

Guidelines for RNA-Seq data analysis



Nicolas Delhomme^{1,*}, Niklas Mähler², Bastian Schiffthaler¹, David Sundell¹, Chanaka Mannapperuma¹, Torgeir R. Hvidsten^{1,2}, Nathaniel R. Street^{1,3}

1) Umeå Plant Science Center, Department of Plant Physiology, Umeå University, 90187, Umeå, Sweden

2) Department of Chemistry, Biotechnology and Food Science, Norwegian University of Life Sciences, 1432 Ås, Norway

3) Computational Life Science Cluster (CLiC), Umeå University, Umeå, Sweden

Email feedback to: nicolas.delhomme@umu.se

Last reviewed: 17 November 2014 by Michael Love, Department of Biostatistics, Harvard School of Public Health, Boston, USA.

Key words: Next-Generation Sequencing, RNA-Seq, Data pre-processing, Data analysis

Introduction

RNA-Seq (RNA-Sequencing) has fast become the preferred method for measuring gene expression, providing an accurate proxy for absolute quantitation of messenger RNA (mRNA) levels within a sample (Mortazavi *et al*, 2008). RNA-Seq has reached rapid maturity in data handling, QC (Quality Control) and downstream statistical analysis methods, taking substantial benefit from the extensive body of literature developed on the analysis of microarray technologies and their application to measuring gene expression. Although analysis of RNA-Seq remains more challenging than for microarray data, the field has now advanced to the point where it is possible to define mature pipelines and guidelines for such analyses. However, with the exception of commercial software options such as the CLCbio CLC Genomics Workbench, for example, we are not aware of any fully integrated open-source pipelines for performing these pre-processing steps. Both the technology behind RNA-Seq and the associated analysis methods continue to evolve at a rapid pace, and not

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

all the properties of the data are yet fully understood. Hence, the steps and available software tools that could be used in such a pipeline have changed rapidly in recent years and it is only recently that it has become possible to propose a *de-facto* standard pipeline. Although proposing such a skeleton pipeline is now feasible there remain a number of caveats to be kept in mind in order to produce biologically and statistically sound results.

Here we present what is, in our opinion, a mature pipeline to pre-process and analyze RNA-Seq data. As often as possible we identify the most obvious pitfalls one will face while working with RNA-Seq data and point out caveats that should be considered. An overview of the pipeline is presented in **Figure 1** and is detailed below. Briefly, the first step upon receiving the raw data from a sequencing facility is to conduct initial QC checks. These QC results will inform whether the data requires filtering to remove ribosomal RNA (rRNA) contamination, if sequence reads require ‘trimming’ to remove low quality bases and if there is a need to trim reads to remove sequencing adapters. These data pre-processing steps must be performed with care to ensure that the required data cleaning is adequately performed while avoiding the introduction of any potential bias, for example removing sequences of interest. Once the data is deemed of sufficient quality, it is aligned/mapped (both terms are considered synonyms in the following) against the chosen reference; this can be a model organism genome, a novel draft genome or a *de-novo* assembled transcriptome. Each of these alternatives has advantages and caveats, some of which are detailed below. Having obtained the mapping of the RNA-Seq reads to the genome, the subsequent analysis steps to be performed will be determined by the project goals and the scientific questions that one wishes to address. Distinctly different analysis methods are required depending on whether interest lies in identifying sequence variants or in exploring expression level differences between samples groups *i.e* differential expression (DE), for example. These are the two most popular uses of RNA-Seq data and are hence briefly introduced at the end of the current protocol. However, as these analyses are complex, we redirect the reader to more complete literature. There are many additional analyses that RNA-Seq data can be used for, including examining allele-specific expression and RNA editing, among others.

The pipeline we describe in the following is made publicly available and we aim to shortly release a worked example using a representative dataset as a companion to this protocol. The worked example will exemplify all of the steps detailed below and will demonstrate the influence of different biases and steps taken to mitigate them. The guideline will be available at <https://bioinformatics.upsc.se/>.

Before reading on, we wish to stress that as the analysis of RNA-Seq data is still a rapidly maturing field, one must always keep an open mind, challenging the results obtained to be sure a possible technical artifact does not underlie an observed difference, particularly where unexpected results are obtained. Only when such possibilities have been considered and eliminated can one assume that observed results are likely of biological origin.

