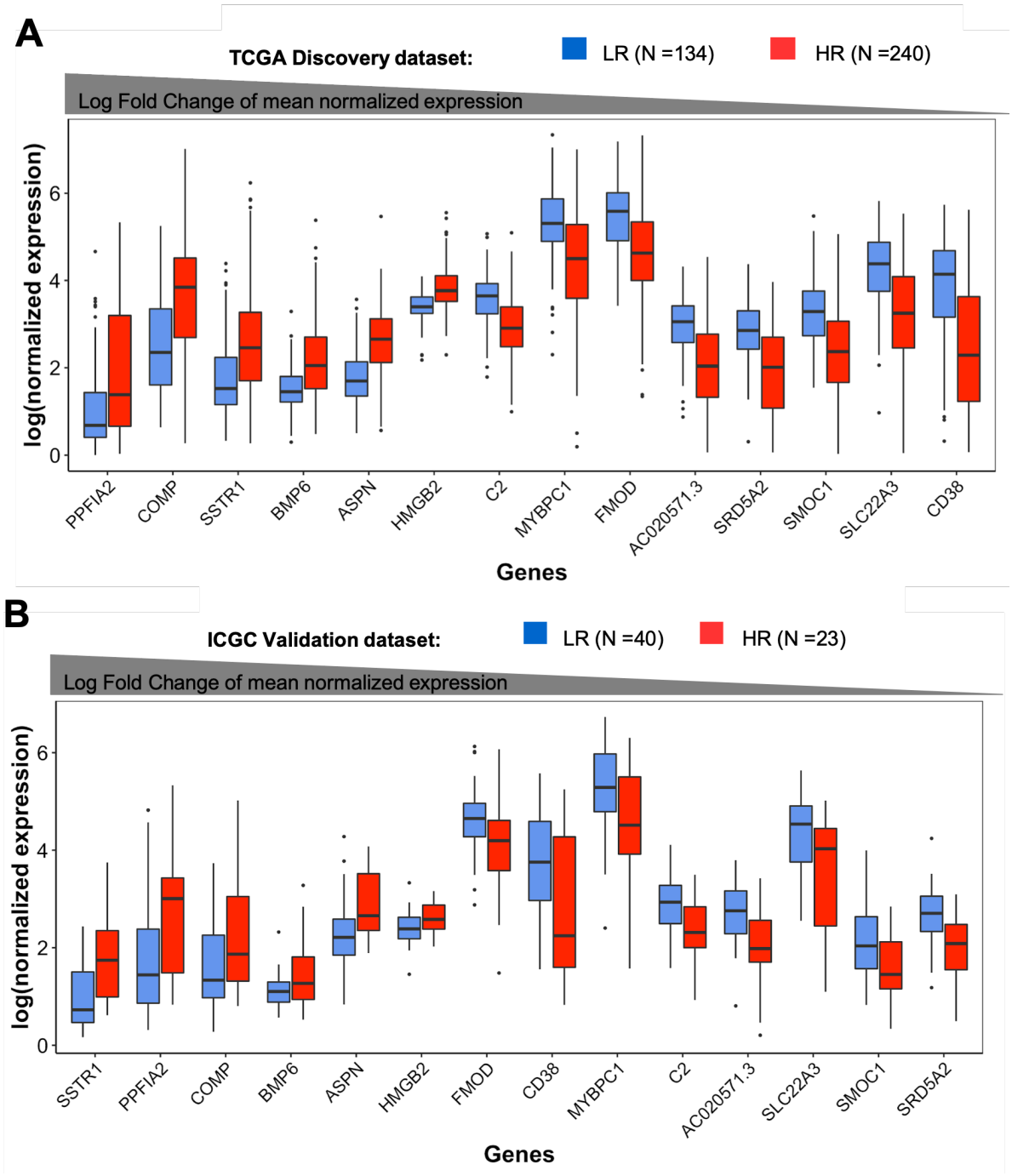


Figure S5. Expression of risk signature genes in relapse/non relapse samples. A: TCGA-PRAD discovery cohort. B: ICGC-PRAD validation cohort.

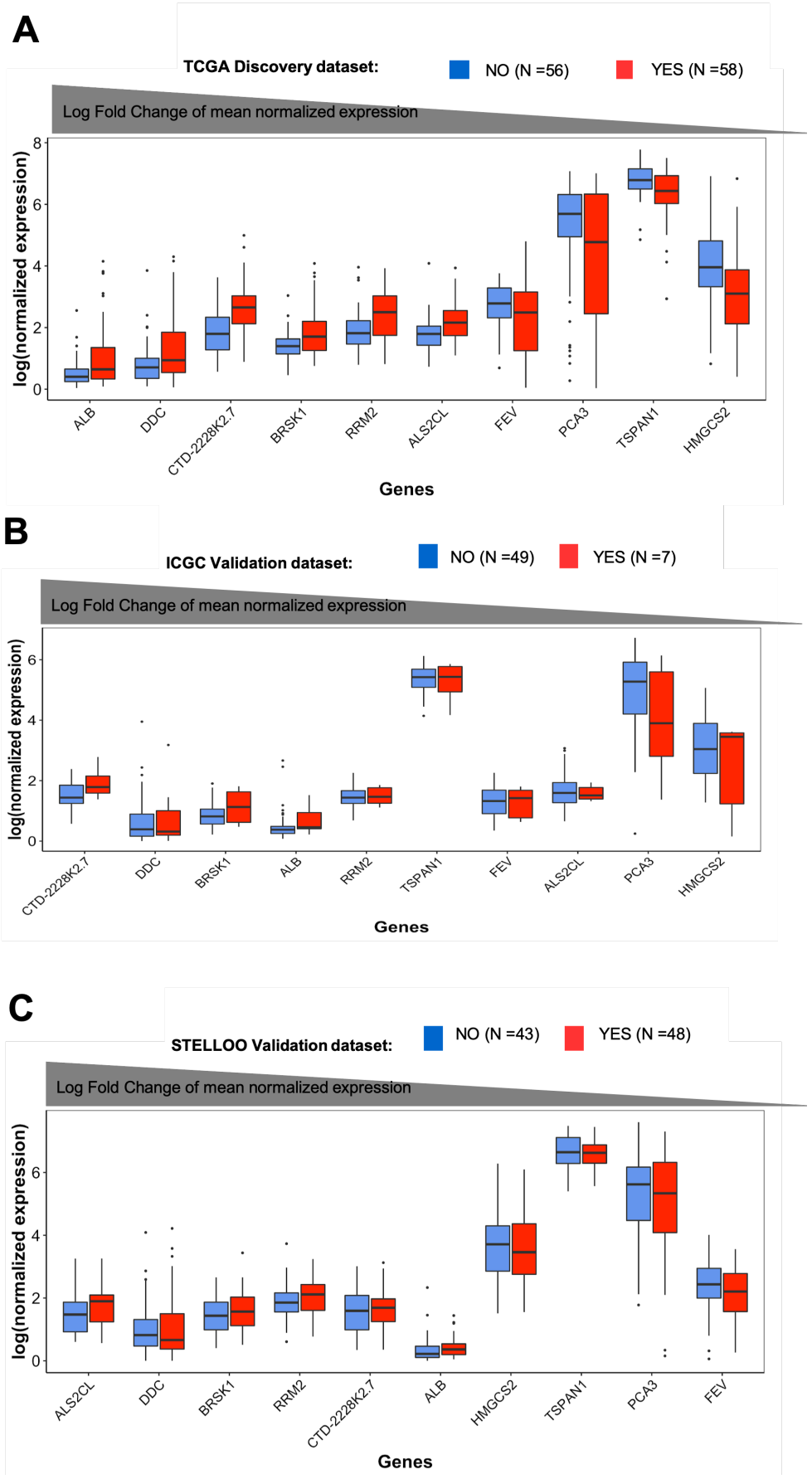


Figure S6. Expression of relapse signature genes in relapse/non relapse samples. A: TCGA-PRAD discovery cohort. B: ICGC-PRAD validation cohort. C: Stelloo validation cohort.

Part III

Discussion

Chapter 8

Discussion and perspectives

8.1 Applying unsupervised filtering methods with contigs extension count data

The results obtained by filtering strategies in Chapter 5 show that unsupervised filtering methods, such as variance, MAD, and especially entropy filter methods are useful to remove low-expressed k-mers prior differential analysis. We have benchmarked several filtering methods, and the entropy filter seems the most appropriate to remove low-expressed k-mers.

Additionally, an advanced contig extension applied in Chapter 7 showing the effectiveness in reducing drastically k-mers while preserving predictive accuracy for further analysis. As a result, we would apply the entropy filtering method with contig extension counts instead of k-mer count matrix. We would also experiment on a more complex design with multiple conditions.

8.2 Other unsupervised learning algorithms for clustering k-mers

The density-based clustering obtained undesirable results, *e.g.*, k-mers within the same clusters showed a lack of homogeneous. Though, all our experiments were performed only with one density-based algorithm. It should be fair to try several algorithms, either other density-based methods or other idea-based clustering methods, *e.g.*, partition, hierarchy . . . , which unfortunately, I could not implement during my thesis.

Besides, as 34% of the total 150,000 k-mers were clustered within a cluster by density-based DBSCAN, these k-mers were heterogeneous; this may suggest that our dataset had several different density clusters. Therefore, future clustering algorithms will tackle with varying density problem, *e.g.*, **L**ocally **S**caled **D**ensity-based **C**lustering (LSDBC) (Biçici and Yuret, 2007).

8.3 The characteristics of reference-free signatures

In Chapter 6, we found a predictive signature made of only nine unannotated lncRNA that outperformed the lncRNA-based biomarker PCA3 in the prediction of high-risk tumors. The reference-free signature consisting of 26 contigs inferred in Chapter 7 had a similar AUC-based performance as the reference-based signature for risk prediction. The contigs in this signature were relevant to PCa (*e.g*, MYBPC1, ASPN, COL1A1) and overlapped with other PCa signatures found in previous articles, *e.g.*, Erho et al. (2013); Shahabi et al. (2016); Jhun et al. (2017); Sinnott et al. (2017). Noticeably, our signature also included two unannotated contigs which could not have been found by a conventional RNA-seq analysis.

