9-6-2020

Tess van Rens and Gino Raaijmakers

Version 1

Prediction model for cancer through cfDNA fragmentation patterns

Minor Bioinformatics

**Prediction model for cancer detection by cfDNA fragmentations patterns**

**Student Student nr. E-mail**

Tess van Rens 2114576 ta.vanrens@student.avans.nl

Gino Raaijmakers 2058347 gja.raaijmakers@student.avans.nl

**Supervisor (external commissioner)**

*Name E-mail*

Dr. Marleen Gosens mjem.gosens@avans.nl

Dr. Tim Hulsen tim.hulsen@philips.com

**Supervisor (Avans University of Applied Sciences)**

*Name E-mail*

Dr. Miaomiao Zhou [m.zhou1@avans.nl](mailto:m.zhou1@avans.nl)

# Abstract

A liquid biopsy is a non-invasive way to diagnose cancer. In this procedure, the cell-free DNA (cfDNA) is isolated from the blood plasma and sequenced. We tested a program called DNA evaluation of fragments for early interception (DELFI), which examines fragmentation profiles from sequencing data. The machine learning algorithm can be trained to separate the fragmentation profiles into cancer and non-cancer groups. The algorithm can also differentiate between the origins of cancer tissue, indicating the location of cancer.

Three different datasets, including the original dataset from the Cristiano et al. paper, were run through the DELFI program. The quality of these datasets was tested and varied greatly between the sets. The program was evaluated script by script, to test the diagnostic capabilities. At the fifth script, it was not capable to join the data of a sample reference to the processed data from the set. More script editing and testing are needed to make DELFI reach its potential.

Index

[Abstract 2](#_Toc42596556)

[1. Introduction 4](#_Toc42596557)

[2. Methods 5](#_Toc42596558)

[2.1 Data generation 5](#_Toc42596559)

[2.2 Data collection 5](#_Toc42596560)

[2.3 Implementation 6](#_Toc42596561)

[2.3.1 Processing the Snyder dataset 6](#_Toc42596562)

[2.3.2 Processing the Hao dataset 7](#_Toc42596563)

[2.3.3 the Cristiano dataset 7](#_Toc42596564)

[2.3.4 The DELFI scripts 8](#_Toc42596565)

[3. Results 9](#_Toc42596566)

[3.1 Quality of the datasets 9](#_Toc42596567)

[3.1.1 FastQC 9](#_Toc42596568)

[3.1.2 BAMQC 9](#_Toc42596569)

[3.2 Human reference genome hg19 or hg38 10](#_Toc42596570)

[3.3 Load gaps missing 11](#_Toc42596571)

[3.4 Task ID change 11](#_Toc42596572)

[3.5 Final error messages 12](#_Toc42596573)

[3.5.1 Cristiano dataset 12](#_Toc42596574)

[3.5.2 Snyder and Hao datasets 13](#_Toc42596575)

[4. Discussion and Conclusion 14](#_Toc42596576)

[4.1 Discussion 14](#_Toc42596577)

[4.1.1 Missing code in the DELFI scripts 14](#_Toc42596578)

[4.1.2 Array jobs system in two out of the seven DELFI scripts 14](#_Toc42596579)

[4.1.3 Size Cristiano data set 14](#_Toc42596580)

[4.1.4 Quality of the datasets 15](#_Toc42596581)

[4.1.5 Accuracy of the DELFI algorithm 15](#_Toc42596582)

[4.1.6 Recommendations 15](#_Toc42596583)

[4.2 Conclusion 16](#_Toc42596584)

[4.3 Availability of data and materials 16](#_Toc42596585)

[4.4 Acknowledgments 16](#_Toc42596586)

[References 17](#_Toc42596587)

# 1. Introduction

Early detection of cancer is crucial for a successful treatment, unfortunately, most new tools for early detection suffer from insufficient clinical and technical validation.1

Cell-free DNA (cfDNA) is a potential non-invasive biomarker for early diagnosis of patients with cancer. It can be isolated from patient-derived blood samples and contains DNA fragments released from tumor cells or cells going through apoptosis.2 Obtaining cfDNA through isolation from blood samples has proven to be challenging due to only small amounts of circulating cfDNA, and often already degraded.3 Because of this, cancer detection methods by cfDNA need to be very sensitive and specific. Machine learning methods have become a popular tool for medical researchers. These techniques can discover and identify patterns and relationships in complex datasets. Predicting premature cancer types and future outcomes with high precision could be within reach, because of machine learning.4

Previous research indicated that screening of fragmentation patterns of cfDNA with machine learning models is a good method to distinguish premature cancer patients from healthy individuals.5,6 To be able to detect these differences Cristiano et al.6 developed a machine learning approach called the DNA evaluation of fragments for early interception (DELFI). DELFI can detect abnormalities of cfDNA fragmentation profiles at the genome-wide level with a specificity up to 95%.7 This study was initiated to investigate the DELFI algorithm, and whether it could be reproduced with the data set used in the Cristiano et al. paper. At the same time, DELFI was used for other cfDNA datasets from Snyder et al.8 and Hao et al.9 to see if DELFI could be used for other cfDNA datasets that still need cancer diagnosis.

