UNIVERSITY OF ZAGREB
FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

MASTERS'S THESIS no. 740

Identify pathogen organisms from a stream of RNA sequences

Matija Šošić

Zagreb, June 2014.

Many thanks to my mentor Mile Šikić, family an everybody who provided me support and guidance during creation of this thesis.

CONTENTS

List of Figures

LIST OF TABLES

1. Introduction

One of the currently most popular topics in both science and medicine is definitely genome study - either humane or of other species. This knowledge can be used to gain deeper understanding of biological processes and interactions on various levels in human body which directly means developing new disease treatments and prevention methods.

The biggest barrier into entering this field of science was cost and rarity of technology for analysing data on molecular level. But, with rapid development of genetic material sequencing technology which literally exploded in the past few years, the interest of academia has been growing in the same manner.

Although advancement of sequencing technology is far from the end, actual and very challenging task is also to interpret acquired data and get the greatest amount of useful information. This is very diverse problem, greatly varying from one scenario to another, and it is very hard to find one general way to address them all with equal success.

Besides analysing genetic material of only one species at a time, the problem can be extended to mixed genetic material of multiple species, often of uknown origin. While the first scenario is typical for samples created in laboratory conditions, the second one considers real-world, environmental samples.

Study considering such complex samples is called metagenomics. By definition, it is "the application of modern genomic techniques to the study of communities of microbial organisms directly in their natural environments, bypassing the need for isolation and lab cultivation of individual species"[8]. Besides bypassing the lab isolation, the great value of metagenomics is in fact that majority of organisms cannot be cultivated in laboratory[4], therefore revealing us a broad range of previously unknown organisms.

The general problem of metagenomics could be described as follows: given genetic material recovered directly from environmental sample, characterize its living content. In simple terms, two main questions we are asking here are:

- Who is in there? - with taxonomy analysis we want to recognize present organisms
- What are these organisms doing there? - with functional analysis we want to determine their role in the community

Metagenomic sample is always extracted from some microbial community. All present organisms are not there accidentally - each of them fills its role in sustaining the balance in that community. For example, the same bacteria may express different behaviour in different conditions. Thus it can be very important to recognize that behaviour and see if it is in some way unwanted, besides solely confirming its the presence in the sample.

Having such complex problem and many different approaches to it calls for a good method of evaluation. Unfortunately, it is a challenge by itself to achieve that. There is still no "gold standard" for evaluating metagenomic analysis methods, as nobody can for sure know what is in some given sample. Creating synthetic samples of known origin is very common, but it is very hard to create a sample that will be a good imitation of a environmental one, and then also to verify that. Another way of evaluation is to compare acquired results with outputs of other systems for the same sample input.

This thesis is addressing previously described general metagenomics problem in a following scenario: Metagenomic sample is a tissue sample taken from an eukariotic host. From that tissue data is acquired by RNA sequencing. It is expected that majority of sample will contain host sequences. As host is known in advance, (we know who the sample was taken from), the task is to isolate non-host sequences and identify organisms of origin, potentially pathogens.

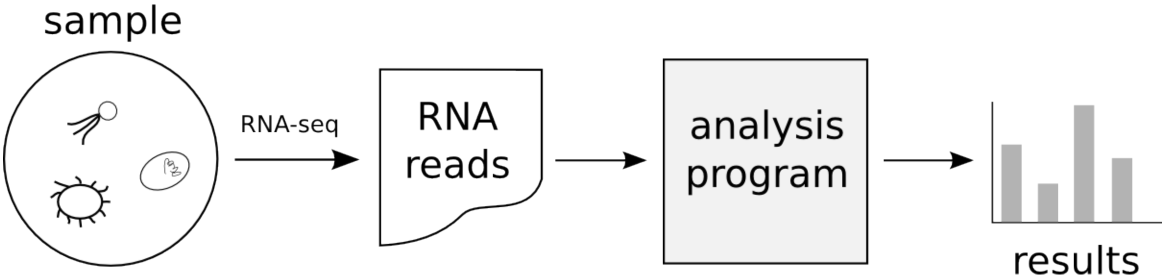


Figure 1.1: High-level overview of pipeline of metagenomic sample RNA-seq analysis

What is the value of such system? Consider a following example: tissue sample is taken from a human (e.g. urine or blood). By traditional means the sample would be sent to targeted analysis - doctor would examine the symptoms and assume on certain bacterial or viral infection. Results of analysis would confirm or deny his suspicions.

Using the proposed system there is no targeted but total analysis, with only one sample all present organisms would be identified. Advantage over traditional flow is obvious as it saves both time and needed amount of sample and provides a much more detailed analysis. Also, in long term, the money is saved so cost of analysis could be reduced. Instead of answering to "Is certain bacteria present in the sample?", it answers to "Who is present in the sample?" question.

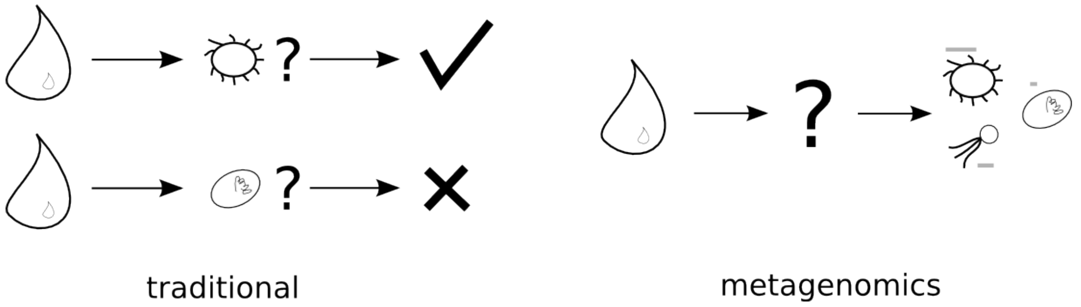


Figure 1.2: Illustration depicting advantages of metagenomic analysis over the traditional one.

Chapter *Methods* provides an insight into principles and ideas on which this solution is built. In chapter *Implementation* are explained implementation details of the algorithm. Chapter *Results and discussion* presents the result of testing on real environmental sample. *Conclusion and outlook* makes short summary and discusses future directions of development.

2. Overview of methods for classification of metagenomic reads

This chapter presents an brief overview of existing principles and methods in classifying metagenomic reads, and as well of tools utilising them. Metagenomic classifiers can differ on various levels, starting from the initial requirements, applied methods and expected output to the way of testing and evaluation.

Regarding the generation of input, there are basically two approaches[16, 28]. One is shotgun sequencing, and the other is focused on isolation, extraction and sequencing of amplicons corresponding to phylogenetic marker genes as 16S rRNA. As focus of this thesis is on analysing sequenced RNA reads, amplicon based themes reviewed in[25] won't be further explained.

Other principal differences are whether classification is done with respect to taxonomy hierarchy or not, and then further if it is done by direct comparison to some reference base or by extracting and comparing some defined set of features.

Sometimes may be difficult to compare different evaluations because testing conditions may differ in a lot of parameters, e.g. size and content of reference databases or read length. In addition, the simulated shotgun reads in the test sets may have been simulated under different error models.

Hereafter are given basic ideas and analysed advantages, limitations and challenges of various methods. Further details for each of them can be found in the corresponding publications.

2.1. Taxonomy dependent methods

Most of binning solutions for metagenomic data obtained by shotgun sequencing falls into this category. This methods are also often referred to as assignment dependent, as their goal is to assign each read to some node of a taxonomy tree. Therefore, the assignment process is driven by the level of similiraty with sequences from the refer-

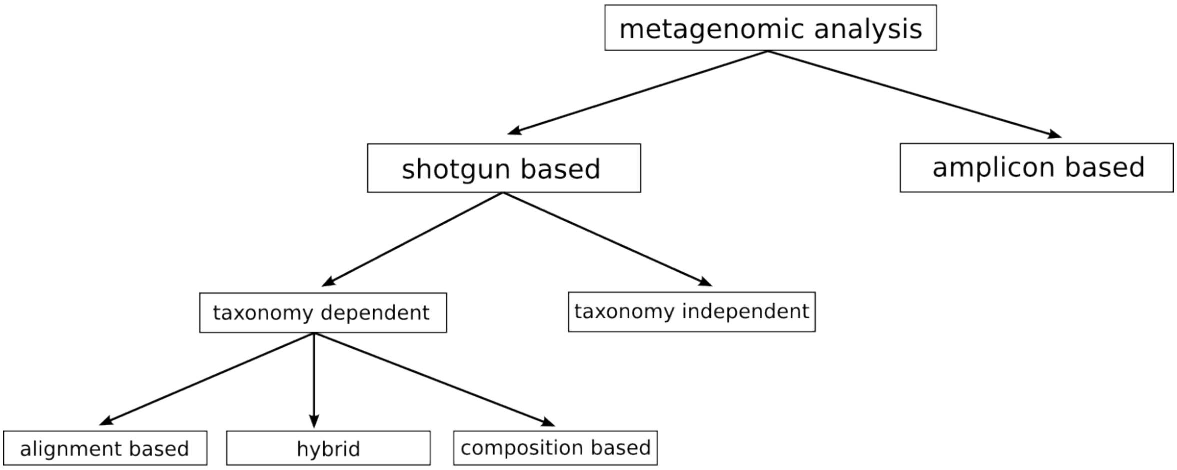


Figure 2.1: According to [16], a schematic representation of various categories of algorithm principles for binning metagenomic datasets.

ence database or some features precomputed again from the reference database. Reads that cannot exceed some given similiraty treshold are categorized as "unassigned". According to these two ways of similarity definition, taxonomy dependent methods can be further separated into alignment based and composition based methods.

2.1.1. Alignment based methods

Alignment based methods first compare given reads to the sequences from reference database, and typically for that employ algorithms as BLAST[3], BWA [14] or BOWTIE[13]. Each read is processed individually. Finally, reads are assigned to taxonomic nodes based on different strategies depending on the quality of their alignment to the "hit" sequences.

The simplest form of such strategy is to assign each read to the taxonomy node of organism to which sequence it had the alignment with the highest score. This is employed by MG-RAST[18] and CAMERA[27], and for alignment they are using BLAST.

The drawback of this assignment method is that there is no way to assign reads coming from organisms whose sequences are not already in a database. Also, the best BLAST hit may not always guarantee the origin of read, especially if it is coming from some organism who has sequenced similar relatives on lower taxonomic ranks.