The sensitivity when transferring a signature to an external dataset is a potential weakness of k-mer approaches. This weakness derives from k-mer contents which

are affected by differences in library preparation protocols and sequencing platforms used in discovery and external validation sets. Two solutions for measuring signature contigs in an external independent cohort were proposed in Chapter 6 and Chapter 7. The solution proposed in Chapter 7, designed to match exactly each nucleotide, could be upgraded to allow a fixed number of nucleotide mismatches k_m. This solution would require change in the procedure for gathering k-mers related to signature contigs in the new independent dataset. The number of mismatches k_m could be a tuning parameter, allowing a trade-off between the number of k-mers found in the independent cohort and the consistence in the abundance levels of k-mers within the same signature contig. However, this procedure would amount to erasing single nucleotide variants in contigs. Such variants may be part of important biomarkers, as shown in the relapse and risk signatures in Chapter 7.

8.4 Performances of reference-free signatures

In our study presented Chapter 7, the predictive performances of the reference-free RNA signatures did not significantly surpass the predictive performance of the reference-based signatures. Especially, the relapse signatures in Chapter 7 performed poorly in external independent datasets.

Our current approach aims to discover RNA signatures that classify samples into groups. This performs well if patients within the same group (i.e. tumor, recurrence) show similar RNA expression levels and the average expression levels between different groups (i.e. tumor or recurrence versus normal or non-recurrence) are dissimilar (as shown in the case of risk signature in Chapter 7). This means samples are homogeneous within the same group. The low performance of relapse signatures observed in Chapter 7 may suggest that samples are highly heterogeneous, which is a well known problem in prostate cancer (see for instance Berglund et al. (2018)). One solution would be to stratify patients into subgroups before applying further analysis. Several stratification solutions have been proposed for gene expression analysis (de Ronde et al., 2013; Campos-Laborie et al., 2019). An adaptation of

these solutions to reference-free approaches using k-mers would require non-trivial developments.

Additionally, the way we grouped patients into discrete relapse and non relapse categories lead us to drop a large number of samples from the analysis, as described in Section 7.2.1. Future relapse models should take into account of the time-to-relapse using survival models described in Section 1.3.7. To do so, we will need to adapt survival models to the context of high-dimensional k-mer analysis.

Another factor that should be considered is adequacy of the model building strategy to the size of the sample available. In Chapter 6, DE-kupl was applied to a small discovery cohort (8 normal *vs.* 16 tumor speciments). In Chapter 7, our pipeline was used on a larger cohort (more than 500 samples). To rank the features, we adopted an univariate feature ranking based on the F1-scores obtained by 5-fold cross validation with a Naïve Bayes classifier. This type of ranking was suitable for large cohorts but would not be effective on smaller cohorts. For small datasets, we still advise to combine DE-kupl (hence a strategy based on differential expression analysis) with supervised learning as presented in Chapter 6.

In summary, if our reference-free approach did not surpass the the reference-based approach in one specific application (PCa prognosis), it is important to notice that we found predictive events without using prior knowledge on the genome and the genes. Therefore, the approach can be useful for the analysis of species with no reference genome or transcriptome.

Finally, a follow-up to this study should involve testing other reference-free classifiers such as GECKO on the same datasets in order to compare predictive performances and signature contents of the different methods.

Résumé en français

Introduction: Chapitres 1 à 4

La technologie de séquençage d'ARN à haut débit (RNA-seq) a révolutionné notre vision de l'expression génique grâce à sa capacité à capturer toute la diversité des transcrits produits par chaque cellule. Les données RNA-seq sont de plus en plus utilisées en médecine de précision pour établir les profils moléculaires des tumeurs ou pour étudier les réseaux de gènes régissant l'adaptation d'une cellule à son environnement. Cependant, l'analyse informatique des données RNA-Seq, qui repose généralement sur la comparaison avec des séquences de référence, ne parvient pas à identifier une grande partie des ARN résultant d'altérations génomiques ou de traitement de l'ARN. En outre, les méthodes existantes ne s'adaptent pas bien à des centaines de milliers de bibliothèques générées par les études actuelles de transcriptome à grande échelle.

Le projet général de l'équipe vise à développer un nouveau concept d'analyse du transcriptome, en s'appuyant sur une information de "tags", ou k-mers, sélectionnés pour représenter des événements spécifiques de variation d'ARN, et un système d'indexation efficace permettant une identification rapide de ces variants dans n'importe quelle banque de transcrits, sans nécessiter d'alignement. Ce système présente deux avantages majeurs: Premièrement, il peut identifier et quantifier tout type de variant de transcription (variants d'épissage, transcrits de fusion, ARN circulaires, ARN de répétitions ou même ARN provenant de pathogènes) ainsi que

les variations génomiques telles que les SNP, agissant soit au niveau de la séquence des protéines (mutations non synonymes) ou de la structure secondaire (ARNm, UTR). Deuxièmement, il est suffisamment efficace pour permettre la réanalyse de grands ensembles publics de données RNA-seq. Ces propriétés nous permettront d'identifier de nouveaux biomarqueurs et signatures (structurelles) qui ont échappé à toutes les études précédentes.

La classification moléculaire des sous-types de maladies est une tâche essentielle en médecine de précision. Les données de transcriptome sont probablement l'une des méthodes les plus puissantes pour obtenir une telle classification. Cependant, ces données sont le plus souvent résumées dans une liste de gènes surexprimés et sous-exprimés. En utilisant notre approche de décomposition en k-mers, nous montrons qu'une grande quantité d'informations génétiques et d'expression peut être récupérée et contribuer fortement à enrichir les signatures de la maladie et à obtenir une classification plus précise des patients. Dans ce but, le manuscrit est organisé en trois parties principales: introduction, résultats et discussion.

La première partie du manuscript est introductive: elle présente le contexte biostatistique (Chapitre 1) et bioinformatique (Chapitre 2) de l'analyse des données transcriptomique (comprenant l'analyse de type microarray et RNA-seq), les spécificités du cancer de la prostate (Chapitre 3) ainsi que les défis de la thèse et contributions associées (Chapitre 4).

Si la technologie RNA-seq a révolutionné la mesure de l'expression génique à l'échelle du transcriptome et a permis de comprendre les éléments fonctionnels et structurels du génome, mener correctement une étude RNA-seq reste un défi en raison du grand nombre de protocoles d'analyse RNA-seq publiés. Nous présentons deux stratégies principales pour l'analyse des données RNA-seq: basée sur les références et sans référence. Il y a trois étapes principales dans l'analyse des données de référence RNA-seq qui sont le contrôle de qualité, l'alignement des reads et la quantification des niveaux de gène et de transcription. Nous introduisons des outils logiciels impliqués dans chaque étape dans le chapitre 2. Les logiciels "sans référence" présen-

tés diffèrent par leurs objectifs: DE-kupl et MINTIE identifient de nouvelles formes de transcription, par exemple, des variants d'épissage, des transcripts de fusion, ... qui sont spécifiques ou surreprésentées dans un ensemble d'échantillons, tandis que GECKO crée des classificateurs prédictifs composés de fragments de séquence sans référence. Les résultats présentés dans les publications suggèrent que l'approche sans référence est faisable à la fois pour la découverte de transcriptions et pour la modélisation prédictive. Un gros avantage de ces approches est qu'elles ne sont pas limitées par l'alignement des reads sur le génome ou le transcriptome de référence. Cependant, il existe encore plusieurs limites dans les approches actuelles, par exemple, DE-kupl a été initialement proposé pour découvrir des événements biologiques non référencés dans les données RNA-seq, et n'a pas été conçu pour effectuer la prédiction de l'état des échantillons, alors que l'ensemble des k-mers découvert par GECKO a été validé sans jeu de données indépendant.

Le chapitre 3 présente les aspects cliniques du diagnostic des cancers de la prostate: diagnostic par PSA, évaluation du grade et stratification des tumeurs. Nous présentons aussi diverses publications utilisant des approches d'apprentissage pour l'identification des signatures diagnostiques ou pronostiques de PCa. Cependant, aucune des signatures publiées n'a exploré la possibilité de trouver de nouveaux gènes ou des isoformes de transcription associés au risque ou à la rechute. La nouvelle génération de prédicteurs utilisant des approches transcriptomiques sans référence pourrait potentiellement identifier tout nouveau transcrit présent dans un échantillon, par exemple, de nouveaux variants d'épissage, des lncRNA ou des ARN issus de rétroéléments répétés. Ma thèse vise donc à appliquer ce type de prédicteurs pour l'identification des signatures PCa.

Résultats

Cette section résume les trois chapitres de contributions de cette thèse. Le chapitre 5 détaille mes contributions à la réduction de la dimension. Le chapitre 6 détaille mes contributions à la problématique d'analyse du transcriptome sans référence

par rapport à l'analyse basée sur la référence (conventionnelle) pour révéler de nouvelles signatures d'ARN pour les cancers de la prostate. Le chapitre 7 détaille la comparaison des performances entre ces méthodes de prédiction (sans référence versus approche conventionnelle) pour la prédiction du status des patients.

Chapitre 5: Méthodes de réduction de dimension dans l'analyse k-mer

L'analyse k-mer implique un grand nombre de variables à prendre en compte, généralement des dizaines à des centaines de millions de k-mers par banque RNA-seq. Beaucoup de ceux-ci peuvent résulter d'erreurs et / ou d'artefacts technologiques (tels que la contamination de l'adaptateur), ou peuvent être fortement corrélés dans leur expression, conduisant à des variables peu informatives ou redondantes. Un très grand nombre de variables redondantes et non pertinentes peut entraîner un surajustement, une faible précision et de longs temps d'exécution. En conséquence, la réduction de la matrice est une étape essentielle avant d'appliquer les techniques d'apprentissage pour l'analyse en aval, telles que l'analyse de l'expression différentielle, l'analyse de survie ou la classification du transcriptome. Une partie de ma thèse de doctorat était consacrée au développement de différentes stratégies de réduction de dimension basées sur le nombre de k-mer. J'ai étudié une gamme de stratégies de filtrage et de clustering basées sur le nombre de k-mer et les ai testées avec des ensembles de données réels. J'ai exploré différentes solutions pour réduire la dimension des matrices k-mers en utilisant leurs comptages. Les stratégies de filtrage ont démontré leur efficacité en réduisant de manière significative le nombre de k-mers à faible expression avant l'analyse d'expression différentielle. Parmi les approches de filtrage, la méthode signal-bruit supervisé a produit la méthode la plus rapide et la plus efficace pour réduire les k-mers faiblement exprimés ou non pertinents avant l'analyse différentielle. Cependant, cette approche de filtrage n'est pas un filtre indépendant et ne peut pas être utilisée en toute sécurité avant l'expression différentielle. Parmi les filtres qui n'utilisaient pas les informa-

tions préalables sur l'échantillon, l'entropie normalisée s'est avérée la plus efficace. J'ai également exploré l'utilisation potentielle de techniques de clustering non supervisées pour regrouper les k-mers en fonction de la similitude de leurs nombres. L'idée de clustering des k-mer a montré des résultats indésirables, avec des k-mers dans le plus grand cluster montrant une hétérogénéité entre les échantillons alors que les k-mers "bruit de fond" devraient être regroupés dans un cluster réel. Cependant, mes résultats de clustering étaient basés sur un algorithme de clustering sur critère de densité. Je n'ai pas testé d'autres algorithmes de clustering, c'est-à-dire à partir d'autres méthodes basées sur la densité ou d'autres catégories, telles que la hiérarchie, les algorithmes de grille. Ces résultats indiquent que DBSCAN n'était pas une méthode de clustering appropriée pour notre ensemble de données qui pouvait avoir des densités variables. Les densités variables conduisent à l'échec de DBSCAN qui rassemble probablement des k-mers "faussement semblables", c'est-à-dire une expression de dissimilarité lors de l'expansion de cluster, dû au fait DBSCAN n'utilise qu'un seul seuil de densité global ϵ. Cependant, chaque k-mer est caractérisé non seulement par ses comptages, mais aussi par sa séquence. Ceci m'a conduit à proposer une autre approche pour réduire la taille de la matrice k-mer: fusionner les k-mers en contigs en fonction de la similitude de leurs comptages et du chevauchement des séquences k-mer. Ce processus de réduction permet de réduire la matrice de comptage de k-mer à une matrice de comptage de contig plus petite avec des caractéristiques moins corrélées et redondantes que dans la matrice initiale.

