Comenius University, Bratislava

Faculty of Mathematics, Physics and Informatics

Computational Design of Probes for the Hyb-Seq Protocol

Bachelor thesis

2017

Michaela Šandalová

Comenius University, Bratislava
Faculty of Mathematics, Physics and Informatics

Computational Design of Probes for the Hyb-Seq Protocol

Bachelor thesis

Study programme:	Bioinformatics
Study field:	2508 and 1536 Computer science and biology
Department:	Department of Computer Science
Supervisor:	Mgr. Matúš Kempa

Bratislava, 2017
Michaela Šandalová

72481876

Comenius University in Bratislava
Faculty of Mathematics, Physics and Informatics

THESIS ASSIGNMENT

Name and Surname:	Michaela Šandalová
Study programme:	Bioinformatics (Joint degree study, bachelor I. deg., full time form)
Field of Study:	Computer Science, Informatics Biology
Type of Thesis:	Bachelor´s thesis
Language of Thesis:	English
Secondary language:	Slovak

Title: Computational Design of Probes for the Hyb-Seq Protocol

Aim: The goal of the thesis is to design probes for the Hyb-Seq protocol for Alyssum genus based on raw sequencing data. The starting point is an existing software pipeline called Sondovač (https://github.com/V-Z/sondovac). The goal is to explore and describe methods used in this tool, propose and test apropriate parameter settings and potentially propose improvements of the pipeline.

Supervisor:	Mgr. Matúš Kempa
Department:	FMFI.KI - Department of Computer Science
Head of department:	prof. RNDr. Martin Škoviera, PhD.
Assigned:	26.10.2016
Approved:	26.10.2016

doc. Mgr. Bronislava Brejová, PhD.
Guarantor of Study Programme

....
Student

....
Supervisor

Univerzita Komenského v Bratislave
Fakulta matematiky, fyziky a informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Michaela Šandalová
Študijný program: bioinformatika (Medziodborové štúdium, bakalársky I. st., denná forma)
Študijné odbory: informatika
biológia
Typ záverečnej práce: bakalárska
Jazyk záverečnej práce: anglický
Sekundárny jazyk: slovenský

Názov: Computational Design of Probes for the Hyb-Seq Protocol
Automatizovaný dizajn prób pre Hyb-Seq protokol

Cieľ: Cieľom práce je vytvoriť próby pre Hyb-Seq protokol pre rod Alyssum zo surových sekvenačných dát. Východiskom práce je existujúci nástroj Sondovač (https://github.com/V-Z/sondovac), pričom cieľom je naštudovať a popísať metódy použité v tomto nástroji, navrhnúť a otestovať vhodné nastavenia parametrov, prípadne navrhnúť vylepšenia nástroja.

Vedúci: Mgr. Matúš Kempa
Katedra: FMFI.KI - Katedra informatiky
Vedúci katedry: prof. RNDr. Martin Škoviera, PhD.

Dátum zadania: 26.10.2016

Dátum schválenia: 26.10.2016

doc. Mgr. Bronislava Brejová, PhD.
garant študijného programu

....
študent

....
vedúci práce

Poďakovanie: Thanks Obama.

Abstrakt

Slovenský abstrakt v rozsahu 100-500 slov, jeden odstavec. Abstrakt stručne sumarizuje výsledky práce. Mal by byť pochopiteľný pre bežného informatika. Nemal by teda využívať skratky, termíny alebo označenie zavedené v práci, okrem tých, ktoré sú všeobecne známe.

Kľúčové slová: jedno, druhé, tretie (prípadne štvrté, piate)

Abstract

Abstract in the English language (translation of the abstract in the Slovak language).

Keywords:

Obsah

Zoznam obrázkov

Zoznam tabuliek

Introduction

Computational biology is currently a blooming discipline, its methods and tools having wide use among scientists, especially in the subject of genetics. One such use lies in taxonomy: a determination of phylogenetic relationships and evolutionary history among various species or families [15].

Research in the fields of phylogenesis and taxonomy allows us better understanding of biodiversity, evolution or ecology and aids in identification and classification of living organisms, effectively showing their differences and similarities. The analysis of evolutionary history is called phylogeny and is represented by a tree diagram called a phylogenetic tree [16]. We will provide a more detailed description of phylogenetic trees in chapter 1.

Creation of phylogenetic trees and comparing organisms in general requires a large amount of data, usually in the form of DNA or RNA sequences. In the last few yars, the price of sequencing has gone down rapidly. However, phylogenetic trees require data from several organisms, and can prove to be time and money consuming. Moreover, we are sequencing plants, which tend to have much bigger and more complex genomes, along with additional genetic information from chloroplasts and mitochondria [18].

A modern aproach to sequencing - the next generation sequencing offers reduced time and is affordable even with more data. The next generation sequencing, or NGS for short is a name for several methods that are more effective than the previously used Sanger sequencing. Specifically, Hyb-Seq protocol [21] is a method that utilizes target enrichment and genome skimming to forego sequencing all of the genome. Hyb-Seq uses specifically designed probes to find conserved places within the genome, and thus enabling comparison among the species.

These probes are usually determined from orthologous low-copy nuclear loci combined with other types of information, for example mitochondrial and plastid genomes. However, finding these loci for non-model organisms is a difficult task. Loci can be selected from transcriptomes (set of all messenger RNA molecules), genomes, gene expression studies, or the literature. There is a lack of automated bioinformatic pipelines for selection of low-copy nuclear loci.

Sondovač is a script that offers relatively easy and automated creation of othologous low-copy nuclear probes from transcriptome and genome skim data for target enrichment [19]. Purpose of this thesis is to understand this tool, describe methods used in it and to create probes for plants from genus Alyssum using raw sequencing data. The resulting data are intended to be further used by the Hyb-Seq protocol.

In the first chapter, we will take a look at the biological motivation behind probe design and we will explain the most common terms used through the thesis and provide some information on phylogenetic trees. Chapter **??** offers insight into the biological data we are using and what we are trying to achieve with it. Chapter **??** will explain the Hyb-Seq protocol in more detail. Next, chapter **??** will take us through a detailed description of Sondovač and its methods. Chapter **??** will entail the results we got from creating probes for Alyssum. In conclusion, chapter **??** will go through possible improvements of the pipeline that is Sondovač, and their implementation.

Kapitola 1

Biological motivation and background

In this chapter we will introduce the terms and concepts of biology, genetics and computational biology that are commonly used through this thesis. We will explain phylogenetic trees, their significance in evolutionary biology and their connection to the Hyb-Seq protocol and probes.