To address this issue, some solutions utilize a Lowest Common Ancestor(LCA), approach which assigns a read to to the lowest common taxonomic ancestor of the organisms corresponding to the set of significant alignment hits. This way decision of assignment is taken to the higher taxonomic ranks (e.g. phylum or genus) and as such is much more reliable in cases when alignment score is not good enough. Although this approach reduces specificity of the output, it also reduces the number of wrong assignments.

MEGAN[12], one of the most popular binning algorithms is also built on this principle, and also uses BLAST. One of the most important steps in this strategy is identifying significant alignment hits in order to find the lowest ancestor. MEGAN is defining as significant all hits that have a bit score equal or higher than 90% of the bit score of the best hit. Some other algorithms like MetaPhyler[15] and MARTA[11] are apart from bit scores in account also taking other alignment parameters like the fraction of identities, positives and gaps to evaluate the quality of alignment.

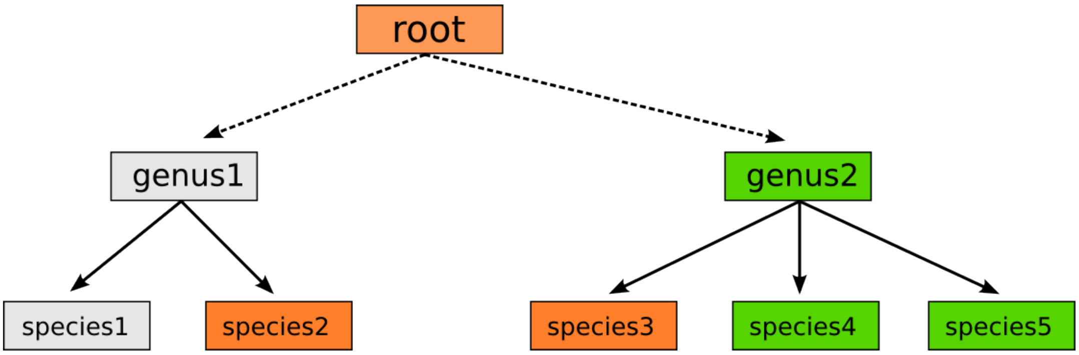


Figure 2.2: Illustration of LCA assignment method. While green hits stay assigned within a genus 2 clade, orange hits get assigned higher to the root level

Reference database used here can contain either nucleotide sequences or protein sequences of known taxonomic origin. Protein sequences may be used as they are better conserved than nucleotide ones, but the disadvantage is that they cannot be used for reads originating from non-coding DNA sequences.

The primary limitation of all alignment based methods is very heavy computational complexity of alignment phase and therefore requirement of a huge computer power. The advantage is it may be used even for shorter read lengths.

2.1.2. Composition based methods

Composition based methods extract features from sequences like GC percentage[10], codon usage[20] or *k*-mer[17, 29] frequencies cite. Those features are also precomputed for sequences in reference database so comparison can be made. Machine learning techniques like Self Organizing Maps[2] or Support Vector Machines[9] can be used to train classifer on that data. As these methods are not directly dependent on alignment, they require less computing power than alignment based methods and are therefore much faster. But, they usually need the initial traning step and do not work so well with shorter reads.

2.1.3. Hybrid methods

Hybrid methods combine ideas from both alignment based and composition based methods. For example, algorithm SPHINX[19] combines it into two phases. First phase is composition based where read is compared with reference tetra-nucloetide free frequency vector. Goal of this phase is to quickly identify groups of similar sequences which are then going to be aligned against in second, alignment phased phase. First step can significantly reduce the search space for the second step which still ensures the reliability of assignment. Another popular solution with two-step idea is PhymmBL[7] algorithm.

2.2. Taxonomy independent methods

Methods in this category do not have a taxonomy knowledge and are not trying to assign reads to some specifix taxonomic unit. Instead, they are doing unsupervised learning and clustering given reads by species of origin. Basic ideas include grouping of reads by similarity defined over some set of features, which requires for the reads of sufficient length. AbundanceBin[30] using Poisson distributions models the number of reads originating from different species, but requires that species present in the sample are not of similar abundance, which is often a case in the nature.

3. Methods

In this chapter is more in details described input data, its origin and developed biological model. Overall pipeline is described step by step, from taking input data to the final results output. Each step is thorougly explained and analysed while also presented with illustrative data.

3.1. Biological model

As this thesis' theme is very closely connected to biology and organic molecular processes, it is neccessary to have some understanding about it to be able to create a model, define some base space and set boundaries within which we are looking for the solution. It is connection to the real-world application of proposed solution. Assumptions made here are the basis upon which this solution is built and directly reflect on its capabilities and finally, successfulness.

3.1.1. Taxonomic classification

In order to characterize the origin of given sequences most commonly is used NCBI taxonomy [6, 26]. It is a phylogenetic taxonomy with a tree-like structure that reflects estimated revolutionary relationships between organisms. The tree itself is defined by ranks, starting in root from the most general to the more specific ones. While higher ranks define groups of organisms, leaves of the tree usually refer to some specific species or strain.

Table 3.1: Taxonomic lineage of *Streptococcus oralis* according to NCBI

rank	name
superkingdom	Bacteria
phylum	Firmicutes
class	Bacilli
order	Lactobacillales
family	Streptococcaceae
genus	Streptococcus
species	Streptococcus oralis

Also, each node is accompanied by its unique taxonomy id so it can be easily identified.

3.1.2. The sequencing process

For our needs, DNA sequencing can be described as a function that takes a sample of tissue as input containing genetic material, DNA or RNA sequences of all present organisms. Each sequence is cut into multiple smaller pieces, read with some accuracy and returned as output. Output is a set of readings of those pieces, called *reads* and stored in well known biological formats *FASTA* or *FASTQ*. It is important to say that output consists solely of reads, without any information suggesting their original positions. Also, this overall process is defined by a lot of parameters such as length of read, accuracy rate and others dependent on the sequencing technology and device model itself.

3.1.3. RNA-seq: Monitoring host-pathogen interaction

Samples that are being analysed are originating from a known eukaryotic host, usually human or some other mammal. When working with RNA-seq data, as is the case here, we are in fact analysing gene expression of present organisms, which is a part of a transcriptome. Compared with pure DNA sequencing data, RNA-seq data is more dimensional, dynamic, as it is a function of a time and continuosly changing. Knowing which genes get expressed and proteins produced and their function, we can have an idea what is currently happening inside the host's cell. That way it is possible to get better understanding of the life-cycle of infections and host-pathogen interplay.

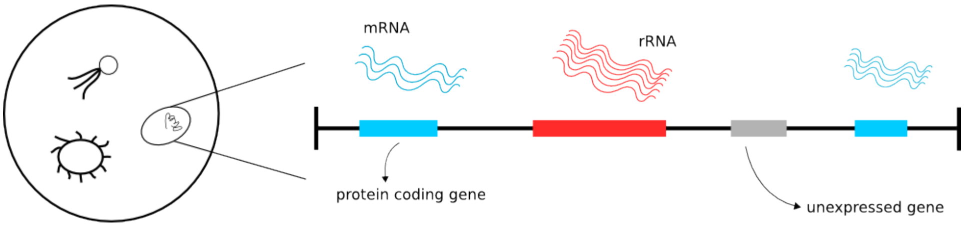


Figure 3.1: Illustration of transcriptome expression

On the whole genome, there are certain active parts which by transcription produce RNA. Such parts are called coding sequences, in further text abbreviated as CDS. CDS can either code for a mRNA which then gets translated into a protein, or for a rRNA which constitutes a predominant material within the ribosome.

Typically there is a significantly more rRNA present (up to 90%) in the sample than mRNA. Because of that, in studies that analyse only gene expression rRNA is often depleted from the sample before the processing.

The set of all RNA molecules produced in a cell is called transcriptome. In a certain conditions only a subset of existent coding sequences will transcribe to RNA, in other words get expressed. That is the reason why is transcriptome dynamically changing.

3.1.4. Using ribosomal proteins as species indicators

The main assumption used to create here presented binner is that all species present in the given sample can be identified by their ribosomal proteins. If an organism is present in the sample then it is expected its ribosomal proteins will also be present[24].

This assumption is based on a fact that ribosomal proteins are appropriate for prokariotic systematics. That is so because they are well conserved throughout evolution and taxonomy lineages, but also unique enough to ensure reliable systematics at least on a species level.

At the end, formal definition of the problem this thesis is solving would be: given set of reads (*FASTA/FASTQ* file) from a known eukaryotic host, for each read determine its species of origin. Also, determine all expressed genes and proteins they are coding.

3.2. Data processing pipeline

It this section, each step of a data processing pipeline is described, along with inputs and outputs.

3.2.1. Aligning reads to the reference genome

As this is an alignment-based binning method, first step is to align given reads to some reference genome database. Each read is processed independently as alignment program tries to find which part of the known sequences it resembles the most. One read can be aligned to the multiple sequences or to multiple locations in one sequence or can have no alignments at all. Each alignment is paired with some quality data or measure, depending on the used alignment algorithm.

There is a whole palette of available alignment algorithms to choose from, starting from famous BLAST to the bleeding-edge Bowtie2. We chose to use Burrows - Wheeler Aligner software package (BWA), as it is much faster than BLAST although it has a bit lesser sensitivity.

The other needed component in this step is a reference genome database. It is a set of already existing sequences, systematized by taxonomy ranks. Its purpose is to serve as a knowledge base against which input reads are being compared. Here we are using NCBI nt database, resource holding nucloetide sequences from more or less all known organisms, including human genome.

Depending on the size of input, this step can be computationaly very demanding and thus requiring the large fraction of the overall execution time. Because of that it is a good idea to reduce the reference genome to only the parts that are nedded, if that is possible. For example, as a reference was also used database holding only extracted bacterial sequences as they were expected in a given sample.

For the sake of the algorithm in this step, reference genome can be thought of as a very long sequence created by concatenating all sequences from the database together. Having this large string, aligner algorithm basically performs a text search on it with some additional rules defined by biological model.

To summarize, input of this phase is a set of reads, and output is a file containing possible alignments for each read to the given reference genome. Output file is written in a standard *SAM* format.