The methods and the implementation of the tools are presented in chapter two. Chapter three presents the results of the methods that were performed during this research. The following subjects are discussed: Quality of the datasets, Human reference genomes hg19 or hg38, Load gaps missing, Task ID change, and Final error messages. The discussion and conclusion are described in chapter four. References used throughout the report are included at the end of this document.

# 2. Methods

## 2.1 Data generation

The original data set of the Cristiano et al. paper6 was used to verify if the DELFI algorithms can be re-performed. At the same time, the DELFI algorithms were tested for other datasets from Snyder et al.8 and Hao et al.9. All the sequence data was generated by an Illumina platform (Illumina HiSeq 2000/2500) and will already be processed by the software Illumina CASAVA (Consensus Assessment of Sequence and Variation), including demultiplexing and masking of dual-index adapter sequences.

## 2.2 Data collection

The Cristiano et al. dataset was obtained via the European Genome-phenome Archive (EGA) (Dataset ID EGAD00001005339) after a data access agreement. This dataset contained cfDNA extracted from blood plasma samples from healthy individuals and cfDNA extracted from blood plasma samples from patients with breast, lung, ovarian, colorectal, bile duct, or gastric cancer. Healthy individuals were considered healthy if they had no previous history of cancer and negative screening results.

The Snyder et al. dataset was obtained via the National Center of Biotechnology Information (NCBI) short read archive (SRA) (project number PRJNA291063). This dataset consisted of cfDNA extracted from pooled blood plasma samples of patients with the same type of cancer and separately cfDNA extracted from pooled blood plasma samples of healthy individuals. This made sure it didn’t have to abide by privacy limitations.

The Hao et al. dataset was also obtained via the NCBI SRA (project number PRJNA383370). This dataset consisted of cfDNA extracted from pooled blood serum samples of lung cancer patients.

## 2.3 Implementation

The dataset from Cristiano et al. is the dataset for which the DELFI scripts were created. In the dataset, there are 537 bam files of individual patients, aligned to the GRCh37/hg19 human reference genome maintained by the University of California Santa Cruz (USCS)10. The datasets from both Snyder et al.8 and Hao et al.9 are publicly available but are in fastq format. To use the DELFI scripts with all three datasets these fastq files have to be processed to bam files first, following the steps described in the flowchart (figure 1).

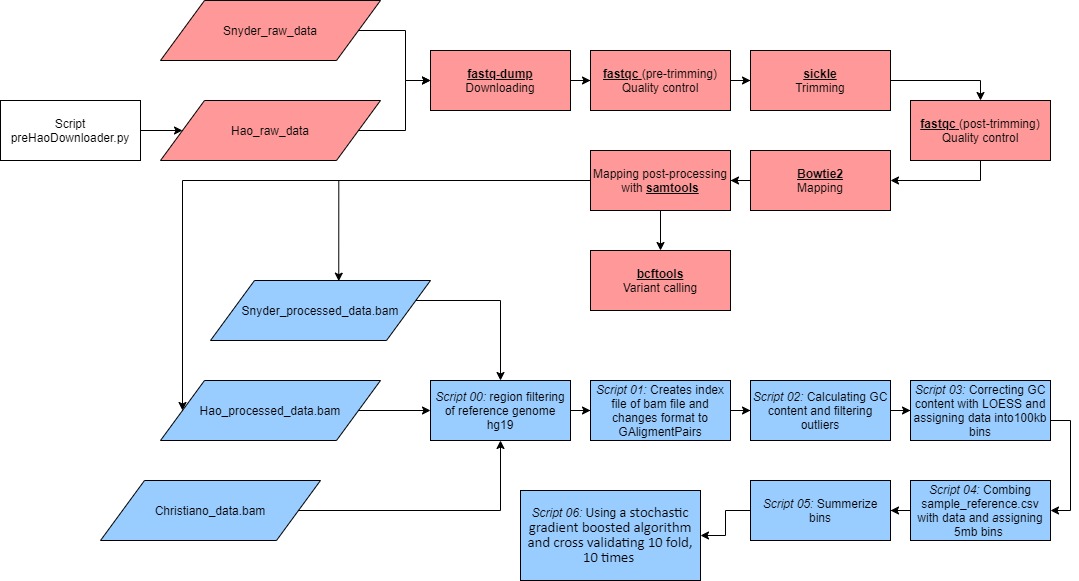


Figure 1, Flow chart describing through which steps the data was processed. The Snyder and Hao datasets were downloaded and the quality determined. The downloaded reads were trimmed and the quality re-checked. The data were mapped to the human reference genome hg19 with the tool bowtie2 and processed with samtools and bcftools (red). These datasets, and the Cristiano datasets, ran through the 7 different DELFI scripts (blue).