Chapitre 6: L'exploration du transcriptome sans référence révèle de nouveaux ARN pour le diagnostic du cancer de la prostate

Un objectif méthodologique majeur de cette thèse était de faire progresser les méthodes k-mer sans référence en appliquant la décomposition k-mer à des modèles prédictifs avec des résultats évalués dans des ensembles de données indépendants. J'ai utilisé les résultats produits par DE-kupl dans un ensemble de données PCa fourni par des collaborateurs de l'institut Curie pour effectuer la prédiction de l'état de

l'échantillon. Puisque DE-kupl a été développé comme un pipeline qui capture tous les k-mers présentant des abondances significativement différentes selon les conditions, il n'a pas été conçu pour la modélisation prédictive. J'ai appliqué une procédure pour calculer et tester une signature prédictive obtenue à partir des contigs k-mer produits par DE-kupl, et pour évaluer cette signature dans des ensembles de données indépendants. Dans une cohorte de validation indépendante, ce modèle a atteint un score AUC de plus de 90% pour le diagnostic de la PCa.

Le pipeline DE-kupl , combiné à la visualisation des contigs d'expression et à la sélection manuelle, a permis de récupérer des sous-séquences d'ARN ayant une capacité prédictive aussi puissante que la signature dérivée de gènes annotés par GENCODE. Cela démontre la capacité d'une approche sans référence à récupérer des contigs intéressants non référencés.

Cependant, les deux approches (sans gène et basée sur les gènes) ne sont pas équitablement comparées dans ce travail: l'approche sans gène implique une sélection manuelle de contigs, basée sur des connaissances d'experts. La sélection des contigs dans l'approche sans gène implique l'utilisation de l'ensemble de données d'expressions mesurée par la téchnologie NanoString, sans équivalent dans l'approche basée sur les gènes. J'ai voulu comparer les deux approches en utilisant un pipeline aussi similaire que possible. La raison pour laquelle je voulais comparer les deux approches (sans gène et RNA-seq conventionnel) est que les approches sans gène augmentent fortement le nombre de variables à considérer dans l'ensemble de données (de 50000 gènes dans une analyse RNA-Seq conventionnelle à des millions de k -mers ou contigs dans les approches sans gène). Par conséquent, je pouvais soupçonner que l'approche k-mer est plus sujette au surajustement. Pour répondre à cette question, j'ai proposé de comparer un classificateur sans gène et un classificateur basé sur le gène. En outre, dans les deux approches (sans gène et à base de gène), l'étape préliminaire avant la sélection des variables et l'apprentissage supervisé a été réalisée à l'aide d'une analyse différentielle (recherche de k-mers ou de gènes différentiellement exprimés). J'ai donc souhaité présélectionner les variables en fonction de leurs performances prédictives au lieu du résultat de l'analyse différen-

tielle. Pour les deux raisons mentionnées ci-dessus, mon objectif du travail mené et décrit dans le chapiptre suivant était donc double: proposer un pipeline pour effectuer des prédictions à l'aide de k-mers et comparer ce pipeline à un pipeline RNA-seq conventionnel pour découvrir une signature transcriptomique à l'aide de données RNA-seq.

Chapitre 7: Une analyse comparative des signatures de transcriptome sans référence et conventionnelles pour le pronostic du cancer de la prostate

Comme présenté dans la partie Introduction, il existe de nombreuses techniques de sélection de fonctionnalités et d'apprentissage supervisé. J'ai décidé d'utiliser la régression logistique lasso combinée à la pénalité `LASSO` car elles ont été utilisées dans des articles récents pour découvrir des signatures PCa à l'aide de données RNA-seq conventionnelles (Shahabi et al., 2016; Jhun et al., 2017). Dans les deux cas, basé sur le gène et sans gène, le nombre de variables est trop grand pour appliquer directement la régression logistique `LASSO` sur la matrice de comptage. Pour cette raison, j'ai adopté une étape préliminaire de criblage drastique conçue pour réduire le nombre de variables à un nombre inférieur et éviter d'exécuter la régression logistique `LASSO` dans un cadre dimensionnel ultra-élevé. Pour extraire les variables importantes, j'ai utilisé une classification univarié basée sur la capacité des variables à prédire de nouvelles données à l'aide d'une approche Bayesienne, calculant un F1-score par cross-validation. Compte tenu du grand nombre de k-mers à classer, j'ai choisi la règle de `Naïve Bayes` comme suggéré par Thomas et al. (2019) car l' implémentation C++ de Naïve Bayes était la plus rapide à exécuter parmi l'ensemble des outils disponibles. D'autres solutions sont possibles, comme l'utilisation d'autres algorithmes que Naïve Bayes pour classer les caractéristiques, et d'autres techniques de sélection de variable et d'apprentissage supervisé multivarié que la régression logistique LASSO. Cependant, pour effectuer une comparaison équitable entre les approches génique et sans gène, j'ai sélectionné les outils avant de lancer la compara-

ison, indépendamment des considérations externes: je n'ai pas cherché à optimiser l'ensemble des outils utilisés pour biaiser la comparaison vers une approche ou une autre. Un autre problème est le choix des outils pour déduire la signature k-mer: Thomas et al. (2019) ont proposé d'utiliser un algorithme génétique et j'ai proposé le pipeline résumé ci-dessus et décrit plus en détail au chapitre 7.

Dans cette thèse, je me suis principalement concentrée sur la comparaison des approches sans gènes et conventionnelle. Dans le pipeline proposé, j'ai utilisé une technique de réduction matricielle basée sur l'extension des k-mers en contigs pour éviter de travailler directement sur une matrice de k-mers pleine de k-mers corrélés et redondants. Je n'ai pas utilisé les stratégies de filtrage avant l'étape de dépistage univarié pour les deux raisons suivantes. Premièrement, les stratégies de filtrage sont principalement conçues pour éliminer les k-mers peu abondants avant l'analyse de l'expression différentielle. Deuxièmement, l'étape de filtrage conduit à une réduction drastique de l'espace des variables (de milliers ou millions à quelques centaines). Dans ce contexte, le filtrage préalable au criblage ne réduirait que légèrement le temps d'exécution de l'étape de criblage et ne changerait pas l'ensemble final de variables retenues pour la sélection de variables ultérieure et l'apprentissage supervisé.

La signature sans référence composée de 26 contigs déduits au chapitre 7 avait une performance basée sur l'AUC similaire à la signature basée sur la référence pour la prédiction des risques. Les contigs de cette signature étaient pertinents pour PCa (par exemple, MYBPC1, ASPN, COL1A1) et se chevauchaient avec d'autres signatures de PCa trouvées dans des articles précédents, par exemple Erho et al. (2013); Shahabi et al. (2016); Jhun et al. (2017); Sinnott et al. (2017). Notamment, notre signature comprenait également deux contigs non annotés qui n'auraient pas pu être trouvés par une analyse classique d'RNA-Seq.

Discussion

Dans la partie Introduction, j'ai montré que l'analyse k-mer pouvait, en principe, capturer la variation de transcription complète présente dans un échantillon RNA-seq. Cette variation peut ensuite être attribuée à des événements biologiques tels que les ARNnc, les variants d'épissage et de polyadénylation, les introns, les répétitions, qui sont ignorés par les protocoles standard basés sur les annotations de gènes de référence. Il existe de nombreuses preuves de présence d'ARN non référencé dans les tissus malades et celui ci peut former des biomarqueurs cliniquement utiles. La signature transcriptomique que j'ai identifiée pour le diagnostic du cancer de la prostate (chapitre 6) consistait en seulement neuf lncRNAs et s'est avérée plus efficace que le biomarqueur commercial PCA3 pour la détection des tumeurs à risque élevé. Mes signatures sans référence pour le pronostic du cancer de la prostate (chapitre 7) contenaient un ensemble de séquences d'ARN contenant des ARN non annotés et de nouveaux variants d'ARN annotés qui ne faisaient pas partie des signatures basées sur des gènes.

Dans notre travail sur les modèles pronostiques pour le PCa, j'ai dérivé des signatures de contigs à partir d'un jeu de données de découverte. Comment évaluer la robustesse et la généralisation de ces signatures? Pour ce faire, il faut transférer les informations contig à une cohorte clinique différente et obtenir une mesure d'expression quantitative comparable. Cependant, cette tâche pose un réel défi car cela nécessite un mécanisme qui permet une correspondance exacte de chaque nucléotide pour garantir que les contigs de la signature sont correctement identifiés dans le nouvel ensemble de données. J'ai proposé deux solutions pour la mesure des contigs de signature dans un nouvel ensemble de données. Ces deux solutions ont été conçues pour s'adapter à différents contextes d'étude. Dans le diagnostic du cancer de la prostate, l'ensemble des candidats pour les signatures contig était un panel donné extrait par des connaissances d'experts. En conséquence, les contigs dans la signature dérivée de cet ensemble étaient fortement exprimés et la tâche de les trouver dans d'autres ensembles de données était relativement facile. À l'inverse,

les signatures contig inférées dans le pronostic du cancer de la prostate ont été automatiquement identifiées à partir d'environ 94 millions de k-mers générés à partir de plus de 400 banques RNA-seq (modèle de prédiction du risque) et avaient des niveaux d'expression très variables. Cela nécessitait une procédure d'appariement minutieuse pour garantir que le plus de contigs possible puisse être quantifié dans l'ensemble de données indépendant.