1.1 Terminology overview, basic terms and concepts

1.1.1 Nucleic acids

Nucleic acids carry the genetic information of all known living things. Two nucleic acids are called RNA (ribonucleic acid) and DNA (deoxyribonucleic acid). They both consist of sequence of nucleotides - monomers that are made of a pentose - a sugar with 5 carbons (ribose in RNA, deoxyribose in DNA), a phosphate group, and a nitrogenous-base. The five most common bases are cytosine (C), guanine (G), adenine (A), thymine (T) and uracil (U). RNA contains C, G, A, U and DNA contains C, G, A, T. These bases create hydrogen bonds between each other as follows: C-G, A-T in DNA, and C-G, A-U in RNA. RNA is usually single stranded and the bases pair with each other within the same strand, creating 3D-structures. Most DNAs is a double helix - it has two complementary strands that pair with each other. [9]

1.1.2 Genes

A gene is a basic unit of heredity, a region in DNA that encodes some function, usually a protein. Several genes can encode a single trait or one gene can encode multiple traits. A position, or a place of a gene in DNA is called locus (plur. loci). Homologous genes are genes that share a common ancestor. More specificaly, homologous sequences are called orthologs, if two copies of the same gene are in two different species and they are called paralogs, if the gene was duplicated within the same genome. During evolution, orthologs retain the same function while paralogs (or one of them) can can gain new functions. Sequences of DNA that are converted into mature mRNA are called exons. The sequences that are between exons are called introns. Introns do not code proteins and their sequences can change frequently over time. Exons, on the other hand, are much more conserved. [9]

1.1.3 Transcription, translation

When making proteins, regions of DNA are transcribed into a shorter RNA that is complementary to the original DNA sequence. This RNA is called messenger RNA or mRNA. mRNA is then translated into proteins. The complete set of all mRNAs from a cell or a population of cells is called transcriptome. The transcriptome represents all genes that are being actively expressed.

Analogicaly, the DNA from mitochondria is called mitochondrion and the DNA from chloroplast is a plastome. [9]

1.1.4 Phylogenetic trees

In biology, the study of evolutionary history amongst organisms, species, populations, etc. is called phylogenetics. Heritable traits are evaluated to determine a phylogenetic relationship. Earlier, only morphologic traits could be used, but nowadays, DNA sequences or other genetic characteristics are also a valuable genetic markers used in phylogeny. [15]

A phylogenetic tree is a representation of such relationships. It's a branching diagram, where the taxa that are joined together have descended from a common ancestor.

In this thesis, we tried to find genetic markers which can be used to infer relations in phylogeny of a group of plants. [16]

Kapitola 2

Hyb-Seq Protocol

In this chapter we will describe in detail the Hyb-Seq Protocol and next generation sequencing. We will address their position in the process of creating a phylogenetic tree and their connection to the Sondovač.

2.1 Next generation sequencing

Next generation sequencing refers to faster and cheaper approaches to sequencing and acquiring phylogenetic information. The effectiveness of NGS is due to using better technology - sequencing platforms such as Illumina, Roche 454 sequencer and others, which can sequence many shorter sequences at the same time. The process where the sequences from diffent individuals are sequenced subsequently is known as pooling. [8] When sequencing a whole genome, it's divided into small fragments which are then sequenced subsequently. Several copies of genome are used and thus each base is sequenced multiple times; often in a different fragment. The resulting data is then put together using bioinformatics analyses. [10]

2.2 Hyb-Seq Protocol

Hyb-seq protocol is the combination of target enrichment and genome skimming. It enables data collection for low-copy nuclear genes and high-copy genomic targets for evolution studies and plant systematics. In hyb-seq, suitable probes are first created to serve in target enrichment. The principle of target enrichment is selectively finding regions of interest in a genome before sequencing and only processing those, thus making the following sequencing process more effective. [21]

Genome skimming refers to shallow sequencing approaches that aim to find conversed ortholog sequences. The data for Sondovač thesis was acquired by using genome skimming and thus getting paired-end genome data. [12] From the sequenced data, a phylogenetic tree can be built. The script Sondovač, we are using in this thesis, is a tool that selects orthologous low-copy nuclear genes from provided data. The goal is to find effective markers to use in target enrichment. [19]

2.3 Orthologous genes

Orthologous genes are fundamental for phylogeny, since they can be used as markers. Selecting orthologous genes that are effective as markers is difficult as gene duplication and deletion is making it hard to tell orthologs from paralogs. Single-copy paralogs that have undergone lineage-specific changes can be mistaken for orthologs. [20] Distinguishing orthologs from paralogs is especially difficult in angiosperms, where polyploidization is a common occurence. [22]

2.4 Low-copy nuclear genes

Low-copy nuclear genes are genes from nucleus that can be found in the genome in few or single copies. Highly conserved orthologous low-copy nuclear genes have found their use as a source of phylogenetic information. They proved to be effective markers to track organismal evolution. [22] Most of the protein-coding cell genes can be found in nucleus. Compared to genes from other organelles, nuclear genes from eukaryotic organisms consist from several chromosomes. Because of this, genes of eucaryotic organisms are evolutionary unlinked; either on different chromosomes or sufficiently far apart. [20]

Finding low-copy nuclear genes has been constrained by technical limitations. High sequencing throughput of current platforms, such as Illumina, combined with target enrichment enables us to sequence large amounts of low-copy nuclear loci effectively. [17]

Kapitola 3

Existing work - Sondovač

In this chapter, we will take a closer look at the pipeline and the workflow of Sondovač. We will also specify the input and output data format and consider them from the informatic and more formal point of view. Finally, detailed description of the tools and software that are used by Sondovač will be listed.

3.1 Sondovač origins

Sondovač is a Czech neologism standing for "Probe-maker". It was created by Roswitha Schmickl, Aaron Liston, Vojtěch Zeisek and others. Sondovač is an interactive and automated script to create orthologous low-copy nuclear probes for further use by the Hyb-Seq protocol. It uses transcriptome and genome skim data. Sondovač does not require strong bioinformatic skills nor high-performance computer. It is intended for either Linux or Mac OS X.

3.2 The workflow overview

Sondovač is written in BASH, an Unix shell and a command language [13]. It has three parts: Sondovač_part_a.sh, the Geneious [3] intermediate part and Sondovač_part_b.sh. Between the parts a and b of Sondovač it is necessary to manualy input the output from the part a into another software - Geneious, for data processing and then run the part b on the output from the software.

3.2.1 Sondovač part a

3.2.2 The input and output data

3.2.3 Input data for part a

Input data for part a of Sondovač consist of 5 files, 1 of them being optional.

1. Transcriptome input file in FASTA format

   The file consists of several blocks with same format: On the first line of a block there is a '>' character followed by an unique description of the sequence, in this case a number. On the next few lines, there is the actual sequence composed of 'A', 'C', 'G' or 'T' characters.

2. Plastome reference sequence input file in FASTA format

   This file is used in both parts. It consists of a single long sequence and the starting line with '>' and the sequence's unique description.

3. Mitochondriome reference sequence input file in FASTA format (Optional)

   The file contains a single sequence along with the first line describing it. It is optional, because the size of a plant mitochondrial genome can vary greatly and have high rearrangement rates.

