UNIVERSITY OF TARTU

FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Institute of Computer Science

Computer Science Curriculum

Tõnis Tasa

Re-using public RNA-Seq data

Master's Thesis (30 EAP)

Supervisor: Priit Adler, MSc

Tartu 2015

Re-using public RNA-Seq data

Abstract:

Next Generation Sequencing (NGS) methods are rapidly becoming the most popular paradigm for exploring genomic data. RNA-Seq is a NGS method that enables gene expression analyses. Raw sequencing data generated by researchers is actively submitted to public databases as part of the requirements for publishing in academic journals. Raw sequencing data is quite large in size and analysis of each experiment is time consuming. Therefore published raw files are currently not re-used much. Repetitive analysis of uploaded data is also complicated by negligent experiment set-up write-ups and lack of clear standards for the analysis process. Publicly available analysis results have been obtained using a varying set of tools and parameters. There are biases introduced by algorithmic differences of tools which greatly decreases the comparability of results between experiments. This is due because of lack of golden analysis standards.

Comprehensive collections of expression data have to account for computational expenses and time limits. Therefore collection set-up needs an effective pipeline implementation with automatic parameter estimation, a defined subset of tools and a robust handling mechanism to ensure minimal required user input. Aggregating expression data from individual experiments with varying experimental conditions creates many new opportunities for data aggregation and mining. Pattern discovery over larger collections generalises local tendencies. One such analysis sub-field is assessing gene co-expression over a broader set of experiments.

In this thesis, we have designed and implemented a framework for performing large scale analysis of publicly available RNA-Seq experiments. No separate configuration file for analysis is required, instead a pre-built database is employed. User intervention is minimal and the process is self-guiding. All parameters within the analysis process are determined automatically. This enables unsupervised sequential analysis of numerous experiments. Analysed datasets can be used as an input for co-expression analysis tool *MEM* which was developed by BIIT research group and was originally designed for public microarray data. RNA-Seq data adds a new application field for the tool. Other than co-expression analysis with *MEM*, the data can also be used in other downstream analysis applications.

Keywords:

NGS, RNA-Seq, pipeline, data-analysis, bioinformatics, gene expression analysis

Avalike RNA-Seq andmete taaskasutamine

Lühikokkuvõte:

"Järgmise põlvkonna sekveneerimismeetodid"(NGS) on geeniandmete analüüsil kiiresti populaarsust kogumas. RNA-Seq on NGS tehnika, mis võimaldab geenieksprессiooni tasemete hindamist. Eksperimentidest kogutuid andmeid arhiveeritakse jõudsalt avalikesse andmebaasidesse, kuna toorandmete neisse edastamine on üheks eeltingimuseks akadeemilistes ajakirjades avaldamiseks. RNA-Seq toorandmed on mahult üsna suured ja üksikute eksperimentide analüüs üsnagi aeganõudev. Sekveneerimise toorandmeid taaskasutatakse praegu veel üsna vähe. Andmebaasidesse leiduvate andmete taaskasutamisele avaldavad pärssivat mõju ebatäpsed katseplaneerimise kirjeldused ja kindlate standardite puudumine analüüsimeetodites. Tööriistade vahelised algoritmilised eripärad tähendavad erinevatel meetoditel teostatud analüüside vähest võrreldavust. Lihtne kollektsioonide agregeerimine ei tööta, kuna analüüsitud andmed pole võrreldavad. Seega tuleb analüüs kõikide eksperimentide jaoks teostada alates toorandmetest. Iga eksperimendi analüüs on aga üsna aeganõudev ning nõuab kuldsete standardite puudumisel konkreetseid valikuid.

Suuremahuliste analüüsiandmete kollektsiooni nõuab seega efektiivset töövoo implementatsiooni. Toimimise tingimusteks on minimaalne inimsekkumine, fikseeritud tööriistade valik ja robustne eksperimentide käsitsemismetoodika. Väga erinevates tingimustes teostatud eksperimentide ekspressiooniandmete agregeerimine loob võimaluse andmekaeve meetodite rakendamiseks. Lokaalselt ilmnevad mustrid võivad taustsüsteemis osutuda signaaliks. Üheks analüüsivallaks, mis selliseid mustreid uurib on koekspressioonianalüüs.

Selles magistritöös arendasime ja implementeerisime raamistiku suuremahuliseks avalike RNA-Seq andmete analüüsiks. Analüüs ei vaja eksperimentide analüüsimisele eelnevalt konfiguratsioonifaili vaid toetub ühekordselt konstrueeritud andmebaasile. Kasutaja poolne sekkumine on minimaalne, kõik parameetrid määratakse andmetest lähtuvalt. See võimaldab järjestikulist analüüsi üle arvukate eksperimentide. Loodavat RNA-Seq ekspressiooniandmete kollektsiooni kasutatakse sisendina BIIT töörühma poolt arendatud koekspressiooni uurimise tööriistas - *MEM*. Algselt oli see ehitatud üksnes mikrokiip andmetelt sondide koekspressiooni hindamiseks, kuid RNA-Seq ekspressiooniandmed laiendavad selle rakendusampluaad.

Võtmesõnad:

NGS, RNA-Seq, töövoog, andmeanalüüs, bioinformaatika, geenieksprессiooni analüüs

Contents

Introduction

DNA/RNA sequencing determines the ordering of the base level subunits of these organic molecules. Next Generation Sequencing (NGS) platforms perform these operations using massively parallel techniques. RNA-Seq is a NGS technique that profiles the transcriptome. Applications of RNA-Seq data analysis contain gene expression studies, phenotyping, splice- and variant detection. A lot of RNA-Seq data is uploaded to public repositories such as ArrayExpress because of requirements to publishing the results in academic journals [1]. The amount of experiments conducted is continuously growing due to decrease in cost, accessibility and development of analysis techniques. However, data re-usage is still difficult because of computational requirements for processing and ambiguities in experimental set-up descriptions. This means under-usage of information [2].

This master's thesis gives an overview of the RNA-Seq with special emphasis on data analysis. RNA-Seq data processing work-flow consists of independent discrete steps. Accurate processing of the data requires preliminary information about this and of its library preparation methods. Misinterpretations of analysis are expressed in skewed results.

Difficulties are exacerbated in case of processing multiple experiments. Large scale data processing can only be effective when handled automatically via a pipeline due to costs of active human input. Extensive collections of RNA-Seq data are still infrequent with notable exceptions such as ExpressionAtlas [3]. An efficient pipeline combines tools that integrate well with each other by compatible formats and is optimised for performance and quality. The overview section of this thesis gives special consideration to all data analysis processes and individual tools selected for the work-flow. We also intend to describe other existing pipeline solutions and highlight opportunities and complexities in making public RNA-Seq data more usable.

The practical aim of the thesis is an implementation of a pipeline that automatically processes and handles publicly available RNA-Seq experiments. Tool selection looks for balance between speed and performance quality [4]. Final analysis output is a NetCDF formatted expression matrix combined with metadata from the experiment. Management of analysis is handled through a pre-constructed database that guides the operations. Collections are separated by species. Experiments are analysed in order of increasing raw data file sizes for each such collection. The design of the database containing experimental data and the data processing pipeline are also described in detail.

A separate section in the thesis is dedicated on evaluating automatic library type selection methods. This is an important parameter in quantification step that can not be directly inferred from metadata nor from data that is generated in analysis steps preceding

its usage. We also perform a runtime analysis and profile the implemented pipeline's actual performance based on collected logs. Also, a simulation study for predicting the future performance of the pipeline is conducted.

Final expression matrices are used within an existing framework initially developed for microarray data - *MEM*. It is a multi-experiment co-expression discovery tool that was developed by BIIT group in University of Tartu [5]. The aim is to use expression data from different experimental conditions to investigate and quantify co-expression on a gene level.

The pipeline was implemented in *Python* programming language integrating a number of special purpose bioinformatics tools. The flow charts were designed using an online chart maker *Draw.io* [6]. All other visualizations and underlying simulations were performed using *R* software [7].

1 Framework

1.1 Overview

1.1.1 Overview of sequencing methods

DNA/RNA sequencing is the process of ordering the atomic subunits of named organic molecules. First major development in the field was the introduction of Sanger sequencing method in the mid-seventies. It is an early-generation DNA sequencing method that uses DNA polymerase enzyme for forming new copies of DNA. DNA is double-stranded, so a complementary strand is synthesized based on a given one using special enzymes. For two and a half decades the algorithm was the most popular method of sequencing and culminated in laying out the structure of full human genome by 2003. Sequencing was performed one base at a time at a very slow rate. This method is still employed for sequencing either very short regions or genomic regions containing many repeated sub-sequences. Such scenarios are still not well handled by automatic and parallelised algorithms.

Subsections of DNA/RNA molecules are known as genomic fragments. A sequenced fragment is called a read. Next Generation Sequencing (NGS) methods perform massively parallel sequencing of millions of fragments which are combined in collections grouped by biological samples. This approach is also known as high-throughput sequencing. NGS technologies can be used for outputs from a variety of sample preparation methods. Sample or library preparation methodologies are selected by genetic material extraction preferences. Material of interest can be RNA, DNA, specific DNA or RNA regions bound by proteins or alike. Therefore sample preparation determines which genomic property is characterised by sequencing.

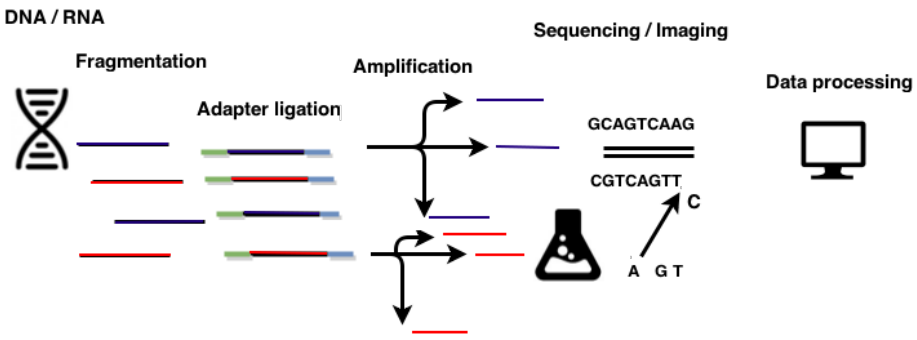

Figure 1: General workflow of NGS experiments

Next Generation Sequencing methodology can be divided into template preparation, imaging, sequencing and data processing as shown in figure 1 [8]. General workflows of

NGS technologies differ by algorithms that are employed by platform providers. Additionally processing methods for outputted sequenced reads were developed for analysing and biologically interpreting obtained large-scale data. Genetic material extraction is performed in template preparation phase. Adapters are synthesized sequences linking the ends of two subsections of genomic molecules. These are connected on both ends of each fragmented sequence. Polymerase chain reaction (PCR) is a widespread technology for amplifying the fragments [9]. These are brought together with a pool of various bases which form a second strand based on complementarity between strands. In imaging, light signalling fluorescent events are recorded from these connecting events. Since single lighting events can not be detected then the fragments need to be amplified. Detected events are digitally recorded as a sequence. A good detailed overview of specific platforms is given by Metzker [10].

1.1.2 Overview of RNA-Seq

Transcriptome is a collection of all different RNA molecules known as transcripts. RNA-Seq uses high-throughput sequencing for quantifying the levels of transcript expression in a tissue at a given time. Main part of this thesis investigates the data analysis process of such sequenced data.

Based on different transcripts RNA-Seq can additionally be divided to mRNA-Seq, small RNA-Seq and Total RNA-Seq . mRNA-Seq profiles coding and non-coding transcripts characterised by messenger RNA. Total RNA-Seq does the same for all RNA transcripts - mostly rRNA, tRNA and mRNA by proportional abundances. Small RNA-Seq transcript types such as piRNA and miRNA have regulatory purpose and are non-coding. Therefore gene level expression can not be investigated using small RNA-Seq [2].

RNA is originally synthesised from DNA by transcription. It is very unstable and degrades easily. Therefore library preparation for RNA-Seq needs transcripts reverse transcribed to complementary DNA (cDNA). After isolation of RNA, primer sequences from which synthesis begins are hybridized onto RNA templates. Reverse transcriptase enzyme is added which starts the synthesis of the first cDNA strand. The strand of the DNA on which a coding sequence is located is called a sense strand. Therefore the newly synthesised strand of cDNA is complementary to the sense strand and matches the template strand of the original DNA sequence. Before second strand synthesis the RNA transcript is degraded which allows second strand synthesis.

Genes consist of coding areas known as exons and non-coding areas known as introns. RNA transcription excludes intronic areas of genes from its product. Exon usage is variable as well so transcripts from the same gene may appear in a range of different isoforms. RNA-Seq is increasingly popular for performing gene expression analyses.

1.1.3 Gene expression analysis

Gene expression analysis investigates in what capacity are expressed the genomic regions annotated as genes. Measured or computed levels expressed in numeric values are mostly directly comparable only against other quantified values within an experiment and have no direct interpretation outside that context. Gene expression is exhibited as a complex aggregate of the functional and regulatory performance but also in other traits of a specimen. Expression can also be quantified in terms of other annotation groupings besides genes such as transcripts, exons or alike.

DNA microarrays are an older technique for assessing gene expression in standardized experimental conditions. Notably they are constrained by only quantifying the expression levels of probes that are included on a specific microarray. This means no novel region discovery in untested tissues or conditions is possible. It is only able to quantify expression for the genomic sequence that attracts another sequence complementary to itself that it is designed for. This sequence is also known as a hybridization target. In microarrays this target does not allow mismatches or any other alternative forms of the same target sequence.

Other older technologies such as quantitative real-time PCR and northern blotting are very slow and restricted to analysing only a single selected genomic region at a time. Sequencing based expression analysis approaches are increasingly popular but aforementioned technologies still retain a wide user base [11]. In sequencing based applications, more extracted genomic material is positively correlated with the average number of times any base is sequenced. This number is called the sequencing depth. More depth offers added sensitivity for detecting varying expression levels. For many applications such as differential analysis, the gene expression analysis is only a preliminary step. It is an important stepping stone into many other fields which use expression data such as gene fusion detection, homology analysis, transcript and variant detection/calling amongst others. Hypotheses can be constructed on the fly based on analysis results in preceding steps which have prompted its growth into fast developing field of active research [12].

1.2 Workflow of RNA-Seq data analysis

The analysis of RNA-Seq data concludes with output of formatted expression matrix. Figure 2 shows the proceedings of an individual RNA-Seq experiment with elements of data analysis highlighted. Library preparation and sequencing are performed in a laboratory setting whereas quality control of the raw reads, genome alignment, quantification and normalization are all part of the data analysis process. Tools for these steps can in most cases be chosen independently from one another depending on the characteristics

of data and aims of the analysis [13].

Raw reads prepared in the laboratory are filtered and trimmed as needed. Preprocessed raw reads are then aligned on a reference sequence. Next, quantification is performed for estimating levels of aligned reads falling in genomic regions specified on the level of genes, transcripts, exons or alike. Raw expression levels are then normalized for comparability between samples. Next, expression data can used for any other downstream analysis processes. These steps of RNA-Seq data analysis are detailed in the following subsections.

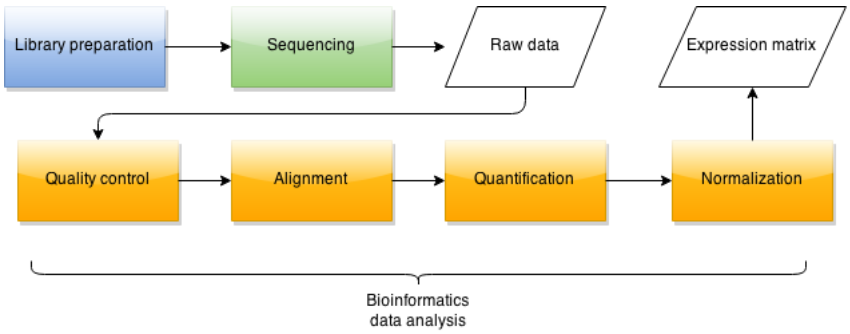

Figure 2: General workflow of RNA-Seq

1.2.1 Quality control

Quality control of RNA-Seq reads is the process of checking for biases that have been induced to reads by library preparation and sequencing. Correction methods aim to improve the quality levels of the following analysis outcome. Common problems emerge in forms of low confidence sequencing quality, over-represented sequences, positional biases, PCR artefacts or unremoved ligation adapters. Many software packages have been developed to identify and quantify these problems. Even though some elements can not be directly corrected, biases can nevertheless be accounted for in the interpretation of final analysis results.