### 2.3.1 Processing the Snyder dataset

To convert the Snyder et al. dataset to bam files, a shell script was created to download the fastq files, assess the quality, and process it to produce bam files. The tool used to download the files from the NCBI Sequence Read Archive (SRA) is called fastq-dump version 2.8.2, which is part of the SRA-toolkit 11. The quality of the fastq files was assessed using the tool FastQC, version 0.11.912 created by the Babraham Institute. The computation was spread over 16 threads, the quiet mode was activated, and the output was formatted into an HTML file.

To trim the data the program sickle, version 1.33 of the University of California was used13. The options for pair-ended sanger sequenced input files were used, combined with the gunzipped output option. After this, the FastQC program was used again to assess the effectiveness of the trimming.

The full analysis set of the GRCh37/hg19 human reference sequence was downloaded14 and indexed using bowtie2 version 2.3.4.1 64-bit. The bowtie2-build function was used with the options telling the program that the reference input was a fasta file, the quiet option, and the option to split the computing over 16 threads. This reference index was needed to run the bowtie2-align function. The function has to be told what kind of input files is used, in this case, fastq files. Bowtie2-align was told not to output progress, and an option to increase the maximum fragment length size to 1000bp. When the exact insert size is unknown this could prevent the tool from skipping longer insert fragments15. The output is a file in the Sequencing Alignments Map (SAM) format.

The tool samtools version 1.716 of Genome Research Limited has numerous functions that were used to process the sam files from Bowtie2. Firstly the sort function, which sorts the alignments. The option to sort by name instead of coordinates was used, as well as the option to multithread using 16 threads. This multithread option was used in all samtools functions.

The fixmate function was used to insert mate coordinates in a name sorted alignment. Mate score tags were added for the future markdup function, the option for multiple threads was used and the output was changed to a Binary Alignment Map (BAM) format. This bam file was sorted again, this time not by name but by mate coordinates, multiple threads were used and the output was a bam file. The markdup function was used to mark duplicate alignments from a coordinates sorted file with mate score tags. The option to include supplementary reads of duplicates as duplicates, the options to use multiple threads, and the options to remove all reads marked as a duplicate was used. To filter the reads which had a Phred score of 30 or higher the samtools function view was used. The options for the Phred score quality filter, multiple threads, and the option to keep the bam format were used.

To assess the quality of the bam file, the program QualiMap version 2.2.2 tool called BAMQC17 was used. This created a txt file with a quality report assessing among other things the mapping quality, GC content, mismatches and indels, and both the mean coverage and the coverage per contig.

The variants of the bam file were called by first indexing the same hg19 reference genome sequence, which was used with the bowtie2-build tool earlier, but this time with the samtools faidx tool. The bam file was indexed with the program bamtools index18 made by the Boston College. Samtools also has a tool called mpileup which generates a BCF file, based on the indexed hg19 reference genome. This inputs into a program called bcftools version 1.919 also made by Genome Research Limited. It has a tool named call which selects the variants. The option for the variant only input, the VCF output format, and the multiallelic caller was selected. The multiallelic caller locates and counts the amount multiallelic sites, which are sites where multiple alleles can stack and cause SNV’s20. This was the final table where the variants were shown. The quality was checked with the bcftools tool named stats.

### 2.3.2 Processing the Hao dataset

The dataset from the Hao et al. paper has the same problem as the dataset from Snyder et al. It has not been aligned yet and is in the form of fastq files. The difference between the two is that while the Snyder data set is grouped according to cancer types, the Hao dataset is grouped according to biosample. This means that where a single cancer group has a single fastq file in the Snyder set, the cancer group in the Hao set has 654 files. To download all these files and concatenate them into a single pair-ended fastq set, a python3 script was created called preHaoDownloader.py. This script downloads all the 654 files. The files are concatenated in the modified shell script used for the Snyder paper. The rest of the shell script is identical to the Snyder script.

### 2.3.3 the Cristiano dataset

The Cristiano dataset did not need pre-processing as it was downloaded as bam files. The details of the processing from fastq files to bam files by Cristiano et al. are not disclosed in the paper.6

### 2.3.4 The DELFI scripts

To predict whether the patients within the three cfDNA datasets had cancer, a pipeline of scripts had to be followed written in the programming language R. There are 7 different scripts in the DNA EvaLuation of Fragments for early Interception (DELFI) pipeline.