La dernière partie du manuscrit est consacrée à la discussion et aux perspectives. Les résultats obtenus par les stratégies de filtrage montrent que les méthodes de filtrage non supervisé, telles que la variance, **M**edian **A**bsolute **D**eviation, et en particulier les méthodes de filtrage d'entropie sont utiles pour éviter l'analyse différentielle préalable de k-mers faiblement exprimés. J'ai comparé plusieurs méthodes de filtrage et le filtre d'entropie semble le plus approprié pour éliminer les k-mers faiblement exprimés.

En outre, un algorithme avancé d'extension de contig appliquée dans le pronostic PCa a montré son efficacité à réduire considérablement les k-mers tout en préservant la précision prédictive pour une analyse plus approfondie. En conséquence, j'ai appliqué la méthode de filtrage d'entropie suivie de la routine avancée d'extension des contig, plutôt que d'utiliser directement 'une matrice de comptage de k-mer. Nous expérimenterons également une conception plus complexe avec plusieurs conditions. Le regroupement basé sur la densité a produit des résultats indésirables, par exemple, des k-mers au sein des mêmes groupes ont montré un manque d'homogénéité. Cependant, toutes mesexpériences n'ont été réalisées qu'avec un seul algorithme basé sur la densité. Il serait justifié d'essayer plusieurs algorithmes, soit basés sur la densité, soit d'autres méthodes de regroupement basées sur des concept tels que la partition ou la hiérarchie, ce que je n'ai malheureusement pas pu mettre en œuvre au cours de ma thèse. Le fait que 34% des 150 000 k-mers totaux étaient regroupés par DBSCAN dans un seul cluster de densité et que ces k-mers étaient hétérogènes peut suggérer que notre ensemble de données comprenait plusieurs aggrégats de densité différentes. Par conséquent, les algorithmes de regroupement de futurs devraient aborder ces différents problèmes de densité, par exemple **L**ocally

Scaled **D**ensity-based **C**lustering (LSDBC).

La sensibilité lors du transfert d'une signature vers un ensemble de données externe est une faiblesse potentielle des approches k-mer. Cette faiblesse provient du contenu k-mer qui est affecté par les différences dans les protocoles de préparation des banques et les plates-formes de séquençage utilisées dans les jeux de données de découverte et de validation externe. Deux solutions pour mesurer les contigs de signature dans une cohorte externe indépendante ont été proposées au chapitre 6 et au chapitre 7. La solution proposée au chapitre 7, conçue pour correspondre exactement à chaque nucléotide, pourrait être améliorée pour permettre un nombre fixe de mésappariements de nucléotides k_m . Cette solution nécessiterait une modification de la procédure de collecte des k-mers liés aux contigs de signature dans le nouvel ensemble de données indépendant. Le nombre de mésappariements k_m pourrait être un paramètre de réglage, permettant un compromis entre le nombre de k-mers trouvés dans la cohorte indépendante et la cohérence des niveaux d'abondance des k-mers dans le même contig de signature. Cependant, cette procédure reviendrait à effacer les variants de nucléotides uniques dans les contigs. Ces variants peuvent constituer d'importants biomarqueurs, comme le montrent mes signatures de rechute et de risque dans le pronostic de la PCa. Dans notre étude présentée au chapitre 7, les performances prédictives des signatures ARN sans référence n'ont pas dépassé significativement les performances prédictives des signatures basées sur la référence. En particulier, les signatures de rechute ont mal fonctionné dans les ensembles de données externes indépendants. Notre approche actuelle vise à découvrir des signatures ARN qui classent les échantillons en groupes. Cela fonctionne bien si les patients du même groupe (c.-à-d. Tumeur, récidive) présentent des niveaux d'expression d'ARN similaires et les niveaux d'expression moyens entre différents groupes (c.-à-d. Tumeur ou récidive versus normale ou non-récidive) sont différents (comme indiqué dans le cas de la signature du risque dans le pronostic du PCa). Cela signifie que les échantillons sont homogènes au sein d'un même groupe. La faible performance des signatures de rechute observée dans le pronostic du PCa peut suggérer que les échantillons sont très hétérogènes, ce qui est un problème

bien connu dans le cancer de la prostate (voir par exemple Berglund et al. (2018)). Une solution serait de stratifier les patients en sous-groupes avant d'appliquer une analyse plus approfondie. Plusieurs solutions de stratification ont été proposées pour l'analyse de l'expression génique (de Ronde et al., 2013; Campos-Laborie et al., 2019). Une adaptation de ces solutions à des approches sans référence utilisant des k-mers nécessiterait des développements non triviaux. De plus, la façon dont j'ai regroupé les patients en catégories discrètes de rechute et de non-rechute m'a conduit à supprimer un grand nombre d'échantillons de l'analyse. Les futurs modèles de rechute devraient prendre en compte le délai de rechute en utilisant des modèles de survie. Pour ce faire, je devais adapter les modèles de survie au contexte de l'analyse k-mer à haute dimension. Un autre facteur à prendre en compte est l'adéquation de la stratégie de construction du modèle à la taille de l'échantillon disponible. Au chapitre 6, DE-kupl a été appliqué à une petite cohorte de découverte (8 échantillons normaux contre 16 tumeurs). Au chapitre 7, notre pipeline a été utilisé sur une plus grande cohorte (plus de 500 échantillons). Pour classer les caractéristiques, j'ai adopté un classement univarié des caractéristiques basé sur les scores F1 obtenus par validation croisée 5 fois avec un classifieur Naïve Bayes . Ce type de classement convenait aux grandes cohortes mais ne serait pas efficace pour les cohortes plus petites. Pour les petits jeux de données, je conseille de combiner DE-kupl (d'où une stratégie basée sur l'analyse différentielle des expressions) avec un apprentissage supervisé comme présenté au chapitre 6. En résumé, si notre approche sans référence n'a pas surpassé l'approche basée sur la référence dans une application spécifique (pronostic PCa), il est important de noter que j'ai trouvé des événements prédictifs sans utiliser les connaissances préalables sur le génome et les gènes. Par conséquent, l'approche peut être utile pour l'analyse d'espèces sans génome ou transcriptome de référence. Enfin, un suivi de cette étude devrait impliquer de tester d'autres classificateurs sans référence tels que GECKO sur les mêmes jeux de données afin de comparer les performances prédictives et le contenu des signatures des différentes méthodes.

Bibliography

M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. Tensorflow: A system for large-scale machine learning. In 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16), pages 265–283, 2016. URL `https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf`.

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. Agarwal. Data mining: Data mining concepts and techniques. In 2013 International Conference on Machine Intelligence and Research Advancement, pages 203–207. IEEE, 2013.

R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan. Automatic subspace clustering of high dimensional data for data mining applications. In Proceedings of the 1998 ACM SIGMOD international conference on Management of data, pages 94–105, 1998.

A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. Nature, 403(6769):503–511, 2000.

C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. Proceedings of the National Academy of Sciences, 99(10):6562–6566, may 2002. ISSN 0027-8424. doi: 10.1073/pnas.102102699.

S. Anders and W. Huber. Differential expression analysis for sequence count data via mixtures of negative binomials. Nature Precedings, 11:R106, 2010. ISSN 1756-0357. doi: 10.1038/npre.2010.4282.2.

S. Anders, P. T. Pyl, and W. Huber. Htseq—a python framework to work with high-throughput sequencing data. Bioinformatics, 31(2):166–169, 2015.

G. L. Andriole, E. D. Crawford, R. L. Grubb III, S. S. Buys, D. Chia, T. R. Church, M. N. Fouad, E. P. Gelmann, P. A. Kvale, D. J. Reding, et al. Mortality results from a randomized prostate-cancer screening trial. New England Journal of Medicine, 360(13):1310–1319, 2009.

M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander. Optics: ordering points to identify the clustering structure. ACM Sigmod record, 28(2):49–60, 1999.

S. A. Armstrong, J. E. Staunton, L. B. Silverman, R. Pieters, M. L. den Boer, M. D. Minden, S. E. Sallan, E. S. Lander, T. R. Golub, and S. J. Korsmeyer. Mll translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature genetics, 30(1):41–47, 2002.

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):243, dec 2017. ISSN 1474-760X. doi: 10.1186/s13059-017-1372-2.

P. L. Auer and R. W. Doerge. A Two-Stage Poisson Model for Testing RNA-Seq Data. Statistical Applications in Genetics and Molecular Biology, 10(1):1–26, May 2011.

M. Aumüller, E. Bernhardsson, and A. Faithfull. Ann-benchmarks: A benchmarking tool for approximate nearest neighbor algorithms. In International Conference on Similarity Search and Applications, pages 34–49. Springer, 2017.

M. Bawa, T. Condie, and P. Ganesan. Lsh forest: self-tuning indexes for similarity search. In Proceedings of the 14th international conference on World Wide Web, pages 651–660, 2005.

R. Bellman. Dynamic programming. Science, 153(3731):34–37, 1966.

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.

E. Berglund, J. Maaskola, N. Schultz, S. Friedrich, M. Marklund, J. Bergenstråhle, F. Tarish, A. Tanoglidi, S. Vickovic, L. Larsson, et al. Spatial maps of prostate cancer transcriptomes reveal an unexplored landscape of heterogeneity. Nature communications, 9(1):1–13, 2018.

E. Bernhardsson. `https://github.com/spotify/annoy`, 2015.

M. Bibikova, E. Chudin, A. Arsanjani, L. Zhou, E. W. Garcia, J. Modder, M. Kostelec, D. Barker, T. Downs, J. B. Fan, and J. Wang-Rodriguez. Expression signatures that correlated with Gleason score and relapse in prostate cancer. Genomics, 89(6):666–672, 2007. ISSN 08887543. doi: 10.1016/j.ygeno.2007.02.005.

E. Biçici and D. Yuret. Locally scaled density based clustering. In International conference on adaptive and natural computing algorithms, pages 739–748. Springer, 2007.

J. M. Bland and D. G. Altman. Multiple significance tests: the bonferroni method. Bmj, 310(6973):170, 1995.

A. L. Blum and P. Langley. Artificial Intelligence Selection of relevant features and examples in machine. Artif. Intell., 97(1-2):245–271, 1997.