Notes:

- In this tutorial we focus on RNA-Seq data obtained from mRNA - not total RNA - and

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

generated using the Illumina platform, more specifically data obtained using the now standard protocol using TruSeq adapters for sequencing on an Illumina HiSeq 2000.

- Throughout the tutorial, the different steps are exemplified using our currently preferred tools. However, there exist numerous alternatives for every step of this pipeline and we encourage readers to explore alternatives and to check the literature as updated tools become available to ensure that the best option is selected to match both the data being analyzed and the question being addressed.
- The described pipeline is implemented and made available from our Git repository: <https://bioinformatics.upsc.se/git/UPSCb-public.git>. This repository is constantly being updated and revised as we fix issues, implement new tools and devise new analysis methods.
- Although the term “isoform” is commonly used in the “RNA-Seq” community to refer to transcript splice variants (*i.e.* gene isoforms) arising from alternative splicing of a single gene loci, it is often misunderstood in other communities where it is most commonly understood as protein isoform. Hence, in this document, we refer to multiple transcripts originating from a single gene as splice variants.

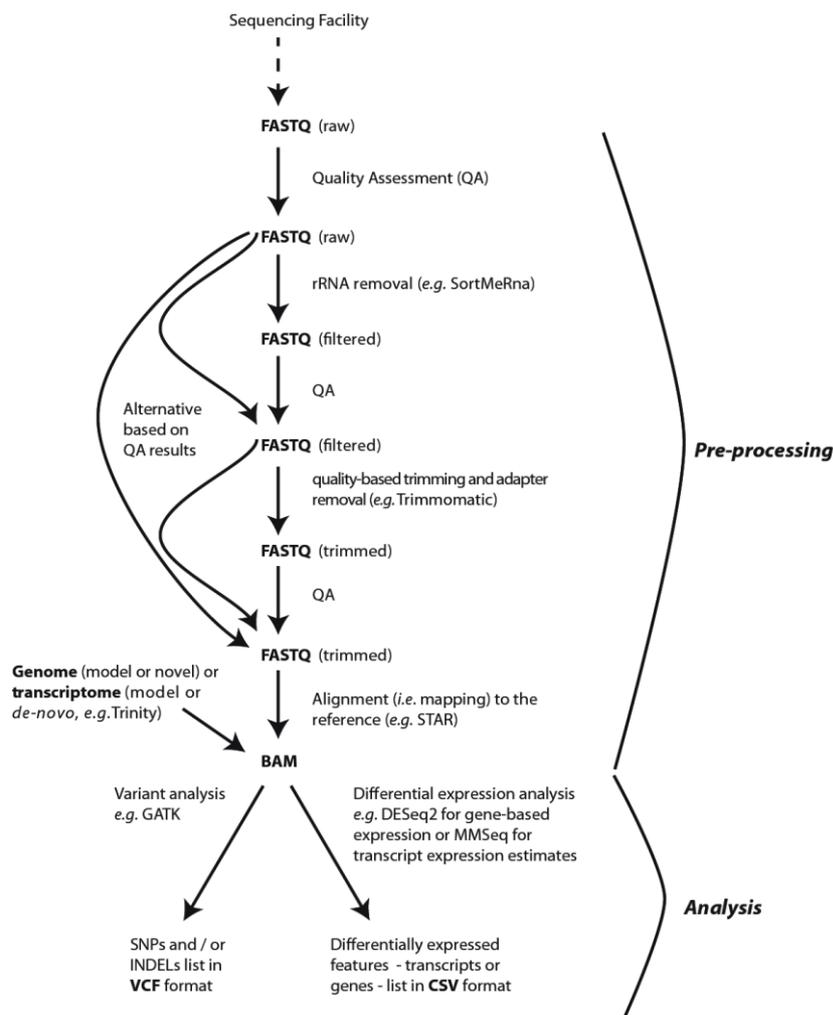


Figure 1. The RNA-Seq pre-processing and analysis workflow. Nodes represent the data format at a given stage; edges represent the process the data undergoes - or the tool used.

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

Procedure

In most of the following, tool command lines are exemplified using a paired-end (PE) FASTQ (Cock *et al*, 2010) formatted file set, named `read_1.fq.gz` and `read_2.fq.gz`. These files contain raw data received from the sequencing facility. We also assume using a computer with 8 cores.