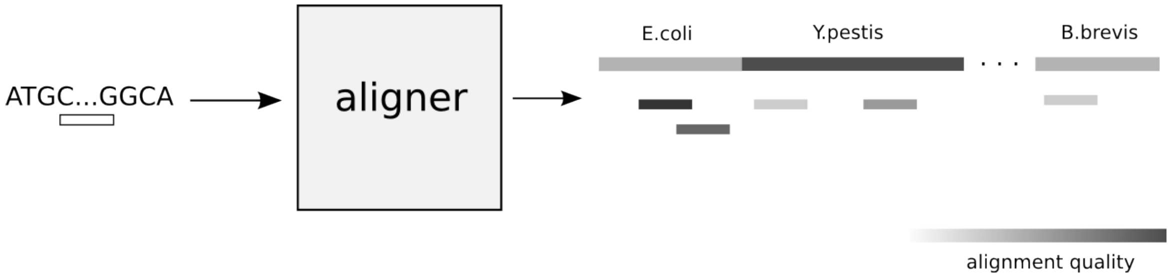


Figure 3.2: Alignment process for a single read - input and output

3.2.2. Identifying and removing host reads

As host is already known in advance, it is needed to remove its reads in order not to interfere with the rest of the process. To consider a read to be a host's one it has to have more than a some defined percentage of alignments to the potential host organism. That defined percentage is a parameter that can be adjusted, although we mostly used the value of 0.5. Also, there is a prepopulated list of values of various taxonomic ranks as potential hosts which is used in this step to classify each read.

Table 3.2: Taxonomy nodes considered to characterize a potential host

rank	name
kingdom	animalia
kingdom	green plants
order	primates
order	rodents
genus	rats
species	human
species	mouse

For each alignment is determined taxonomy id of its reference genome and by that is checked if it is contained in a potential host node. It is expected that majority of reads will originate from host so after this step amount of data to process is significantly reduced, which also reduces the running time of the further steps.

3.2.3. Determining and locating coding sequences (CDSs)

As illustrated on figure 3.1, sequenced RNA originates only from coding sequences of a genome, while NCBI NT database contains all sequences, both coding and non-coding. That means we are actually aligning given reads against database with unwanted data that should never be matched. As we are interested only into coding sequences, the next step is to identify and locate all CDSs that were associated by at least one read during the alignment phase. For each sequence of the reference genome database it is known position and length of each CDS.

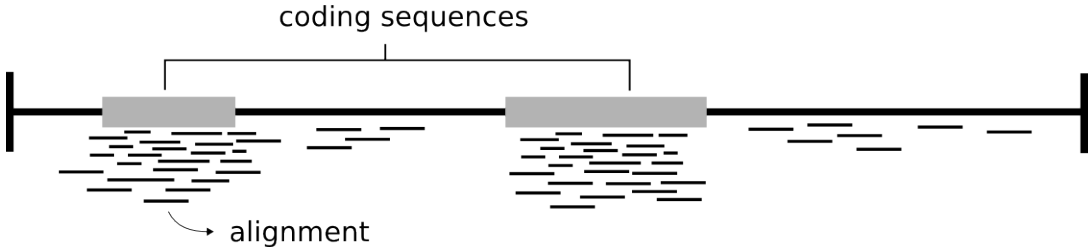


Figure 3.3: Expected distribution of coverage between coding and non-coding sequences.

It is expected that majority of alignments will be aligned to the coding sequences, that way creating alignment groups in these areas of the reference genome. While portion of alignments will associate with non-coding parts, they can be ignored since they represent an untrue information.

Having identified coding sequences, the next step is to determine significant ones, in terms that they really are the origin of the analysed reads. Even if a CDS has a portion of reads aligned to it, it does not automatically mean that it really is the true connection. It is important to take a look at a given alignment group, extract some interesting features and define some measure that will tell evaluate its quality.

For such purposes is often used a *read coverage* of a coding sequence which is being analysed. Read coverage of a CDS is a function of a single nucletoide in the sequence which tells us by how many reads (or in this case alignments) is that nucloetide covered.

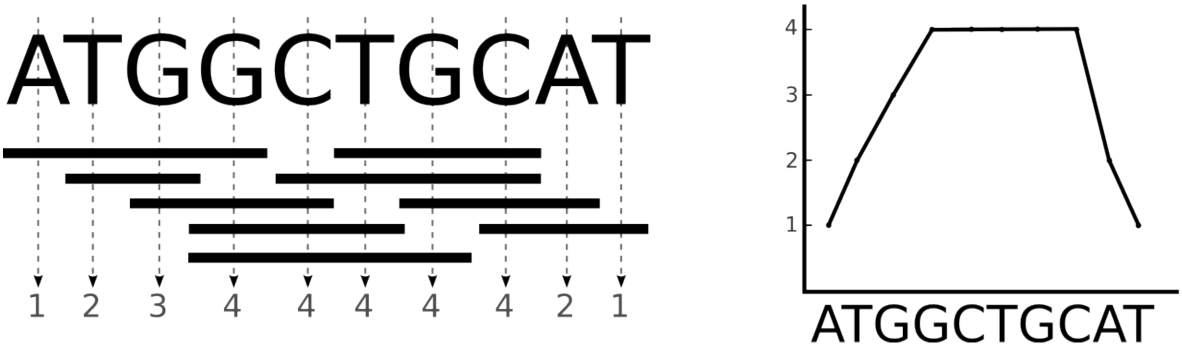


Figure 3.4: Illustration of read coverage of a sequence.

For a CDS that is really an origin of an analysed sample, shape of a coverage function is expected to be somewhat similar to one on the figure 3.4. At the end and the beginning of the CDS coverage should be lower while middle area should be most densely covered. Also, in the middle area there should not be any holes or major fluctuations. As we already isolated eukaryotic host sequences and are dealing with prokaryotic sequences, only exons are present therefore CDS should be continuously covered.

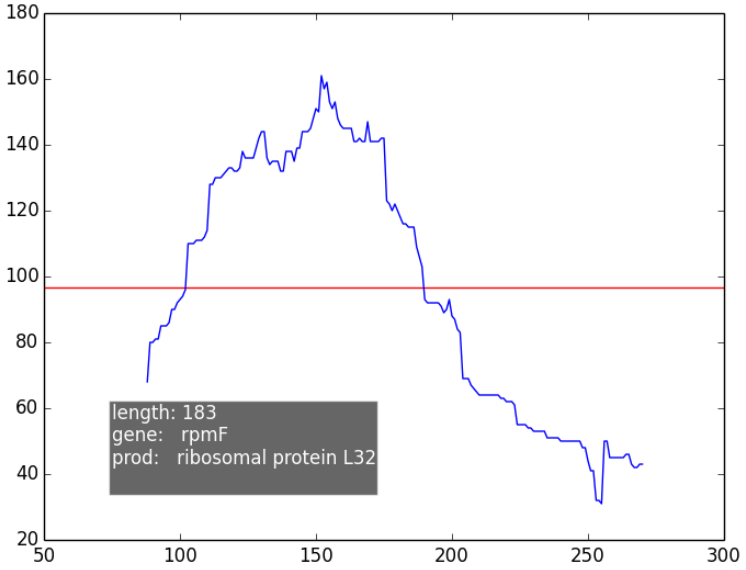
Used measures to check for these attributes are mean value of coverage, and then upon it the standard deviation, to evaluate the fluctuation. The length of CDS is also put in consideration as CDSs of greater length considered to be of greater importance.

3.2.4. Isolating ribosomal CDSs and determining present species

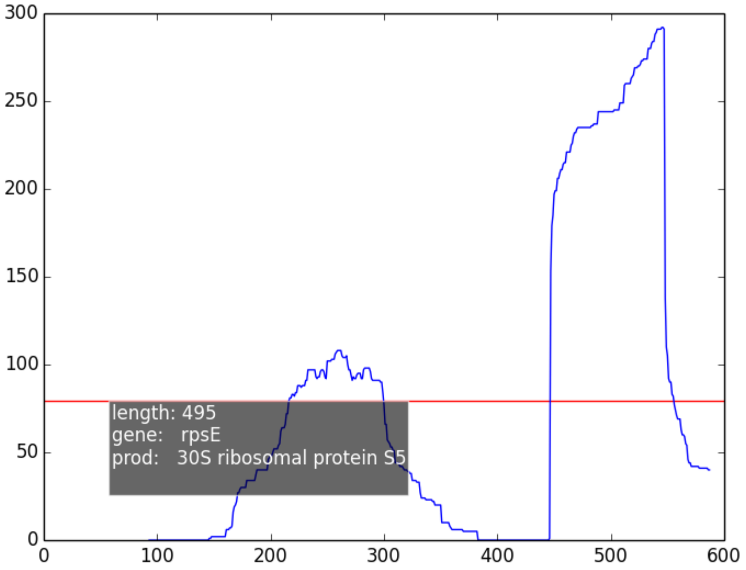
After all coding sequences with at least one read aligned are determined, the next step is to determine species that are present in the sample. The idea here is to use ribosomal CDSs as indicators of the present species, make initial assumption and then to go further on that. Ribosomal coding sequence can be either genes transcribing to mRNA which afterwards gets translated into ribosomal proteins or areas that encode rRNA which than builds up an ribosome itself. Both of them are regarded as highly conserved and unique to ensure the good classification.

Ribosomal coding sequences are recognized among all CDSs by text search for associated keywords. Every CDSs encodes for a certain final product (protein or rRNA), whose data is also stored in a database. This data also includes a textual description of the product and its functions. By searching for keywords as "ribosomal" and "16S" in this description ribosomal CDSs are identified.

The next step is to group those ribosomal CDSs by species they are originating from. Having that done, we have the first idea of what species may be present in the sample, but still don't know much about their abundance. To this initial groups can now also be joined all other CDSs, those that aren't ribosomal, also by species. In each group should be at least one ribosomal CDS, otherwise it's CDSs are discarded. Having that done, we have determined groups of CDSs by species while in each group is at lest one ribosomal CDS. It is also important each represented species has a significant amount, e.g. it is biologically not possible that some organism has only one CDS which would mean it expresses only one gene. Therefore there is parameter which determines how many CDSs should at least a species have in order to be considered invalid.



(a) CDS considered to have a good quality of coverage, and therefore to be origin of aligned reads. Coverage is continuous, with lower values on the borders.



(b) CDS considered to have a bad quality of coverage. There are sudden fluctuations and even holes which shouldn't exist in a prokaryotic coding sequence.

Figure 3.5: Illustration of CDSs coverage quality in terms of determining origins of reads.