FastQC and *RSeQC* are packages in popular and widespread use for general quality control assessment [14, 15]. Raw reads are often processed using *Trimmomatic* or *Cutadapt* [16, 17].

Successful sequencing process outputs FASTQ files containing millions of individual raw reads. These Reads can either be single or paired end. Paired end reads contain two mates from relative genomic proximity and because of this internal check mechanism, these are considered less prone to misalignments.

FASTQ format is human readable. Each read is described on four lines as shown in table 1.

Odd lines give a unique identifier for each read in a file. The header also contains

Table 1: FASTQ file format

	Example	Meaning
Line 1	@SRR1061357.1 SOUFRE_0103:2:1101:1245:2448/1	Unique identifier
Line 2	TTCGTCGCAGTAGTAAAACCCAAGGTTA	Sequence of bases
Line 3	+	Unique ID repetition
Line 4	cd[ccYcfcceaedbdfbbdgffhhf	Quality values for line 2

information about the sequencing process' characteristics which ensures uniqueness of reads. The second line contains actual nucleotide sequences and the last line gives a qualitative representation for the original sequence. Probability of correct nucleotide assignment is assigned on each base. Quality value interpretations vary depending on the used encodings. Numerical value of quality is retrieved by subtracting offset 33 or 66 from the ASCII value of the character. These values are calculated from probabilities of incorrect assignment using the Phred scale [18].

$$Q = -10 \log_{10} P,$$

where P is the probability of incorrectly calling the base and Q is the quality value for a position of a read.

Trimming reads by quality and discarding those smaller than minimum length is most effective for reducing sequencing failures and directly improves alignment rates. Reads in sequencing are laid out in directional manner. Both forward and reverse strand sequences are represented directionally. The leftmost part of the reads is called 5' and rightmost 3' end. On most sequencing platforms it is common to see lower quality values towards the 3' end of the read. Checking mean quality values by every position over all reads implies general quality of sequencing process. If uniform read sizes are required then these values may guide decision making from which position all reads should be cut.

Popular strategies for trimming include sizing the read to a fixed length or cutting a proportion of all bases in a read or bases with quality lower than an arbitrary cut-off level from either ends. Quality value checking towards the 3' end of the read identifies the first value below the threshold and cuts bases towards 5' from the identified base. In reverse direction the strategy is more lenient. Approaching from 3' end the first quality value exceeding the threshold is set as the boundary and everything back towards 3' direction is cut. A sliding window approach extends this method of identifying arbitrary thresholds over mean values within a range of qualities. This increases insensitivity to sharp local variations.

A different approach used by *Cutadapt* tool uses cumulative summing of differences with a given threshold. Trimming is performed downstream from the worst position.

Thresholds for both minimal and maximal read length are often set and inappropriate reads are discarded from the sample. This requires careful controlling for paired-end reads. Removing one end of a pair brings about the other's removal.

Longer reads are less susceptible to being aligned on multiple locations. Also they are better in detecting exon-intron boundaries or splices. Small reads may align to multiple locations on the genome by chance. This motivates filtering for too small reads and for sequences of low-complexity mainly from repetitive regions containing only one to three nucleotides. Generally, low complexity reads are less likely to be uniquely mapped because of originating from regions of low mappability.

Deviations in some quality measures are often difficult to control for even if the bias is clearly identified. Over-representation of bases C and G or any other low complexity sequences regularly means contamination by foreign sequences. Over-representation of longer sequences can derive from clonal over-amplification or from library preparation kits in the form of adapters. However, unless over-represented sequences are clearly identified by contaminant source or adapters, reads should not be directly filtered based on this property. Expression profiling depends on the quantification of transcript levels and directly removing over-represented sequences may flatten true expression signal. It is also very common to have uneven nucleotide content on the first 12 bases over all reads because of non random primer positioning in commonly used Illumina library prep kits [19].

Removing reads that are duplicated more than a 100 times is a common technique to identify PCR amplification artefacts which are expressed as extreme deviations from the Poisson distributed alignment profile. Any duplication removal is prone to errors because of inaccuracies induced by upstream sequencing errors and SNP variations slightly modifying the alignment.

Unremoved adapters lower alignment precision. Used ligation adapters vary greatly between technologies and between their versions. Therefore comprehensive lists for automatic adapter checking are difficult to construct. This phenomenon is emphasized for publicly available data. Information about types of adapters is not directly accessible nor it is known whether these have been removed before data had been upload. Providers of data such as Illumina only provide information about adapters on request. This causes difficulties even if it is known that adapters do exist. In addition, like other sequences adapters suffer from sequencing errors and mismatches which complicates their removal.

Identification of whole adapters is easier than partial adapters. Therefore their removal has to precede other trimming operations. Removal is facilitated by many tools. Ad-hoc approach looks for matches between over-represented sequences and adapters based on a list of known sequences used as adapters. This feature is provided by some

quality control packages such as *FastQC*. These however do not say anything about their positions relative to the actual read which complicates choosing appropriate trimming strategy.

Quality requirements and cut-off levels for the reads are dependent both on the downstream analysis application and of the characteristics of analysed data. Comprehensive analyses on optimal thresholds are largely missing from literature. Sensitivities to quality variation are specific to used tools which affects approaches to data processing. Generally, a very aggressive trimming policy may yield high alignment percentages with the cost of fewer total available reads and coverage. However, no trimming at all causes low quality reads to be unsuccessfully aligned but no information is inadvertently removed.

Practical standards for quality control in RNA-Seq are also largely missing. Quality control assessment tools offer a wide range of measures for quantifying certain parameters based on raw reads but solid references for consequential action are largely missing. This is in part because the field has only started to develop recently and lacks universal cases [20].

1.2.2 Alignment using *STAR*

Genomic alignment is the process of locating short length nucleotide sequences on species specific genomes. The most common approach matches sequences or reads to an existing reference sequence using some scalable algorithmic technique. In case of a missing reference, it is first constructed from individual reads in a *de novo* manner. Reads are then located on this new *de novo* reference. Large alignment data requires compact storage formats. Many have been developed for storing sequences and their qualitative properties. BAM format is the most popular. Human readable format of this is SAM. Properties of alignment formats have been discussed in the author's bioinformatics research seminar paper linked in appendix 1.

Raw RNA-Seq samples consist of millions 50-100 base long reads. Reference genomes are much longer. Human genome is about 3.5 billion bases and also includes a high number of repeated regions.

Reference genome is built using an aggregated consensus of many ordered sequences. Perfect reference construction however is not feasible. Sequencing errors, single nucleotide polymorphisms (SNPs) and indels all play a role in variability in full genomes. Even though a large proportion of the whole genome is universally shared, local biological variation between individuals is a considerable factor for low mappability of reads. Therefore some naturally occurring misalignment between a read and a reference is anticipated.

The ultimate aim of an alignment tool is to uniquely align and map sequences from non-contiguous regions. Researchers have developed a large number of different alignment

tools. All aim to increase the speed and precision of alignment. Advances are made in indexing schemes for the reference annotation and heuristic approaches for finding the matching regions.

DNA sequences do not contain splices. A major difference between aligners is the ability to detect splicings or exon-intron junctions. However, transcribed RNA sequences do not contain introns and some exons may have been skipped in transcription. This phenomena is known as alternative splicing. A read may consist of multiple separated exons. so aligners suitable for RNA-Seq need to be able to detect exon-intron junctions. That is a much more difficult problem to solve automatically, emphasized by very long spanning intronic areas [21].

Aligners have been subjected to comprehensive comparative studies. There is great variation in speed, resource requirements and accuracy of results [22]. Many aligners with low user numbers are not actively maintained after publication in academic journals. Algorithmic properties running the alignment process cause great variation in tool-specific parametrization requirements. Therefore parameters are often contextually linked to tools and not based on inherent alignment problem characteristics. Practical application of this thesis uses *STAR* aligner by default. Its speed and high accuracy of dealing with spliced data offers a good balance. *TopHat* aligner is implemented as a secondary aligner [23, 24].

The alignment algorithm of *STAR* consists of two discrete steps. A Maximal Mappable Prefix (MMP) corresponds to the longest sequence in read (R_i, R_{MML-1}) where MML is the maximal mappable length and R is the read aligned. Reference genome for alignment is deconstructed into suffix arrays which are indexed. Binary search on indexed suffix arrays identifies the MMP location which can also be multiply aligned. It is assumed that reads can be aligned contiguously. Unmapped fragment of the read is then matched for another MMP. A single read can therefore be constructed in multiple non-overlapping sequences. Binary search in suffix arrays naturally retrieves MMPs. There is no computational add-ons which makes *STAR* very fast. This search is performed on both forward and reverse strands.

Mismatches are handled by gaps between local MMPs. These are handled naturally when MMP matches from different reads are nested together as a whole. MMP construction in aforementioned manner does not require a pre-specified junction database. Minimal intron length specification distinguishes between regions labelled introns and indels. Gaps longer than specified size are considered as introns whereas smaller gaps are labelled as mismatches. MMP alignment has no regard to biological accuracy of underlying annotation. This is advantageous for being able to discover multiple alignments with little overhead.

In the second algorithmic step aligned MMPs are clustered on genomic proximity.

Seeds are first selected by their relative distance to each-other. It is known that reads can not overlap and are ordered sequentially. Each linked group of MMPs allows a single indel per pair of sequences and some number of mismatches. Paired end reads are handled naturally. Inner regions between mates can contain splices.

Combined MMPs are validated with a scoring scheme. The highest scoring combination is selected as the final alignment. The score is a combination of scores from mismatches, insertions, deletions, gaps and matches. *STAR* gives high quality results very fast and with few assumptions such as known splices [25].

1.2.3 Quantification using *HTSeq*

Locations of genes and transcripts generally do not vary over individuals of same species. Pre-existing transcriptome is grouped by genes, transcripts or by other annotation groups. Quantification of expression levels is generally performed within these pre-set regions, unless existence of new transcripts or genes under some experimental condition is hypothesized. mRNA transcripts mostly characterise coding or non-coding genes. RNA-Seq reads are expected to produce more regions in genomic areas where expression during library preparation was higher. Therefore quantifying levels of reads found within gene level annotation regions characterises how genes are expressed. Levels of aligned reads falling in these locations can be described by some aggregative function, simplest of these is counting.

A common method for estimating expression is looking at raw counts. Estimating gene expression directly from counts is however not completely trivial. Interpretations need to be made carefully. Expression estimation on transcript level is especially ambiguous due to partially overlapping and differing exon usage. Not all regions are equally mappable because of repeated areas, sequencing- or library preparation errors or for faulty reference annotations in less described species.

Genes with multiple isoforms also may be under-represented in terms of expression. More advanced modelling based approaches of expression estimation that are more effective in combining expression levels from multiple isoforms have been developed to overcome this drawback.

In the practical application of this thesis a raw count method is used because of its speed and since only gene level expressions are estimated. This significantly reduces the downsides of count based methods. Also using gene level annotations derived from specific transcriptome guarantees uniform genomic structure over all samples. This property is integral for comparability [20].

Quantification can be performed in robust conditions of reference data availability. Availability of both the reference genome and transcriptome enables identification of new

transcripts in addition to existing transcriptomic sequences. Missing genome restricts expression estimation to the domain of known annotation regions from known transcriptome. However, in case of no trancsriptome and genome a whole new transcriptome needs to be constructed using a *de novo* assembler.

HTSeq is a framework written in Python which aims to ease writing custom analysis solutions for analysing high-throughput sequencing data [26]. *HTSeq* also includes stand-alone scripts enabling comparison of aligned reads to transcriptomic regions and outputting counts with chosen annotation level precision. Such quantification is used for expression matrix building and follow-up analyses. Therefore we only count reads that do not overlap multiple genes and are uniquely aligned. Otherwise genes in close proximity to each other may benefit from double counting. Compared to true levels this means over-expression. In worst case, differential expression analysis may be skewed in favour of such regions.

Counting in *HTSeq* is performed for a read if it uniquely matches any portion of an annotated gene and does not overlap any others. Construction of raw counts in union mode is given by algorithm 1,

Quantification process needs information about library type strandedness. This property specifies whether reads originate from sense, anti-sense or from both of the two strands relative to the coding strand of the initial transcribed sequence. This property is determined by kits used in laboratory sample preparation. For public data this information is not directly available. Therefore analytic methods for determining library type are required. A case study for automatic library type assessment based on raw counts is performed in section 3.

Different samples within an experiment need to be normalized for different library sizes and transcript/gene lengths. Normalized expressions have the added quality of comparable values between samples and removed technical artefacts. Further downstream analysis methods such as differential expression analysis take advantage of this property.

Trimmed Mean of M-values (TMM) is a normalization method shown to perform well on raw-counts in comparison with other used expression normalization methods [27]. The main motivation of the method is biologically relevant correction for differing library sizes. A factor that is often left under-corrected is the varying proportional distribution of transcript abundances in different samples. This variability affects the differential expression outcome. Simple proportional scaling often disregards underlying biological reasons and just directly transforms data. This may drive under- or oversampling. TMM intends to correct for this phenomena.

TMM also considers the RNA population of origin for the genomic sample. Normalization factors are derived from log-fold changes between experimental conditions. One

Algorithm 1: Counting unique genomic IDs for aligned reads

Input: aligned_reads.SAM, transcriptome file (gtf)

Output: Alignment counts by genomic IDs

```
Initialize counts
Transcriptome ← Transcriptome.gtf
for all alignments ∈ aligned_reads.SAM do
  if aligned_reads.SAM == paired_end then
    Find all left_pair.gene_features ∈ transcriptome for left_pair.alignment
    Initialise counter for gene_id
    for all (gene_interval, gene_id) ∈ left_pair.gene_features do
      gene_ids ← (gene_el ∪ gene_ids)
    end for

    Find all right_pair.gene_features ∈ transcriptome for right_pair.alignment
    for all (gene_interval, gene_el) ∈ right_pair.gene_features do
      gene_id ← (gene_ids ∪ gene_el)
    end for
  else {aligned_reads.SAM is single_end }
    Find all single_end.gene_features ∈ transcriptome for single_end.alignment
    Initialise counter for gene_ids
    for all (gene_interval, gene_id) ∈ single_end.gene_features do
      gene_ids ← (gene_ids ∪ gene_el)
    end for
  end if
  if gene_ids.size ≥ 1 then
    counts.ambiguous+ ← 1
  else if gene_ids.size==0 then
    counts.nofeature+← 1
  else
    counts.gene_ids[0]+ ← 1
  end if
end for
```

of the samples needs to be selected as a reference. For all non-reference samples this one sample is considered as reference and factor value is calculated for each [28].

Normalization factors are calculated as

$$f_k = \frac{\sum_{g \in G} w^r_{gk} M^r_{gk}}{\sum_{g \in G} w^r_{gk}},$$

where the ratio of log-fold change for sample k on reference r is

$$M^r_{gk} = \frac{\log_2(\frac{Y_{gk}}{N_k})}{\log_2(\frac{Y_{gr}}{N_r})},$$

Y_{gk} and Y_{gr} are two different experimental conditions with raw gene counts, N_k and N_r are respective library sizes and w^r_{gk} are weights for gene g on sample k for reference r. Genes g in geneset G are doubly trimmed separately for all samples. Top 30 % of gene-wise log fold changes and 5% of the extremal absolute expression levels between a library and a reference library are excluded from the calculations.

Counts are assumed to be binomially distributed in a library. Weights are calculated as an inverse of counts' asymptotic variances. Logarithmic transformation of binomially distributed variable converges to normal distribution. Such variance can be exactly evaluated using Delta method [29].

Lemma 1. *Let θ_n be a sequence of random variables such that $\sqrt{n}(\theta_n - \theta_0) \xrightarrow{D} N(0, \sigma^2)$ where θ_0 is an asymptotic mean constant of X_n. Let $f : R \to R$ be a differentiable function. Then $\sqrt{n}(f(\theta_n) - f(\theta_0)) \xrightarrow{D} N(0, \sigma^2 * \frac{\partial f(\theta_0)}{\partial \theta})$.*

Delta method can be applied to the binomially distributed and logarithmically transformed counts. Using Delta method we derive the variance of logarithm. We use the independence between proportions.

$$Var(\log(p1/p2)) = Var(\log(p1) - \log(p2)) = Var(\log(p1)) - Var(\log(p2)),$$

where $p_1 = \frac{Y_{rg}}{N_r}$ and $p_2 = \frac{Y_{kg}}{N_k}$. Y_{rg} is a binomially distributed variable of sample k for gene g and N_r is the library size of sample r. Variables in p_2 are analogous but for sample k.