1. Raw Data QC Assessment

Upon receiving the RNA-Seq FASTQ files from the sequencing facility, it is essential that some initial QC assessments be performed. Most importantly, one should check the overall sequence quality, the GC percentage distribution (*i.e.* the proportion of guanine and cytosine bp across the reads) and the presence/absence of overrepresented sequences. FastQC (<http://www.bioinformatics.babraham.ac.uk/projects/fastqc/>) has become the *de-facto* standard for performing this task. FastQC is run for every sequencing file independently as follows:

```
fastqc -o qa/raw -t 8 --noextract read_1.fq.gz read_2.fq.gz
```

The output of FastQC is a zip archive containing an HTML document, which is sub-divided into sections describing the specific metrics that were analyzed. These sections are:

a) Basic Statistics

Most metrics within this section are self-explanatory. For PE reads, the total number of sequences should match between the forward and reverse read files. It is good practice to take note of the FASTQ Phred encoding, as some downstream tools require the user to specify whether Phred64 or Phred33 encoding should be used. Finally, the %GC should lie within the expected values for the sample species.

Note: the Phred scale value is a “best guess” by FastQC and there is always a very small possibility that it may be miss-identified. However the sequencing facility data delivery report should contain this information. If in doubt we suggest consulting the relevant Wikipedia page (http://en.wikipedia.org/wiki/FASTQ_format).

b) Per base sequence quality

The Phred scale quality represents the probability p that the base call is incorrect. A Phred score Q is an integer mapping of p where $Q = -10 \log_{10} p$. Briefly, a Phred score of 10 corresponds to one error in every 10 base calls or 90% accuracy; a Phred score of 20 to one error in every 100 base calls or 99% accuracy. The maximum Phred score is 40 (41 for Illumina version 1.8+ encoding). See http://en.wikipedia.org/wiki/FASTQ_format#Quality for more details on the quality and http://en.wikipedia.org/wiki/FASTQ_format#Encoding for information on the corresponding encoding.

The second FastQC section details the Phred scaled quality as a function of the position in the read. It is very common to observe a quality decrease as a function of the read length (**Figure 2C**) and this pattern is often more pronounced for read2 than it is for read1; this is

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

due to cumulative stochastic errors of the sequencing progresses, largely as a result of the enzyme ‘tiring out’, and the increasing likelihood that a read cluster becomes out of sync, for example.