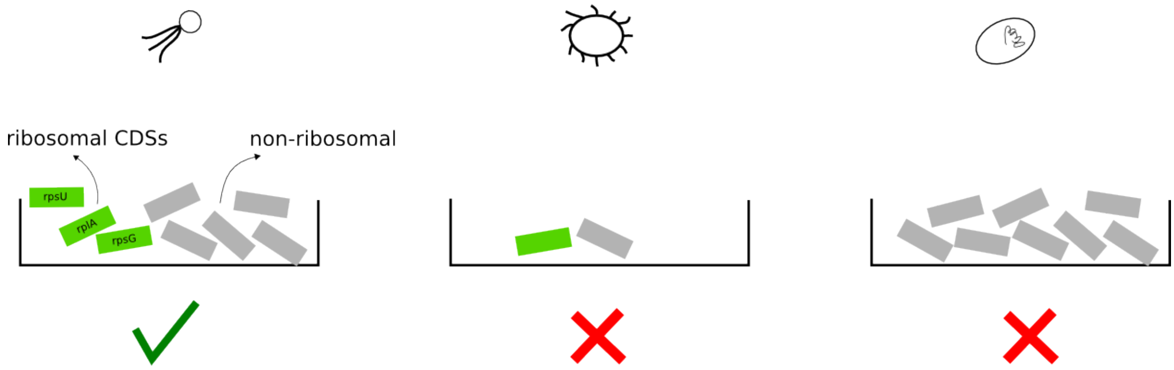


Figure 3.6: Illustration of species grouping by coding sequences identified in a sample. The leftmost species has both ribosomal and non-ribosomal CDSs and their total number is over the given treshold. The middle species has also both kinds of CDSs, but their total number is too small to be considered as biologically significant. Finally, the rightmost species has enough CDSs associated with it, but none of them are ribosomal. Therefore, this species is also considered invalid.

3.2.5. Read assignment

Having determined species, the final step is to try to assign each read to some of them. Read is assigned to a certain species if it is aligned to it with the highest score. If read does not have any alignments, it is considered to be unassigned.

4. Implementation

This chapter describes implementation of the proposed binning solution. All main components are introduced, starting from the supporting data and its organisation, existing tools of choice, to the code itself and principles on which it was built. Choices made are analysed and discussed in a terms of both speed and efficiency, just as justified for the sake of more convenient implementation.

4.1. Language and environment

The language of implementation is Python, while solution was developed on the operating system Linux. Code is compiled for and follows the standards of Python's version 2, although it is written in compliance with Python3 where possible in order to ensure easier transition if it will be needed in the future.

We chose Python as it is one of the best languages for developing prototypal solutions due to its flexibility, openess and a vast amount of existing libraries. The especially useful was BioPython, a library designed specifically for bioinformatic tasks and implementing a lot of current bioinformatics standards, just as manipulation with their formats, e.g. *SAM* alignment file.

As Python supports an object-orient paradigm, the program is also writen in a such manner. This is especially convenient when dealing with data from external sources as they can be directly represented in the code the same way.

While writing code Google style docstring[21] for comments was used, with added types of parameters. As Python itself is not a typed language, this has proven to be very useful as there is a lot of different classes from the inputs and outputs of various phases of the algorithm.

Although Python is a scripting language, meaning it significantly is slower than its programming counterparts as it is interpreted rather than compiled, the speed of execution has proven not to really be an issue here. The execution accent is not on the heavy algorithms, but more on loading big amounts data which takes majority of time.

Also, other parts of the described data processing pipeline are several times slower, so gaining speed here would only minorly reflect on the overall execution time.

Program is created to be used as a command-line tool under Linux. User can specify input file, choose among available options and then set path to the folder where the output will be stored.

4.2. Third-party tools

For the alignment step we use exising solutions of BWA and Tophat. While BWA was mainly used for aligning non-host reads to reference genome, in some cases we checked for existence of host with Tophat. Tophat is especially fit for that task as it is an aligning software built above Bowtie, with an extra knowledge about splicing.
We chose BWA over BLAST as it has proven to be much faster. Also, BWA performs semi-global alignment while BLAST does a local one. Semi-global alignment may be more convenient in this scenario, when we are looking to match small read to a much bigger reference genome. Both of these tools are well known, tested and commonly used for similar bioinformatic tasks. This phase is one that lasts for a longest amount of time. It is directly dealing with all input data and doing computationally heavy work. Also, before the first aligning to some reference genome it is needed to build an index upon it.

4.3. Supporting data

As our solution is an alignment-based and taxonomy-dependent binning method, it heavily depends on preexistent knowledge for both of these features. Because of that, we are using two external SQL databases - one holding relevant data for each sequence to which reads can get aligned, and other holding tree-structured knowledge of NCBI taxonomy.

4.3.1. Reference sequences data - Unity database

This database is holding data about each sequence from the reference genome. It is called Unity as is holds sequences from EMBL, GenBank and DDBJ. Those three sources are today's main sources of sequenced genetic material from variuos experiments. They are also mutually collaborating and exchanging acquired data on a daily basis, but location of origin is still noted for each sequence.

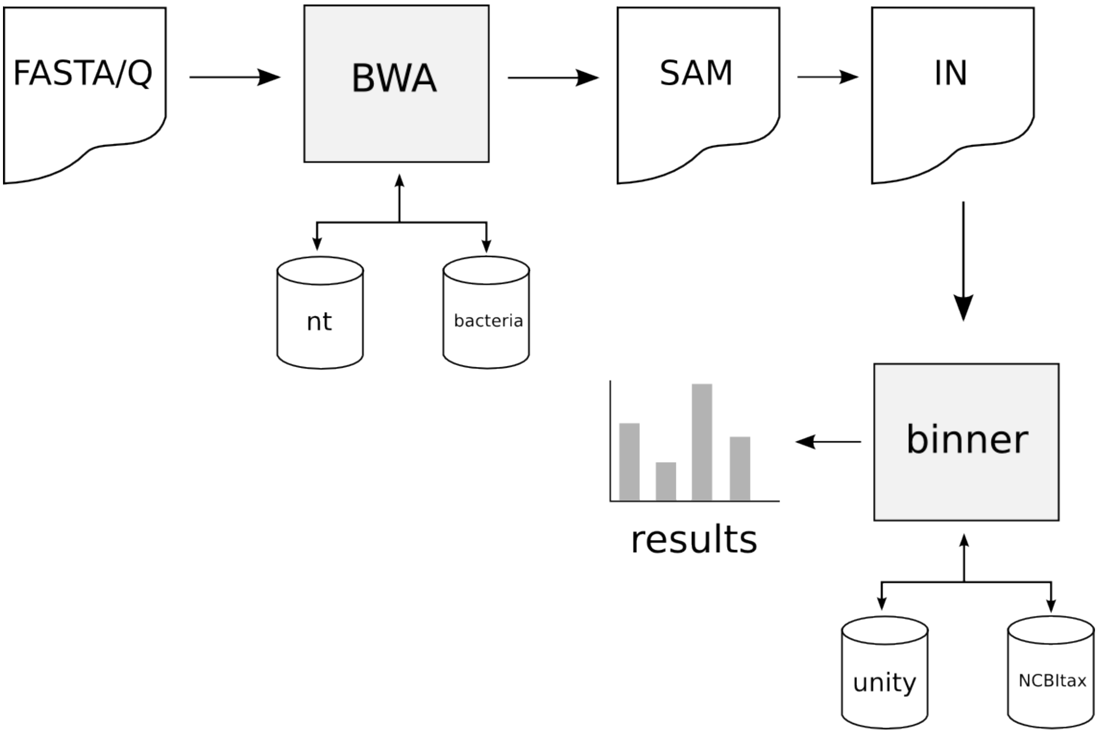


Figure 4.1: Illustration of data flow along with databases used by certain components.

In order to conduct the binning process, the algorithm needs to know for each sequence the location of its coding sequences and that is, among other data, what is stored in this database.

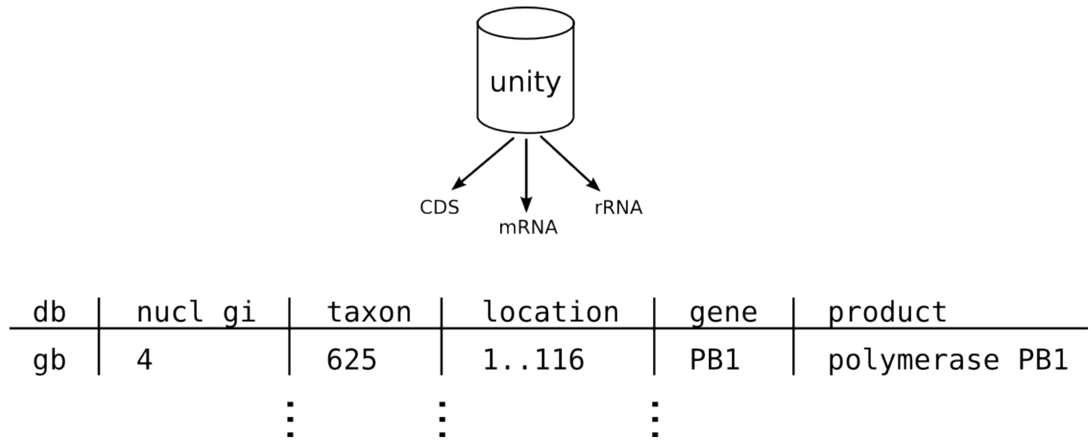
Other data include nucleotide genInfo identifier(commonly known as gi number), which is a unique integer that identifies a particular sequences, and also consistently changes if the sequence changes.

NCBI compliant taxonomy identifier of a sequence is also stored marking a rank of strain or species. Location attribute is a complex string from which is directly loaded an associated object during runtime. It allows for multiple joined intervals and also strand specification.

Where it makes sense, gene name and protein product data is also stored.

4.3.2. NCBI taxonomy database

The other needed database is the one holding taxonomy data what is then applied to establish relations between multiple sequences, and also between CDSs. As already described and illustrated in 3.1.1, it is a tree structure of taxonomy identificators, and just that is stored in a SQL database.
NCBI tree database holds several important tables:



db	nucl gi	taxon	location	gene	product
gb	4	625	1..116	PB1	polymerase PB1
	⋮	⋮	⋮		

Figure 4.2: Unity database holds data from both mRNA and rRNA sequences in separate tables, and then CDSs for each of them in a third table. Here is given example of one row of a table with attributes important for explaining the implementation.