4. Paired-end genome skim input file in FASTQ format (first file, the forward reads)

5. Paired-end genome skim input file in FASTQ format (second file, the reverse reads)

   Other than the input files, Sondovač requires a minimum total locus length to be set.

   Output data from part a are the input files for Geneious.

3.2.4 Input data for part b

Input data for part b of Sondovač consist of 3 files and include the output data from Geneious.

(a) Input file in TSV format (output of Geneious assembly)

(b) Input file in FASTA format (output of Geneious assembly)

(c) Plastome reference sequence input file in FASTA format

This file is used in both parts. It consists of a single long sequence and the starting line with '>' and the sequence's unique description.

3.3 Workflow, pipeline

The part a of the script covers 6 steps:

(a) Removing the transcripts that share $\geq 90\%$ sequence similarity

We want to get low-copy nuclear orthologous probes. To minimize the enrichment of multi-copy loci, the Sondovač script removes transcripts that are too similar; share $\geq 90\%$ sequence similarity. This is done using BLAT and UNIX commands. From this we get unique transcripts that we match against processed reads.

(b) Removing the reads of plastid origin

Since we want only nuclear probes, the raw paired-end genome data is stripped of the reads that have plastid origin, utilizing the reference input sequences. Tools used for this are Bowtie 2 and Samtools.

(c) Removing the reads of mitochondrial origin

In the same manner, the reads of mitochondrial origin are removed from the paired-end genome data, if the list of mitochondrial sequences is present. Bowtie 2 and Samtools are used.

(d) Combining the paired-end reads

Subsequently, the paired-end reads without plastid and mitochondrial reads are combined using FLASH.

(e) Matching the unique transcripts and the filtered, combined genome skim reads sharing $\geq 85\%$ sequence similarity.

Sequences that are well-preserved and therefore present amongst several related species make good genetic markers. Since transcripts are the sequences that are translated into proteins, they rarely change their genetic composition, eg. the bases they consist of. The Sondovač script matches the unique transcripts with the processed paired-end genome skim data. Using BLAT and Unix commands, only sequences that have $\geq 85\%$ similarity are kept.

(f) Filtering the BLAT output

   i. Choosing the transcript or genome skim sequences for further processing Either transcript or genome sequences are used as the basic sequences for designing the probes. The choice depends on the phylogenetic depth that should be obtained, but it doesn't matter if the researched taxa are closely related. Defaultly, the genome skim data is used.

   ii. Removing the transcripts with more than 1000 BLAT hits

While making an alignment, BLAT makes hits - short similar sequences. The transcripts that achieve $\geq$ 1000 BLAT hits while matching them with filtered combined genome skim reads are removed to avoid repetitive elements. Unix commands are used for filtering and the amount of hits can be adjusted; it can be an integer ranging from 100 to 10000.

iii. Removing the transcript or genome skim BLAT hits containing masked nucleotides

Hits that contain masked nucleotides (nucleotides that are unknown or have various options) are removed as well.

3.3.1 Geneious

After filtering the BLAST output, de novo assembly of BLAT hits into larger contigs commences. This part is done by Geneious, a desktop software platform that can analyze, asslemble or align sequences. The user has to take output of Sondovač part a and manually process the data with Geneious using the medium sensitivity / fast setting.

3.3.2 Sondovač part b

Sondovač part b covers 4 steps. The output data from Geneious assembly and the plastome reference are the input files for part b.

(a) Retention of those contigs that comprise exons greater or equal than bait length and have a certain total locus length

Sequences that are too short aren't good genetic markers, because it's more likely that their presence in the genome is coincidental. Thus, the script picks those contigs that comprise exons with a minimum bait length greater than 120 base pairs and have a set minimum total locus length (the recommended length is 600bp and it has to be a multiple of the bait length), although these values can be adjusted. The selection is done using Unix commands.

(b) Removal of probe sequences sharing greater or equal than 90% sequence similarity

We don't want the probes to target multiple similar loci, so similar sequences or duplicates are removed using CD-HIT.

(c) Retention of those contigs that comprise exons greater or equal than bait length and have a certain total locus length

A second filtering for sequences that are too short commences. The parameters are the same as before.

(d) Removal of probe sequences sharing greater or equal than 90% sequence similarity with the plastome reference

Lastly, the sequences that are present in the plastome reference are removed, since we want to ensure we are targeting only nuclear probes. This is done by BLAT and Unix commands and only sequences that have similarity $\geq$ 90% are removed.

3.3.3 Additional removal of plastid sequences

If any remaining plastid sequences are detected, they have to be removed manually from the final output of part b of Sondovač script, since we preffer nuclear probes and plastid genes would occupy too much space on the Illumina lane during target enrichment.

3.4 Output data

Each part of the Sondovač pipeline has its own output data. Some of them are further used in the pipeline and other files are purely for user. In this section, we will take a look at the output data from various parts of the Sondovač script. An asterisk ($*$) in the name of file indicates the part of the file name that is specified by the user with the '-o' flag. Default value is 'output'.