Applying the logarithm as the function $f(\theta) = \log(\theta)$ approximates $Var(\log(p1))$ to $\frac{1-p_1}{p_1 N_r}$ through

$$\sqrt{N_r}(\log(\frac{Y_{rg}}{N_r}) - \log(p_1)) \xrightarrow{D} N(0, \frac{p_1(1-p_1)}{p_1^2}),$$

$$Var(\sqrt{N_r}(\log(\frac{Y_{rg}}{N_r})) = \frac{p_1(1-p_1)}{p_1^2}$$
$$N_r Var(\log(\frac{Y_{rg}}{N_r})) = \frac{p_1(1-p_1)}{p_1^2}$$
$$Var(\log(\frac{Y_{rg}}{N_r})) = \frac{1-p_1}{p_1 N_r}$$

From this result we can formulate the weights $w^r_{gk} = \frac{1-p_1}{p_1 N_r} + \frac{1-p_2}{p_2 N_k} = \frac{(N_{gr}-Y_{gr})/N_{gr}}{Y_{gr}N_{gr}/N_{gr}} + \frac{(N_{gk}-Y_{gk})/N_{gk}}{Y_{gk}N_{gk}/N_{gk}} = \frac{N_{gr}-Y_{gr}}{N_{gr}Y_{gr}} + \frac{N_{gk}-Y_{gk}}{N_{gk}Y_{gk}}$.

These sample specific normalization factors are then used to retrieve final TMM counts. Raw counts for every gene g in all samples k are transformed as

$$\frac{Y_{kg}}{N_k * f_k}$$

1.3 Public RNA-Seq data

The number of experiments conducted using NGS techniques and respective submissions to public data repositories are increasing faster than those of older technologies such as microarrays. A lot of journals are MIAME or MINSEQE compliant [30, 31]. Data submitted to compliant journals needs to be publicly available and have specified at least a minimum set of information enabling re-analysis of generated data. Method choices for data analysis on such experiments depend on the application and used tools that have different algorithmic properties and use cases.

Databases originally designed for storing microarray data are now increasingly used for NGS experiments storage. Official MINSEQE requirements are broadly stated and therefore enforcement is tricky. Number of RNA-Seq experiments submitted to primary archives has been increasing about two fold on a year-on-year basis. Most important centers of storage are ArrayExpress at EBI (European Bioinformatics Institute) [1] and GEO in the United States [32]. European centres provide faster connection rates so for our analysis purposes data needs to be made available through EBI servers.

Conventional RNA-Seq data analysis process is mostly performed in collaboration with the lab that prepared the reads. Standard protocol of minimal re-analysis does not cover all use-cases. Data stored in public repositories is therefore difficult to use as a separate entity. Tools used in the analysis pipeline do need to infer specifications of data for parametrization directly from metadata. Public data re-analysis needs to account that data providers are not available and metadata is of unknown value.

Each sequencing experiment contains at least two files of metadata. An example set of SDRF and IDF files can be downloaded from a link given in appendix 1. Sample and data

format (SDRF) file is a tab separated human readable format containing properties of all individual samples in an experiment. Experimental factors and properties are specified for each condition. ArrayExpress handling standard classify sample properties into factors, characteristics, commentary and identification pieces. Type of SDRF field is noted in the header together with the field value. Some of the compulsory header values present in every SDRF file are source name, technology type and term source references. Aim of these fields is to describe each sample uniquely as required by MINSEQE platform but also to give as much information about the investigated samples as possible. Investigation Description Format (IDF) is a human readable format containing general information about the experiment. It contains information that is consistent for all samples and adds comparability in relation to other experiments. Values for these include experiment descriptions, identifications, database submission identifiers etc.

ArrayExpress experiment submissions include both raw and processed data. Preparing a comparable set of expression data needs fresh re-analysis starting from raw reads. Because of its extensive scale the analysis has to use a reference framework to guide its operations. Information for developing such a structure and distinguishing suitable experiments for analysis can be found in SDRF and IDF metadata files. Inconsistencies in metadata contents and in the availability of raw data mandate only processing a proportion of all available public data. Databases interchangeably store microarray data among NGS experiments. RNA-Seq experiments also need to be distinguished from experiments conducted by other NGS technologies.

Distinguishing RNA-Seq experiments with accessible links and accessions is based on SDRF and IDF files. All metadata files are categorized into sequencing, microarray and hybrid experiments. The aforementioned contains input from both microarrays and NGS. RNA-Seq experiments which profile transcription are filtered from other sequencing assays. Selected experiments are additionally checked for suitable accession codes to European Nucleotide Accession(ENA) database which are parsed for raw data downloading sources. This second ENA metadata page is used for validating available raw data and species-wise selection. Superseries are experiments performed under a different general hypothesis from the original experiment that used the same data. These experiments are excluded from the final eligible set.

Experiments are categorized by species. Compatibility of performed RNA-Sequencing analysis is ensured by using a fixed set of tools, common reference transcriptomes and genomes within a species. A collection of datasets of the same species are grouped together in a platform. Processing within these is performed in order of increasing raw file sizes because downloading is time consuming and analysis is performed on local servers. Larger collections are this way constructed in less time and by using less resources. Con-

struction of a database framework enables analysis in multiple separate processes, status tracking and also maintaining an overview of general progress.

1.3.1 Available RNA-Seq experiments(05.2015)

There are 9553 submissions of NGS experiments metadata on EBI's FTP server. Of these 4435 experiments contain RNA-Seq sequencing data. For improved accession speeds only experiments with FASTQ files made available through ENA were considered. Raw data files are to be analysed in full and bandwidth is a consideration. Superseries experiments matching previously published data were identified in additional 481 cases. ENA accession codes were missing in 183 experiments with suitable data. This means no accessible download links. After removing Small RNA-Seq and Chip-Seq experiments 3645 RNA-Seq experiments were retained in total.

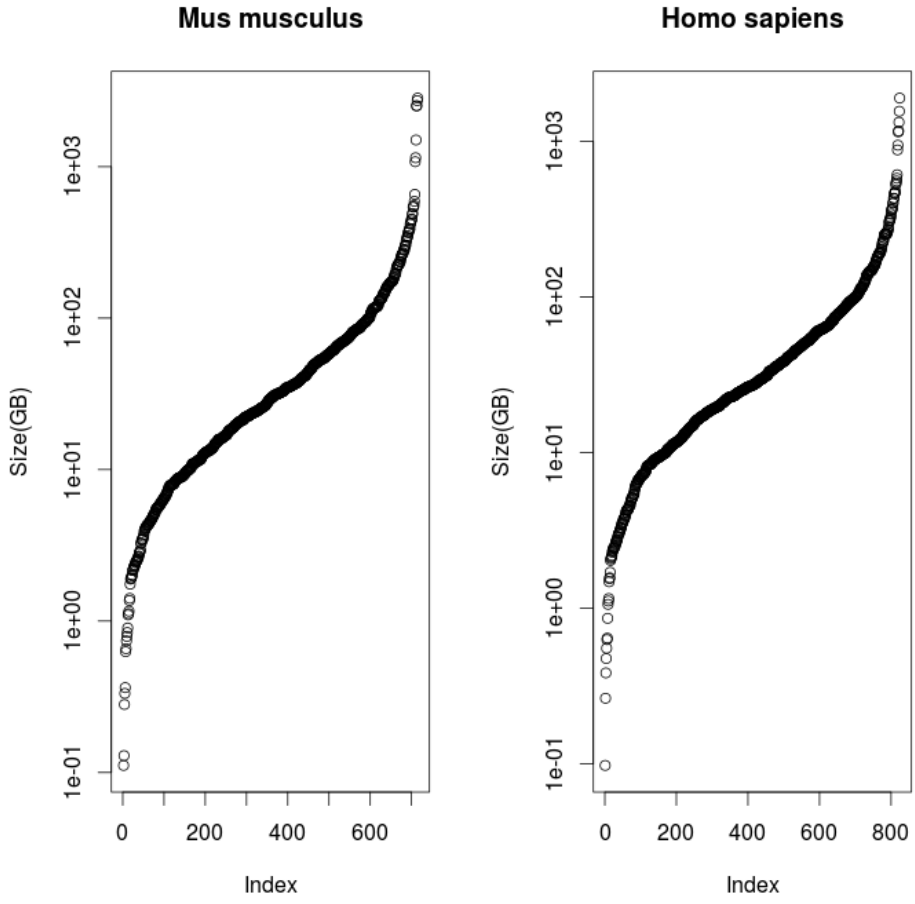

Figure 3: Logarithmically scaled sized of all publicly available RNA-Seq experiments for *Homo sapiens* and *Mus musculus*

Some experiments include samples from multiple species. Since expression data is gathered into collections partitioned by species such data needs to be analysed separately.

A minimum of 5 samples per experiment are necessary for successful normalization. Altogether 2630 experiments contained such number of samples for a minimum of one species. With occasional multiple partitioning, 2847 expression sets were discovered in total as some experiments had enough samples to be analysed for multiple species. This final number of publicly available RNA-Seq experiments suitable for analysis represents the standing by the accocunt of 01.05.2015. These experiments involve 421 different species with highly uneven distribution between platforms. Only 23 species have more than 10 experiments available, seven more than 50 and three more than 100. *Mus musculus* (mouse) and *Homo sapiens* (human) have 824 and 715 experiments containing more than five samples. Among *Homo Sapiens* experiments, raw data of 597 experiments is smaller than 100GB in size, whereas 708 out of 715 are smaller than 1TB. For corresponding mouse data analogous numbers are 700 and 819 out of total 824. Figure 3 demonstrates that experiment size distributions are quite similar. Larger experiments do not add much more information to the collection than smaller ones because they often aggregate signal from many different sources.

1.4 *MEM*

MEM also known as Multi Experiment Matrix is a tool for finding similarly expressed probes over multiple microarray experiments. It was developed by BIIT group in University of Tartu [33]. *MEM* was initially designed to work with microarray data for discovering coexpressed features on the level of probes. Microarrays that carry expression data are built by a company called AffyMetrix. These platforms are mostly species specific and many exist for a single species. Information concerning individual platforms and experiments conducted in the framework are available through ArrayExpress database [1].

MEM uses collections of processed public experiment data. Expression levels are calculated for all experiments and these are aggregated into collections grouped by underlying AffyMetrix platforms. The software aims to find probesets of greatest similarity to one query probeset by profiling expressions over expandable sets of experimental data.

MEM only allows for querying a single probe/gene of interest. Aggregation of information from individual experiments is performed over expression value arrays within an experiment. Distance measure values over all arrays are expressed in a single data point and significance value calculations assume equal weights for all experiments. Distance to the query gene is found for each of the probesets on a platform and these values are ranked by order. Numerically closest expression values are given the smallest ranks. This ranking operation is performed separately for all experiments. After ranking operations each probe has a vector of ranks attached to it whose length equals to the number of

experiments. Ranks are then sorted for each probe. Significance p-values for all probesets are calculated using a BetaMEM method based on Rank Aggregation method with properties of beta-distribution. Smaller p-values express lesser probabilities of random rank values on a specific probeset. Probes with smallest p-values are said to be most closely co-expressed with the query probe [5]. Experiments with more coverage or samples from multiple tissues do not therefore necessarily add more value than smaller ones.

Microarray experiment expression values are only meaningful within the context of a given experiment and in comparison with other samples. Probe effects cause incomparable expression values between experiments. There are also biases in hybridised probe sequences and from array platforms. *MEM* disregards these absolute values and only uses similarities from distance measures such as Pearson correlation. This frees the analysis from many constraints. Normalization and missing probe effects make data comparable over and across genes from different experiments.

Methodology of *MEM* for co-expression discovery requires standardized underlying platforms. Collections used in *MEM* are organized by microarray platforms which by default contain uniform pre-specified probesets. Similarly, such structure needs to be defined for RNA-Seq expression analysis. In RNA-Seq microarray platforms translate to data from experiments grouped by species and analysed using a fixed transcriptome and reference genome. Based on common structure, RNA-Seq expression matrices can be mined for co-expression patterns using existing *MEM* framework.

1.5 ExpressView data representation format

NetCDF is a self-describing binary format suitable for storing multidimensional data. The format stores variables across multiple dimensions maintaining a uniform structure for cross-linking and reference between variables. This is achieved by element-wise correspondence between different elements of variables at same dimension points. Available suite of tools for manipulating, retrieving and storing that data makes it especially popular for scientific data naturally exhibiting dimensional characteristics.

ExpressView data representation format is a NetCDF file format with fixed architecture. Files formatted as such are used as a portable medium meant to work with a variety of in-house tools built by BIIT group. Contents in files are mostly expression matrices with experimental metadata from microarray and next generation sequencing experiments. Each file contains a processed data matrix with expression values laid out in two dimensions representing dimensions of the number of genes and samples. Identification values for all of these are set in a separate variable field with one-to-one correspondence to corresponding numerical values.

Metadata and non-sample specific fields such as the title of investigation, experiment

description and other optional non-track fields also include experiment's pubmed ID, secondary accession, dataset IDs and links to ArrayExpress webpage. Sample specific metadata is extracted from SDRF files. An example of a SDRF file is given in appendix 1. All fields internally represented as factor values, characteristics and comments are extracted from SDRF files and transformed to accompany expression data. Sample specific metadata is asserted to positionally correspond to expression data matrix values. Individual generated data files are handled as separate entities within a set of all ExpressView formatted files on a platform. An example of a dataset in ExpressView data representation format can be downloaded from a link given in appendix 1.

2 RNA-Seq pipeline solutions

All processing pipelines aim to reduce manual workload of the user by combining several sequential steps. A workflow produces final output without intermediate intervention. For RNA-Seq analysis many counterparts have developed their own general purpose pipelines using a selection of available tools and methodologies. In the following we highlight the extent of usage of such solutions. Also limitations that are unilaterally common among work-flows with variable outcomes and application fields.

High heterogeneity between different tools limits the accessibility and usability of pipelines made available for the public. Majority of RNA-Seq pipelines perform sequential analysis of a single experiment. Running each of these often requires a specification of a complex configuration file where parametrizations need to be set separately for all tools part of the analysis process. Scaling the pipeline to multiple experiments makes individual file configuration infeasible. Often a highly hierarchic file layout system is required. Configuration files are time consuming to prepare and require just as many decisions as using individual tools thus not decreasing expected knowledge from the user. This additional file specification removes some of the advantages of employing a pipeline. Choosing suitable parameters for any of the used tools in the process is complicated and mandates acquaintancing with manuals for optimal decisions [35, 36]. Pipelines as well as single-step user-guided analyses need to be able to decrease the number of parametrization decisions for the user. This means that automatic multiple experiment analysis creates the need for on the fly parametrization.

In every pipeline exist intermediate steps for integrating transmission from one process to another. Generally the user has no overview of control on how the data is managed in these steps. This means decreased transparency over result processing [37]. Goals for every pipeline is to increase simplicity of usage and maintain a good overview over processing steps which could attract users with less experience.

RNA-Seq pipelines share only a limited number of similarities beside the name. Differences emerge from the choice of analysis tools, formats of output, methods of maintaining infrastructure, requirements for system prerequisites, modes of running code and most importantly the field of application to which the RNA-Seq pipeline applies. Many tools have completely different requirements for parameters distinguishing them from all others such as the paired end distribution specification in *TopHat* [24]. Each alignment and quantification algorithm has its own customizable properties. However, use cases for all different selections has not always been offered sound biological interpretation. Commonly a lot of trust is placed on default values of the algorithm developers. Most developed pipelines that combine various sets of tools have been developed to solve a very specific problem. General release is often a side-effect not a separate goal. This guides potential users with clear analysis goals to building their own that better suits to available computation resources and infrastructure.

iRAP is a pipeline developed in EBI. Its purpose is to prepare collections of analysed data for ExpressionAtlas project [34, 3]. It contains a large number of tools to be selected for data analysis steps making the process highly customizable. It also displays characteristics similar to many other pipelines and is optimized for solving a problem of a single type which means troubles generalising to other types of systems. Preliminary usage indicated complex installation procedures, a missing mechanism for handling dependencies conflicting with the existing software and signs of very limited parallelization outside a very concrete computational system. Analysing multiple experiments/samples concurrently requires LSF queue management system which is proprietary. Configuration file requirements include specifications of read pairedness, the distribution of gaps between paired end mates, pre-specified quality encoding and strictly uniform read sizes in samples. Preliminary quality control needs to be performed for obtaining require values so many of the pipeline's gains are lost for general users.