The first script produces two RDA files, gaps, and filters. The gaps.rda file is a GRanges object that provides locations of gaps in the hg19 reference genome. The filters.rda file is also a GRanges object but with blacklisted regions of the hg19 reference genome. These regions have what is called low mappability. Mappability is the measure of the ability to map the region in the genome, defined by alignability and uniqueness. For further reading see Dave Tang’s blogpost on ENCODE mappability21.

The second script takes a singular bam file, indexes it and creates a GAlignmentsPairs object from the bam file applying a MAPQ filter at 30. The third script reads the galp file, fragments it by splitting the chromosomes, and filters the outliers. The fourth script splits the data into 100kb bins. The genomic locations of these bins are assigned in the data provided by Fortin et al 22. The bins are split into open and closed chromatin, which affect the physical interactions between nucleosomes and thus have to be taken into account in the processing. The fifth script joins the sample\_reference.csv file, provided on the GitHub (4.3 Availability of data and materials), creating larger, 5mb bins. The sixth script condenses and summarizes the data in the 5 Mb bins. The seventh and final script trains a stochastic gradient boosted machine learning algorithm to distinguish cancer sequences from non-cancer sequences. The prediction error is estimated by 10-fold cross-validation.

# 3. Results

In this chapter, all of the issues that were experienced throughout this research will be presented. Within each paragraph one issue will be explained and which approach was used to solve the issue.

## 3.1 Quality of the datasets

The quality of the Snyder and Hao datasets were checked in two different ways. First, the fastq files were assessed with the FastQC tool. Then the BAM file was checked after alignment with the hg19 reference genome. For the control, the BAMQC tool from the Qualimap package was used. The BAM file from the Cristiano dataset was also checked to determine the difference between these datasets.

### 3.1.1 FastQC

The FastQC tests were performed both before, and after the trimming of the fastq files as a control on the effectiveness of the trimming. Within the Snyder dataset, three files were tested, the healthy patients file, the breast cancer file, and the prostate cancer file. The Hao dataset was tested and compared as well.

According to the result report, there weren’t many low-quality sequences. The adapters from sequencing were already trimmed, and the GC-content was between 37% and 41% in all cases. The total amount of sequences did differ significantly, with the healthy file containing >5x as many sequences as the cancer files, and the Hao file contained over 20x as much. The percentage of sequences trimmed also differed between these groups. The healthy and the Hao files had 12% of the sequences trimmed, while the breast and prostate files had, on average, 25% of their sequences trimmed. (table 1)

Tabel 1, Difference in total sequence count and trimming results between the Hao and Snyder data files.

|  |  |  |
| --- | --- | --- |
|  | Average sequence count | Phred score >30 |
| Hao | 681.964.640,00 | 88,31% |
| Snyder healthy | 184.450.153,50 | 88,76% |
| Snyder prostate | 30.085.864,50 | 77,48% |
| Snyder breast | 36.569.002,00 | 71,67% |

### 3.1.2 BAMQC

The BAMQC results show a clear difference between the Cristiano, Snyder, and Hoa data. The percentage of mapped reads did not show this, with the Cristiano data resulting in 83% of reads mapped. Hao topped the list with 100% of reads mapped, and both breast and prostate Snyder datafiles had 76% of their reads mapped. The mapping quality was 231 in the case of Cristiano, and only 32, 31, 27, and 32 for Hao, breast, prostate, and healthy, respectively. The coverage was different as well, Cristiano covered the genome 2,2X. The Snyder breast and prostate data only covered 0,4X each, while the healthy data covered the reference genome 7,5X. The Hao data covered the reference genome a total of 0,02X.

## 3.2 Human reference genome hg19 or hg38

Bowtie2 was used for the mapping of the alternative datasets and needs a human reference genome. The reference genome hg19 was used for this to match the workflow of the Cristiano et al. paper. At first, this was not an issue. Then a certain error message was generated after running script 03 (bin\_compartments.r). This error message stated that the script wasn’t able to locate the different chromosomes of the hg19 reference genome. This can be explained by the fact that different annotations are used for human reference genomes. The hg19 reference genome that was used, had no clear way for the script to locate chromosome 1 to 22.

After these findings, there was concluded that a different reference genome should be used with a suitable design for the scripts. When exploring other options of human reference genomes we found hg38, which is an updated version of hg19. According to the literature of Pan et al.23, hg38 is recommended for next-generation sequencing Single Nucleotide Variant (SNV) analyses. Aligning cfDNA data to a reference genome with an improved SNV knowledge could contribute to better screening of abnormalities within this data. Because variants that are already present are more updated.