3.4.1 Output data - part a

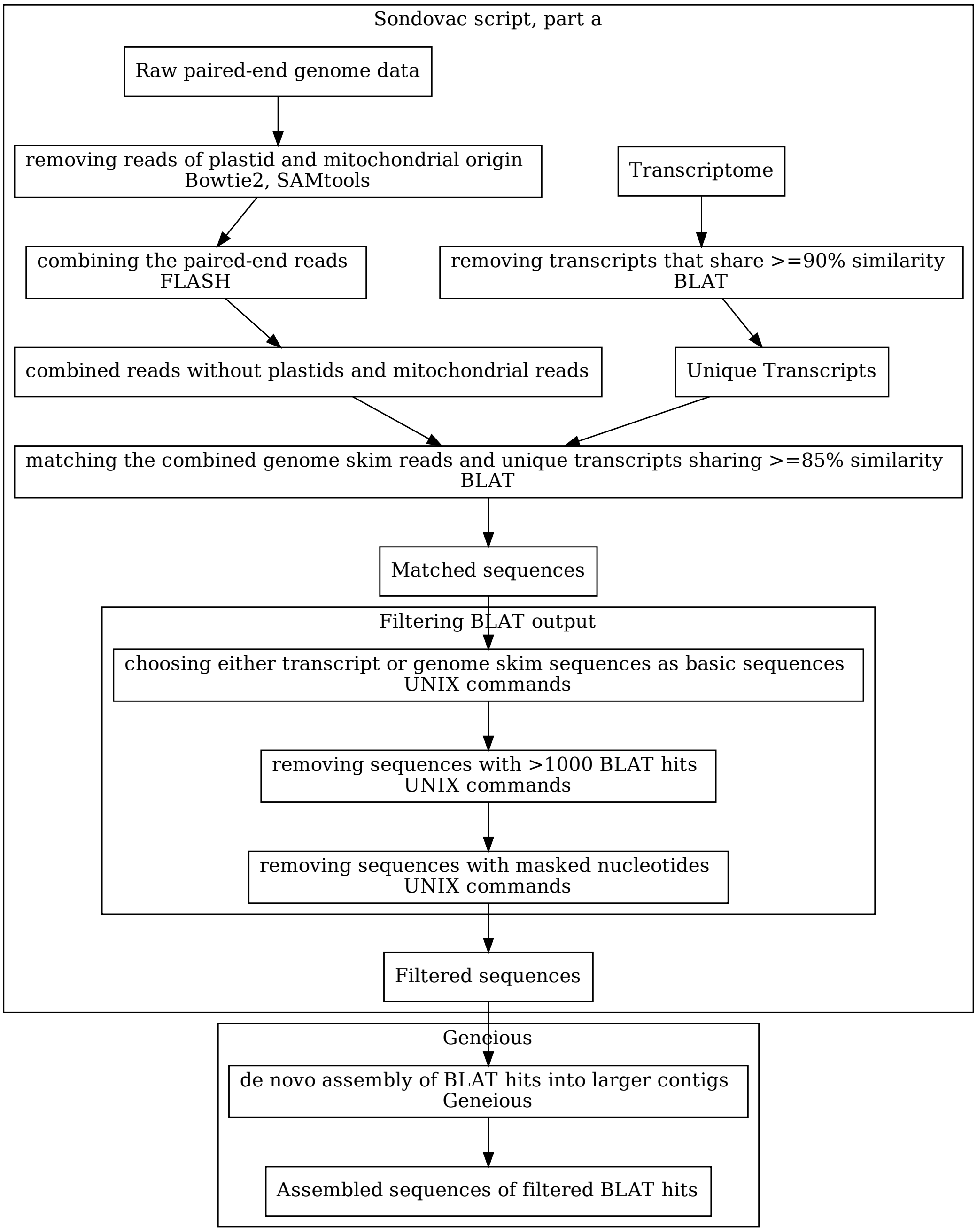
Only the last file is necessary for further processing as the input for Geneious. However, other files may be useful for the user. Sondovač, part a, creates the following files:

1. $*$_renamed.fasta

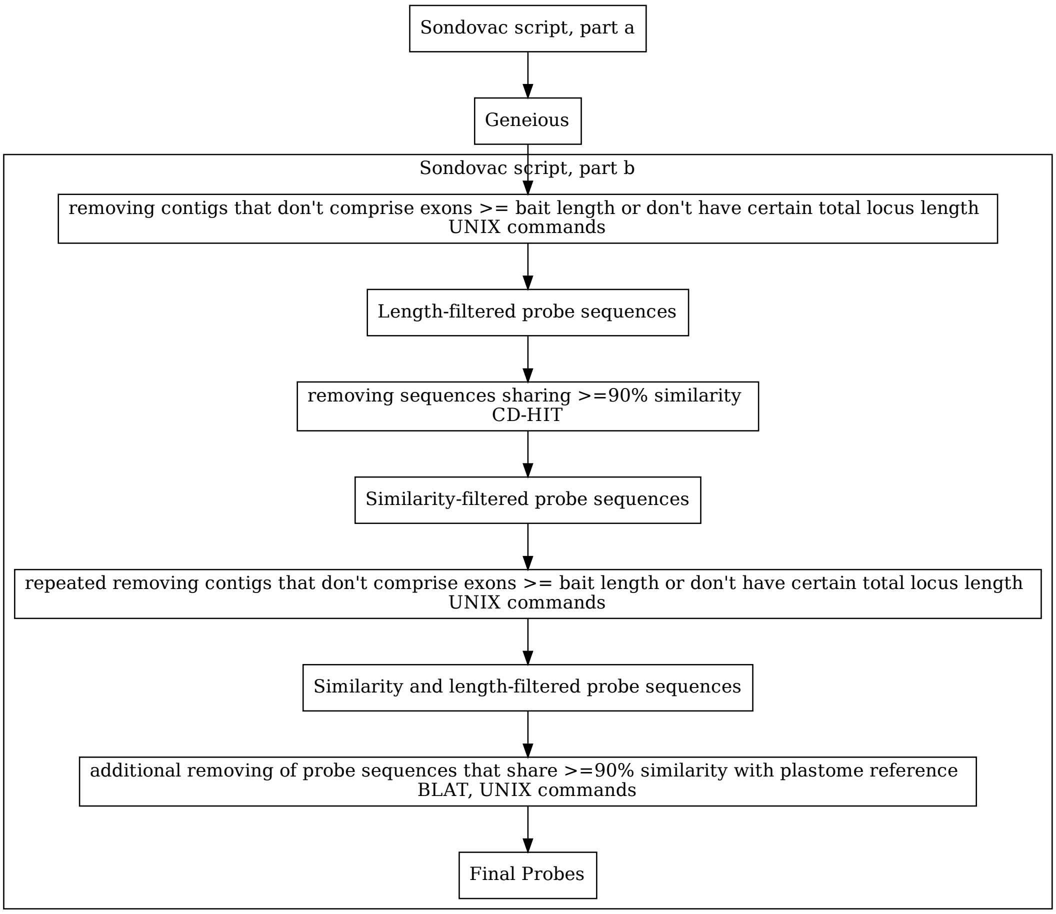
   Copy of the transcriptome file where the labels of FASTA sequences were changed - unique numbers now correspond to the original file's line numbers.

2. $*$_old_and_new_names.tsv

   This file contains two columns - labels of the original sequences as it was in the transcriptome file provided by the user in the first, and new sequence labels in the second. This file and the $*$_renamed.fasta file can be used to trace back certain sequences or probes.


Sondovac script, part a
Raw paired-end genome data
removing reads of plastid and mitochondrial origin
Bowtie2, SAMtools
Transcriptome
combining the paired-end reads
FLASH
removing transcripts that share >=90% similarity
BLAT
combined reads without plastids and mitochondrial reads
Unique Transcripts
matching the combined genome skim reads and unique transcripts sharing >=85% similarity
BLAT
Matched sequences
Filtering BLAT output
choosing either transcript or genome skim sequences as basic sequences
UNIX commands
removing sequences with >1000 BLAT hits
UNIX commands
removing sequences with masked nucleotides
UNIX commands
Filtered sequences
Geneious
de novo assembly of BLAT hits into larger contigs
Geneious
Assembled sequences of filtered BLAT hits

Obr. 3.1: Flowchart illustrating the workflow of Sondovač script, part a and step with the Geneious software.


Sondovac script, part a
Geneious
Sondovac script, part b
removing contigs that don't comprise exons >= bait length or don't have certain total locus length
UNIX commands
Length-filtered probe sequences
removing sequences sharing >=90% similarity
CD-HIT
Similarity-filtered probe sequences
repeated removing contigs that don't comprise exons >= bait length or don't have certain total locus length
UNIX commands
Similarity and length-filtered probe sequences
additional removing of probe sequences that share >=90% similarity with plastome reference
BLAT, UNIX commands
Final Probes

Obr. 3.2: Flowchart illustrating the workflow of Sondovač script, part b.