There are also a number of other inefficiencies. Configuration files are created for every experiment separately. File and sample naming systems are quite complex and indexing of genome files is performed at every run. It is highly probable that these problems on developer's system locally have been solved with custom scripts and no human intervention is needed in swift processing of ExpressionAtlas data.

However, mentioned drawbacks mean that this solution like many others does not scale outside its very specific domain. By employing existing solutions potential users do not encounter reductions in required user input. Especially if given specifications only apply to a single experiment. For adjusting a solution to a different problem, separate wrapper functions would need to be written which is not suitable to everyone. The number of potential users of RNA-Seq analysis tools/pipelines is quite limited. The time

and monetary effort needed to put into developing a scalable and generalizable automatic pipeline is probably difficult to economically justify. Problem based approaches into building pipelines on local systems are therefore probable to persist in the future as developing one's own very specific method will probably give better results in faster time.

Our aim is to create a collection of analysed expression datasets from maximal number of experiments available through ArrayExpress. Hundreds of experiments analysis requires a different approach on managing the data and processing capacities. This is not offered by any of the existing solutions. Special emphasis is to be set on automatic parameter optimization and real time speed of the analysis. A large number of experiments needs to optimize the performance of analysis on the level of a single experiment and performance, automatic handling and simplicity of usage. It also has to be tuned by the available computation resources in UT servers. Pipeline is supported locally by a database framework that manages data, recollects results and tracks performance. As many others, this pipeline emerges out of specific purpose which is comparative co-expression mining. With this approach this pipeline does not intend to change the paradigm of building pipelines in niche fields. However, it is unique by the nature of its processing task which existing solutions effectively fail to address.

3 Automatic library type inferral

Central part of many RNA-Seq work-flows is expression analysis. Quantifying transcriptome expression levels requires knowledge about library preparation protocol which is independent of the used tools. Wide variety of different proprietary kits and ad-hoc methods exist. The value of this parameter depends on the methods used in sample preparation [19]. These kits can be partitioned into three categories. Produced reads in FASTQ files are either unstranded or stranded. Unstranded reads can originate from both sense and anti-sense strands relative to the coding strand of the original transcript. Strand specific directional reads are only sequenced from one of these strands [2].

Annotations for cDNA sequencing from only one of the strands are defined in different contextual terms. cDNA strand synthesis process is described in further detail in section 1.1.1. *TopHat* and *HTSeq* both use this information in their processes. Quantification based on exclusive first-strand sequencing of cDNA which is also anti-sense to original transcript is noted as *first-strand* in *TopHat* and *reverse* in *HTSeq*. Here reverse notes the over-turned direction compared to the original transcript strandedness. Second strand cDNA synthesis is analogously *second-strand* (*TopHat*) and *yes* (*HTSeq*) meaning that reads originate exclusively from the coding strand of the original transcripts. Unstranded protocols preserve no strand information so the sequenced reads can emerge from synthe-

sis of both strands (*TopHat* - *unstranded*, *HTSeq* - *no*). Inherent drawback of unstranded libraries is the incapability of resolving two overlapping transcripts transcribed on different strands. This means ambiguous matching of many transcripts.

Strand types of public data are automatically assessed. For this type of data the layout and data formats available in databases such as ArrayExpress which are compliant to the MIMISEQ protocol are assumed.

Transcriptome region comparison with read alignment strands and quantifying the observed results is similar to common ad-hoc method of using *BLAT*. Only a handful of reads are aligned on the genome to infer the directionality of the reads through visual inspection. Corresponding implementation of this method in *RSeQC* quality control package samples a portion of total reads and checks the alignments for directionality patterns. This method directly requires the availability of the annotation file, alignment data and additional software (*RSeQC*) or a new implementation of the same technique on the underlying computational system [14].

Second method uses a count based classifier to establish a comparative method. Count based quantifiers such as *HTSeq* and most self-made implementations are very common. The method is based on quantification output from using different strand specific protocols. New features that enable comparison are constructed and classified. This approach requires no new software and the method generalizes well for further downstream analysis paths.

Thirdly, protocol data available for each experiment is the basis data for a classifier using a simple Naive Bayes approach. Its accuracy is assessed over average of leave-one-out classifiers' evaluation results. In total 57 protocols/experiments were used. This method requires neither extra data analysis during processing nor any additional software. Also the accuracy of this classifier exhibits information on how well are researchers describing protocol data. Especially in conditions of inexplicit requirements for library type specifications.

Automatic library type inferring has an important application in pipelines where successful intergrations remove user input requirements to set-up configurations.

3.1 Methods

Aligned reads are compared to the alignment annotation at same chromosomal locations. Strand directionality patterns are reported as ratios. All experiments were available through ENA. Alignment data was generated for 23 experiments totalling 266 samples with true library type information available. Quality control was performed using *FastQC* and *Cutadapt* [15, 16]. Trimmed reads were aligned using *STAR* mapper [25]

3.1.1 Alignment fit to transcriptome

All conducted experiments had true labels pre-established. K-nearest neighbour classifier was constructed on the two generated features indicating the ratio of reads aligned to either sense or anti-sense strand. Classification was performed both by individual samples and also by experiments. Ratios were obtained using *infer_experiment.py* script that is part of *RSeQC* package. This classifier was trained on data due to its suitability for multi-label classification problems and few assumptions on the distribution of the data. Classifier was validated using a 10-fold cross validation over all samples.

3.1.2 Multiple quantification inferral

Quantification results using two different strand settings are also combined for features to infer true library type. Method is built upon *HTSeq*, one of the most common raw-count based quantifiers. However, required features can be extracted from any count based quantification tool. *HTSeq* library type setting is specified as one of the two strand specific parametrization types ("yes"/"reverse"). By specifying one of these all sequenced reads are assumed to originate exclusively from sense or anti-sense strand of the coding region.

Feature generation for both parametrizations requires the number of uniquely aligned reads matching gene or transcript regions and also the number of reads uniquely mapped outside selected annotated regions. It relies on assumption that only a small ratio of overlapping genes on different strands. This is in the region of 5-10% for both human and mouse genomes [38].

The total number of reads aligned to regions annotated by transcriptome using two different strand-specific parametrizations should be similar to the number obtained from quantification using an unstranded library type. Some reads are lost due to ambiguity between strands and in feature construction will be accounted for by scaling. Single descriptive numeric values are obtained from differences of stranded feature values divided by the sum of the same feature counts.

$$\frac{\sum m_{str2} - \sum m_{str1}}{\sum m_{str1} + \sum m_{str2}},$$

where m_{str1} marks the feature counts for the first-strand and m_{str2} for second-strand library type. This simple feature transformation can be used for both accounting for uniquely aligned reads and no feature regions. Transformed features are fitted a K-nearest neighbour classifier with a 10-fold cross-validation.

3.1.3 Protocol classifier

ArrayExpress stores a list of protocols used for each stored experiment among other data. We scraped hand-written *nucleic acid library construction protocol* and *nucleic acid extraction protocol* protocols for 57 experiments with known labels. Based on these we constructed a Naive Bayes classifier with Laplacian smoothing where the probability of a protocol document D_j belonging to library type C_i is

$$P(D_j|C_i) = \frac{P(C_i|D_j)P(C_i)}{P(D_j)}$$

Also

$$P(C_i|D_j) = \prod_{w \in V_j} P(w|C_i) = \prod_{w \in V_j} \frac{n_w + 1}{\sum\limits_{w \in V_j} (n_{(w,i)}) + |V|},$$

where $n_{(w,i)}$ is the number of word w occurrences in class i and and V is the vocabulary of all protocols and V_j is the vocabulary of protocol j. Also $P(C_i) = \frac{n_{D_i}}{n_D}$ where n_{D_i} is number of protocols D with label i. From this assigned class for each protocol would be $\arg\max_{i \in n_i} P(D_j|C_i)$. After classification we performed leave one out cross validation. Classification was carried out in two different cases firstly discriminating only labels between stranded and unstranded library types and secondly also distinguishing the two stranded protocols.

3.2 Results

3.2.1 Transcriptome and alignment correspondence (*RSeQC*)

First and second strand alignments also form definitive clusters of three per individual samples of all included experiments as shown in figure 4. KNN-classifier with 10-fold cross validation was fitted and results were averaged for the multi-label classifier. Results are shown in table 2.

Figure 4 indicates discernible patterns for correspondence between strands of alignments and annotation regions by true library type directionality. Library types are linearly separable on the measured sample.

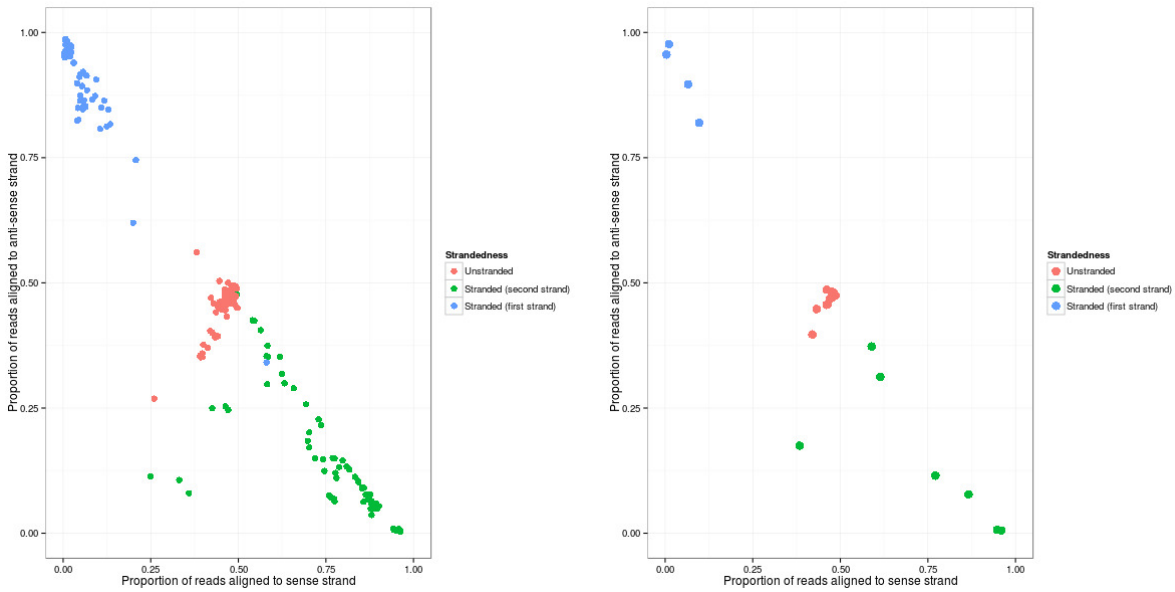

Figure 4: Proportions of reads aligned to sense and anti-sense strands for all individual 266 samples (left) and for all samples averaged by grouping into 23 experiments (right) with known true library strandedness types

Table 2: Recall and precision values for sample based KNN-classifier in library type identification

	Recall	Precision
Unstranded	0.97	0.98
First strand	0.98	1
Second strand	0.98	0.97

The classifier gives near perfect results for all labels on multiple binary comparisons.

3.2.2 Classification of raw count quantification results

The difference between rows matched to annotated regions and reads with no matches in annotated regions of two library types were used as classification features. Figure 5 shows that likewise three definitive clusters are formed with two variables plotted against each-other. For evaluation 10-fold cross validation was used on all samples as in previous section.

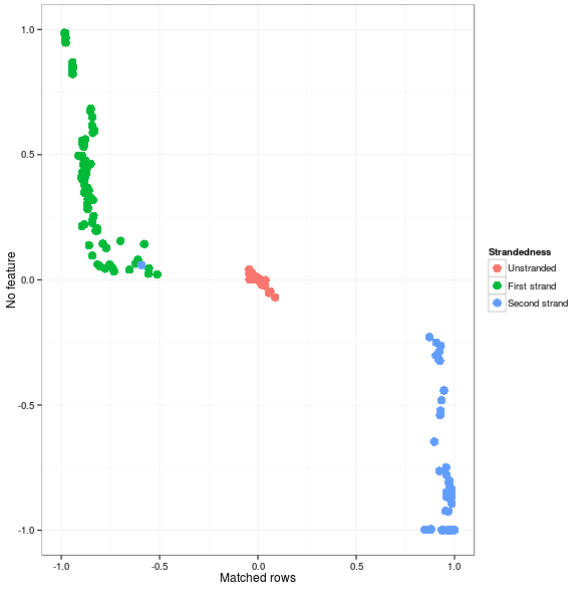

Figure 5: Strand-specific normalized differences of matched and mismatched reads on annotation regions by library types

Table 3 shows that the recall and precision of each library type versus all others are again near perfect. Only a single data-point between strands was misclassified. This corresponds to implications from the visualization on figure 5.

Table 3: Recall and precision of multi-labelled KNN-classification of normalized differences between stranded library types with matched and mismatched reads on annotation data as features

	Recall	Precision
Unstranded	1	1
First strand	0.99	1
Second strand	1	0.99

3.2.3 Protocol classifier

In protocol based library type assessment no sample specific discrimination can be made. A Bayesian classifier with leave-one-out cross validation was trained. Protocol sample size was increased from 23 experiments to 57. Naive Bayes classifier for text classification separated stranded protocols from unstranded protocols with 0.77 accuracy. Table 4 shows that the classifier suggested experiments belonging to stranded library types conservatively misclassifying many such experiments. Experiments classified as stranded were correct in 94% of cases.

Additionally, multi-label classification in separating between the two strand specific

Table 4: Naive Bayes text classifier's recall and precision in separating stranded and unstranded library types

	Recall	Precision
Stranded	0.59	0.94
Unstranded	0.96	0.69

Table 5: Naive Bayes text classifier's recall and precision in distinguishing library types from one another in a multi-labelling problem

	Recall	Precision
Unstranded	0.96	0.69
First strand	0	0
Second strand	0.39	0.64

protocols does not work as well. Total accuracy fell to 0.63 due to complete inability to distinguish between the two strand specific protocols. These results are shown in table 5.

4 Pipeline

4.1 Database development

Only experiments that have been selected as eligible for analysis are processed by the pipeline. This necessitates a construction of systematic database containing such info. The process for this is demonstrated in the current section.

First, raw metadata files for all experiments are scraped from ArrayExpress FTP databases. Retrieved IDF and SDRF metadata files are downloaded if missing and updated when newer versions compared to local file variants are available. This process is tracked in a separated database table and enables fast and effective updates of the system.

Locally downloaded metadata files are the basis for consequential identification of processing eligible RNA-Seq experiments. SDRF files contain data about individual samples of experiments. Example of a SDRF file is in appendix 1. In the following, all flowcharts describe the retrieval or analysis of a single experiment unless looping over experiments is specifically remarked.

As new data is continuously released databases need to be updated to corroborate the actual status of submissions to ArrayExpress database online.

First step in database construction is distinguishing NGS experiments from other experimental settings as shown on figure 6. Separation may be based on metadata fields

only present in a certain assay types or directly inferred from values from within parsed file contents. All experiments available are separated into four broad categories and classified either as array assays(microarrays), hybrid assays, sequencing assays or other experiments which have too little data for accurate separation. Tables for experiments are updated with corresponding decisions.