The error message was solved by this change of the reference genome to hg38 but also brought new error messages. It was concluded that all of the scripts together were built for only reference genome hg19 and it wasn’t an option to change this. After the next search for a solution, we found a hg19 analyses set which were slightly different than the hg19 that was used before. This hg19 analysis set was put together to use in next-generation sequencing alignment pipelines. When changing the reference genome again all the error messages for script 03 (bin\_compartments.r) were finally solved.

## 3.3 Load gaps missing

In script 00 (filtered\_regions.r) two files are generated to obtain blacklisted regions: filters.hg19.rds and gaps.hg19.rds. The files are used in script 03 (bin\_compartments.r) to find overlapping and matching fragments between these two files and the “\_frags.rds” file generated in script 02 (fragment\_gc.r).

When trying to run script 03 (bin\_compartments.r) the following error message appeared. (figure 2)

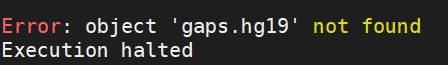


Figure 2, Error message object gaps.hg19 after running script 03-bin\_compartments.r

The error message got generated because only the filters.hg19.rds file was loaded in the original script. To solve this problem a line was added, equal to the line that loads the filters.hg19.rds file.

## 3.4 Task ID change

In the third and fourth original DELFI scripts, a system called Array Jobs was used. This is a way to run almost identical jobs in parallel. In scripts where, for example, twenty input files have to be run through the script. Instead of running the script twenty different times, a Task ID number can be set to run the twenty scripts with twenty different input files in parallel.

While this system works well for an entire dataset, when there is a singular input file it is not necessary. When the scripts were edited and optimized for the Snyder, Hao, and Cristiano datasets, the Array Job code was replaced with a commandArgs function. This function allows arguments, like the input file name, to be submitted into the command line, without changing the script. This makes testing a script with a singular input file easier. When multiple files have to be run through the script, a simple for loop can be utilized.

3.5 Main error messages

### 3.5.1 Cristiano dataset

The first four DELFI scripts worked fine after some small adjustments. Then after running script 04 (5mb\_bins.r ) an error message was generated. (figure 3)

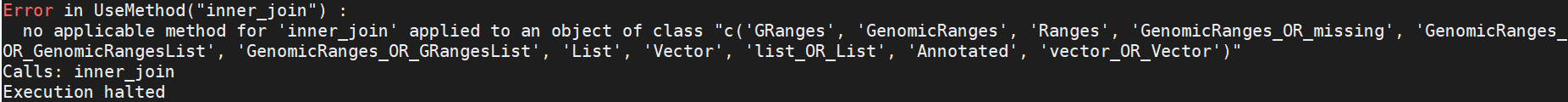


Figure 3, Error message inner\_join function classes after running script 04-5mb\_bins.r

The error message reported that the inner\_join function wasn’t applicable between two objects of different classes. This inner\_join function in script 04 (5mb\_bins.r ) was meant to join parts of a .rds file (output from script 03) and the sample\_reference.csv file that was already provided by the DELFI script makers. This occurred because the .rds file was a GRanges object and the sample\_reference.csv file a tibble data frame. One of these files needed to change class to make the inner\_join function work. When the .rds file was changed into a data frame the error message was solved, but the next error message emerged after running script 04 (5mb\_bins.r ) again. (figure 4)

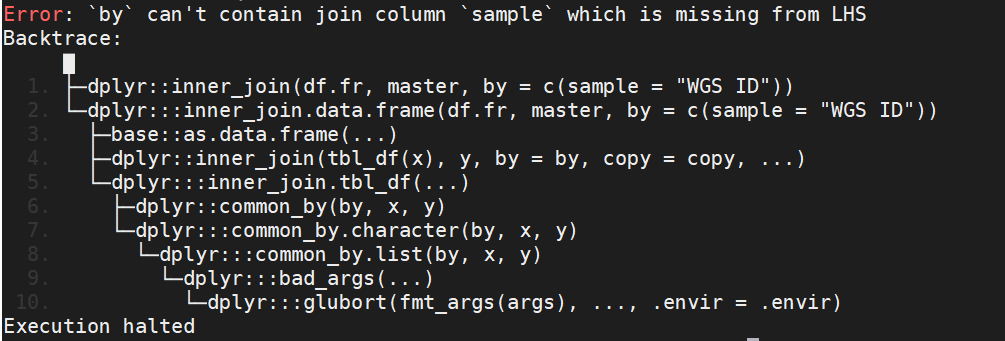


Figure 4, Error message inner\_join function column sample not found after running script 04-5mb\_bins.r

This next error message pointed out that the column ‘sample’ was missing from the .rds file, which contributed to the inner\_join function to fail and the execution of the script to get halted. It’s possible that during the first scripts the data got sorted a bit different than it should be sorted, but for now, there is no explanation of how this column ‘sample’ got generated wrongly. It’s also not clear how this error message could be solved.