3. *_blat_unique_transcripts.psl

   Unique transcripts that are the output of BLAT after removal of the sequences that have $\geq$ 90% similarity.

4. *_unique_transcripts.fasta

   Unique transcripts that are the output of BLAT in FASTA format.

5. *_genome_skim_data_no_cp_reads

   Genome skim data without the cpDNA reads - after the chloroplast DNA is removed.

6. *_genome_skim_data_no_cp_no_mt_reads

   Genome skim data without the mtDNA reads - after the mitochondrial DNA is removed. This file is present only if the mitochondriome reference data was provided.

7. *_combined_reads_co_cp_no_mt_reads

   Paired-end genome skim reads that are combined.

8. *_blat_unique_transcripts_versus_genome_skim_data.pslx

   Output of BLAT after matching the unique transcripts and combined paired-end genome skim reads that have $\geq$ 85% similarity.

9. *_blat_unique_transcripts_versus_genome_skim_data.fasta

   Output of BLAT after matching the unique transcripts and combined paired-end genome skim reads in FASTA format.

10. *_blat_unique_transcripts_versus_genome_skim_data-no_missing_fin.fsa

    Final sequences for further use in Geneious. This is a FASTA file and the only file that is used further in the pipeline.

3.4.2 Output data - Geneious

The output data from Geneious is the input data for the Sondovač script, part b, along with the plastome reference. Geneious output consists of two files:

1. Assembled sequences from Geneious - a file exported in TSV format

2. Assembled sequences from Geneious - a file exporded in FASTA format

3.4.3 Output data - part b

The Sondovač script, part b creates the following files:

1. ∗_prelim_probe_seq.fasta

   Preliminary probes in FASTA format.

2. ∗_prelim_probe_seq_cluster_100.fasta

   Unclustered exons and exons that have 100% sequence identity - the bases match exactly between two different sequences. The file is in FASTA format.

3. ∗_prelim_probe_seq_cluster_90.clstr

   Unclustered exons and exons that have more than certain sequence identity in CLSTR format.

4. ∗_unique_prelim_probe_seq.fasta

   Unclustered exons and exons that have less than a certain sequence similarity.

5. ∗_similarity_test.fasta

   Contigs that comprise exons with greater or equal minimum bait length and have a certain minimum total locus length.

6. ∗_target_enrichment_probe_sequences_with_pt.fasta

   Probes in FASTA format, that contain putative plastid sequences. If any BLAT hits were present, the possible plastid sequences are listed in the ∗_possible_cp_dna_gene_i file.

7. ∗_possible_cp_dna_gene_in_probe_set.pslx

   A list of putative chloroplast sequences in case of any BLAT hits. These sequences are best to be removed from the final probe list, as we preffer nuclear probes.

8. ∗_target_enrichment_probe_sequences.fasta

   The final list of probes in FASTA format.

3.5 Used software and tools

Sondovač uses a broad variety of tools and scientific software packages, both freeware and payware. It is mainly coded in BASH, but it also uses smaller python scripts. We will take a closer look at what each of the tools is and what it does in the Sondovač script.

3.5.1 Programming languages

Here we will take a look at programming languages the Sondovač script uses.

1. BASH

   BASH is a command line interpreter (or shell) and a command language for Unix. It's a programming scripting language accessible through a terminal in any Unix-based operating system. It is a free software. Scripts written in BASH usually have the extension *.sh.

   The Sondovač script is written in BASH. Using this language, it runs other programms and scripts. It is also used to work with files or manipulate and filter the data.

2. Python

   Python is an interpreted programming language. Several scripts that Sondovač uses are coded in Python.

3.5.2 Tools and software

In this section, we will list the most inportant tools and software that Sondovač uses. Some of the software changed or was replaced with a newer version of Sondovač, but the tools and software listed here comprise are essential part of Sondovač. Sondovač uses the following software:

1. BLAT

   BLAT – the BLAST like alignment tool – is a pairwise sequence alignment algorithm. [14] It is used by the Sondovač script to match reads to unique transcripts and for other alignments. It is used in both parts of the script.

2. Bowtie2

   Bowtie2 is a memory-efficient tool used for aligning sequencing reads to longer reference sequences. It keeps an FM index to save memory. Bowtie2 has local, gaped and paired-end alignment modes. [11] It is used in Sondovač, part a to find plastome or mitochondrione sequences in the genome.

3. CD-HIT

   CD-HIT is a program for clustering and comparing protein or nucleotide sequences. It can compare two databases and identify sequences that are similar above a threshold. [1] It is used in Sondovač, part b, to remove sequences that share similarity above 90%.

4. FLASH

   FLASH – Fast Length Adjustment of SHort — is an accurate and fast tool to combine or merge paired-end reads. It works the best on fragments that are shorter than twice the length of reads. The longer the reads, the better the result of assembly. [2]

   It is used in Sondovač, part a to combine paired-end reads.

5. Geneious

   Geneious is a payware software that can be used for organizing, analyzing, assembling or aligning DNA. It can run in interactive or non-interactive mode.

   Geneious is an intermediate step between Sondovač part a and part b. It requires the data to be put in it manually. It is used for de novo assembly of the genome or transcript skim BLAT hits. It creates larger contigs from the data. There are plans to replace the part that Geneious does by another free open-source command line tool and thus make the Sondovač pipeline fully automated.

6. Grab_singleton_clusters.py

   Grab_singleton_clusters.py is a python script designed in the paper "K. Weitemier, S.C.K. Straub, R. Cronn, M. Fishbein, A. McDonnell, R. Schmickl, and A. Liston. 2014. Hyb-Seq: Combining target enrichment and genome skimming for plant phylogenomics Applications in Plant Sciences 2(9): 1400042"[21]. It finds clusters from a CD-HIT *.clstr file that only one sequence with 100% identity. If it contains multiple sequences with 100% identity, it will choose the longes sequence possible. As an output, it creates a *.clstr format. [4] This program is used in the Sondovač, part b.

7. Samtools

   Samtools is a collection of programs for manipulating with high-throughput sequencing data. Three separate repositories are present:

   (a) Samtools - Working with files in SAM/BAM/CRAM format

   (b) BCFtools - Working with files in BCF_2/VCF/gVCF format

   (c) HTSlib - A C library for reading and writing high-throughput sequencing data

   [7]

   In Sondovač script, SAMtools is used in part a to convert SAM files to BAM files.

3.6 Use of Sondovač

Here we will write about how to use Sondovač, what modes it runs and what flags or settings can be used.

DO I EVEN WANT THIS PART HERE?

Kapitola 4

Practical work and results

In this chapter, we will specify the data we are using, provide a biological background for them and describe the way in which they are processed. We will describe the practical work and process of creating suitable probes. We will take a look at parameters and specifications we used during the run of the Sondovač script.