– **gi number to taxonomy id mapping** - This table connects each sequence to the species from which it originates. This table is also the basis for interaction between Unity and NCBI tree database.

– **NCBI nodes** - Information about each node is stored, like its rank, name and id.

– **NCBI tree** - Holds relation data between nodes and is used for navigation throughout the tree.

Interface implemented to interact with the tree uses those tables and also offers some additional advanced functionalities built upon them. Because of that it is possible to access analysed node's parent on a specified rank or get the lowest common ancestor of a group of nodes.

4.3.3. Reference sequence header format

In order to connect sequence to the Unity database, one must know gi number of the sequence, as shown on figure 4.3. Because of that, in header of each sequence in reference file is written gi number. There also exists an standard FASTA header defined by NCBI:

>gi|*gi_number*|*db_id*| *description*

This implementation assummes that a file containing reference sequences has headers exactly in this format.

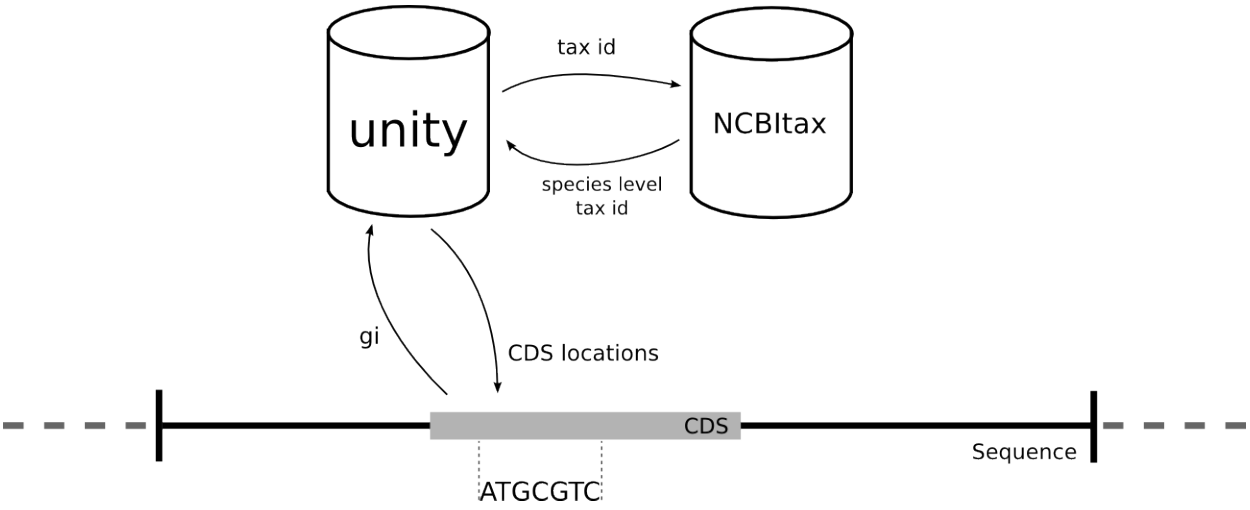


Figure 4.3: Illustration of data flow through the databases during single read analysis. As read is aligned to the specific sequence, unity database is looked up with its gi number in order to retrieve locations of CDSs. Taxonomy id is also retrieved, and then in taxonomy tree databse is looked up for its parent on a species level.

4.4. Runtime data structures

Starting from the input alignment file, we are in fact moving through different data structures and analysing them until the final result is acquired.

First, data from alignment file is loaded into structure `ReadContainer`, object that connects each read with its alignments, regardless on which sequences are they aligned. This objects interface also allows for fetching all reads, and when interacting with a single read (object `Read`) one cas access specific data like taxonomy id and all its alignments.

The most computationally demanding step in creating this structure in memory is determining coincident CDSs for each read alignment. For given alignment it is needed to loop through all CDSs of sequence to which it is aligned and test for intersection. But, as CDSs for a single sequence are in database already sorted ascending by starting position, it is not neccessary to loop through all coding sequences. We employed binary search algorithm to determine the first CDS overlaping with a given alignment and checked consecutive CDSs as long as an intersection existed. When there is no more intersection, it is not needed to continue with checking as all other CDSs will be only more "away" from the alignment location.

As database is built only once, it is affordable to previously sort coding sequnces of each sequence. Introducing this method significantly reduced running time of this part opposed to the usual "check all" principle, but is still the most time expensive portion

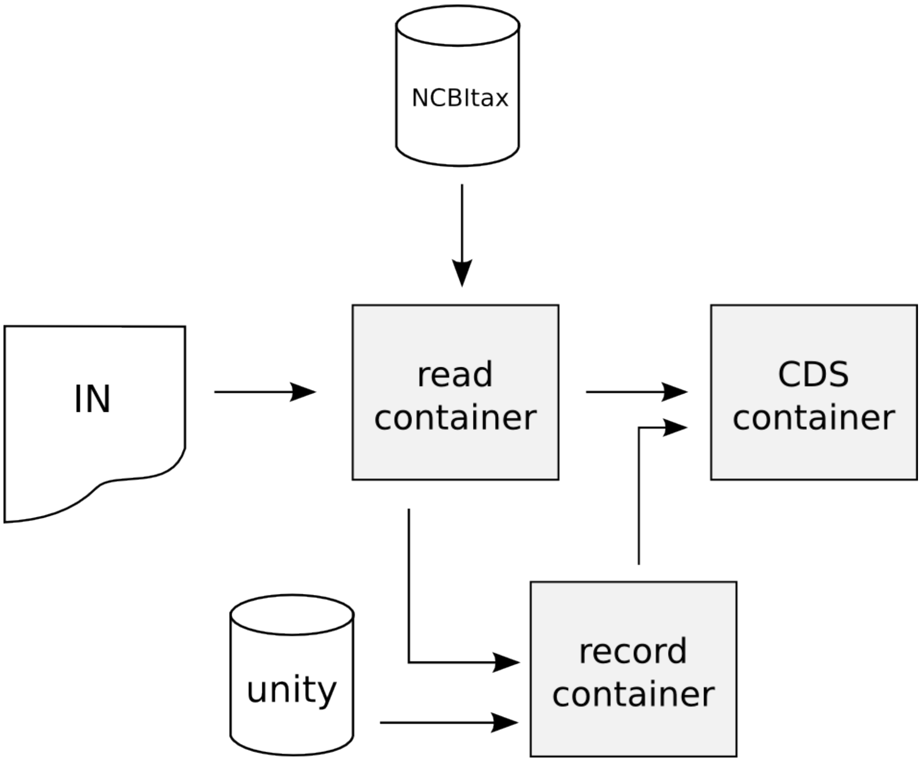


Figure 4.4: Construction relations of data structures used in the implementation. Read container is initially constructed from input alignment file and through NCBItax database taxonomy id to each alignment is assigned. All referenced sequnces are loaded from Unity into `RecordContainer`. `CDSContainer` for each CDS holds alignments to it.

of code.

This directly reflects data organization from the input file. One read has more possible alignments, and each alignment may coincide with one or more coding sequences, so this forms a three-leveled structure starting with reads and ending with CDSs. In order to get a view on all present CDSs and calculate their coverage, it is needed to "flip" this structure around, putting CDSs on the first level and associated alignments beneath. Having that done, data structures creating is finished and algorithm is ready to advance to analysis phase.

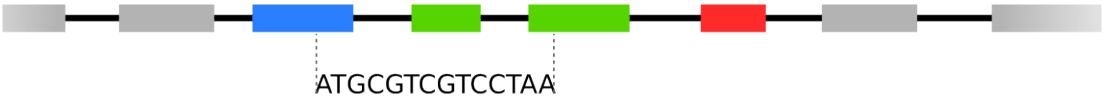


Figure 4.5: Problem of determing intersections of alignment and CDSs of aligned equence. Firstly, with binary search is determined first intersecting CDS from the left. The following CDSs are being consecutively checked until no intersections occurs. Because CDSs list is sorted, searching stops here.

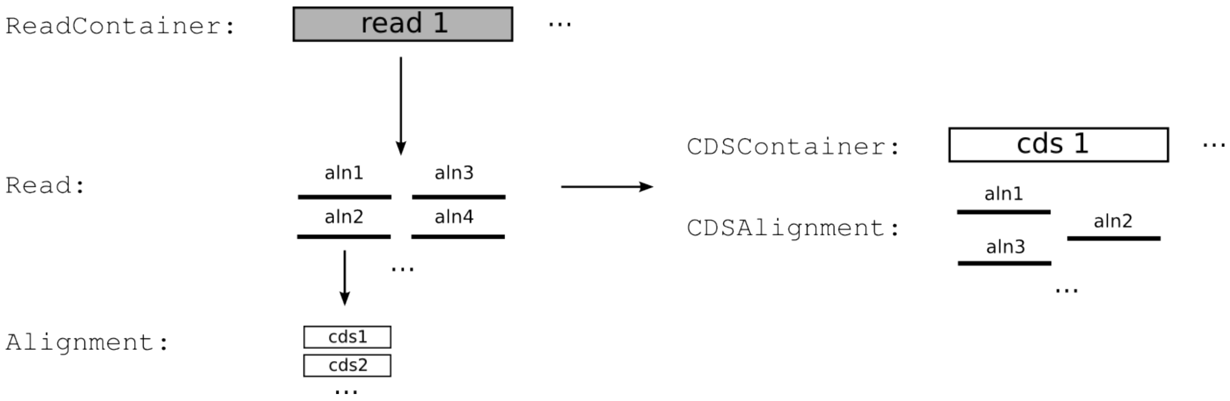


Figure 4.6: Transition of data structures used in implementation, starting from three-leveled read-based to the two-leveled CDS-based structure.

`RecordContainer` is an interface to sequences from Unity. It is optimal in a way that only data of sequences that were referenced (having one on more read aligned to them) gets loaded into memory.

4.5. Preparing the alignment input format

As there exist multiple file formats for storing alignment data, our binner also has its own simplified format with idea of mobility and easy transition to other formats. So far *SAM* and *BLAST* formats are supported, and for each of them there is appropriate script(`sam2input.py` and `blast2input.py`) for transition to our format, called *IN* (as *input*).

5. Results and discussion

This chapter presents testing of described implementation on the environmental dataset. We are interested in acquired results and its content and as well in total time spent on execution and its distribution over different phases of the algorithm.

All tests were conducted on a workstation with 24 cores of Intel Xeon E5645@2.40GHz processor and 200GB of RAM.