### 3.5.2 Snyder and Hao datasets

Scripts 00 (filtered\_regions.r) , 01 ([read\_galp.r](https://github.com/cancer-genomics/delfi_scripts/blob/master/01-read_galp.R" \o "01-read_galp.R)) and 02 (fragment\_gc.r) generated no error messages with Snyder and Hao data as input. Untill script 03 (bin\_compartments.r) got halted at the same point for both datasets and produced an error message. (figure 5)

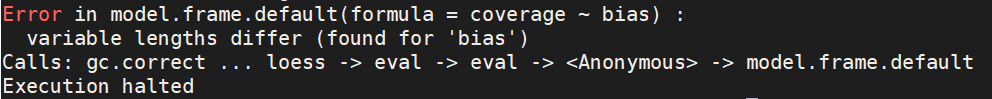


Figure 5, Error message within the gc.correct object after running script 03- bin\_compartments.r

The error message calls an error within the gc.correct object and states: ‘variable lengths differ’ and an error in model.frame.default. This could mean that one object has two different data frame lengths. Within the code lines the argument ‘na.rm=TRUE’ is used, which makes sure all the NA’s are removed from the data frame where the argument is used for. It is possible that when these NA’s are removed from the data frame at one point and the data frame was defined before that, the script will give the error message mentioned earlier. Removing the NA’s before the data frame was defined didn’t solve this error message.

In the end, the best explanation for this error to occur was that the datasets Snyder and Hao weren’t suited for the parameters that are set within the DELFI scripts. It is not impossible to make the DELFI scripts work for these alternative datasets, the script would just need a lot of adjustments.

# 4. Discussion and Conclusion

## 4.1 Discussion

The DELFI algorithm is a machine learning approach that can distinguish cancer patients from healthy individuals through abnormalities of cfDNA fragmentation profiles. DELFI consist of a pipeline of seven scripts that need cfDNA as input. Within this research, we tried to recreate the DELFI algorithm by using the original cfDNA dataset as input for the DELFI pipeline. This would have provided an understanding of how the algorithm should work. At the same time, we worked on implementing the DELFI algorithm on alternative cfDNA datasets from Snyder et al. and Hao et al. This would have contributed to making the algorithm work for other patients that still need a diagnosis.

### 4.1.1 Missing code in the DELFI scripts

Several findings within this research consisted of code lines that were missing in the scripts, e. g. load gaps.hg19. This created doubts about what else could be missing in the scripts. Another example was the transitions of the output of script 03 (bin\_compartments.r) to input for script 04 (5mb\_bins.r), wherein script 04 the output of script 03 wasn’t in the right class to be input for script 04. These issues had relatively simple solutions but still contributed to dubiety about the accessibility of the DELFI script to other researchers than the original script makers. For instance, the GitHub page does only include the scripts but no explanation or guidance about the code lines. Whereas other GitHub pages24,25 of published bioinformatic papers do.26,27

### 4.1.2 Array jobs system in two out of the seven DELFI scripts

Due to this lack of explanation about the code lines of code, it is still not clear why only in scripts 02 ([fragment\_gc.r](https://github.com/cancer-genomics/delfi_scripts/blob/master/02-fragment_gc.r" \o "02-fragment_gc.r)) and 03 (bin\_compartments.r) an Array Jobs system was used. The original dataset for the DELFI scripts consists of data per patient. When trying to make all the scripts work for the data, it seemed that till script 03 (bin\_compartments.r) It was not needed to use multiple data files as input. In script 04 (5mb\_bins.r) this could be needed, this is still vague.

The Array Jobs system is used to make it easier to run big datasets through the DELFI scripts at the same time. It would make more sense if this system was used for at least script 01 ([read\_galp.r](https://github.com/cancer-genomics/delfi_scripts/blob/master/01-read_galp.R" \o "01-read_galp.R)) to 06 ([gbm\_full.r](https://github.com/cancer-genomics/delfi_scripts/blob/master/06-gbm_full.r" \o "06-gbm_full.r)), because script 00 ([filtered\_regions.r](https://github.com/cancer-genomics/delfi_scripts/blob/master/00-filtered_regions.r" \o "00-filtered_regions.r)) doesn’t need input. Because of this indistinctness, It is not evident if changing this system to a commandArgs function had any impact on the final error messages.