A. M. Bolger, M. Lohse, and B. Usadel. Trimmomatic: a flexible trimmer for illumina sequence data. Bioinformatics, 30(15):2114–2120, 2014.

B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. Bioinformatics, 19(2):185–193, 2003.

R. Bourgon, R. Gentleman, and W. Huber. Independent filtering increases detection power for high-throughput experiments. Proceedings of the National Academy of Sciences, 107(21):9546–9551, may 2010. ISSN 0027-8424. doi: 10.1073/pnas.0914005107.

F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal. Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries. CA: A Cancer Journal for Clinicians, 68(6):394–424, nov 2018. ISSN 00079235. doi: 10.3322/caac.21492.

N. L. Bray, H. Pimentel, P. Melsted, and L. Pachter. Near-optimal probabilistic rna-seq quantification. Nature biotechnology, 34(5):525–527, 2016.

J. H. Bullard, E. Purdom, K. D. Hansen, and S. Dudoit. Evaluation of statistical methods for normalization and differential expression in mrna-seq experiments. BMC bioinformatics, 11(1):1–13, 2010.

M. J. Bussemakers, A. van Bokhoven, G. W. Verhaegh, F. P. Smit, H. F. Karthaus, J. A. Schalken, F. M. Debruyne, N. Ru, and W. B. Isaacs. DD3: a new prostate-specific gene, highly overexpressed in prostate cancer. Cancer research, 59(23): 5975–9, dec 1999. ISSN 0008-5472.

D. Bzdok. Classical Statistics and Statistical Learning in Imaging Neuroscience. Frontiers in Neuroscience, 11(OCT):1–23, oct 2017. ISSN 1662-453X. doi: 10.3389/fnins.2017.00543.

F. J. Campos-Laborie, A. Risueño, M. Ortiz-Estévez, B. Rosón-Burgo, C. Droste, C. Fontanillo, R. Loos, J. M. Sánchez-Santos, M. W. Trotter, and J. De Las Rivas.

DECO: decompose heterogeneous population cohorts for patient stratification and discovery of sample biomarkers using omic data profiling. Bioinformatics, 35 (19):3651–3662, 03 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz148.

D. Castel, C. Philippe, R. Calmon, L. Le Dret, N. Truffaux, N. Boddaert, M. Pagès, K. R. Taylor, P. Saulnier, L. Lacroix, et al. Histone h3f3a and hist1h3b k27m mutations define two subgroups of diffuse intrinsic pontine gliomas with different prognosis and phenotypes. Acta neuropathologica, 130(6):815–827, 2015.

N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. Journal of artificial intelligence research, 16: 321–357, 2002.

C. D. Chen, D. S. Welsbie, C. Tran, S. H. Baek, R. Chen, R. Vessella, M. G. Rosenfeld, and C. L. Sawyers. Molecular determinants of resistance to antiandrogen therapy. Nature medicine, 10(1):33–39, 2004.

C.-L. Chen, D. Mahalingam, P. Osmulski, R. R. Jadhav, C.-M. Wang, R. J. Leach, T.-C. Chang, S. D. Weitman, A. P. Kumar, L. Sun, et al. Single-cell analysis of circulating tumor cells identifies cumulative expression patterns of emt-related genes in metastatic prostate cancer. The Prostate, 73(8):813–826, 2013.

W.-Y. Chen, Y.-C. Tsai, H.-L. Yeh, F. Suau, K.-C. Jiang, A.-N. Shao, J. Huang, and Y.-N. Liu. Loss of spdef and gain of tgfbi activity after androgen deprivation therapy promote emt and bone metastasis of prostate cancer. Science Signaling, 10(492):eaam6826, 2017.

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, 2020.

A. Conesa, P. Madrigal, S. Tarazona, D. Gomez-Cabrero, A. Cervera, A. McPherson, M. W. Szcześniak, D. J. Gaffney, L. L. Elo, X. Zhang, et al. A survey of best practices for rna-seq data analysis. Genome biology, 17(1):13, 2016.

A. Criscuolo and S. Brisse. Alientrimmer: a tool to quickly and accurately trim off multiple short contaminant sequences from high-throughput sequencing reads. Genomics, 102(5-6):500–506, 2013.

R. R. Curtin, M. Edel, M. Lozhnikov, Y. Mentekidis, S. Ghaisas, and S. Zhang. mlpack 3: a fast, flexible machine learning library. Journal of Open Source Software, 3:726, 2018. doi: 10.21105/joss.00726.

A. V. D'Amico, R. Whittington, S. B. Malkowicz, D. Schultz, K. Blank, G. A. Broderick, J. E. Tomaszewski, A. A. Renshaw, I. Kaplan, C. J. Beard, et al. Biochemical outcome after radical prostatectomy, external beam radiation therapy, or interstitial radiation therapy for clinically localized prostate cancer. Jama, 280 (11):969–974, 1998.

J. J. de Ronde, G. Rigaill, S. Rottenberg, S. Rodenhuis, and L. F. A. Wessels. Identifying subgroup markers in heterogeneous populations. Nucleic Acids Research, 41(21):e200–e200, nov 2013. ISSN 1362-4962. doi: 10.1093/nar/gkt845.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. Journal of the Royal Statistical Society: Series B (Methodological), 39(1):1–22, 1977.

Y. Di, D. W. Schafer, J. S. Cumbie, and J. H. Chang. The NBP negative binomial model for assessing differential gene expression from RNA-Seq. Statistical Applications in Genetics and Molecular Biology, 10(1), 2011. ISSN 15446115. doi: 10.2202/1544-6115.1637.

M.-A. Dillies, A. Rau, J. Aubert, C. Hennequet-Antier, M. Jeanmougin, N. Servant, C. Keime, G. Marot, D. Castel, J. Estelle, et al. A comprehensive evaluation of normalization methods for illumina high-throughput rna sequencing data analysis. Briefings in bioinformatics, 14(6):671–683, 2013.

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.

A. Drouin, S. Giguère, M. Déraspe, M. Marchand, M. Tyers, V. G. Loo, A.-M. Bourgault, F. Laviolette, and J. Corbeil. Predictive computational phenotyping and biomarker discovery using reference-free genome comparisons. BMC Genomics, 17(1):754, dec 2016. ISSN 1471-2164. doi: 10.1186/s12864-016-2889-6.

O. J. Dunn. Multiple comparisons among means. Journal of the American statistical association, 56(293):52–64, 1961.

J. G. Dy, C. E. Brodley, A. Kak, L. S. Broderick, and A. M. Aisen. Unsupervised feature selection applied to content-based retrieval of lung images. IEEE transactions on pattern analysis and machine intelligence, 25(3):373–378, 2003.

L. Ein-Dor, O. Zuk, and E. Domany. Thousands of samples are needed to generate a robust gene list for predicting outcome in cancer. Proceedings of the National Academy of Sciences, 103(15):5923–5928, apr 2006. ISSN 0027-8424. doi: 10.1073/pnas.0601231103.

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.

L. Ertöz, M. Steinbach, and V. Kumar. Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In Proceedings of the 2003 SIAM international conference on data mining, pages 47–58. SIAM, 2003.

M. Ester, H.-P. Kriegel, J. Sander, X. Xu, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd, volume 96-34, pages 226–231, 1996.

J. Fan and J. Lv. Sure independence screening for ultrahigh dimensional feature space. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 70(5):849–911, nov 2008. ISSN 13697412. doi: 10.1111/j.1467-9868.2008.00674.x.

T. Fawcett. An introduction to ROC analysis. Pattern Recognition Letters, 27(8): 861–874, jun 2006. ISSN 01678655. doi: 10.1016/j.patrec.2005.10.010.

J. Ferlay, M. Colombet, I. Soerjomataram, C. Mathers, D. Parkin, M. Piñeros, A. Znaor, and F. Bray. Estimating the global cancer incidence and mortality in 2018: GLOBOCAN sources and methods. International Journal of Cancer, 144 (8):1941–1953, apr 2019. ISSN 0020-7136. doi: 10.1002/ijc.31937.

M. Fraser, V. Y. Sabelnykova, T. N. Yamaguchi, L. E. Heisler, J. Livingstone, V. Huang, Y.-J. Shiah, F. Yousif, X. Lin, A. P. Masella, et al. Genomic hallmarks of localized, non-indolent prostate cancer. Nature, 541(7637):359–364, 2017.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. Journal of Statistical Software, 33(1):1–22, 2010.

A. Gionis, P. Indyk, R. Motwani, et al. Similarity search in high dimensions via hashing. In Vldb, volume 99-6, pages 518–529, 1999.

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, C. D. Bloomfield, and E. S. Lander. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science (New York, N.Y.), 286(5439):531–537, 1999. ISSN 0036-8075. doi: 10.1126/science.286.5439.531.

I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In Advances in neural information processing systems, pages 2672–2680, 2014.

I. Guyon and A. Elisseeff. An introduction to variable and feature selection. Journal of machine learning research, 3(Mar):1157–1182, 2003.

I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. Machine learning, 46(1-3):389–422, 2002.

I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh. Feature extraction: foundations and applications, volume 207. Springer, 2008.

J. Han, J. Pei, and M. Kamber. Data mining: concepts and techniques. Elsevier, 2011.

G. Hannon. http://hannonlab.cshl.edu/fastx_toolkit/, 2010. [Online; accessed 19-July-2008].

A.-C. Haury, P. Gestraud, and J.-P. Vert. The Influence of Feature Selection Methods on Accuracy, Stability and Interpretability of Molecular Signatures. PLoS ONE, 6(12):e28210, dec 2011. ISSN 1932-6203. doi: 10.1371/journal.pone.0028210. URL https://dx.plos.org/10.1371/journal.pone.0028210.

A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. Technometrics, 12(1):55–67, 1970.

K. Hua and X. Zhang. Estimating the total genome length of a metagenomic sample using k-mers. BMC genomics, 20(2):183, 2019.

P. A. Humphrey. Gleason grading and prognostic factors in carcinoma of the prostate. Modern pathology, 17(3):292–306, 2004.

G. James, D. Witten, T. Hastie, and R. Tibshirani. An Introduction to Statistical Learning, volume 103 of Springer Texts in Statistics. Springer New York, New York, NY, 2013. ISBN 978-1-4614-7137-0. doi: 10.1007/978-1-4614-7138-7. URL https://www.springer.com/gp/book/9781461471370{%}0Ahttp://www.springer.com/us/book/9781461471370http://link.springer.com/10.1007/978-1-4614-7138-7.