5.1. Dataset: Dental microbiome analysis

This dataset has been published as a part of Human Microbiome Project(HMP)[22] initiative. It consists of RNA-seq samples of the dental microbiome in twin pairs with and without dental Caries.

RNA was extracted from dental plaque scrapings from 38 patients. Amplified cDNA was created and rRNA sequence sequence was removed by subtractive hybridization. After that, each individual patient's sample was ran on a single lane of an Illumina Genome Analyzer. Also, for each patient's sample it is marked if it is characterized as caries positive or caries negative.

5.1.1. Sequence file information

File containing reads was downloaded from HMP's RNA-seq data resources website[1] in *FASTQ* format.

Table 5.1: Information about dental microbiome *FASTQ* file

Instrument model	Illumina
read length	101 bp
read count	36 626 798
file size	9.8 GB
file name	`2011_CP_DZ_PairTo_2012_fastqc`

5.1.2. Alignment phase

Alignment was performed by BWA-MEM algorithm using multithreaded option with 10 cores, and lasted about 18 hours. As a reference genome was used file with bacterial sequences extracted from NCBI nt database.

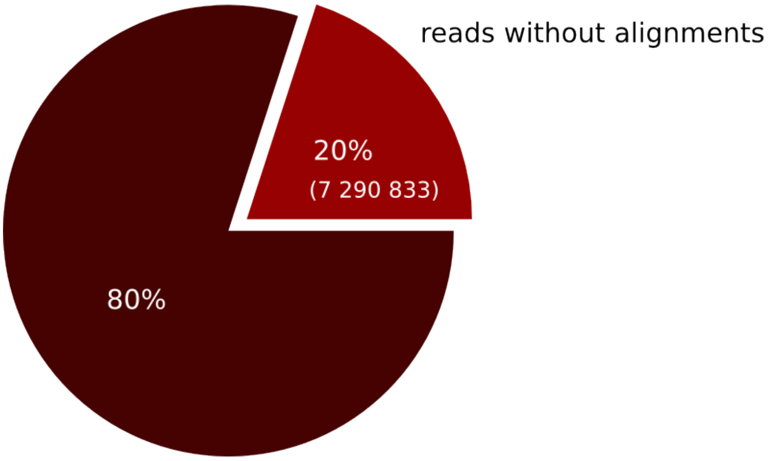


Figure 5.1: Fraction of reads without alignments

20% of all reads, around 7 million of them, does not have any alignment at all. We also tried employing BLAST on a sample of 10000 reads from that set against both NT and described bacteria databases, but that also did not produce any hits.

As can be seen on figure 5.2, majority of reads have only one alignment. Also, average value of number of alignments per read is 0.86, and if we are looking only at the reads that have at least one alignment then it is 1.08. Maximum number of alignments per read is 4.

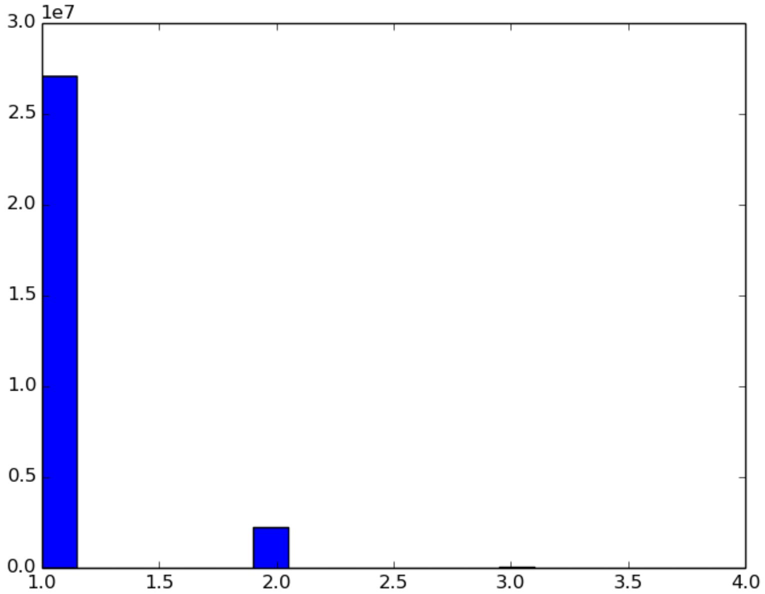


Figure 5.2: Histogram of the number of alignments per read. Only reads with more than zero alignments are taken into consideration.

5.1.3. Host filtering

There were basically no reads aligned to a host. Besides BLAST-ing a 10k sample, whole dataset analysed with Tophat, which runs upon Bowtie 2, against human genome. Only 1% of all reads(about 370 000) got aligned.

5.1.4. CDS analysis phase

In total were identified 34 411 CDSs, both ribosomal and non-ribosomal, initially spanning about 15 000 different species.

As can be seen on figure 5.3, majority of coding sequences has mean coverage lower than 25 reads per nucloetide, while length is fluctuating around a few thousand bases.

5.1.4.1. Ribosomal coding sequences

Of all CDSs, 2627 of them were identified as ribosomal ones. They are originating from 380 different species. That means that about 8% of all CDSs, are spanning 2.5% of all identified species.

Also, ribosomal CDSs tend to have a slightly larger mean coverage which is 16.52

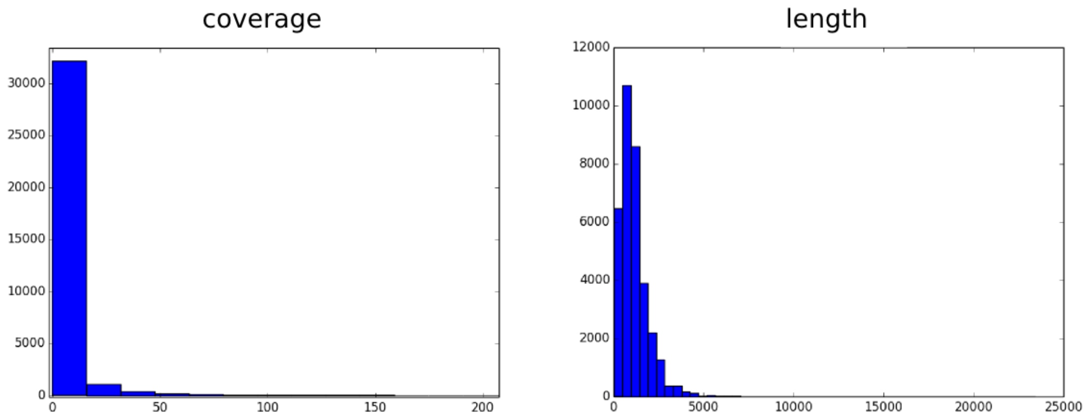


Figure 5.3: Length and coverage histogram of all coding sequences identified in a dental microbiome sample.

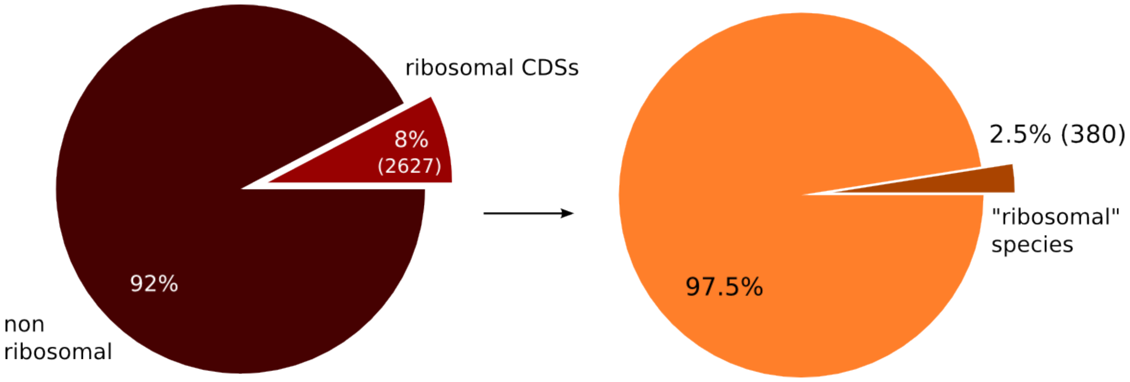


Figure 5.4: Influence of selecting ribosomal CDSs on the selection of ideintified species.

reads per base, while the maximum mean coverage is 1465.97 reads per base.

Taking all ribosomal reads into consideration and species they are defining, majority of reads could be assigned. Besides the group of reads without alignments, only 13% of reads could not be assigned to any of the species estimated by ribosomal coding sequences.

For the idea of selecting only ribosomal CDSs of "good" coverage quality, we tried a criterium of taking only those CDSs whose mean coverage value is equal or greater to the mean value of all ribosomal CDSs' mean coverage values. It determined 72 species as present and has proven to select CDSs with the desired shape of coverage, but reduced percentage of assigned reads from 67% to 53%.

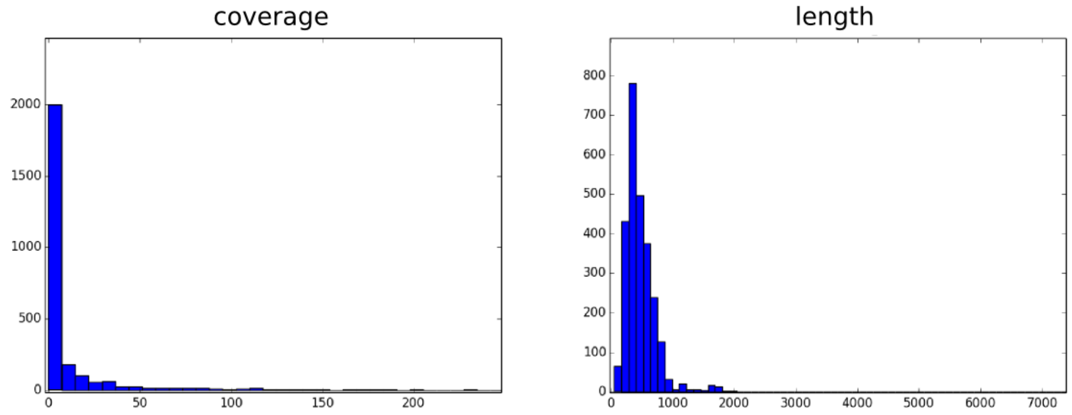


Figure 5.5: Length and coverage histogram of ribosomal coding sequences identified in a dental microbiome sample.