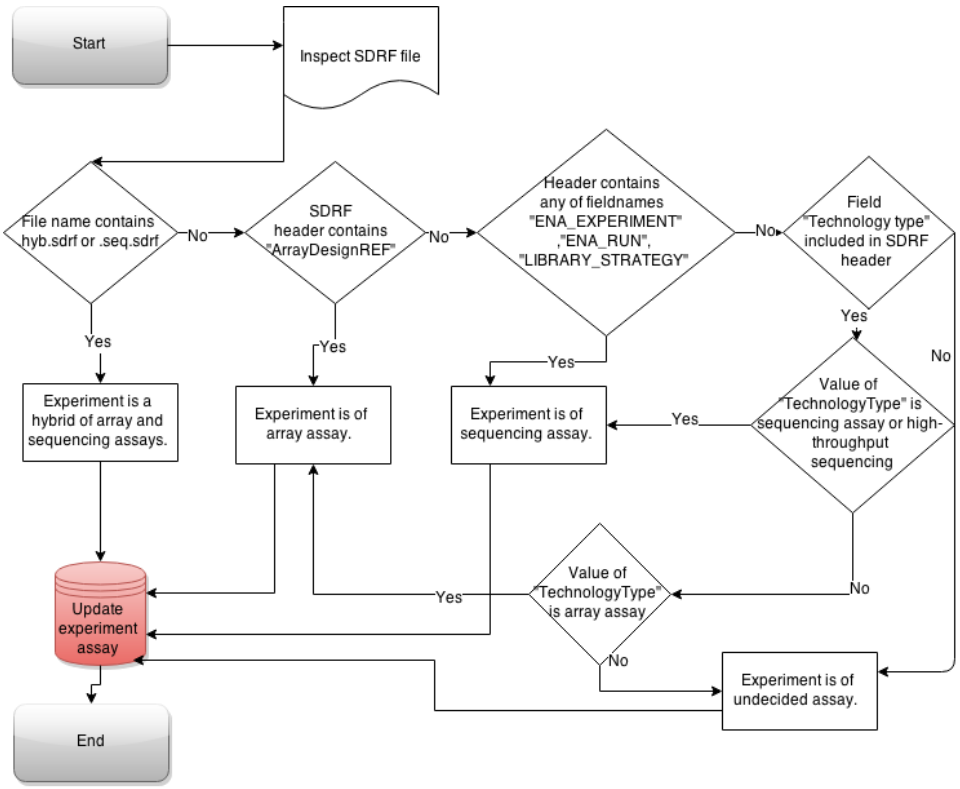

Figure 6: Flowchart of experiment assay classification in public experiment database construction

RNA-Seq is one technique out of many NGS based methodologies. Sequencing experiments therefore also need to be filtered for experiments that were conducted using RNA-Seq library construction techniques. Some NGS methods explore other genomic features besides gene expression. Such information can be inferred from fields specified in IDF metadata describing general overview of the experiment as outlined on figure 7.

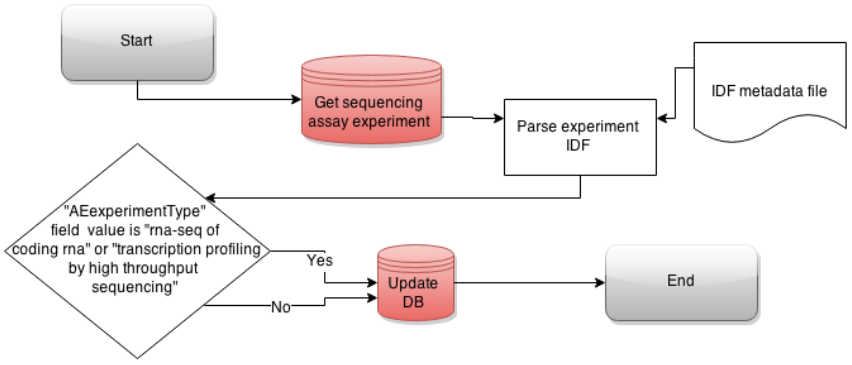

Figure 7: Flowchart of RNA-Seq experiment retrieval amongst NGS experiments in public experiment database construction

Experiments are further filtered by the availability of raw data and some other terms for downloading process as outlined on figure 8. For all individual samples, ENA experiment- and individual run accessions have to exist. This is needed to match metadata with the outputted expression matrices. A valid ENA accession code is also a requirement for all experiments. Links for raw data download are parsed from ENA metadata accessions. Metadata constructions and file specifications published by ENA and SDRF/IDF do not always match so practical solutions need early difference detection mechanisms. Processing superseries experiments would replicate existing results. Duplications need to be eliminated for removing over-representation of some experimental conditions.

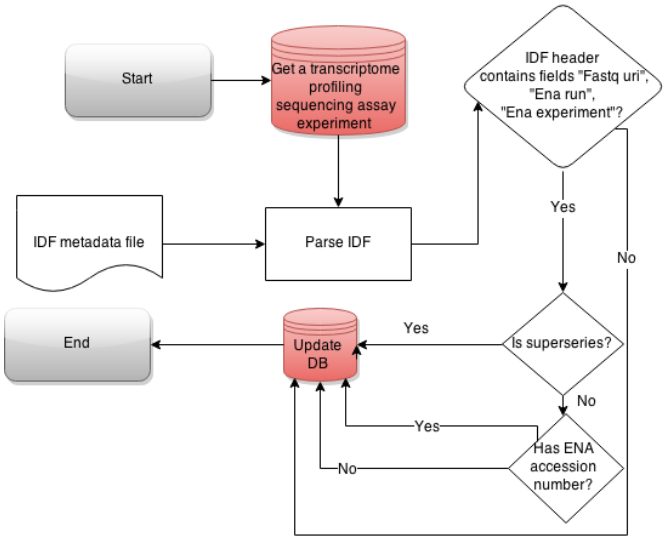

Figure 8: Flowchart of data availability classification for RNA-Seq experiments in public experiment database construction

Analysis process for RNA-Seq experiments is time critical and analysed by species in the order of experiment's raw data sizes. This number is retrieved by counting all

individual samples and summing their size grouped by species in an experiment as shown on figure 9. This completes the content constructions to the database which can be used as a basis for querying, maintaining the automatic analysis and structural composition of the data.

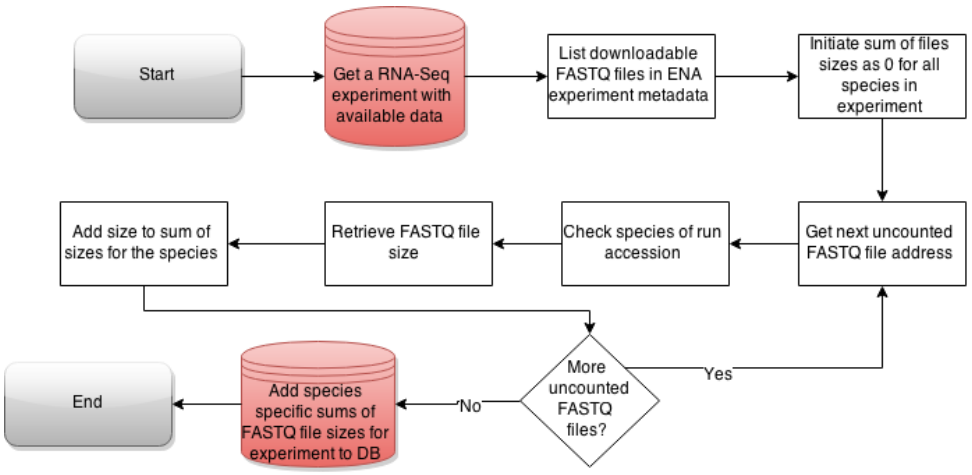

Figure 9: Flowchart of grouping RNA-Seq experiments with available raw data by species and sizes of raw files in public experiment database construction

4.2 Design decisions

Only experiments with more than four samples are included in analysis. This is verified through ENA metadata sheet. Processed results are outputted in ExpressView data representation format. Besides expression data the file also consists of metadata concerning information about the experiment and its samples. It requires both SDRF and IDF meta-data files. First process of the analysis deals with extraction and formatting of this metadata. All preprocessing in outlined in figure 10.

Metadata construction in detail is shown in figure 11. SDRF and IDF files are parsed and transformed through a series of steps. IDF files are parsed for 4 different fields describing the experiment. These are experiment description, investigation title, pubmed ID and secondary accession. The first two fields in the list are compulsory to exist in IDF. Secondary accessions are used for connecting the experiment with the submission to ENA. This accession ID is used as a gateway to ENA metadata page for obtaining the raw data.

In parsing metadata, SDRF and IDF field names and their values making up the body of the file are separated. SDRF value columns of interest are extracted. A single sample may constitute of data from multiple sequencing lanes that is specified individually. Thus different replicates have to be distinguished based on ENA experiment and data from different lanes needs to be merged into a coherent sample. Experimental metadata also

needs to be separated by species as samples are analysed in groupings by organisms.

It is common for SDRF files to miss sample run accessions. However, they invariably exist in ENA metadata. It is possible to transfer these from ENA metadata as these guide data mergers originating from different sequencing lanes and metadata information matching with expression data.

In the final step headers and processed values are merged together. To this, additional tags are added for ExpressView format such as the fileformat tag along with experiment dataset IDs and links to online datasets. This info is created from external inputs. Metadata file is stored separately from numerical data but positional correspondence to field values needs to be asserted. This is required by ExpressView data representation format.

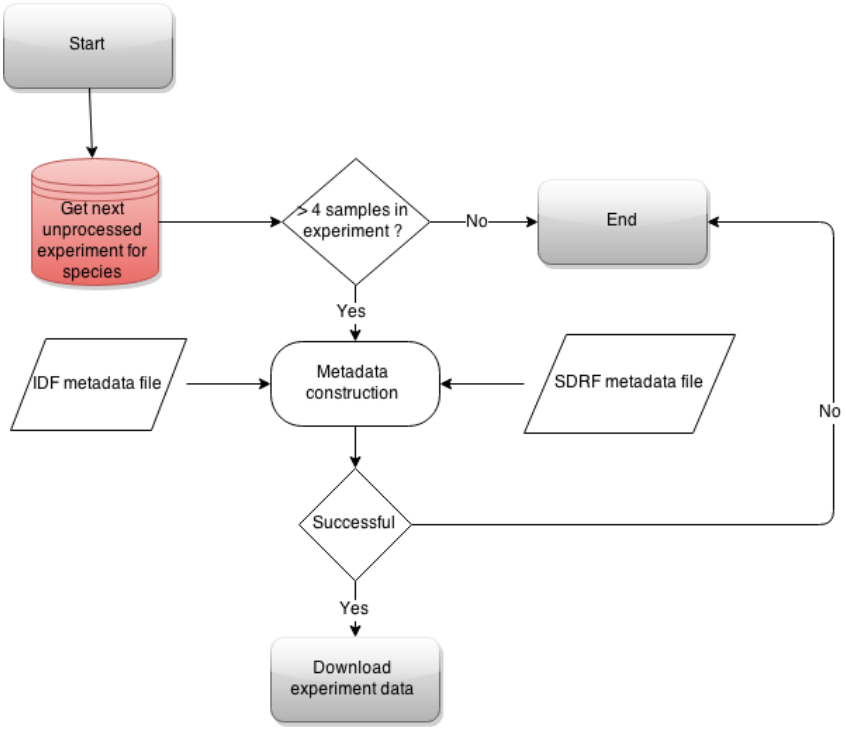

Figure 10: Flowchart of general preprocessing schema for public experiment data analysis

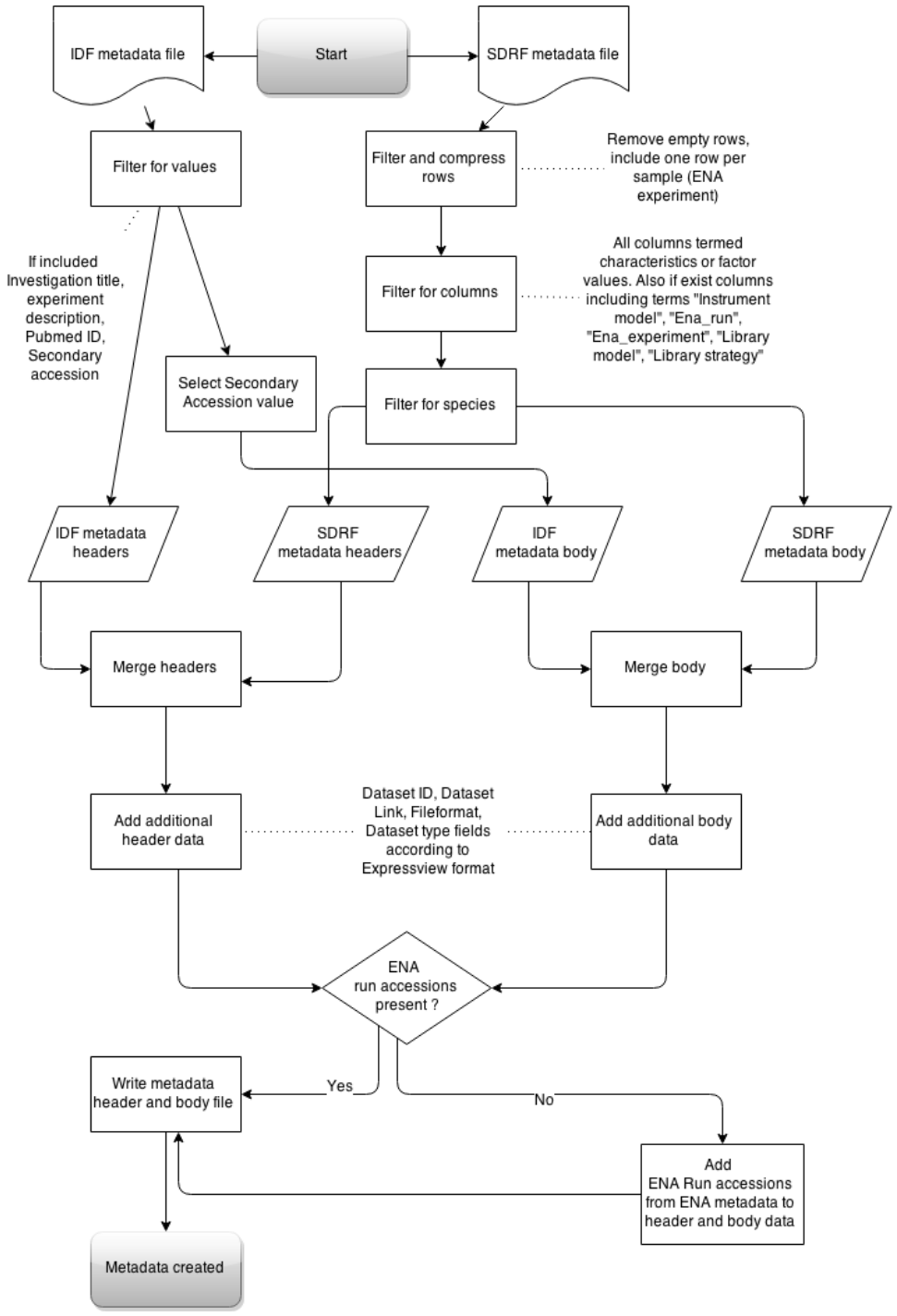

Figure 11: Metadata construction flowchart for public experiment data analysis

Data processing starts after creation of metadata. There is occasional incompatibility between metadata specified in SDRF files and metadata in ENA. Ahead of downloading the samples the pipeline needs to check that ENA experiment accessions match those created in metadata construction step. This is to ensure compatibility of created data with data that is actually downloaded. As illustrated in section 1.3.1, raw file sizes may reach up to hundreds of gigabytes so assertion failure detection in early stage of analysis

saves a lot of time. FASTQ files of sequenced reads are downloaded if metadata fields match. Downloading process flowchart is shown in figure 12.