M. A. Jhun, M. S. Geybels, J. L. Wright, S. Kolb, C. April, M. Bibikova, E. A. Ostrander, J.-B. Fan, Z. Feng, and J. L. Stanford. Gene expression signature of gleason score is associated with prostate cancer outcomes in a radical prostatectomy cohort. Oncotarget, 8(26):43035, 2017.

D. Jiang, C. Tang, and A. Zhang. Cluster analysis for gene expression data: a survey. IEEE Transactions on knowledge and data engineering, 16(11):1370–1386, 2004.

W. B. Johnson and J. Lindenstrauss. Extensions of lipschitz mappings into a hilbert space. Contemporary mathematics, 26(189-206):1, 1984.

E. L. Kaplan and P. Meier. Nonparametric estimation from incomplete observations. Journal of the American statistical association, 53(282):457–481, 1958.

R. J. Karnes, E. J. Bergstralh, E. Davicioni, M. Ghadessi, C. Buerki, A. P. Mitra, A. Crisan, N. Erho, I. A. Vergara, L. L. Lam, R. Carlson, D. J. Thompson, Z. Haddad, B. Zimmermann, T. Sierocinski, T. J. Triche, T. Kollmeyer, K. V. Ballman, P. C. Black, G. G. Klee, and R. B. Jenkins. Validation of a Genomic Classifier that Predicts Metastasis Following Radical Prostatectomy in an At Risk Patient Population. Journal of Urology, 190(6):2047–2053, dec 2013. ISSN 0022-5347. doi: 10.1016/j.juro.2013.06.017.

G. Karypis, E.-H. Han, and V. Kumar. Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8):68–75, 1999.

D. Kim, J. M. Paggi, C. Park, C. Bennett, and S. L. Salzberg. Graph-based genome alignment and genotyping with hisat2 and hisat-genotype. Nature biotechnology, 37(8):907–915, 2019.

E. A. Klein, M. R. Cooperberg, C. Magi-Galluzzi, J. P. Simko, S. M. Falzarano, T. Maddala, J. M. Chan, J. Li, J. E. Cowan, A. C. Tsiatis, D. B. Cherbavaz, R. J. Pelham, I. Tenggara-Hunter, F. L. Baehner, D. Knezevic, P. G. Febbo, S. Shak, M. W. Kattan, M. Lee, and P. R. Carroll. A 17-gene assay to predict prostate cancer aggressiveness in the context of gleason grade heterogeneity, tumor multifocality, and biopsy undersampling. European Urology, 66(3):550–560, 2014. ISSN 18737560. doi: 10.1016/j.eururo.2014.05.004.

E. A. Klein, K. Yousefi, Z. Haddad, V. Choeurng, C. Buerki, A. J. Stephenson, J. Li, M. W. Kattan, C. Magi-Galluzzi, and E. Davicioni. 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. European Urology, 67(4):778–786, 2015. ISSN 18737560. doi: 10.1016/j.eururo.2014.10.036.

R. Kohavi, G. H. John, et al. Wrappers for feature subset selection. Artificial intelligence, 97(1-2):273–324, 1997.

G. Koutalellis, K. Stravodimos, M. Avgeris, K. Mavridis, A. Scorilas, A. Lazaris, and C. Constantinides. L-dopa decarboxylase (ddc) gene expression is related to outcome in patients with prostate cancer. BJU international, 110(6b):E267–E273, 2012.

M. Kubat, S. Matwin, et al. Addressing the curse of imbalanced training sets: one-sided selection. In Icml, volume 97, pages 179–186. Citeseer, 1997.

M. Kuhn. Building predictive models in r using the caret package. Journal of Statistical Software, Articles, 28(5):1–26, 2008. ISSN 1548-7660. doi: 10.18637/jss.v028.i05.

B. Langmead, C. Trapnell, M. Pop, and S. L. Salzberg. Ultrafast and memory-efficient alignment of short dna sequences to the human genome. Genome biology, 10(3):R25, 2009.

A. Latil, I. Bièche, L. Chêne, I. Laurendeau, P. Berthon, O. Cussenot, and M. Vidaud. Gene Expression Profiling in Clinically Localized Prostate Cancer: A Four-Gene Expression Model Predicts Clinical Behavior. Clinical Cancer Research, 9 (15):5477–5485, 2003. ISSN 10780432.

C. W. Law, Y. Chen, W. Shi, and G. K. Smyth. Voom: Precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biology, 15(2): 1–17, 2014. ISSN 1474760X. doi: 10.1186/gb-2014-15-2-r29.

C. Lazar, J. Taminau, S. Meganck, D. Steenhoff, A. Coletta, C. Molter, V. de Schaetzen, R. Duque, H. Bersini, and A. Nowe. A Survey on Filter Techniques for Feature Selection in Gene Expression Microarray Analysis. IEEE/ACM

Transactions on Computational Biology and Bioinformatics, 9(4):1106–1119, jul 2012. ISSN 1545-5963. doi: 10.1109/TCBB.2012.33.

G. Lemaître, F. Nogueira, and C. K. Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. The Journal of Machine Learning Research, 18(1):559–563, 2017.

J. Lever, M. Krzywinski, and N. Altman. Points of significance: model selection and overfitting, 2016.

G. H. Leyten, D. Hessels, F. P. Smit, S. A. Jannink, H. de Jong, and W. J. Melchers. Identification of a candidate gene panel for the early diagnosis of prostate cancer. Clinical Cancer Research, 21(13):3061–3070, 2015.

B. Li and C. N. Dewey. Rsem: accurate transcript quantification from rna-seq data with or without a reference genome. BMC bioinformatics, 12(1):323, 2011.

L. Li, D. M. Umbach, P. Terry, and J. A. Taylor. Application of the ga/knn method to seldi proteomics data. Bioinformatics, 20(10):1638–1640, 2004.

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.

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

J. Liu, J.-X. Shen, H.-T. Wu, X.-L. Li, X.-F. Wen, C.-W. Du, and G.-J. Zhang. Collagen 1a1 (col1a1) promotes metastasis of breast cancer and is a potential therapeutic target. Discovery medicine, 25(139):211–223, 2018b.

X. Liu, T. R. Grogan, H. Hieronymus, T. Hashimoto, J. Mottahedeh, D. Cheng, L. Zhang, K. Huang, T. Stoyanova, J. W. Park, et al. Low cd38 identifies

progenitor-like inflammation-associated luminal cells that can initiate human prostate cancer and predict poor outcome. Cell reports, 17(10):2596–2606, 2016.

Q. Long, J. Xu, A. O. Osunkoya, S. Sannigrahi, B. A. Johnson, W. Zhou, T. Gillespie, J. Y. Park, R. K. Nam, L. Sugar, A. Stanimirovic, A. K. Seth, J. A. Petros, and C. S. Moreno. Global Transcriptome Analysis of Formalin-Fixed Prostate Cancer Specimens Identifies Biomarkers of Disease Recurrence. Cancer Research, 74(12):3228–3237, jun 2014. ISSN 0008-5472. doi: 10.1158/0008-5472.CAN-13-2699.

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):550, 2014.

G. Lunter and M. Goodson. Stampy: a statistical algorithm for sensitive and fast mapping of illumina sequence reads. Genome research, 21(6):936–939, 2011.

J. MacQueen et al. Some methods for classification and analysis of multivariate observations. In Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, volume 1, pages 281–297. Oakland, CA, USA, 1967.

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.

M. Martin. Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnet.journal, 17(1):10–12, 2011. ISSN 2226-6089. doi: 10.14806/ej.17.1.200.

Y. Z. Mazzu, J. Armenia, G. Chakraborty, Y. Yoshikawa, A. C. Si'Ana, S. Nandakumar, T. A. Gerke, M. M. Pomerantz, X. Qiu, H. Zhao, et al. A novel mechanism driving poor-prognosis prostate cancer: overexpression of the dna repair gene, ribonucleotide reductase small subunit m2 (rrm2). Clinical Cancer Research, 25 (14):4480–4492, 2019.

N. Meinshausen and P. Bühlmann. Stability selection. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 72(4):417–473, jul 2010. ISSN 13697412. doi: 10.1111/j.1467-9868.2010.00740.x.

G. Menardi and N. Torelli. Training and assessing classification rules with imbalanced data. Data Mining and Knowledge Discovery, 28(1):92–122, 2014.

S. Michiels, S. Koscielny, and C. Hill. Prediction of cancer outcome with microarrays: a multiple random validation strategy. The Lancet, 365(9458):488–492, feb 2005. ISSN 01406736. doi: 10.1016/S0140-6736(05)17866-0.

S. Michiels, S. Koscielny, and C. Hill. Interpretation of microarray data in cancer. British Journal of Cancer, 96(8):1155–1158, apr 2007. ISSN 0007-0920. doi: 10.1038/sj.bjc.6603673.

S. Mo, L. Zhang, W. Dai, L. Han, R. Wang, W. Xiang, Z. Wang, Q. Li, J. Yu, J. Yuan, et al. Antisense lncrna ldlrad4-as1 promotes metastasis by decreasing the expression of ldlrad4 and predicts a poor prognosis in colorectal cancer. Cell death & disease, 11(2):1–16, 2020.

A. Morillon and D. Gautheret. Bridging the gap between reference and real transcriptomes. Genome biology, 20(1):1–7, 2019.

A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold. Mapping and quantifying mammalian transcriptomes by rna-seq. Nature methods, 5(7): 621–628, 2008.

J. Munkley, U. L. McClurg, K. E. Livermore, I. Ehrmann, B. Knight, P. Mccullagh, J. Mcgrath, M. Crundwell, L. W. Harries, H. Y. Leung, et al. The cancer-associated cell migration protein tspan1 is under control of androgens and its upregulation increases prostate cancer cell migration. Scientific reports, 7(1): 1–11, 2017.

U. Nagalakshmi, Z. Wang, K. Waern, C. Shou, D. Raha, M. Gerstein, and M. Snyder. The transcriptional landscape of the yeast genome defined by rna sequencing. Science, 320(5881):1344–1349, 2008.

S. Nagaraja, N. A. Vitanza, P. J. Woo, K. R. Taylor, F. Liu, L. Zhang, M. Li, W. Meng, A. Ponnuswami, W. Sun, et al. Transcriptional dependencies in diffuse intrinsic pontine glioma. Cancer cell, 31(5):635–652, 2017.

N. A. O'Leary, M. W. Wright, J. R. Brister, S. Ciufo, D. Haddad, R. McVeigh, B. Rajput, B. Robbertse, B. Smith-White, D. Ako-Adjei, et al. Reference sequence (refseq) database at ncbi: current status, taxonomic expansion, and functional annotation. Nucleic acids research, 44(D1):D733–D745, 2016.