4.1 Input data

In this chapter we will take a closer look at the data we are using. We will cover the biological aspect of the data. We will also look at how such data is obtained and how it is further processed.

Sondovač uses a transcriptome, a genome and possibly a chloroplast and mitochondrion. We worked with two different sets of input data. These data shared most of the files, differing only in genome.

First set used the following data:

1. Transcriptome: Alyssum alyssoides
2. Genome: Odontarrhena tortuosa
3. Mitochondrion: Arabidopsis thaliana
4. Chloroplast: Arabidopsis thaliana

Second set used the following data:

1. Transcriptome: Alyssum alyssoides
2. Genome: Alyssum gmelinii
3. Mitochondrion: Arabidopsis thaliana
4. Chloroplast: Arabidopsis thaliana

4.1.1 Nickel and metallic genes

We were especially interested in genes that bind nickel and metal that might be present in these genomes. Therefore, we matched the possible probes against nickel and metallic reference and primarily picked those with the greatest similarity.

4.1.2 What kind of data we are using

All the data we are using are from plants, specifically, alyssum alyssoides, alyssum gmelinii, odontarrhena tortuosa and arabidopsis thaliana, which are all species of flowering plant from the brassicaceae (mustard) family. This thesis is a part of the process to infer the phylogenetic relationship between them.

The hyb-seq protocol allows for lower quality of data, which means that dried plants from herbarium can and were be used. The genetic data was obtained from another company after sending them the basic plant material.

The nickel and metallic genes are from the web page https://www.arabidopsis.org/tools/bulk/se The used dataset was AGI transcripts and AGI coding sequences. It was searched against sequences for all gene models/splice forms.

4.2 Selection of probes

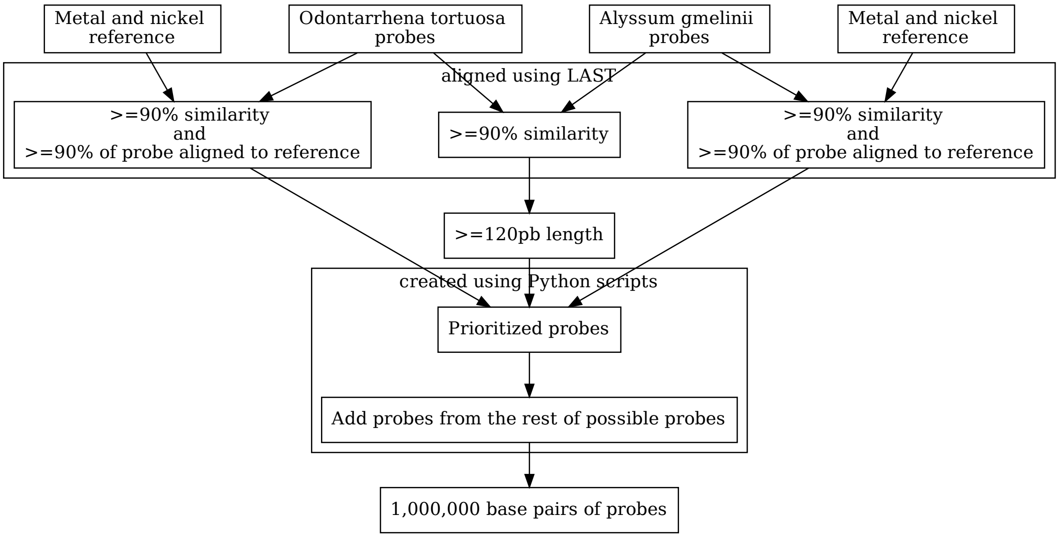
When making probes, we usually want to get exact number of bases, because the created probes are later passed to another company, such as MYcroarray (USA) [6] that will create the actual probes from them. These companies usually charge for processing of a precise number of bases. To save money, it is best to fill this number to the brim. Therefore, when we have several sequences available, we want to select such sequences, that the sum of their lenghts makes the highest number possible, but does not exceed the given limit, let us say $1,000,000$ of base pairs.

Aside from this, we often have other requirements on the selected sequences, usually based on biological evidence. There might be sequences we do not want in our selection, or even sequences we might want to add.

This selection is usually done by hand or a series of programmes. It would be beneficial, if this part too, was automated.

We want to create a script, that would work with the second part of sondovac and would be able to select the best possible sequences to fill up the limit. This script would take into account any sequences we want to keep, or those we do not want in the result.

We need specific genes - responsible for nickel and metal binding. If these genes are present among our sequences, they have to be in the result. We will take the largest possible output file from sondovac_part_b: the one with the lowest minimum total


Metal and nickel reference
Odontarrhena tortuosa probes
Alyssum gmelinii probes
Metal and nickel reference
aligned using LAST
>=90% similarity and >=90% of probe aligned to reference
>=90% similarity
>=90% similarity and >=90% of probe aligned to reference
>=120pb length
created using Python scripts
Prioritized probes
Add probes from the rest of possible probes
1,000,000 base pairs of probes

Obr. 4.1: Flowchart illustrating the workflow of picking the final probes from all possible probes.

locus length. From this file, we will select the required genes. Any additional probes will be chosen by a combination of greedy algorithm and a knapsack.

4.3 Process overview

After several tries, it was estabilished that we will make probes from two sets of datas to cover broader spectrum of plants they can be used for. We ran the Sondovač script for each set and got two resulting sets of possible probes. From these possible probes, we proceeded to pick those that fit the criteria the best. Consecutively, the probes have to make up to $1,000,000$ base pairs, due to restrictions of the probe-making company.

To pick suitable probes, we matched each data set to nickel and metal reference and chose those that aligned with them. We matched the data sets against each other and picked probes by similarity. We chose only one of the probes that matched. The other one wasn't included in the final probes to avoid duplicates. Finally, the rest was picked so they would make up the $1,000,000$ bases to make the most of the space.

4.4 Using the pipeline

The possible probes were produced by running the script Sondovač on two different data sets. We then picked suitable probes from the resulting probes.

We set the minimal total locus length on 360 for both datasets - the least possible minimal total locus length - so we could get the largest amount of possible probes to pick from.

4.5 Probe picker

Probe picker is a script that we coded to help with picking the final probes. It is coded in python. The probe picker requires a list of all possible probes to pick from as an input. It can be also given a list of probes we definitely want in the result - such as the ones that aligned with the nickel or metallic genes or intersection of the two genomes - and a list of probes we do not want in the result - such as duplicates or sequences that are too similar to each other, for example the second of the pair of aligned genes from the two genomes we had.

Aside from lists of probes, the Probe Picker requires a target number of bases to be set and a threshold after which the program should change approaches. After reaching this threshold, the program will start picking the probes so they make up to the $1,000,000$ bases instead of taking probes in order.

After the input files are obtained, the Probe Picker will match a sequence's name to sequence using dictionary data type and the sequence's length to its name using an array.