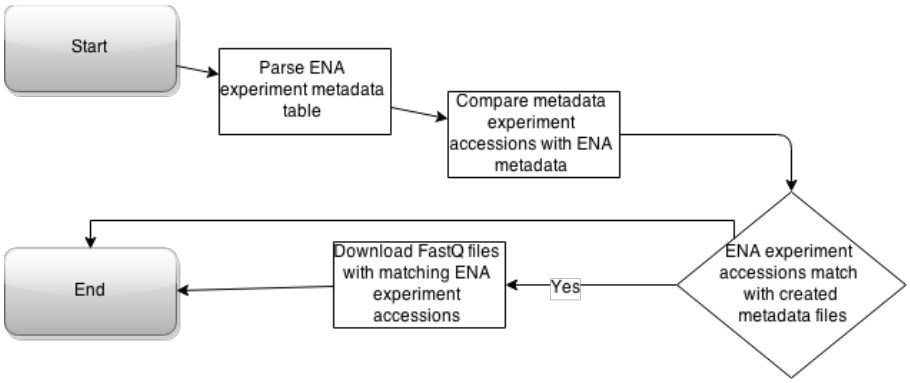

Figure 12: Flowchart of raw data download process for public experiment data analysis

Figure 13 gives an overview of the dependencies in input files and software in order of the analysis pipeline proceedings. Downloaded FASTQ files are first grouped and matched by pairedness. In the alignment step paired-end reads need to be analysed in unison. Quality control uses *FastQC* for general overview of the quality of sequencing and library preparation. *Cutadapt* tool filters and trims the reads based on parsed general quality control results to increase alignment performance. Trimmed reads are aligned using either *STAR* or *TopHat* aligners. These two aligners are procedurally somewhat different so their set-up and parametrizations are called on a per-experiment basis. Experiment analysis is invariably linked to a specific platform. Thus one specific genome in FASTA format which is obtained and indexed before the alignment can be used whenever an experiment for this genome platform is analysed.

Samtools is used for merging samples from different lanes into single alignments of one replicate. Sample identification is performed using ENA experiment accessions for separate FASTQ files. All alignments are then checked for library type strandedness using a script from *RSeQC* quality control package. This method was demonstrated as highly effective in section 3. An arbitrary cut-off for library type decision was set at one directional proportion surpassing the other two-fold. If no difference occurs the sample is called as unstranded. Strand inferring process employs a BED format file constructed from species transcriptome for calling strand-wise alignment matches to transcriptomic sequences. Library type information is used in the quantification step. *HTSeq* is used to count reads uniquely matching genomic regions on strands specified by library type. Count info is gathered for all samples and additionally bound to create a count matrix holding gene-wise information in rows and samples in columns. A single transcriptome is used for extracting annotated regions within a platform. This ensures a uniform structure for the experiments to be gathered into a collection. Normalization within samples

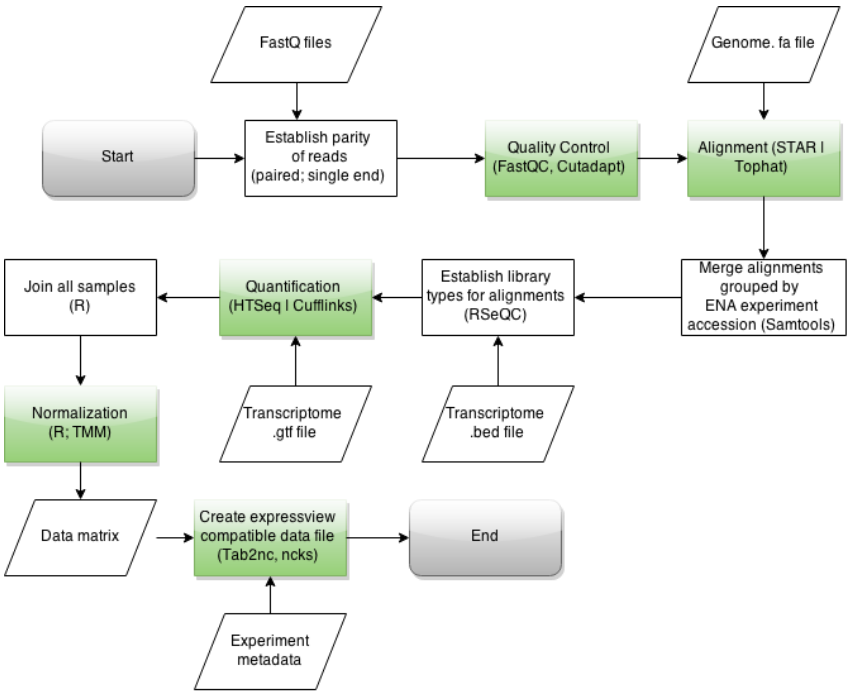

Figure 13: General flowchart of analysis steps, computational dependencies and input files for public RNA-Seq experiments data analysis

by library sizes and gene lengths is performed using TMM. This has an existing implementation as part of the *edgeR* package in R. The resulting normalized data is the final expression matrix which retains a standardized structure. Uniform column structure is not important as distance measures are calculated over gene-wise expression values. In the final step, output file in ExpressView data representation format is created by combining the expression matrix with initially constructed metadata file. Final form is achieved with some additional data manipulation with R and transformation using *tab2nc* and *ncks* tools. These are appropriate tools for handling NetCDF formatted files [39]. These output files are very similar to those obtained when preparing microarray data collections to be used within *MEM*.

Data analysis is handled without manual interference. There is high heterogeneity between experiment data and disagreements about optimal settings. Occasionally experiments of other sequencing types may pass through filters of exclusion. Compared with careful manual and individual processing some discrepancies are bound to happen in large scale data analysis. Automatic processing needs to find a common ground for maximally efficient results. *MEM* has a self-filtering mechanism which enables it to discard data of low quality. Sometimes gene-wise expression levels over samples are not varying more than set threshold because of low coverage or biases from analysis. Such experiments are automatically excluded from ranking and p-value calculations.

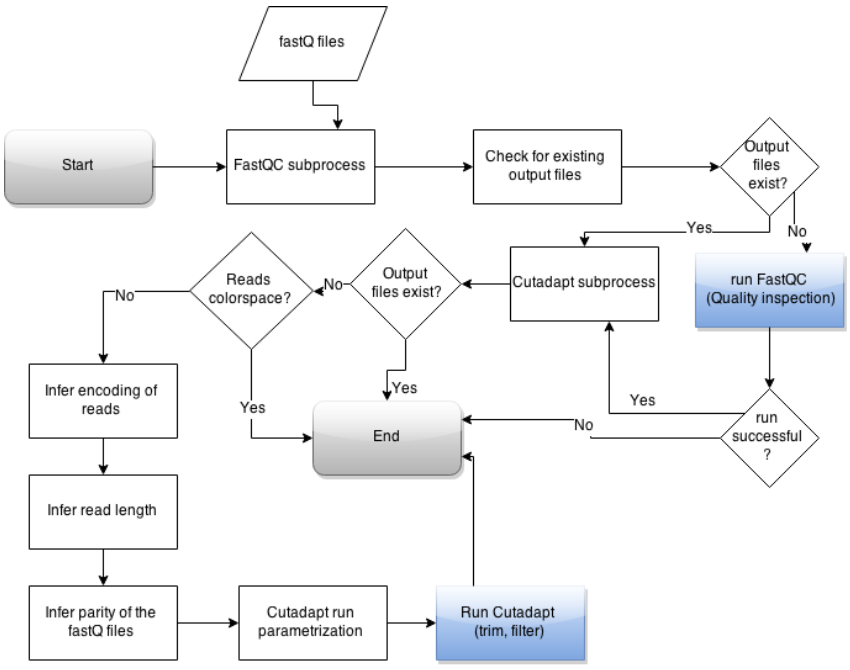

Figure 14: Quality control flowchart of public RNA-Seq experiments data analysis

Each FASTQ file is first quality controlled by running quality control as shown on figure 14. Quality levels of individual read collections are identified using *FastQC*. Outputted reports are used to infer the encoding of read quality levels as well as the range of read lengths within a file. Paired end read mates need to be trimmed and filtered synchronously. Downstream processing expects reads to have the two files' reads matching in sequential order. Color space reads describe nucleotides positioned relative to one another instead of explicitly specifying each base. These are currently not supported by *STAR* alignment tool so processing of such experiments is halted. Both *FastQC* and *Cutadapt* run on 10 simultaneous threads where a thread is dedicated for a single sample.

The minimum quality threshold for trimming is set at 20, additionally 5% of the original read length size from the 3' end is trimmed. The minimum length for a read not to be discarded is at minimum 70% of the original read length or 16 bases. Maximal read length for alignment is set as a fixed size of 240. Encoding of the reads is identified during quality control.

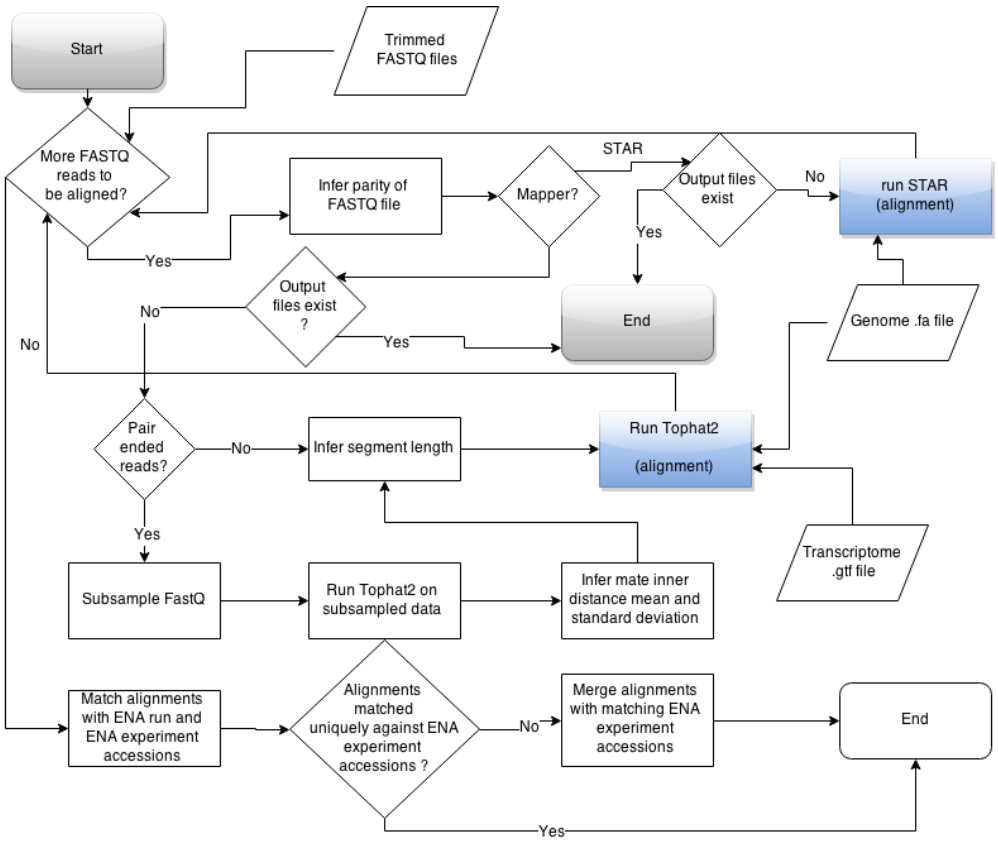

Figure 15: Alignment process flowchart of public RNA-Seq experiments data analysis

Trimmed reads are aligned using *TopHat2* or *STAR*(default). Paired-end files are handled separately from single-end reads. Data sequenced on different lanes may however be aligned in separate steps. Data from different lanes is merged into a single alignment so that the number of total alignment files equals to the number of samples in constructed metadata. *STAR* requires a supplied reference genome. Using this aligner is more straightforward to set-up and about 30 times faster than *TopHat2*.

TopHat2 for paired end reads requires a number of preceding steps to infer some parameters necessary for efficient alignment. To do so, paired end reads need to be down-sampled randomly. Selection of reads are next aligned using standard parameters of *TopHat2*. From resulting alignments the user infers normal distribution parameters for paired-end mate inner distances. *TopHat2* also requires setting segment length which is the size of the fragment that is aligned at a single run. For existing smaller reads this should be set at about half the size of the reads. Aligners use RAM and CPU extensively. In our server configuration we set a maximum of four threads for alignment step. Each sample then runs on five individual threads. Smaller genomes need to be given additional RAM for BAM sorting procedures because by default the memory is limited by the genome size. *STAR* needs to output alignments in BAM format. It keeps only canonically annotated high confidence junctions, also filtering out the novel splices

identified with low confidence.

Quantification tool *HTSeq* brings together most of the results from previous steps to produce expression value estimates. Required elements are the alignment files for all samples, library type information for each sample and a transcriptome file based on which annotation groups are identified for expression analysis. *HTSeq* is run on 15 multiple threads as despite its time usage it is quite low on RAM consumption. Parametrizations are designed to be fast and reliable. We assume reads ordered by name and perform quantification on gene-level. Counting is performed in union mode requiring a read to overlap a single gene for unique count. In every analysis step existing output files are checked for so that in case of possible reruns time critical parts are not repeated.

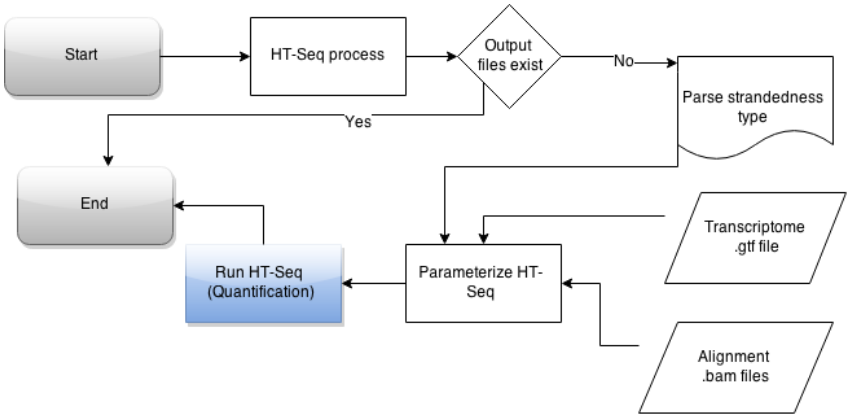

Figure 16: Quantification process flowchart of public RNA-Seq experiments data analysis

Besides the final NetCDF formatted file, we store quality control reports before trimming, alignment and quantification summaries per sample. All processing is logged for debugging and status overview.

4.3 Runtime analysis

A runtime analysis based on 235 full RNA-Seq experiments to test implemented pipeline operations was conducted. Motivation for this is to identify performance issues and to describe and highlight time usage of independent pipeline operations. Calculations were run on an Alligaator.at.mt.ut.ee server [1] and Caiman.at.mt.ut.ee [2] servers. The analysis process is ongoing so all results of analysis are reflecting the current state. Experiments included in runtime analysis were performed over the course of around two and a half months. Analysis success rates vary for different species with around every fifth experiment unsuccessfully terminated before completion with logged error messages. Common

[1] 8 Intel(R) Xeon(R) CPU E7- 2860 @ 2.27GHz, 10 cores, 80 CPU cores in total, 1024 GB RAM
[2] 4 Intel(R) Xeon(R) CPU E7-4870 v2 @ 2.30GHz, 15 cores, 60 CPU cores in total, 1024 GB RAM

reasons for failure include incompatible metadata, broken or incomplete data contents, partial metadata, reads being colorspace etc.

All included experiments had successfully completed in a single runtime. Experimental data concerned three analysis platforms differing by species - mouse (*Mus musculus*), human (*Homo sapiens*) and thale cress (*Arabidopsis thaliana*). These are species that have been most extensively analysed. Purpose of profiling a pipeline performance and time usage is to accurately distribute computational resource usage operations and time management.

A single experimental run is divided into eight time-points at which time usage is measured. The steps are in sequential order: start of analysis (Start), end of initial metadata processing (Metadata), downloading of raw data from the database (Download),quality control (QC), alignment (STAR), library type specification (RSeQC), quantification (HT-Seq) and post-processing (Completed). Figure 17 shows progression through all these steps for all individual experiments in a timeline format. Processing steps invariably consuming the most time are downloading, alignment and quantification. Even though the pipeline also supports *TopHat* all alignments were conducted using *STAR*.

Right side of figure 17 describes the proportion of the whole runtime completed by the end of time measurement. Based on these, estimations could be made for how much runtime is still to be expected for ongoing experiment analysis.