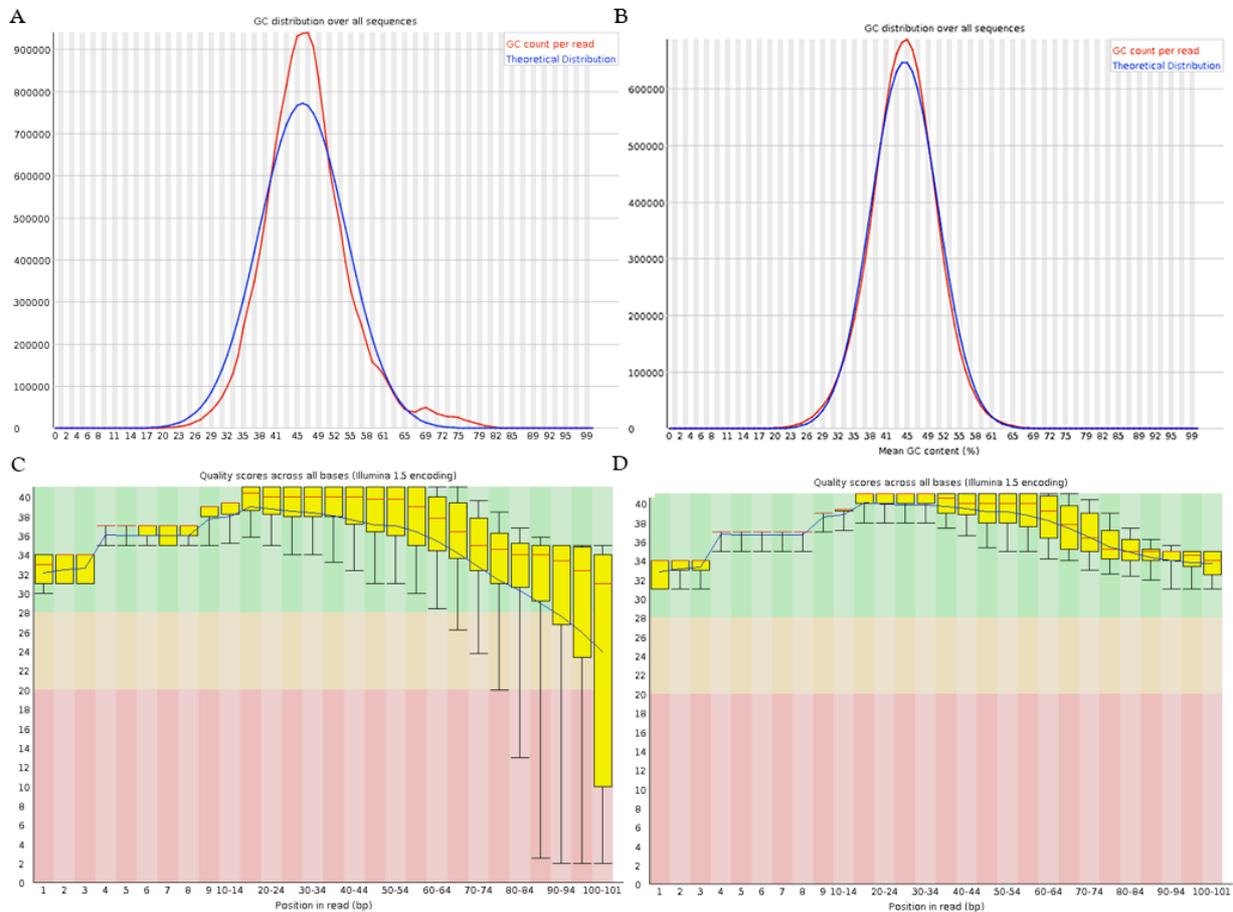


Figure 2. QA plots extracted from FastQC report at different stages of the data pre-processing. **A)** The “Per sequence GC content” of the raw data. **B)** The same data shown in A but after rRNA filtering. **C)** “Per base quality score” of the raw data. **D)** The same data after quality-based trimming has been performed.

c) Per sequence quality scores

This section details the quality distribution at the read level, in contrast to the quality per base position of the previous section. If the data is of good quality, the histogram will be skewed to the right.

d) Per base sequence content

In this section, the average proportion of individual bases (A, C, G and T) is plotted as a line across the length of the reads. The 12 first bases often show a bias that is characteristic of Illumina RNA-Seq data. This is in contrast with the DNA-Seq protocol, which does not show the same bias. The difference between protocols lies in three additional steps performed during the conversion of mRNA to cDNA, which is subsequently sequenced as if it were

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

genomic DNA. Several hypotheses have been proposed as to the cause of this bias: during reverse transcription of the captured cDNA, random hexamer primers are used and these may introduce a positional bias of the reads; artifacts from end repair; and possibly a tenuous sequence specificity of the polymerase may each play a role either singularly in, most likely, in combination.

Note: In multiplexed samples using the Lefrançois *et al.* method (Lefrançois *et al.*, 2009), the barcode may still be present and may also affect the base composition distribution of the first bases of the read. This protocol is now used infrequently as Illumina has developed a proprietary protocol (using two additional sequencing reactions) and the reads are now demultiplexed directly by the sequencing facilities.

e) Per base GC content

Similar to the previous section, the GC content is shown as a function of the position in the read. As previously observed, a bias for the first base pairs (once more in contrast to DNA sequencing data) will often be observed. In addition, for non-strand specific RNA-Seq data, the amount of G and C and of A and T should be similar, as an average, at any position within reads. Moreover the proportion of G+C should match the expected GC content of the sample. For strand-specific data, if the RNA was selected using poly-dT beads, enrichment for T over A may be observed.

f) Per sequence GC content

The plot in this section (see **Figure 2A** for an example) represents the distribution of GC content per read, where the data (red curve) is expected to approximately follow the theoretical distribution (blue curve). If the curve presents a shoulder in a region of high GC content, this is usually an indication that rRNA is present in the sample. However, it may also represent contamination by an organism with a higher GC content (such as bacteria or fungi). In contrast, a peak on the left hand side would indicate enrichment for A/T rich sequences. In particular a sharp peak for very low GC content (in the 0-3 range) is usually indicative of the sequencing of the mRNA poly-A tails. If this plot still shows issues after quality and rRNA filtering, additional steps would have to be taken to filter contaminants.