R. Ounit, S. Wanamaker, T. J. Close, and S. Lonardi. CLARK: fast and accurate classification of metagenomic and genomic sequences using discriminative k-mers. BMC Genomics, 16(1):236, dec 2015. ISSN 1471-2164. doi: 10.1186/s12864-015-1419-2.

H.-S. Park and C.-H. Jun. A simple and fast algorithm for k-medoids clustering. Expert systems with applications, 36(2):3336–3341, 2009.

J. S. Parker, M. Mullins, M. C. Cheang, S. Leung, D. Voduc, T. Vickery, S. Davies, C. Fauron, X. He, Z. Hu, et al. Supervised risk predictor of breast cancer based on intrinsic subtypes. Journal of clinical oncology, 27(8):1160, 2009.

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.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12:2825–2830, 2011.

K. L. Penney, J. A. Sinnott, K. Fall, Y. Pawitan, Y. Hoshida, P. Kraft, J. R. Stark, M. Fiorentino, S. Perner, S. Finn, et al. mrna expression signature of gleason grade predicts lethal prostate cancer. Journal of Clinical Oncology, 29(17):2391, 2011.

C. M. Perou, T. SÃžrlie, M. B. Eisen, M. van de Rijn, S. S. Jeffrey, C. A. Rees, J. R. Pollack, D. T. Ross, H. Johnsen, L. A. Akslen, O. Fluge, A. Pergamenschikov, C. Williams, S. X. Zhu, P. E. LÃžnning, A. L. BÃžrresen-Dale, P. O. Brown, and

D. Botstein. Molecular portraits of human breast tumours. Nature, 406(6797): 747–752, 2000. ISSN 0028-0836. doi: 10.1038/35021093.

M. Pertea, G. M. Pertea, C. M. Antonescu, T.-C. Chang, J. T. Mendell, and S. L. Salzberg. Stringtie enables improved reconstruction of a transcriptome from rna-seq reads. Nature biotechnology, 33(3):290–295, 2015.

E. F. Petricoin III, A. M. Ardekani, B. A. Hitt, P. J. Levine, V. A. Fusaro, S. M. Steinberg, G. B. Mills, C. Simone, D. A. Fishman, E. C. Kohn, et al. Use of proteomic patterns in serum to identify ovarian cancer. The lancet, 359(9306): 572–577, 2002.

P. A. Pevzner. 1-tuple dna sequencing: Computer analysis. Journal of Biomolecular Structure and Dynamics, 7(1):63–73, 1989. doi: 10.1080/07391102.1989.10507752. URL https://doi.org/10.1080/07391102.1989.10507752. PMID: 2684223.

M. Pinskaya, Z. Saci, M. Gallopin, M. Gabriel, H. T. Nguyen, V. Firlej, M. Descrimes, A. Rapinat, D. Gentien, A. De La Taille, A. Londoño-Vallejo, Y. Allory, D. Gautheret, and A. Morillon. Reference-free transcriptome exploration reveals novel RNAs for prostate cancer diagnosis. Life Science Alliance, 2(6):1–12, 2019. ISSN 25751077. doi: 10.26508/lsa.201900449.

H. Pirim, B. Ekşioğlu, A. D. Perkins, and Ç. Yüceer. Clustering of high throughput gene expression data. Computers & operations research, 39(12):3046–3061, 2012.

J. Rainer. EnsDb.Hsapiens.v79: Ensembl based annotation package, 2017. R package version 2.99.0.

S. Ren, G.-H. Wei, D. Liu, L. Wang, Y. Hou, S. Zhu, L. Peng, Q. Zhang, Y. Cheng, H. Su, et al. Whole-genome and transcriptome sequencing of prostate cancer identify new genetic alterations driving disease progression. European urology, 73(3):322–339, 2018.

M. T. Ribeiro, S. Singh, and C. Guestrin. " why should i trust you?" explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD

international conference on knowledge discovery and data mining, pages 1135–1144, 2016.

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, 2015. ISSN 13624962. doi: 10.1093/nar/gkv007.

A. Roberts, H. Pimentel, C. Trapnell, and L. Pachter. Identification of novel transcripts in annotated genomes using rna-seq. Bioinformatics, 27(17):2325–2329, 2011.

J. T. Robinson, H. Thorvaldsdóttir, W. Winckler, M. Guttman, E. S. Lander, G. Getz, and J. P. Mesirov. Integrative genomics viewer. Nature biotechnology, 29(1):24–6, jan 2011. ISSN 1546-1696. doi: 10.1038/nbt.1754.

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 and G. K. Smyth. Moderated statistical tests for assessing differences in tag abundance. Bioinformatics, 23(21):2881–2887, 2007.

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.

T. Ronan, Z. Qi, and K. M. Naegle. Avoiding common pitfalls when clustering biological data. Science signaling, 9(432):re6–re6, 2016.

F. H. Schröder, J. Hugosson, M. J. Roobol, T. L. Tammela, S. Ciatto, V. Nelen, M. Kwiatkowski, M. Lujan, H. Lilja, M. Zappa, et al. Screening and prostate-cancer mortality in a randomized european study. New England Journal of Medicine, 360(13):1320–1328, 2009.

A. Shahabi, J. P. Lewinger, J. Ren, C. April, A. E. Sherrod, J. G. Hacia, S. Daneshmand, I. Gill, J. K. Pinski, J.-B. Fan, and M. C. Stern. Novel Gene Expression

Signature Predictive of Clinical Recurrence After Radical Prostatectomy in Early Stage Prostate Cancer Patients. The Prostate, 76(14):1239–1256, oct 2016. ISSN 02704137. doi: 10.1002/pros.23211.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. Cancer Cell, 1(2):203–209, 2002. ISSN 15356108. doi: 10.1016/S1535-6108(02)00030-2.

A. Sinha, V. Huang, J. Livingstone, J. Wang, N. S. Fox, N. Kurganovs, V. Ignatchenko, K. Fritsch, N. Donmez, L. E. Heisler, et al. The proteogenomic landscape of curable prostate cancer. Cancer Cell, 35(3):414–427, 2019.

J. A. Sinnott, S. F. Peisch, S. Tyekucheva, T. Gerke, R. Lis, J. R. Rider, M. Fiorentino, M. J. Stampfer, L. A. Mucci, M. Loda, et al. Prognostic utility of a new mrna expression signature of gleason score. Clinical Cancer Research, 23 (1):81–87, 2017.

G. K. Smyth. Linear Models and Empirical Bayes Methods for Assessing Differential Expression in Microarray Experiments. Statistical Applications in Genetics and Molecular Biology, 3(1):1–28, February 2004.

C. Soneson, M. I. Love, and M. D. Robinson. Differential analyses for rna-seq: transcript-level estimates improve gene-level inferences. F1000Research, 4, 2015.

S. Stelloo, E. Nevedomskaya, Y. Kim, K. Schuurman, E. Valle-Encinas, J. Lobo, O. Krijgsman, D. S. Peeper, S. L. Chang, F. Y.-C. Feng, et al. Integrative epigenetic taxonomy of primary prostate cancer. Nature communications, 9(1): 1–12, 2018.

A. Thomas, S. Barriere, L. Broseus, J. Brooke, C. Lorenzi, J.-p. Villemin, G. Beurier, R. Sabatier, C. Reynes, A. Mancheron, and W. Ritchie. GECKO is a genetic algorithm to classify and explore high throughput sequencing

data. Communications Biology, 2(1):222, dec 2019. ISSN 2399-3642. doi: 10.1038/s42003-019-0456-9.

R. Tibshirani. Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58(1):267–288, 1996.

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.

L. J. van 't Veer, H. Dai, M. J. van de Vijver, Y. D. He, A. A. M. Hart, M. Mao, H. L. Peterse, K. van der Kooy, M. J. Marton, A. T. Witteveen, G. J. Schreiber, R. M. Kerkhoven, C. Roberts, P. S. Linsley, R. Bernards, and S. H. Friend. Gene expression profiling predicts clinical outcome of breast cancer. Nature, 415(6871): 530–536, jan 2002. ISSN 0028-0836. doi: 10.1038/415530a.

D. Venet, J. E. Dumont, and V. Detours. Most random gene expression signatures are significantly associated with breast cancer outcome. PLoS computational biology, 7(10):e1002240, 2011. ISSN 1553-7358. doi: 10.1371/journal.pcbi.1002240.

S. Wan, M. Xi, H.-B. Zhao, W. Hua, Y.-L. Liu, Y.-L. Zhou, Y.-J. Zhuo, Z.-Z. Liu, Z.-D. Cai, Y.-P. Wan, et al. Hmgcs2 functions as a tumor suppressor and has a prognostic impact in prostate cancer. Pathology-Research and Practice, 215(8): 152464, 2019.

L. Wang, Z. Feng, X. Wang, X. Wang, and X. Zhang. Degseq: an r package for identifying differentially expressed genes from rna-seq data. Bioinformatics, 26 (1):136–138, 2010.

T. Wang, Z. Liu, S. Guo, L. Wu, M. Li, J. Yang, R. Chen, H. Xu, S. Cai, H. Chen, et al. The tumor suppressive role of camk2n1 in castration-resistant prostate cancer. Oncotarget, 5(11):3611, 2014.

W. Wang, J. Yang, R. Muntz, et al. Sting: A statistical information grid approach to spatial data mining. In VLDB, volume 97, pages 186–195, 1997.

F. Wilcoxon. Individual Comparisons by Ranking Methods. Biometrics Bulletin, 1 (6):80, dec 1945. ISSN 00994987. doi: 10.2307/3001968.

D. M. Witten and R. Tibshirani. Survival analysis with high-dimensional covariates. Statistical Methods in Medical Research, 19(1):29–51, feb 2010a. ISSN 0962-2802. doi: 10.1177/0962280209105024.

D. M. Witten and R. Tibshirani. Survival analysis with high-dimensional covariates. Statistical Methods in Medical Research, 19(1):29–51, feb 2010b. ISSN 0962-2802. doi: 10.1177/0962280209105024.

D. E. Wood and S. L. Salzberg. Kraken: ultrafast metagenomic sequence classification using exact alignments. Genome Biology, 15(3):R46, 2014. ISSN 1465-6906. doi: 10.1186/gb-2014-15-3-r46.