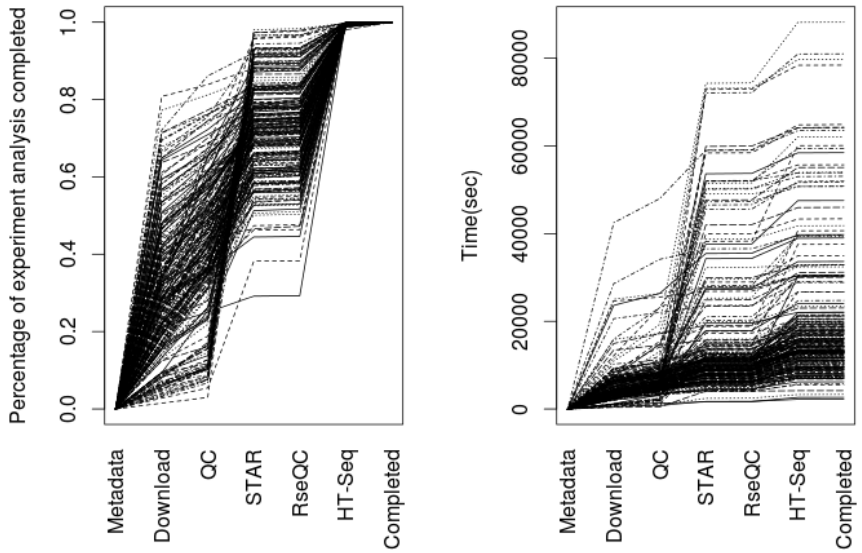

Figure 17: (Left) Timeline of 235 experiments time usage in complete RNA-Seq data analysis. (Right) Proportion of time spent of the whole analysis time by the end of each of the sequential processes over 235 analysed RNA-Seq experiments

Table 6: Time consumption summary statistics for all independent pipeline processes expressed in minutes

	Metadata	Download	QC	STAR	RSeQC	HTSeq	Post
95%	0.10	185.37	87.40	746.81	1.58	170.56	1.96
50%	0.03	64.18	31.28	51.38	0.82	67.13	0.60
5%	0.00	19.47	10.69	12.68	0.40	25.09	0.19

Heatmap on figure 18 shows Pearson correlations experiment process completion times between 4 of the most time consuming processes. The results show that these are not highly correlated between any of the processes. This can be attributed to differing amount of samples and coverage depths besides other unknown quantitative factors.

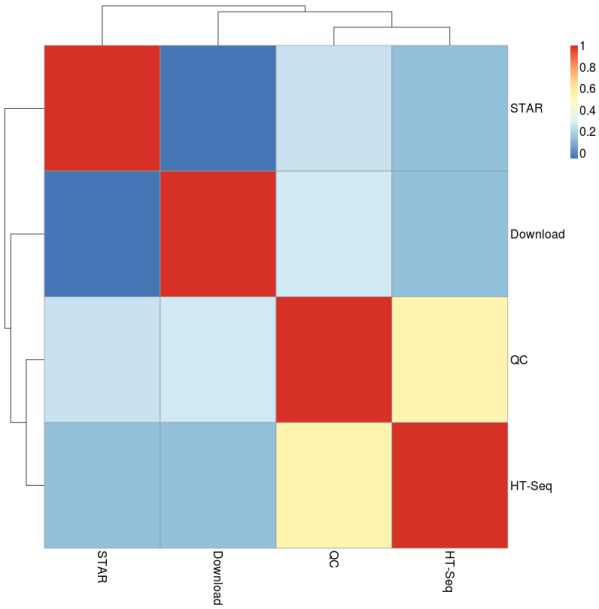

Figure 18: Heatmap of time usage correlations between alignment, downloading, quality control and quantification [40]

Downloading, running *STAR* and *HTSeq* procedures are all relatively equal in their time consumption. The tail of alignment time usage distribution is considerably longer although this property exhibits also in other processes. More experiments deviate by using more time than the mean completion time. Table 6 gives quantiles for a median and 90% empirical analysis confidence intervals of completion times for all individual steps. The median runtime for all processes is 215.42 minutes or around 3 h and 35 min. Upper end of the confidence interval is drawn out by *STAR* taking disproportionally longer to run in some cases. This means that 95% of the experiments are completed in less than 20 hours. Half of this time expense of this is spent on *STAR*.

Experiments were analysed in bottom-to-top order of size from smaller to larger ex-

periments. Left side of figure 19 shows how the number of samples does not corroborate with size. One experiment with 500 samples was removed from the image for scaling. The right side of the figure however illustrates that correlation between the full analysis time and size exists. The number of samples is irrelevant for separate consideration.

All largest size experiments showing up in the sample belong to *Arabidopsis thaliana* species. This is explained by acknowledging that there are only 93 experiments for that species available and experiments of *Arabidopsis thaliana* have just been more extensively analysed than larger platforms.

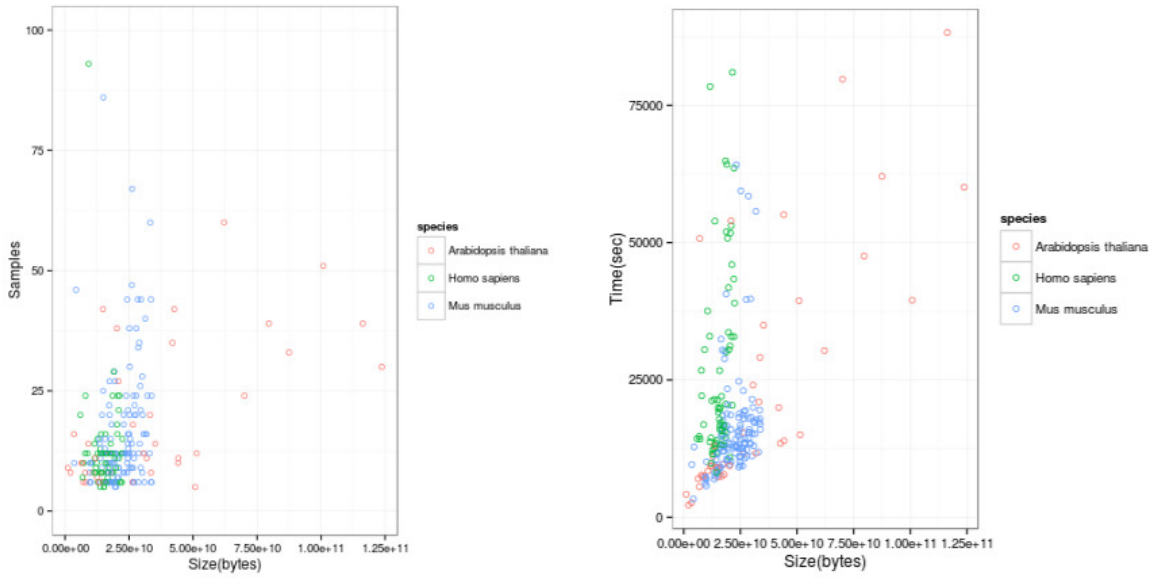

Figure 19: (Left) Number of samples versus the sizes of experiments over 235 analysed RNA-Seq experiments. (Right) Analysis time and size of individual experiments over 235 analysed RNA-Seq experiments

Regardless of limited number of samples, visual inspection affirms that the size of samples and analysis time seem to scale linearly. Experiment sizes were normalized by the time of the total analysis. All data was divided in collections by species to identify any group specific anomalies. Left side of figure 20 notes that on average *Homo sapiens* experiments of the same size are analysed much longer than other species. Right side of the figure additionally illustrates the time normalized by raw file sizes spent specifically in the alignment phase. Human experiment alignment takes on average twice as long. A possible reason for this include the larger genome.

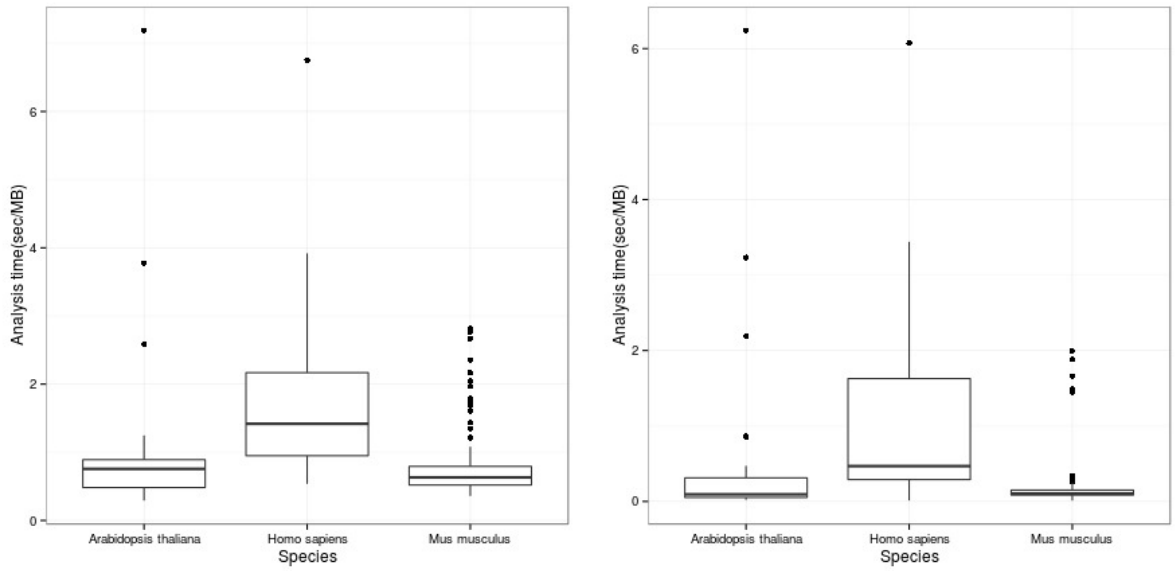

Figure 20: (Left) Analysis time of full experiments normalized by experiment sizes by species. (Right) Analysis time of alignment step normalized by experiment sizes by species.

Figure 21 accounts for all analysed experiments by species and size and shows that a vast majority of experiments take less than 25000 seconds to analyse.

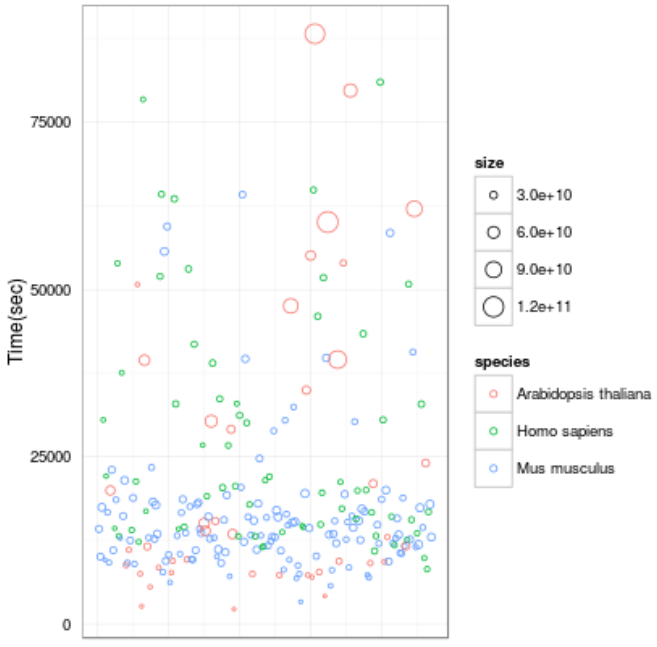

Figure 21: Analysis time of 235 RNA-Seq experiments by size and species

A subsequent future time estimation from analysis needs to account for future exper-

Table 7: Analysis status summaries by species (account of 11.05.2015)

	Analysed experiments	Failed experiments	Immediate failure
Homo sapiens	301	63	37
Mus musculus	480	89	61
Arabidopsis thaliana	87	17	13
Rattus norvegicus	47	8	4
Drosophila melanogaster	91	21	14

iments suffering on average a 20% failure rate. Time used up for failing experiments is highly dependent on the point of failure. The loss is almost negligible when metadata construction fails, worse when failure prevails in some of the final steps due for example to low quality of aligned reads.

4.3.1 Analysis time simulation study

Time spent on analyses in the future can be estimated using data from previously logged information. The database which handles the ordering of experiment analysis will have been constructed in advance so the size and identifications for all experiments are known beforehand. Thus also the experiments to be analysed are precisely known.

Table 7 gives the statistics about failing experiments. If the experiment does not fail immediately because of incomplete or inconsistent metadata then it is assumed that it may fail in a random timepoint. Without regard to species, a general probability of failing is 19.6%. Of those failing, the proportion of those immediately doing so is 65.2%. After normalizing the analysis times for sizes, analysis times were simulated independently for each of the steps assuming distributional fits emerging from log-normal distribution.

Memory constraints on servers and predefined threading settings predispose running three species concurrently on a server with 1024 GB RAM. In simulations separate processes are given for *Mus musculus* and *Homo sapiens*. The third process is stored for smaller platforms. If one is completed, computational resources will be redirected to the next species in line. For simulation we line up yet unprocessed experiments of *Drosophila melanogaster*, *Rattus norvegicus*, *Saccharomyces cerevisiae*, *Danio rerio* and *Caenorhabditis elegans* in this exact order. Assumptions for simulation are extrapolated from *Arabidopsis thaliana* and *Mus musculus* aggregates.

1000 independent simulations for experiment times are conducted. Number of successfully analysed experiments were distinguished for one day, a week, two weeks, three weeks, a month and two months based on known based on actual future analysis sizes and data. Obtained results were also given empirical 90% confidence intervals.

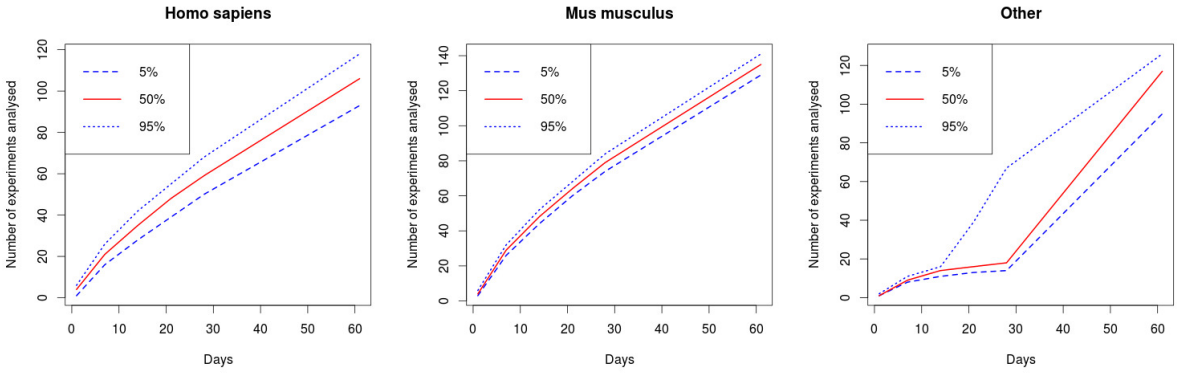

Figure 22: Future analysis time of RNA-Seq experiments by species. (Left) Time expense of analysing*Homo sapiens* experiments. (Center) Time expense of analysing *Mus musculus* experiments. (Right) Time expense of analysing experiments of *Drosophila melanogaster*, *Rattus norvegicus*, *Saccharomyces cerevisiae*, *Danio rerio* and *Caenorhabditis elegans* data.

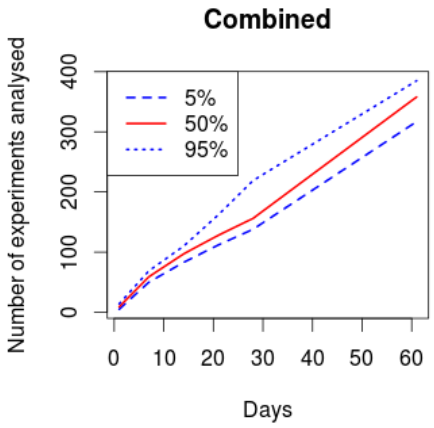

Figure 23: Combined future analysis time of RNA-Seq experiments for three concurrent analysis processes

Figure 22 indicates steady progression of mouse and human threads. The third platform indicates a comparatively lower number of analysed experiments during the first month. This is because only a limited number of relatively large *Drosophila melanogaster* and *Rattus norvegicus* are still to be analysed. These would be completed in about a month. In second month, the analysis is much faster thanks to small sized experiments of following species.

Counts for number of analysed experiments combined on all three processes are displayed in table 8. These indicate that on average we expect 156 experiments to be completed in a month on three simultaneous processes. Most of the gains in absolute

numbers are retrieved from smaller platforms. This information enables us to more accurately quantify time-horizons and deal with resource management.