Note: There is a common misunderstanding concerning the blue theoretical curve. It is NOT devised from the reference genome/transcriptome of your species of interest, as FastQC has no information to this end. It is a computed Gaussian distribution parameterized with the average and variance of the GC proportion of the input reads. As exemplified in **Figure 3**, if a sample contains substantial amounts of rRNA, the data curve would be similar to that of the red curve in panel A, while the computed theoretical curve would be similar to the red curve in panel B. The green curves represent the converse example where there is enrichment for A/T, such as would result from the sequencing of mRNA poly-A tails.

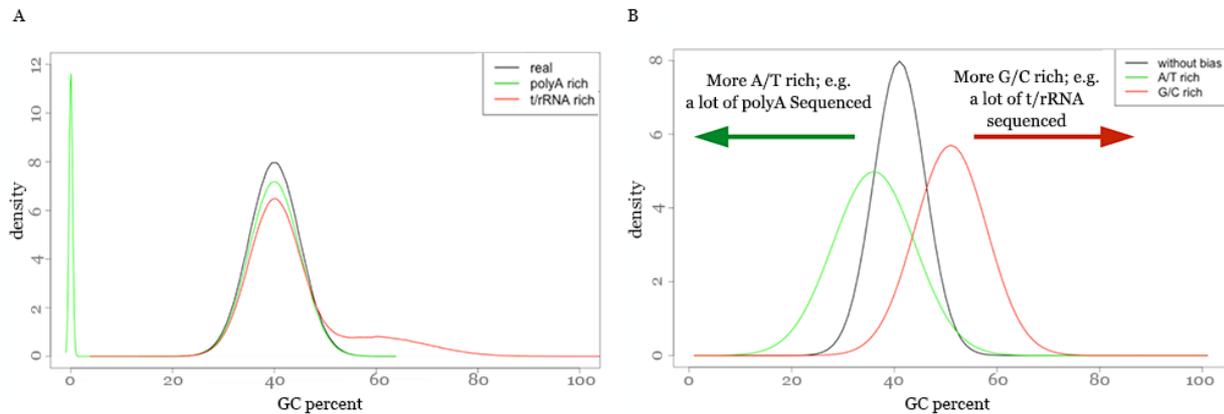


Figure 3. A comparison of the “theoretical” and “observed” GC distribution, *i.e.* the blue and red lines of FastQC “Per sequence GC content” QA plot, see *e.g.* Figure 2B. **A)** Examples of “observed” GC distribution with a poly-A enrichment (green), rRNA enrichment (red) or no (black) bias. **B)** The corresponding “theoretical” curve that FastQC would devise from such read GC content distribution.

g) Per base N content

This plot shows the fraction of indistinguishable bases as a function of the base position in the reads. In high quality sequence data this is expected to be close to zero. Deviations from the expected values indicate problems during the sequencing.

h) Sequence length distribution

This section shows the distribution of read lengths. Prior to trimming, there should be exactly one peak located at the sequenced read length.

i) Sequence duplication level

This plot represents the level of duplicate sequences in the library. FastQC assumes that the library is diverse, with even representation of all sequences, *i.e.* it assumes a uniform coverage as would usually be obtained for DNA-Seq experiments. However, this assumption is not valid for RNA-Seq libraries, which have a large dynamic range, possibly containing a million fold difference between lowly and highly expressed genes. As a result it is common to observe high duplication levels for sequences originating from highly expressed genes. It is worth noting that before version 0.11 of FastQC, all duplication levels ≥ 10 were aggregated into a single bin. In more recent version this has been made more comprehensive in order to provide a more accurate representation of the data.

j) Overrepresented sequences

This table shows sequences that are present at unusually large frequency in the reads. These are most commonly sequencing adapters and will be identified as such. If unidentified sequences are detailed these may originate from rRNA or other contaminants, in which case contaminant filtering should be considered. Often a BLAST search of the unidentified sequence using the NCBI nt database will be informative.