X. Wu, H. Wang, Y. Lian, L. Chen, L. Gu, J. Wang, Y. Huang, M. Deng, Z. Gao, and Y. Huang. Gtse1 promotes cell migration and invasion by regulating emt in hepatocellular carcinoma and is associated with poor prognosis. Scientific reports, 7(1):1–12, 2017.

W. Xie, H. Xiao, J. Luo, L. Zhao, F. Jin, J. Ma, J. Li, K. Xiong, C. Chen, and G. Wang. Identification of low-density lipoprotein receptor class a domain containing 4 (ldlrad4) as a prognostic indicator in primary gastrointestinal stromal tumors. Current Problems in Cancer, page 100593, 2020.

D. Xu and Y. Tian. A comprehensive survey of clustering algorithms. Annals of Data Science, 2(2):165–193, 2015.

N. Yoosuf, J. F. Navarro, F. Salmén, P. L. Ståhl, and C. O. Daub. Identification and transfer of spatial transcriptomics signatures for cancer diagnosis. Breast Cancer Research, 22(1):1–10, 2020.

T. Zhang, R. Ramakrishnan, and M. Livny. Birch: A new data clustering algorithm and its applications. Data Mining and Knowledge Discovery, 1(2):141–182, 1997.

S. Zhao, L. Xi, and B. Zhang. Union exon based approach for rna-seq gene quantification: To be or not to be? PLoS One, 10(11):e0141910, 2015.

W.-D. Zhong, Y.-X. Liang, Y.-K. Liang, Y.-J. Zhuo, J.-H. Ye, X.-J. Zhu, Z.-D. Cai, Z.-Y. Lin, J.-G. Zhu, S.-L. Wu, et al. Tumor suppressor role and clinical implication of the fifth ewing variant (fev) gene, an ets family gene, in prostate cancer. Prostate Cancer (April 15, 2019), 2019.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67 (2):301–320, 2005. doi: 10.1111/j.1467-9868.2005.00503.x.

List of Figures

List of Tables

Acronyms

ANN **A**pproximate **N**earest **N**eighbors. 74, 75, 78, 80

ANNOY **A**pproximate **N**earest **N**eighbors **O**h **Y**eah. 74, 75, 76, 77, 79, 80, 81, 82, 83, 84, 86, 187, 190

AUC **A**re **U**nder the ROC **C**urve. 23, 29, 58, 70, 71, 94, 97, 122, 129, 133, 188

BAM **B**inary **A**lignment **M**ap. 39, 40, 49

BCR **B**io**c**hemical **R**ecurrence. 53, 115, 118

BH **B**enjamin-**H**ochberg. 8, 46, 69

CoxPH **Cox** **P**roportional **H**azards. 27, 55

CV **c**ross-**v**alidation. 14, 15, 21, 99, 122, 124, 125, 128, 133, 188

DBSCAN **D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise. vi, 30, 31, 32, 33, 34, 74, 75, 81, 82, 83, 84, 85, 86, 87, 88, 149, 185, 187, 190

DE **d**ifferential **e**xpression. 3, 4, 5, 10, 46, 48, 49, 57, 64, 65, 69, 73, 88, 185, 186

DGE **d**ifferential **g**ene **e**xpression. iv, 9, 10

DIPG **D**iffuse **I**ntrinsic **P**ontine **G**lioma. 70, 71, 72, 186

DNA **D**eoxyribo**n**ucleic **A**cid. 2, 3, 43, 70, 115

EM **E**xpectation **M**aximization. 40, 41

FDR **F**alse **D**iscovery **R**ate. 8

FPKM **F**ragments **P**er **K**ilobase **M**illion. 42

FPR **F**alse **P**ositive **R**ate. 23, 70

FWER **F**amily-**w**ise **e**rror **r**ate. 7, 21

GA **G**enetic **A**lgorithm. 19, 49, 99, 136

GAN **G**enerative **A**dversarial **N**etworks. 25

GS **G**leason **s**core. 53, 54, 55

GTF **G**ene **T**ransfer **F**ormat. 40

LASSO **L**east **A**bsolute **S**hrinkage and **S**election **O**perator. 15, 16, 20, 28, 55, 92, 93, 97, 99, 122, 128, 158, 188

LIME **L**ocal **I**nterpretable **M**odel-Agnostic **E**xplanations. 27

lincRNA **l**ong **i**ntergenic **n**on-**c**oding **RNA**. 47

lncRNA **l**ong **n**on-**c**oding **RNA**. 45, 55, 59, 60, 91, 93, 94, 97, 114, 116, 129, 131, 136, 149

LSDBC **L**ocally **S**caled **D**ensity-based **C**lustering. 149, 161, 162

LSH **L**ocal **S**ensitive **H**ashing. 77, 78, 187

LSHF **L**ocal **S**ensitive **H**ashing **F**orest. 74, 75, 77, 78, 79, 80, 81, 82, 190

MAC **M**ean **A**bsolute **C**ontrast. 88, 120, 123, 188

MAD **M**edian **A**bsolute **D**eviation. 66, 67, 71, 72, 148, 161

ML **m**achine **l**earning. 11, 13, 14, 19, 22, 28, 29, 44, 50, 55, 57, 136

mRNA **m**essenger **RNA**. 48, 55

MSE **M**ean **S**quared **E**rror. 15, 21, 29

NB **n**egative **b**inomial. 9, 10, 46

PCa **p**rostate **ca**ncer. v, vii, 50, 51, 52, 53, 54, 55, 58, 59, 60, 89, 91, 92, 99, 115, 116, 127, 128, 131, 132, 136, 138, 149, 151

PSA **p**rostate-**s**pecific **a**ntigen. 51, 53, 54, 55, 115, 118, 131

RFE **R**ecursive **F**eature **E**limination. 19

RNA **R**ibo**n**ucleic **A**cid. v, 2, 43, 44, 45, 54, 55, 59, 60, 91, 93, 98, 114, 116, 117, 127, 136, 138, 150

RNA-seq **RNA seq**uencing. v, 2, 3, 4, 7, 9, 10, 35, 37, 40, 42, 44, 48, 49, 50, 54, 57, 59, 60, 68, 70, 79, 91, 92, 97, 98, 99, 114, 115, 116, 117, 118, 120, 138, 149, 185, 190

ROC **R**eceiver **O**perating **C**haracteristics. 23, 29, 70, 71, 94, 97

RPKM **R**eads **P**er **K**ilobase **M**illion. 41, 42

RSS **r**esidual **s**um of **s**quares. 12, 16

TNM **T**umour, **N**ode, **M**etastasis. 52, 53, 115, 118

TPM **T**ranscripts **P**er **M**illion. 42, 43

TPR **T**rue **P**ositive **R**ate. 23, 70

université PARIS-SACLAY

ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Titre : Combiner apprentissage automatique et analyse transcriptomique sans référence pour l'identification de signatures du cancer de la prostate

Mots clés : Apprentissage automatique, transcriptome, k-mers de séquences, cancer de la prostate

Résumé : Par sa capacité à capturer la diversité complète des transcrits produits par chaque cellule, la technologie de séquençage d'ARN à haut-débit (RNA-seq) a révolutionné notre vision de l'expression des gènes. Les données RNA-seq sont de plus en plus utilisées en médecine de précision afin d'établir les profils moléculaires des tumeurs, ou pour étudier des réseaux de gènes régissant l'adaptation d'une cellule à son environnement. Cependant, l'analyse RNA-seq qui classiquement se base sur la comparaison avec des séquences géniques de référence, est incapable d'identifier une grande part des ARN aberrants produits dans les maladies par altération du génome ou des processus de maturation.

Notre projet vise à exploiter un nouveau concept pour l'analyse du transcriptome fondé sur des "étiquettes", ou k-mers, représentant l'intégralité des variations de séquences observées dans un transcriptome. Nous avons appliqué ce concept à la découverte de signatures diagnostiques ou pronostiques à partir de données RNA-seq du cancer de la prostate. A cette fin, nous avons appliqué différentes méthodes de réduction de dimension et de sélection de variable utilisées dans l'analyse transcriptomique classique. En raison de la très grande dimension des matrices de k-mers, ces méthodes ont nécessité des adaptations afin de réduire de manière drastique le nombre de variables à analyser.

Nous sommes parvenu à établir un protocole informatique capable de réduire efficacement une matrice de k-mers issue du séquençage de plusieurs centaines de transcriptomes. A l'aide de ce protocole, nous avons pû produire de nouvelles signatures diagnostiques et pronostiques pour le cancer de la prostate. Ces signatures "sans référence" ne nécessitent pas de connaissance a priori sur le génome ou le transcriptome humain et sont au moins aussi performantes que les signatures géniques conventionnelles. De plus ces signatures contiennent des séquences d'ARN jamais identifiées, correspondant notamment à des variants d'ARNm ou à de nouveaux longs ARN non-codants qui pourront orienter les biologistes vers de nouveaux mécanismes d'oncogénèse.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France

université PARIS-SACLAY

ÉCOLE DOCTORALE
Structure et dynamique des systèmes vivants (SDSV)

Title : Combining machine learning and reference-free transcriptome analysis for the identification of prostate cancer signatures

Keywords : Machine learning, Transcriptome, k-mers sequence, prostate cancer

Abstract : With its ability to capture the full diversity of transcripts produced by each cell, high-throughput RNA sequencing (RNA-seq) has revolutionized our view of gene expression. RNA-seq data are increasingly used in precision medicine to establish the molecular profiles of tumors, or to study gene networks governing the adaptation of a cell to its environment. However, RNA-seq analysis, which is conventionally based on comparison with reference gene sequences, is unable to identify a large fraction of abnormal RNA transcripts produced in disease tissues, through defects in the genome or in RNA processing.

Our project aims to exploit a new concept for the analysis of transcriptomes based on short sequence labels, or k-mers, representing all of the sequence variations observed in a given transcriptome dataset. We applied this concept to the discovery of diagnostic or prognostic signatures from RNA-seq data of prostate cancer. To this end, we applied different dimension reduction and variable selection methods used in classical transcriptomic analysis. Due to the very large dimension of the k-mer matrices, these methods required specific adaptations in order to drastically reduce the number of variables to be analyzed.

We established a computer pipeline capable of effectively reducing a k-mer matrix obtained from the sequencing of several hundred transcriptomes. Using this pipeline, we were able to produce new diagnostic and prognostic signatures for prostate cancer. These "reference-free" signatures do not require a priori knowledge of the human genome or transcriptome and are at least as effective as conventional gene signatures. In addition, these signatures contain novel RNA sequences corresponding to mRNA variants or new long non-coding RNAs. These novel RNAs involved in cancer risk may orient biologists towards new oncogenesis mechanisms.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France