5.1.5. Determined species

According to [5] and [23], which are based on 16S rRNA analysis, specied assigned with greater abundance are also present in their estimation of sample content, especially genus of Streptoccocus. In both cases were analyzed samples from both healthy and diseased individuals and conclusion was that those who have never suffered from caries have significanlty different microbial content from those who had. Both studies agree there are present hundreds of different bacterial species, while some of them are having key role in development of oral diseas.

5.2. Time of execution

Table 5.2: Execution of dental microbiome sample analysis per phases.

phase	duration(s)
taxonomy tree loading	7.15
alignment file loading	627.18
referenced records loading	422.58
mapping alignments to genes	2188.59
creating `CDSContainer`	76.54
total	55.36 min

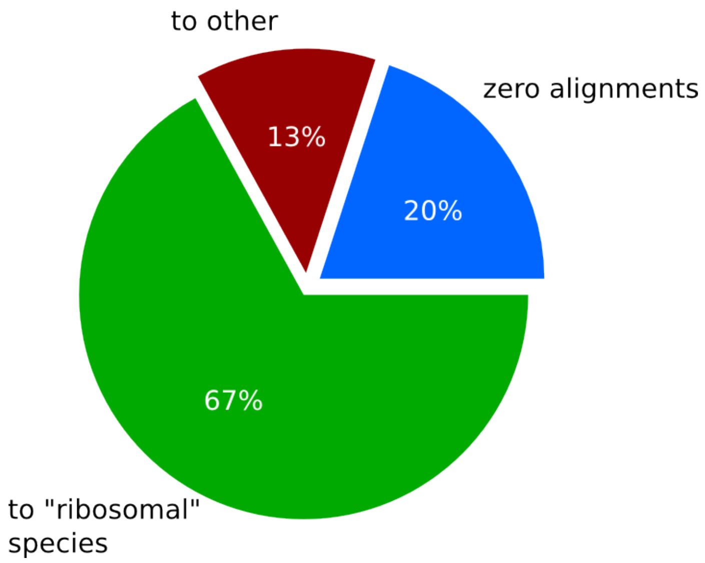


Figure 5.6: Distrubution of read assignment according to species defined by ribosomal CDSs for dental microbiome dataset.

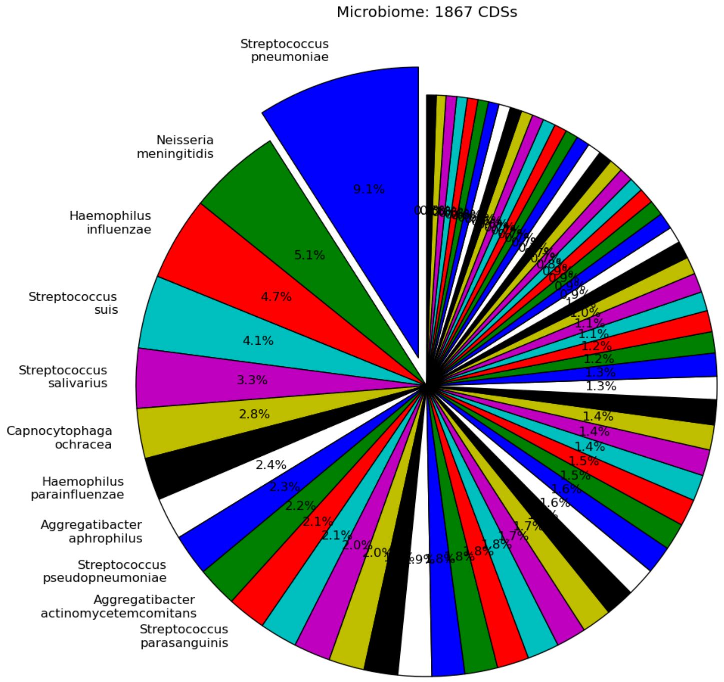


Figure 5.7: Abundance of present species with more than 10 ribosomal coding sequences in dental microbiome dataset.

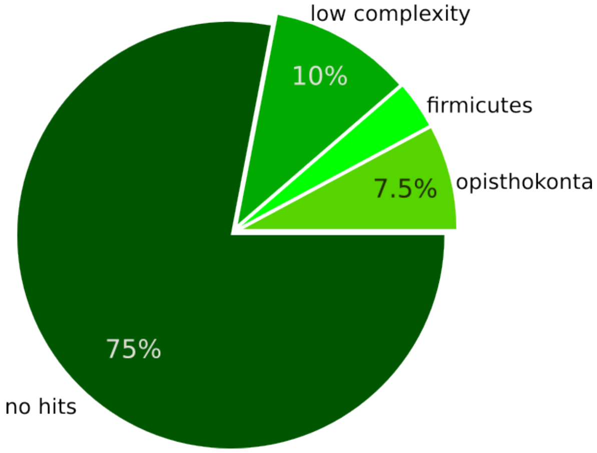


Figure 5.8: MEGAN output for 10k sample of reads without alignment in dental microbiome sample. Majority of reads is assigned to "No hit" and "Low complexity" nodes which is concordant with output of BWA MEM algorithm.

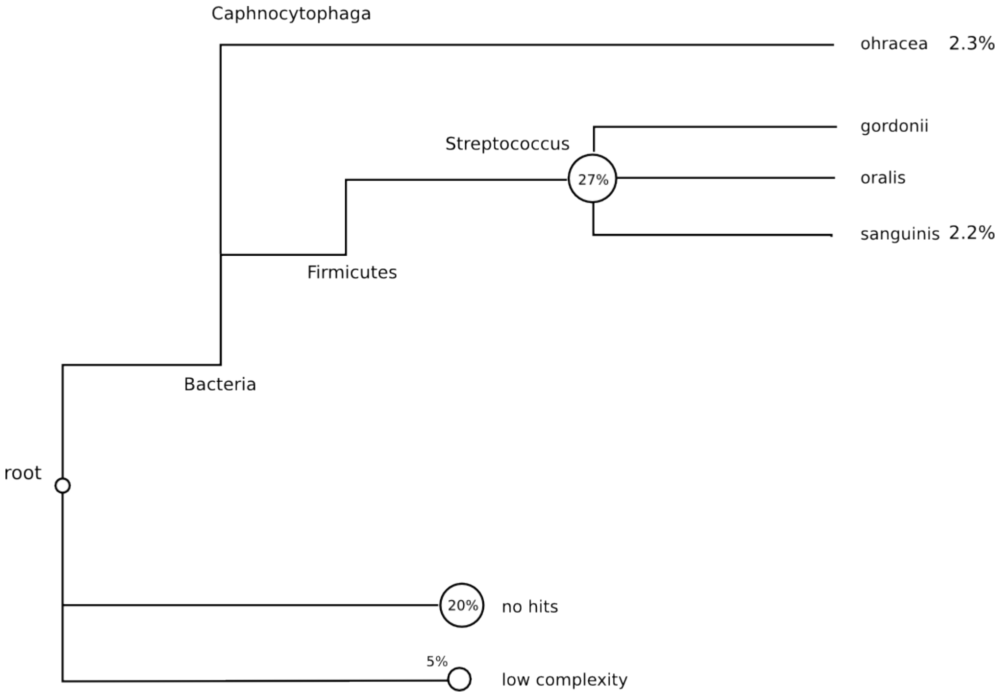


Figure 5.9: MEGAN part of output for the first 10000 reads of a dental microbiome sample. Similar as in our estimation, *Streptococcus* expresses the highest abundance.

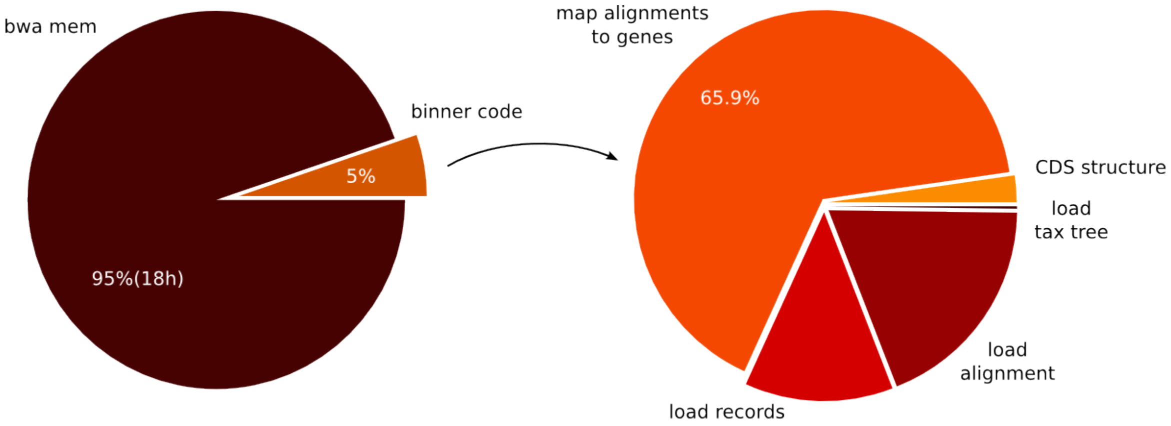


Figure 5.10: Graphical preview of distribution of execution time for dental microbiome dataset.

6. Conclusion and outlook

Metagenomics is a new field of research that still offers many challenges to those who dare to accept them. Research community is still in search for some foundation, base principles that will serve as a stepping stone for a new generation of innovations in both medicine and biology. Often are bioinformaticians compared to cavemen, as they are trying many different ways and boldly experimenting to achieve new conclusions. In this borderless freedom lies both fright and beauty of this area.

This thesis is adressing only a small part of the whole picture and its mission is to test the presented idea of microbiome characterization by analysing presence of ribosomal coding sequences. First is given an overview and existent categorization of related methods by which presented solution fits into taxonomy dependent, alignment based methods. Limitations and advantages of different approaches are given.

Needed reference and taxonomy data is obtained from standard sources, processed and saved into an optimal format for our purposes. On the top of it is built custom-fitted interface to ensure the both efficient and meaningfully structured interaction with underlying data.

Basic biology is briefly explained which allowed us to setup a problem model and get a general idea of the interactions and processes that are happening in this environment and driving our research. Idea and methods are described in details through data processing phases, with accent on bottlenecks and computationally demanding parts. Each phase is described as a separate part, with its own inputs and outpus in order to get a better insight in the overall process. In parallel is also explained implementation and runtime data structures together with their construction relations and whole lifecycle.