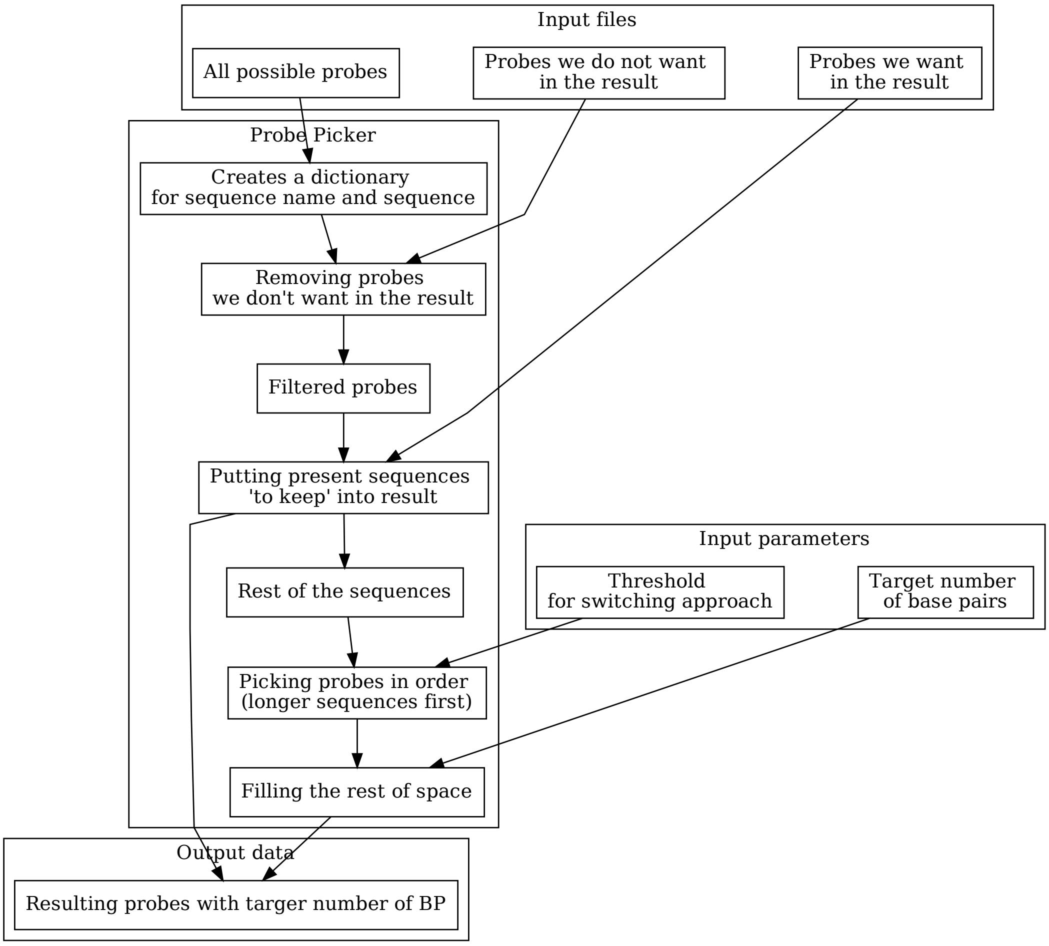
Then, the script puts sequences to keep and sequences to remove into a dictionary, creating a set for each data. Finally, the actual selection ensues. First, the probe is picked if it is not present in the sequences to remove. If it's also in the sequences to keep, it is automatically put into the output. If the sequence isn't amongst the sequences to remove and it isn't amongst the sequences to keep either, it's saved for later processing; for when we are picking sequences in order or picking them so they make up to the target number of bases.

After picking the sequences that are present in the list of sequencest to keep, we start filling the rest of the space by the remaining sequences. We sort the sequences in descending order and pick the largest ones until we hit the threshold. The larger the sequence, the higher the possibility that it's relevant when it matches with something.

After reaching the threshold, we switch to using dynamic programming to make up to the target number of bases with best possible accuracy while not exceeding this number.

4.5.1 Knapsack problem, subset sum problem and dynamic programming

The knapsack problem is a well known problem in combinatorial optimalisation, complexity theory and cryptography. The problem goes as follows: If we are given a knapsack with a weight limit and a set of items where each has a value and weight, what


Input files
All possible probes
Probes we do not want in the result
Probes we want in the result
Probe Picker
Creates a dictionary for sequence name and sequence
Removing probes we don't want in the result
Filtered probes
Putting present sequences 'to keep' into result
Rest of the sequences
Picking probes in order (longer sequences first)
Filling the rest of space
Input parameters
Threshold for switching approach
Target number of base pairs
Output data
Resulting probes with targer number of BP

Obr. 4.2: Flowchart illustrating the workflow of the script "Probe Picker".

is the biggest possible value we can pack into the knapsack while the total weight of items is less or equal to the knapsack's limit.

The subset sum problem is a similar problem, that can be considered a special case of the knapsack problem. The subset sum problem goes like this: If we are given a set or multiset of integers, can we find a non-empty subset of these integers, whose sum is zero? An equivalent problem, closer to what we need, is that we must meet a certain sum instead of a sum of zero.

The Probe Picker presents us with a problem, where we have n integers to choose from and a limit l that we mustn't overstep. This is a variation of the subset sum problem, where length of a sequence, or rather, the number of bases it consists of, can be considered an integer in the initial multiset. As a result, we want a list of sequences that make the most of the possible space; we want to fill the "knapsackäs precisely as possible without overstepping the limit.

The original version of the subset sum is NP complete, which means that we can easily (in polynomial time) confirm, that the solution is valid, but it may be difficult to determine whether a solutions exists at all.

The problem we have is an optiomalisation variation of the subset sum problem. Since subset sum problem is NP complete, the problem we have is NP hard. This means...

However, there is a quicker pseudopolynomial solution for our problem. We can go with an approach that uses dynamic programming. Dynamic programming is about breaking a complex problem into simpler subproblems and solving each of the subproblems only once. From these partial results we can build the final answer. Our problem is finding a subset of sequences whose sum of lengths is the closest possible number to $1,000,000$ and as an input, we have a number of bases (remainging bases we didn't fill up by the greedy approach) we need to fill with probes and a list of probes and their lengths. The definition of a subproblem we are facing goes as follows: For a given sum s and the first k sequences, can the sum be achieved by adding up a subset of these sequences's lengths? We can solve this subproblem by using a result from another subproblem that we have already solved. Let's say we are looking at a specific sequence k with length l and a specific sum s. We want to decide whether we might want the sequence k in our result or not.

If we want it, then how can we achieve the sum s while also using the sequence k? We will take a look at another sum, let's call it $s_{previous}$. This sum $s_{previous}$ is equal to $s - l$. Since lengths of sequences are positive integers, $s_{previous}$ will be smaller than s and therefore already solved (or we can recursively go take a look at the $s_{previous}$ and solve the subproblem for this sum and a sequence $k - 1$.