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

k) Kmer content

The final plot of the FastQC report details the occurrence of kmers - nucleotide sequences of fixed k length – that were present at a higher than expected frequency as a function of their position within the read. Commonly, the early bases show the aforementioned Illumina sequencing bias (see section d), whereas the last bases may show enrichment for sequencing adapters.

Note: FastQC has a significant caveat that users should be aware of: For computational reasons only the first 200,000 reads of a sequencing file are used to determine the Phred quality scale in use. This is generally an acceptable compromise as reads are randomly distributed in the FASTQ file. However, should that assumption not hold true, an alternative solution should be sought. For example, one alternative would be to use the R/Bioconductor package ShortRead and its FastqSampler / qa functionalities (Morgan *et al*, 2009).

2. rRNA filtering

Typically, wet-lab protocols to extract mRNA include a step to deplete the sample of rRNA or to enrich it for poly-adenylated transcripts (rRNA is not poly-adenylated). Common approaches to achieve this are to use RiboMinus™ kits (Life Technologies) or poly-dT beads, respectively or to include a precipitation step that selectively precipitates only long (usually >200 bp) nucleotide fragments. No approach will be fully sensitive and, as a result, some rRNA carryover is to be expected. This is not a problem *per se* as long as the remaining proportion accounts for a low percentage of the reads (commonly between 0.1 and 3%). Larger proportions will have an effect on the usable number of reads obtained from the samples, as fewer sequence reads would originate from expressed mRNAs. This is not to be overlooked as these rRNAs will produce valid alignments (in all reference genomes and for most *de novo* assembled transcriptomes and genomes) and hence other metrics (such as the alignment rate) will fail to identify such a bias. To control for the rRNA quantity in our sample FastQ files, we use SortMeRna, a tool originally developed to identify rRNA in metagenomics analyses (Kopylova *et al*, 2012). The tool accepts FASTQ files (SE or PE) as input and includes a set of reference libraries containing the most common rRNAs (5,5.8,16, 18, 23 and 26-28S). Example command lines for a PE sample are:

```
> find . -name "read_[1,2].fq" | xargs -P 2 -l {} sh -c 'gunzip -c $0 > ${0//.fq/}' {}
```

```
> merge-paired-reads.sh read_1.fq read_2.fq read-interleaved.fq
```

```
> sortmerna -n 6 --db $SORTMERNADIR/rRNA_databases/rfam-5s-database-id98.fasta  
$SORTMERNADIR/rRNA_databases/rfam-5.8s-database-id98.fasta  
$SORTMERNADIR/rRNA_databases/silva-bac-16s-database-id85.fasta  
$SORTMERNADIR/rRNA_databases/silva-euk-18s-database-id95.fasta  
$SORTMERNADIR/rRNA_databases/silva-bac-23s-database-id98.fasta  
$SORTMERNADIR/rRNA_databases/silva-euk-28s-database-id98.fasta
```

```
--l read-interleaved.fq --other read-sortmerna.fastq --log sample.log -a 8 -v --paired-in
```

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

```
> unmerge-paired-reads.sh read-sortmerna.fastq read-sortmerna_1.fq read-sortmerna_2.fq  
> find . -name "read-sortmerna_[1,2].fq" | xargs -P 2 -l {} gzip {}
```

The format conversion step is not required for SE samples, nor is the '--paired-in' argument. The SORTMERNADIR environment variable needs to be set at installation and the '-a' argument details the number of CPUs/threads to be used. The tool manual provides a comprehensive description of all functionalities. SortMeRna does not currently support compressed input, hence the first and last step to (de)compress the data (here we use "find" to identify the files and "xargs" to parallelize the (de)compression; as there are only 2 files to process, we set the corresponding multiple threads argument (-P) accordingly).

Note: This step is not mandatory and could be omitted if the preliminary QC does not reveal any GC biases, in particular enrichment for sequences with GC content over 50% (typical for rRNA). The QC might not always reveal such a bias (e.g. studies in organism with an average GC content similar to rRNA such as most fungi) and if there is any doubt, this step should be performed.

3. Filtered Data QC

The filtered data is again subjected to a QC assessment by FastQC to ensure the validity of the filtering steps. The GC content plot should show the biggest change, now fitting more closely to the theoretical distribution, as shown in **Figure 2A** and **Figure 2B**, which represent the raw and filtered GC content respectively. Shoulders, which were present in regions of higher GC content, should be noticeably smaller or be absent altogether. rRNA overrepresented sequences should no longer be identified in the corresponding table of over-represented sequences. Finally, the theoretical GC curve should be centered closer to the expected GC value of the sample organism.