Table 8: Median and 90% confidence intervals for number of experiments successfully analysed in the future on three simultaneous threads

	1 day	1 week	2 weeks	3 weeks	1 month	2 months
5%	5	50	83	112	138	317
50%	9	59	97	128	156	358
95%	14	69	110	162	219	385

4.4 Data integration within *MEM*

Current newest RNA-Seq data collection dates to 05.05.2015. This collection contains samples from five species. Number of datasets and available species are given in table 9.

Table 9: Number of datasets by species in the RNA-Seq data collection of 05.05.2015)

Species	Analysis genome	Number of datasets
Homo sapiens	GRCh38	227
Mus musculus	GRCm38	378
Arabidopsis thaliana	TAIR10	70
Drosophila melanogaster	BDGP5	56
Rattus norvegicus	Rnor_5.0	35

Figure 24 shows the interface for performing *MEM* queries. A single gene ID in the form of gene short name or ENSEMBL ID is required. Only experiments that have a larger gene-wise standard deviation than the minimum required deviation are included in the significance calculations. Optimal values for standard values for RNA-Seq data are different from microarray experiments because of differing analysis work-flows. Therefore standard deviation filter should be specified with care under "Dataset filters" tab.

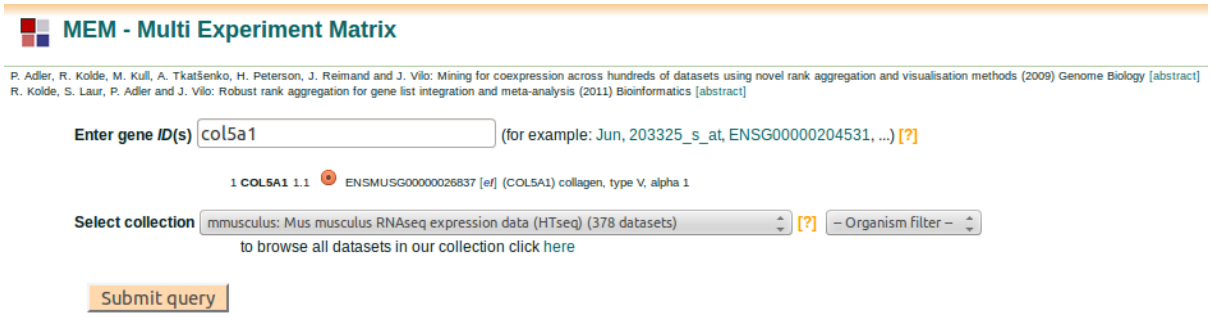

Figure 24: *MEM* query form

Figure 25 shows output results for *COL5A1* query on mouse collection. Closest genes by expression are ranked in top section of the output matrix. Expectedly other collagen genes are amongst the most similarly expressed with very high significance scores.

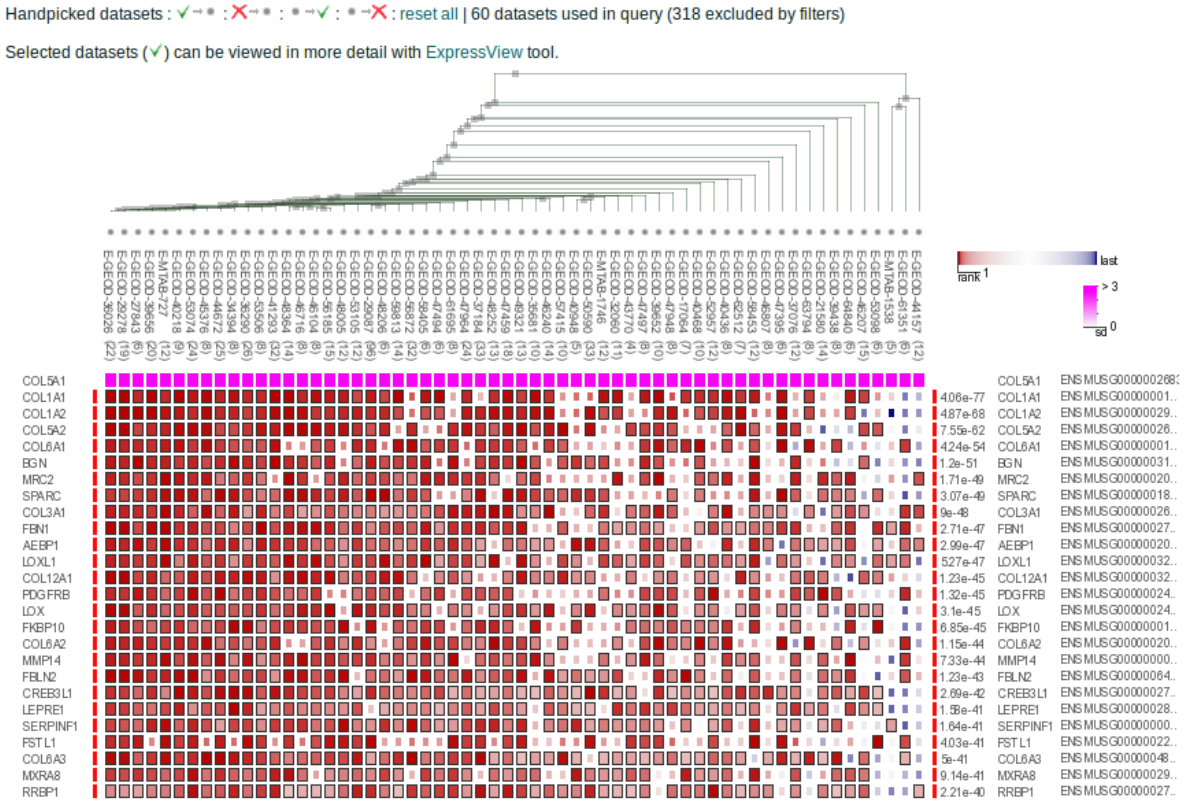

Figure 25: *MEM* interface for mouse COL5A1 gene query. Results are expressed in the form of a matrix. Gene data are found in rows. Columns are experimental datasets that have been selected for co-expression calculations from the set of all available experiments. Ordering of genes is given by similarity to query gene (COL5A1). Significance of the co-expression result is the first column to the right from the coloured data matrix. Individual values in the matrix display ranks on a relative scale. Ranks for red squares are larger and are therefore more similar to the expression profile of the query gene. Opposite applies for blue squares.

Collection has been made available for queries in a beta version of *MEM*. Link for running the tool on the prepared collection is specified in appendix 1.

5 Conclusion

This work aims to give an overview of main processes of RNA-Seq data analysis. It describes a selection of tools to conduct expression analysis and describes their algorithms. Thesis also gives an overview of how the tools were combined and merged into a single workflow as to enable transforming publicly available raw reads into expression data. The main contribution of the work is the construction of a comprehensive analysis framework allowing for analysis of public data from ArrayExpress database. Experimental information is managed and organized into form allowing for coordinated automatic analysis. Underlying database supports the operations of the analysis pipeline. Metadata collections and databases can easily be brought up-to-date with current ArrayExpress status. The tool selection is based on finding a balance between speed and performance allowing most experiments to be analysed in the shortest amount of time. A separate section within the thesis investigated some methods for automatic library inference. This has been a common problem of design copiously affecting analysis results in case of misguided information.

Motivation for design of a new analysis pipeline enabling extensive RNA-Seq data analysis was to prepare data collections for *MEM* - a tool for co-expression discovery over multiple datasets. Existing pipelines serve either very limited analysis purposes or are speed-wise optimised for the underlying system. Available solutions also do not support multiple experiment analysis. Even though mining for information over multiple expression data is becoming more common then the computation expenses and relative infancy of the NGS methods means a lot of work is still to be done.

Performance of the pipeline was evaluated in a runtime analysis study. Results from these were corroborated by a simulation study for predicting future analysis times which assesses expected performance of the pipeline in the future.

Computational resources of BIIT group were used and development was a part of an ELIXIR science infrastructure initiative.

References

[1] Kolesnikov, N. et al. ArrayExpress update-simplifying data submissions, *Nucleic Acids Research*, 43 (Database Issue). D1113-6. available online doi: 10.1093/nar/gku1057

[2] Wang, Z., Gerstein, M. and Snyder, M. RNA-Seq: a revolutionary tool for transcriptomics. *Nature Reviews Genetics*, 10 (January 2009). 57–63. available online doi: 10.1038/nrg2484.

[3] Petryszak, R. et al. Expression Atlas update—a database of gene and transcript expression from microarray- and sequencing-based functional genomics experiments, *Nucleic Acids Research*, 42 (D1). D926:D932. available online doi: 10.1093/nar/gkt1270

[4] Fonseca, N., Marioni, J. and Brazma, A. RNA-Seq Gene Profiling - A Systematic Empirical Comparison, *PLoS ONE*, 9(9). e107026. available online doi:10.1371/journal.pone.0107026

[5] Adler, P. et al, Mining for coexpression across hundreds of datasets using novel rank aggregation and visualisation methods, *Genome Biology* 10 (December 2009). R139. available online doi:10.1186/gb-2009-10-12-r139

[6] JGraph Ltd. *Draw.io* software, project website www.draw.io, accessed 22.04.2015

[7] R Core Team, R: A Language and Environment for Statistical Computing, Vienna, 2014, project website http://www.R-project.org

[8] Grada, A. and Weinbrecht, K. Next-Generation Sequencing: Methodology and Application. *Journal of Investigative Dermatology*, 133 (2013). e11. available online doi:10.1038/jid.2013.248

[9] Saiki, R.K. et al. Primer-directed enzymatic amplification of DNA with a thermostable DNA polymerase, *Science* 239(4839). 487-491. available online doi: 10.1126/science.2448875

[10] Metzker, M.L. Sequencing technologies — the next generation. *Nature Reviews Genetics* 11 (January 2010). 31-46. available online doi:10.1038/nrg2626

[11] Mutz, K-O. et al. Transcriptome analysis using next-generation sequencing. *Current Opinion in Biotechnology*, 24 (February 2013). 22–30. available online doi:10.1016/j.copbio.2012.09.004

[12] Kratz, A. and Carnici, P. The devil in the details of RNA-seq. Nature Biotechnology 32 (September 2014). 882–884. available online doi:10.1038/nbt.3015 Published online 09 September 2014

[13] Oshlack, A., Robinson, M.D. and Young, M.D. From RNA-seq reads to differential expression results, *Genome Biology*, 11 (December 2010). 220. available online doi: 10.1186/gb-2010-11-12-220

[14] Wang, L., Wang, S. and Li, W. RSeQC: quality control of RNA-seq experiments, *Bioinformatics*, 28 (16). 2184-2185. available online doi: 10.1093/bioinformatics/bts356

[15] Babraham Bioinformatics, FastQC software, project website http://www.bioinformatics.bbsrc.ac.uk/projects/fastqc/, accessed 22.04.2015

[16] Martin, M. Cutadapt removes adapter sequences from high-throughput sequencing reads, *EMBnet.journal*, 17 (1). 10-12. available online doi:http://dx.doi.org/10.14806/ej.17.1.200

[17] Bolger, A.M., Lohse, M. and Usadel, B. Trimmomatic: a flexible trimmer for Illumina sequence data, *Bioinformatics*, 30 (15). 2114–2120. available online doi: 10.1093/bioinformatics/btu170

[18] Ewing, B and Green, P. Base-calling of automated sequencer traces using phred. II. Error probabilities, *Genome research*, 8(3). 186-194. available online doi:10.1101/gr.8.3.186

[19] Pease. J, and Sooknanan, R. A rapid, directional RNA-seq library preparation workflow for Illumina® sequencing, *Nature Methods*, 9 (2012). Application notes, available online http://www.nature.com/nmeth/journal/v9/n3/full/nmeth.f.355.html

[20] Korpelainen, E. et al. RNA-seq Data Analysis: A Practical Approach, CRC Press, Boca Raton, 2015.

[21] Engström, P.G. et al, Systematic evaluation of spliced alignment programs for RNA-seq data, *Nature Methods*, 10 (2013). 1185–1191. available online doi:10.1038/nmeth.2722

[22] Grant, G.R. et al. Comparative analysis of RNA-Seq alignment algorithms and the RNA-Seq unified mapper (RUM), *Bioinformatics*, 27 (18). 2518-2528. available online doi: 10.1093/bioinformatics/btr427

[23] Trapnell, C. et al. Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and Cufflinks, *Nature Protocols*, 7 (March 2012). 562–578. available online doi:10.1038/nprot.2012.016.

[24] Trapnell, C. et al. TopHat: discovering splice junctions with RNA-Seq, *Bioinformatics*, 25 (9). 1105-1111. available online doi: 10.1093/bioinformatics/btp120

[25] Dobin, A. et al. STAR: ultrafast universal RNA-seq aligner, *Bioinformatics*, 29 (1). 15-21. available online doi: 10.1093/bioinformatics/bts635

[26] Anders, S., Pyl, P.T. and Huber W. HTSeq — A Python framework to work with high-throughput sequencing data, *Bioinformatics*, 31 (2). 166-169. available online doi:10.1093/bioinformatics/btu638

[27] Dillies M-A. et al. A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis, *Brief Bioinformatics*, 14 (6). 671-683. available online doi: 10.1093/bib/bbs046

[28] Robinson, M.D. and Oshlack, A. A scaling normalization method for differential expression analysis of RNA-seq data, *Genome Biology*, 11 (March 2010). R25. available online http://genomebiology.com/2010/11/3/R25

[29] Casella, G. and Berger, R.L. *Statistical Inference*, 2nd ed. Duxbury, Pacific Grove, 2002.

[30] Brazma, A. et al. Minimum information about a microarray experiment (MIAME) — toward standards for microarray data, *Nature Genetics*, 29(4). 365-371. available online doi:10.1038/ng1201-365

[31] Functional Genomics Data Society. MINSEQE: Minimum Information about a high-throughput Nucleotide Sequencing Experiment - a proposal for standards in functional genomic data, 06.2012. project website http://www.fged.org/projects/minseqe/, accessed 11.05.2015

[32] Domrachev, E.R. and Lash, A.M. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository, *Nucleic Acids Research* 30(1):207-210, available online doi: 10.1093/nar/30.1.207

[33] Kolde, R. et al. Robust rank aggregation f or gene list integration and meta-analysis, *Bioinformatics* 28 (4). 573-580. available online doi: 10.1093/bioinformatics/btr709

[34] Fonseca, N. et al. iRAP - an integrated RNA-seq Analysis Pipeline, *bioRxiv*, released online June 6 2014, available online doi: http://dx.doi.org/10.1101/005991

[35] Goncalves, A. et al. A pipeline for RNA-seq data processing and quality assessment, Bioinformatics 27 (6): 867-869. available online doi: 10.1093/bioinformatics/btr012

[36] Torres-Garcia, W. et al. PRADA: pipeline for RNA sequencing data analysis, *Bioinformatics*, 30(15). 2224-2226. available online doi: 10.1093/bioinformatics/btu169

[37] Rung. J. and Brazma, A., Reuse of public genome-wide gene expression data, *Nature Reviews Genetics*, 14 (February 2013). 89-99. available online:doi:10.1038/nrg3394

[38] Sanna, C., Li, W-H. and Zhang, L. Overlapping genes in the human and mouse genomes. *BMC Genomics*, 9 (April 2008). 169. available online doi: 10.1186/1471-2164-9-169

[39] Arak, T. and Kull, M. TabCDF software, project website http://biit.cs.ut.ee/tabcdf/

[40] Kolde, R. pheatmap: Pretty Heatmaps, *R package version 0.7.7*, project website http://CRAN.R-project.org/package=pheatmap

Appendix 1. Links

1. Download link for experiment E-GEOD-22351 SDRF, IDF and final processed data files

   `https://owncloud.ut.ee/owncloud/index.php/s/IZADyMUFPcYuGh1`

2. RNA-Seq data collection for MEM software, released online 05.05.2015

   `http://caiman.at.mt.ut.ee:1982/index.cgi? project=mem_rna_050515`

3. Research seminar in Bioinformatics. Autumn semester 2014/2015. SAMtools and CRAM format.

   `https://owncloud.ut.ee/owncloud/index.php/s/VDOQMW4gb8xS9XL`

Non-exclusive licence to reproduce thesis and make thesis public

I, Tõnis Tasa (date of birth: 30th of October 1989),

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:

1.1 reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and

1.2 make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

Re-using public RNA-Seq data

supervised by Priit Adler

2. I am aware of the fact that the author retains these rights.

3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, 18.05.2015