### 4.1.3 Size Cristiano data set

Some scripts in the DELFI program are made to run for each of the patients. As discussed in the SGE TASK ID change in the implementation, this was changed in a way that kept that aspect. The problem is that in the 04 (5mb\_bins.r) script, there is no SGE TASK ID change. The only option is that all files should be run in the single script. This has to be done anyway, to train and test the machine learning algorithm. This raises the issue that the entire dataset is 4 TB of data. That is a lot of computing power needed to be able to handle that much data. The dataset needs a specially ordered hard drive to store all the data, how long would it take run a script on that much data, as matrix calculations are heavy computationally.28

### 4.1.4 Quality of the datasets

When the quality of the Snyder and Hao fastq files were checked, the immediate difference between sets was the number of sequences. As seen in Table 1, there were over 20 times as many sequences in the Hao dataset, compared to the prostate or breast data from Snyder.

The Hao paper claims that the cfDNA was sequenced to 96 to 105-fold coverage, while the results from the BAMQC control showed a mean coverage of just 0,02. This could because of the concatenation of the 654 files which were downloaded from the SRA, as this was the only difference between the handling of the Hao and the Snyder data. The pooled cfDNA in the Hao set is 52 GB of data, while the data from the Snyder files run from 133 GB for the healthy group to 1,4 and 2,0 GB for the breast and prostate groups. This does not correlate to the coverage, where the Hao set has 20 times less coverage compared to the breast or prostate cancer sets. The Hao dataset mapped 100% of its reads perfectly to the reference genome, whereas the other datasets mapped only 77% to 95% of the reads.

The mapping quality of the Cristiano data is 231, while the Hao and Snyder data are ~30. Neither the coverage or the alignability affect the MAPQ score, so correlations are difficult to draw. The mapping quality is calculated directly from the bam files, with the following formula:

,

rounded to the nearest integer.29 This scoring system would roughly end with a score between 20 and 40, but Bowtie2, the alignment tool, also sets a MAPQ score of 255 for perfect uniquely mapped reads, which is where the high value for Cristiano comes from.

### 4.1.5 Accuracy of the DELFI algorithm

The viability of the sequencing and processing of cfDNA to diagnose cancer is not accepted everywhere yet. The Cristiano paper itself reported sensitivities of detection ranging as low as 57%, which is almost the same as flipping a coin. Previous research indicates many concerns about the lack of understanding and knowledge of cell-free DNA,30 and the low cfDNA count in plasma.31 Further testing is needed to find a way to increase the accuracy and sensibility of liquid biopsies in cancer diagnostics.

There is some precedence for cfDNA. Cell-free DNA screening already has widespread use in testing for trisomies in pregnant women, although it is not tested with fragmentation patterns, with a karyotype.32

### 4.1.6 Recommendations

In the Cristiano et al. paper the hg19 human reference genome was used for aligning the sequence data. Also within the scripts, hg19 is used for several lines of code. When trying a more updated version of the reference genome, hg38, script 03 (bin\_compartments.r) failed to work. These human reference genome updates happened frequently over the past years and will continue to happen in the future.24 Therefore a recommendation for further research is to alter the script into also accepting human reference genomes other than hg19.

## 4.2 Conclusion

While there have been great strides in development of liquid biopsies by cfDNA, there is no consensus about extraction methods and data analysis. DELFI is one of the pipelines which diagnoses early stages of cancer in patients, but it is not reliable yet. More script editing and testing are needed to make DELFI reach its potential.

## 4.3 Availability of data and materials

Of the three datasets, two are publicly available. The Snyder dataset analyzed during the study is available at the NCBI short read archive (SRA) with the BioProject ID PRJNA291063 . The Hao dataset is also available at the SRA, with BioProject ID PRJNA383370.

The Cristiano dataset is available from the European Genome-phenome Archive (EGA) with dataset ID EGAD00001005339. Restrictions apply to the availability of these data, which were used under license for the current study, and so are not publicly available. Data are however available from the authors upon reasonable request and with permission of Jillian Phallen ([jphalle2@jhmi.edu](mailto:jphalle2@jhmi.edu)).

The scripts used to download and process the Snyder and Hao datasets are available at <https://github.com/GinoRaaijmakers/MachineLearning>, as are the original and modified DELFI scripts.

## 4.4 Acknowledgments

We are grateful to our supervisor Dr. Miaomiao Zhou for guiding us through this research.

# References

1. Babayan, A. & Pantel, K. Advances in liquid biopsy approaches for early detection and monitoring of cancer. *Genome Med.* **10**, 21 (2018).

2. Jahr, S. *et al.* DNA fragments in the blood plasma of cancer patients: quantitations and evidence for their origin from apoptotic and necrotic cells. *Cancer Res.* **61**, 1659–1665 (2001).

3. Wan, N. *et al.* Machine learning enables detection of early-stage colorectal cancer by whole-genome sequencing of plasma cell-free DNA. *BMC Cancer* **19**, 832 (2019).

4. Kourou, K., Exarchos, T. P., Exarchos, K. P., Karamouzis, M. V. & Fotiadis, D. I. Machine learning applications in cancer prognosis and prediction. *Comput. Struct. Biotechnol. J.* **13**, 8–17 (2015).