In the case we don't want the sequence, we will exclude the sequence k and try to solve the subproblem for the same sum s and a sequence $k - 1$.

I WILL ADD MORE LATER.

4.5.2 Picking the genes to keep and remove

We decided to keep probes, that align with nickel and metal genes and also reads that are present – or at least have certain similarity – in the intersection of both Odontarrhena tortuosa and Alyssum gmelinii genome. Naturally, we also want to create a list of probes to remove, where we would put the probes that are too similar to each other, in this case the second of the aligned reads from the intersection of the genomes.

First step was to align probes with metal and nickel reference. We used LAST – a software for finding similar regions between sequences and aligning them. [5] We used the basic settings of LAST and the commands "lastdb"to create a database from reference metal and nickel sequences. After that, we used the command "lastal"to align the probes to the reference.

Both the odontarrhena tortuosa probes and the alyssum gmelinii probes wer were aligned to both metal and nickel reference, resulting in four different alignments.

After this, we aligned the odontarrhena tortuosa probes to the alyssum gmelinii probes, as we wanted an intersection of these probes.

These lists are then filtered using a Python script to pick those, that meet the requirements. In the case of the probes aligned with metal and nickel reference, it was required for at least 90% of the probe to fit into the reference sequence and this part had to have at least 90% similarity. The Python script determined the similarity based on ration of achieved score and the maximal possible score. If at least 90% of the probe fit into the reference was determined on ration of the alignment lenght and probe length.

Another Python script was created to filter the aligned Odontarrhena tortuosa and Alyssum gmelinii genome probes. In this case, the requirements were that the similarity had to be at least 90% and the length of the probe had to be at least 120 bp. Later, the requirement on similarity was changed to $\geq 80\%$ due to low number of aligned probes.

We coded additional script that creates the list of probes we want to have in the result and those we want to remove from these aligned sequences. This script takes the list of names of aligned probes that meet the requirements – the output from the previous script – and a list of probes that aligned to the metal and nickel reference (or probes to probes in case of alignment of Odontarrhena tortuosa and Alyssum gmelinii probes) and creates a list of probes to keep from this file. It then sorts the list of aligned probes and sorts them based on similarity, so the probes with best score are the ones that are taken first.

The script takes probes from this list in order and for each probe, it puts this probe (both sequence and name) into the final list of probes we want to keep. Then it puts the probe it aligned with or any other similar probes into the list of probes we want to remove. This is done to avoid having duplicates or sequences that are too similar to each other amongst the probes.

After the list of probes to keep and list of probes to remove are created, these files are used as an input in the Probe Picker script that picks the final probes.

Kapitola 5

Results

In this chapter, we will take a look at the resulting data. We will also analyze the created probes and summarize the results of the thesis.

5.1 Partial results

5.2 Final probes

Kapitola 6

Possible Improvements

In this chapter we will suggest possible improvements to the pipeline and go through the various approaches to the problem.

6.0.1 Full automatization

One of the main problems of the Sondovač script is the intermediate part where we have to manually input data from Sondovač part a into Geneious and then use the output in the second part. It is planned to replace Geneious by another command line tool that would enable full automatization of Sondovač.

One of the possible tools to replace Geneious is... WRITE MORE HERE.

6.0.2 Incorporating the script we coded into the Sondovač pipeline

BASH could glue it together, programs would need user input

Literatúra

[1] Cd-hit. http://weizhongli-lab.org/cd-hit/.

[2] Flash. https://sourceforge.net/projects/flashpage/.

[3] Geneious. http://www.geneious.com/.

[4] grab_singleton_clusters.py. https://github.com/listonlab/Hyb-Seq_protocol/tree/master/grab_singleton_clusters.

[5] Last.py. http://last.cbrc.jp/doc/last-tutorial.html.

[6] Mycroarray. http://www.mycroarray.com/.

[7] Samtools. http://www.htslib.org/.

[8] Santosh Anand, Eleonora Mangano, Nadia Barizzone, Roberta Bordoni, Melissa Sorosina, Ferdinando Clarelli, Lucia Corrado, Filippo Martinelli Boneschi, Sandra D'Alfonso, and Gianluca De Bellis. Next generation sequencing of pooled samples: guideline for variants' filtering. *Scientific reports*, 6:33735, 2016.

[9] Alberts B, Johnson A, Lewis J, Raff M, Roberts K, and Walter P. *Molecular Biology of the Cell (6th ed.)*, volume 1 of *1*. Garland p, The address, 6 edition, 7 2014. Archived from the original on 14 July 2014.

[10] Sam Behjati and Patrick S Tarpey. What is next generation sequencing? *Archives of Disease in Childhood-Education and Practice*, 98(6):236–238, 2013.

[11] bowtie. Bowtie2. http://bowtie-bio.sourceforge.net/bowtie2/index.shtml.

[12] Dee R Denver, Amanda MV Brown, Dana K Howe, Amy B Peetz, and Inga A Zasada. Genome skimming: a rapid approach to gaining diverse biological insights into multicellular pathogens. *PLoS pathogens*, 12(8):e1005713, 2016.

[13] Free Software Foundation. Bash. https://www.gnu.org/software/bash/.

[14] W James Kent. Blat—the blast-like alignment tool. *Genome research*, 12(4):656–664, 2002.

[15] Colective of authors. Biological dictionary. `http://www.biology-online.org/dictionary/Phylogenetics`.

[16] Colective of authors. Phylogenetic tree. `http://www.biology-online.org/dictionary/Phylogeny`.

[17] Tao Sang. Utility of low-copy nuclear gene sequences in plant phylogenetics. *Critical Reviews in Biochemistry and Molecular Biology*, 37(3):121–147, 2002.

[18] Michael C Schatz, Jan Witkowski, and W Richard McCombie. Current challenges in de novo plant genome sequencing and assembly. *Genome biology*, 13(4):243, 2012.

[19] Roswitha Smickl. Sondovac. `https://github.com/V-Z/sondovac/`.

[20] Baohua Wang, Yan Zhang, Peipei Wei, Miao Sun, Xiaofei Ma, and Xinyu Zhu. Identification of nuclear low-copy genes and their phylogenetic utility in rosids. *Genome*, 57(10):547–554, 2014.

[21] Kevin Weitemier, Shannon CK Straub, Richard C Cronn, Mark Fishbein, Roswitha Schmickl, Angela McDonnell, and Aaron Liston. Hyb-seq: Combining target enrichment and genome skimming for plant phylogenomics. *Applications in Plant Sciences*, 2(9):1400042, 2014.

[22] Ning Zhang, Liping Zeng, Hongyan Shan, and Hong Ma. Highly conserved low-copy nuclear genes as effective markers for phylogenetic analyses in angiosperms. *New Phytologist*, 195(4):923–937, 2012.