Finally, the solution is ran on a real environmental dataset, and it is analysed both in terms of estimated sample content and technical features as running time.

6.1. Future directions

As it was expected for the alignment based method that our binner is, the most limiting factor is the running time of alignment phase, taking up more than 90% of overall amount and being very resource demanding. It would be interesting to employ some ideas and techniques that could reduce initial search space. Optimising other phases in time efficency would currently gain very little improvement.

Another open question is how to further assign reads that do not have any alignments to determined "ribosomal" species. Also, it is pretty limiting factor that BWA has aligned great majority of reads to only one position on reference database, which greatly reduces possibilites.

It would also very interesting to find more RNA-seq sets of high quality and analyse them, and also compare with more different solutions. Another challenge for many of them is to make them mutually comparable, as many different ways of evaluation are employed in different solutions.

Finally, in the future is expected further advancement of sequencing technology, meaning greater read lengths and lesser error rates. That also means more sequenced data, increasing the reference knowledge and directly quality of metagenomic solutions. Nevertheless, we believe that this is a very promising area which will very soon provide new and unexpected answers to a lot of our questions.

BIBLIOGRAPHY

[1] Human Microbiome Project RNA-seq data resources webpage, Jun 2014. URL www.hmpdacc.org/RSEQ/.

[2] T. Abe, H. Sugawara, M. Kinouchi, S. Kanaya, and T. Ikemura. Novel phylogenetic studies of genomic sequence fragments derived from uncultured microbe mixtures in environmental and clinical samples. *DNA Res.*, 12(5):281–290, 2005.

[3] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, and D. J. Lipman. Basic local alignment search tool. *J. Mol. Biol.*, 215(3):403–410, Oct 1990.

[4] R. I. Amann, W. Ludwig, and K. H. Schleifer. Phylogenetic identification and in situ detection of individual microbial cells without cultivation. *Microbiol. Rev.*, 59(1):143–169, Mar 1995.

[5] P. Belda-Ferre, L. D. Alcaraz, R. Cabrera-Rubio, H. Romero, A. Simon-Soro, M. Pignatelli, and A. Mira. The oral metagenome in health and disease. *ISME J*, 6(1):46–56, Jan 2012.

[6] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and E. W. Sayers. GenBank. *Nucleic Acids Res.*, 37(Database issue):26–31, Jan 2009.

[7] A. Brady and S. L. Salzberg. Phymm and PhymmBL: metagenomic phylogenetic classification with interpolated Markov models. *Nat. Methods*, 6(9):673–676, Sep 2009.

[8] K. Chen and L. Pachter. Bioinformatics for Whole-Genome Shotgun Sequencing of Microbial Communities. *PLoS Comput Biol 1(2): e24, doi:10.1371/journal.pcbi.0010024*, July 2005.

[9] N. N. Diaz, L. Krause, A. Goesmann, K. Niehaus, and T. W. Nattkemper. TACOA: taxonomic classification of environmental genomic fragments using a kernelized nearest neighbor approach. *BMC Bioinformatics*, 10:56, 2009.

[10] K. U. Foerstner, C. von Mering, S. D. Hooper, and P. Bork. Environments shape the nucleotide composition of genomes. *EMBO Rep.*, 6(12):1208–1213, Dec 2005.

[11] M. Horton, N. Bodenhausen, and J. Bergelson. MARTA: a suite of Java-based tools for assigning taxonomic status to DNA sequences. *Bioinformatics*, 26(4): 568–569, Feb 2010.

[12] D. H. Huson, A. F. Auch, J. Qi, and S. C. Schuster. MEGAN analysis of metagenomic data. *Genome Res.*, 17(3):377–386, Mar 2007.

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

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

[15] Bo Liu, T. Gibbons, M. Ghodsi, and M. Pop. Metaphyler: Taxonomic profiling for metagenomic sequences. In *Bioinformatics and Biomedicine (BIBM), 2010 IEEE International Conference on*, pages 95–100, Dec 2010. doi: 10.1109/ BIBM.2010.5706544.

[16] S. S. Mande, M. H. Mohammed, and T. S. Ghosh. Classification of metagenomic sequences: methods and challenges. *Brief. Bioinformatics*, 13(6):669–681, Nov 2012.

[17] A. C. McHardy, H. G. Martin, A. Tsirigos, P. Hugenholtz, and I. Rigoutsos. Accurate phylogenetic classification of variable-length DNA fragments. *Nat. Methods*, 4(1):63–72, Jan 2007.

[18] F. Meyer, D. Paarmann, M. D'Souza, R. Olson, E. M. Glass, M. Kubal, T. Paczian, A. Rodriguez, R. Stevens, A. Wilke, J. Wilkening, and R. A. Edwards. The metagenomics RAST server - a public resource for the automatic phylogenetic and functional analysis of metagenomes. *BMC Bioinformatics*, 9: 386, 2008.

[19] M. H. Mohammed, T. S. Ghosh, N. K. Singh, and S. S. Mande. SPHINX–an algorithm for taxonomic binning of metagenomic sequences. *Bioinformatics*, 27 (1):22–30, Jan 2011.

[20] H. Noguchi, J. Park, and T. Takagi. MetaGene: prokaryotic gene finding from environmental genome shotgun sequences. *Nucleic Acids Res.*, 34(19):5623–5630, 2006.

[21] A. Patel, A. Picard, E. Jhong, J. Hylton, M. Smart, and M. Shields. Google Python style guide for comments, Jun 2014. URL `http://google-styleguide.googlecode.com/svn/trunk/pyguide.html#Comments`.

[22] J. Peterson, S. Garges, M. Giovanni, P. McInnes, L. Wang, J. A. Schloss, V. Bonazzi, J. E. McEwen, K. A. Wetterstrand, C. Deal, C. C. Baker, V. Di Francesco, T. K. Howcroft, R. W. Karp, R. D. Lunsford, C. R. Wellington, T. Belachew, M. Wright, C. Giblin, H. David, M. Mills, R. Salomon, C. Mullins, B. Akolkar, L. Begg, C. Davis, L. Grandison, M. Humble, J. Khalsa, A. R. Little, H. Peavy, C. Pontzer, M. Portnoy, M. H. Sayre, P. Starke-Reed, S. Zakhari, J. Read, B. Watson, and M. Guyer. The NIH Human Microbiome Project. *Genome Res.*, 19(12):2317–2323, Dec 2009.

[23] Scott N Peterson, Erik Snesrud, Jia Liu, Ana C Ong, Mogens Kilian, Nicholas J Schork, and Walter Bretz. The dental plaque microbiome in health and disease. *PloS one*, 8(3):e58487, 2013.

[24] H. G. Ramulu, M. Groussin, E. Talla, R. Planel, V. Daubin, and C. Brochier-Armanet. Ribosomal proteins: toward a next generation standard for prokaryotic systematics? *Mol. Phylogenet. Evol.*, 75:103–117, Jun 2014.

[25] P. Ribeca and G. Valiente. Computational challenges of sequence classification in microbiomic data. *Brief. Bioinformatics*, 12(6):614–625, Nov 2011.

[26] E. W. Sayers, T. Barrett, D. A. Benson, S. H. Bryant, K. Canese, V. Chetvernin, D. M. Church, M. DiCuccio, R. Edgar, S. Federhen, M. Feolo, L. Y. Geer, W. Helmberg, Y. Kapustin, D. Landsman, D. J. Lipman, T. L. Madden, D. R. Maglott, V. Miller, I. Mizrachi, J. Ostell, K. D. Pruitt, G. D. Schuler, E. Sequeira, S. T. Sherry, M. Shumway, K. Sirotkin, A. Souvorov, G. Starchenko, T. A. Tatusova, L. Wagner, E. Yaschenko, and J. Ye. Database resources of the National Center for Biotechnology Information. *Nucleic Acids Res.*, 37(Database issue):5–15, Jan 2009.

[27] R. Seshadri, S.A. Kravitz, L. Smarr, P. Gilna, and M. Frazier. CAMERA: A Community Resource for Metagenomics. *PLoS Biol 5(3):e75. doi:10.1371/journal.pbio.0050075*, Mar 2007.

[28] Y. Sun, Y. Cai, S. M. Huse, R. Knight, W. G. Farmerie, X. Wang, and V. Mai. A large-scale benchmark study of existing algorithms for taxonomy-independent microbial community analysis. *Brief. Bioinformatics*, 13(1):107–121, Jan 2012.

[29] H. Teeling, J. Waldmann, T. Lombardot, M. Bauer, and F. O. Glockner. TETRA: a web-service and a stand-alone program for the analysis and comparison of tetranucleotide usage patterns in DNA sequences. *BMC Bioinformatics*, 5:163, Oct 2004.

[30] Y. W. Wu and Y. Ye. A novel abundance-based algorithm for binning metagenomic sequences using l-tuples. *J. Comput. Biol.*, 18(3):523–534, Mar 2011.

Identify pathogen organisms from a stream of RNA sequences

Abstract

Metagenomics is an interesting new field of science focused on characterization of microbial communities from environmental samples. It is especially important as many organisms cannot be cultivated in a laboratory and therefore can reveal previously unknown species. This thesis presents a taxonomy-dependent alignment-based method for determining species present in a given metagenomic sample. Main premise is identification and analysis of ribosomal coding sequences which are considered to be indicators of presence of organisms. The solution works with RNA-seq data and is tested on a real dataset from an environmental sample. **Keywords:** metagenomics,

bioinformatics, RNA-Seq

Određivanje patogenih vrsta iz RNA sekvenciranih podataka

Abstract

Metagenomika je zanimljivo novo znanstveno područje čiji je zadatak karakterizacija mikroskopskih zajednica živih organizama. Posebna važnost je u tome što većina mikroorganizama ne može biti uzgojena u laboratoriju, stoga nam metagenomika otkriva mnoge nepoznate vrste. Ovaj rad predstavlja taksonomski ovisnu i na poravnanju sekvenci na referentni genom temeljenu metodu za određivanje vrsta prisutnih u danom metagenomskom uzorku. Pristup se temelji na identifikaciji i analizi ribosomskih kodirajućih sekvenci za koje se smatra da su indikatori prisutnih vrsta. Rješenje radi s RNA sekvenciranim podacima i testirano je na stvarnom uzorku iz prirodnog okruženja. **Keywords:** metagenomika, bioinformatika, RNA sekvenciranje