5. Zhang, Z. & Zhang, W. Fragmentation patterns of circulating cell-free DNA demonstrate biomarker potential for human cancers. *Biotarget* **3**, (2019).

6. Cristiano, S. *et al.* Genome-wide cell-free DNA fragmentation in patients with cancer. *Nature* **570**, 385–389 (2019).

7. Fiala, C. & Diamandis, E. P. New approaches for detecting cancer with circulating cell-free DNA. *BMC Med.* **17**, 159 (2019).

8. Snyder, M. W., Kircher, M., Hill, A. J., Daza, R. M. & Shendure, J. Cell-free DNA Comprises an In Vivo Nucleosome Footprint that Informs Its Tissues-Of-Origin. *Cell* **164**, 57–68 (2016).

9. Hao, X. *et al.* DNA methylation markers for diagnosis and prognosis of common cancers. *Proc. Natl. Acad. Sci.* **114**, 7414–7419 (2017).

10. UCSC Genome Browser Gateway. https://genome-euro.ucsc.edu/cgi-bin/hgGateway?db=hg19&redirect=manual&source=genome.ucsc.edu.

11. SRA-Tools - Installation and Configuration. https://ncbi.github.io/sra-tools/install\_config.html.

12. Babraham Bioinformatics - FastQC A Quality Control tool for High Throughput Sequence Data. https://www.bioinformatics.babraham.ac.uk/projects/fastqc/.

13. Joshi, N. & Fass, J. Sickle: A sliding-window, adaptive, quality-based trimming tool for FastQ. (2011).

14. Index of /goldenpath/hg19/bigZips/analysisSet. http://hgdownload.cse.ucsc.edu/goldenpath/hg19/bigZips/analysisSet/.

15. Use X with bowtie2 to set minimum and maximum insert sizes for Nextera libraries · Loman Labs. http://lab.loman.net/2013/05/02/use-x-with-bowtie2-to-set-minimum-and-maximum-insert-sizes-for-nextera-libraries/.

16. Samtools. http://www.htslib.org/.

17. Analysis types — Qualimap 2.2.1 documentation. http://qualimap.bioinfo.cipf.es/doc\_html/analysis.html.

18. Bamtools - Bioinformatics. https://bioinformatics.readthedocs.io/en/latest/bamtools/#package-bamtools.

19. bcftools. http://samtools.github.io/bcftools/bcftools.html.

20. Li, H. A statistical framework for SNP calling, mutation discovery, association mapping and population genetical parameter estimation from sequencing data. *Bioinformatics* **27**, 2987–2993 (2011).

21. ENCODE mappability and repeats - Dave Tang’s blog. https://davetang.org/muse/2013/07/08/encode-mappability/.

22. Fortin, J.-P. & Hansen, K. D. Reconstructing A/B compartments as revealed by Hi-C using long-range correlations in epigenetic data. *Genome Biol.* **16**, 180 (2015).

23. Pan, B. *et al.* Similarities and differences between variants called with human reference genome HG19 or HG38. *BMC Bioinformatics* **20**, 101 (2019).

24. jkimlab. *jkimlab/TAMA*. (2020).

25. *mephas/mephas*. (MEPHAS, 2020).

26. Zhou, Y., Leung, S., Mizutani, S., Takagi, T. & Tian, Y.-S. MEPHAS: an interactive graphical user interface for medical and pharmaceutical statistical analysis with R and Shiny. *BMC Bioinformatics* **21**, 183 (2020).

27. TAMA: improved metagenomic sequence classification through meta-analysis | BMC Bioinformatics | Full Text. https://bmcbioinformatics.biomedcentral.com/articles/10.1186/s12859-020-3533-7.

28. Bläser, M. On the complexity of the multiplication of matrices of small formats. *J. Complex.* **19**, 43–60 (2003).

29. Biofinysics: How does bowtie2 assign MAPQ scores? http://biofinysics.blogspot.com/2014/05/how-does-bowtie2-assign-mapq-scores.html.

30. Frontiers | Utility of cfDNA Fragmentation Patterns in Designing the Liquid Biopsy Profiling Panels to Improve Their Sensitivity | Genetics. https://www.frontiersin.org/articles/10.3389/fgene.2019.00194/full.

31. Bronkhorst, A. J., Ungerer, V. & Holdenrieder, S. The emerging role of cell-free DNA as a molecular marker for cancer management. *Biomol. Detect. Quantif.* **17**, 100087 (2019).

32. Cuckle, H., Benn, P. & Pergament, E. Cell-free DNA screening for fetal aneuploidy as a clinical service. *Clin. Biochem.* **48**, 932–941 (2015).