4. Quality trimming and Adapter removal

It is a fact that on Illumina sequencers, the quality of a base pair is linked to its position in the read, *i.e.* bases in the later cycles of the sequencing process have a lower average quality than the earliest cycles (as was observed in the QC report above). This effect depends on the sequencing facility and on the chemistry used and it is only recently that sequencing aligners have integrated methods to correct for this - and not all alignment software does so. A common approach to increase the mapping rate of reads is to trim (remove) low quality bases from the 3' end until the quality reaches a user-selected Phred-quality threshold. A threshold of 20 is widely accepted as it corresponds to a base call error of 1 in a 100, which is approximately the inherent technical error rate of the Illumina sequencing platform.

An additional issue encountered with Illumina sequencing is the presence of partial adapter sequences within sequenced reads. This arises when the sample fragment size has a large variance and fragments shorter than the sequencer read-length are sequenced. As the resulting reads may contain a significant part of the adapter - a bp sequence of artificial origin

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

- earlier generation alignment software (*i.e.* those that do not use Maximum Exact Matching and require global alignment of an entire read) may not be able to map such reads. Being able to identify the adapter-like sequences at the end of the read and clip/trim them - a process termed adapter removal - may consequently significantly increase the aligned read proportion.

There are numerous tools available to perform either or both of these tasks (quality trimming and adapter removal). Here, we exemplify using Trimmomatic, a tool that does both (Bolger *et al*, 2014). The command line for our PE sample example is as follows:

```
java -jar trimmomatic-0.32.jar PE -threads 8 -Phred33 read-sortmerna_1.fq.gz read-sortmerna_2.fq.gz read-sortmerna-trimmomatic_1.fq.gz read-sortmerna-unpaired_1.fq.gz read-sortmerna-trimmomatic_2.fq.gz read-sortmerna-unpaired_2.fq.gz ILLUMINACLIP:"TruSeq3-PE-2.fa":2:30:10 SLIDINGWINDOW:5:20 MINLEN:50
```

Note: The path to the jar file needs to be adapted to wherever you have downloaded the trimmomatic-0.32.jar file. The adapter sequence-containing file: *i.e.* TruSeq3-PE-2.fa in the example is also part of the Trimmomatic installation. The parameters used here for clipping the adaptor and trimming the reads are our default for 101bp PE Illumina HiSeq 2000 sequencing and are set such that reads are trimmed when the average quality over a 5 bp window drops below 20, starting from the 5' end side of the read. The stringency of these parameters may be modified based on the prior QC and would require validation by an *a posteriori* QC (see below). Moreover, note the final criterion, which keeps trimmed/clipped sequences only if they are at least 50bp long. This is simply because shorter sequences are harder to align and are more likely to have originated from technical artifacts.

Note: Trimmomatic uses a trimming sliding window that scans the read from the 5' to the 3' end. Consequently, if a sequencing run had very poor quality in the first cycle(s) the entire read would be discarded. This can be circumvented by *e.g.* hard-clipping the first bp of every read (see the HEADCROP argument). Also note that the clipping/trimming arguments are processed sequentially by Trimmomatic; meaning that the position of the HEADCROP argument would have to be prior to that of the SLIDINGWINDOW one.

Note: Quality trimming and adapter removal are not mandatory and could be omitted if the preliminary QC does not reveal any quality bias or adapter sequence enrichment.

5. Trimmed Data QC

A final FastQC run is performed to ensure that the previous quality trimming and/or adapter removal steps successfully conserved high quality reads without being too stringent and without introducing any newly apparent technical biases. Several changes should be observed in comparison with the previous QC report: first, the per-base quality scores should be noticeably different. As shown in **Figure 2C-D** the per-sequence quality distribution is now shifted towards higher scores (the trimming effect) and sequencing adapters are no longer identified as over-represented (the adapter removal effect). If over-represented sequences

remain, this indicates that an additional kind of contamination may be present and should be investigated.

Note: The overrepresented kmer plot may still show enrichment towards the end of the reads. This is most often due to the presence of short sequencing adapter fragments that are too short to be recognized as such during the removal step.

6. Read Alignment to a reference

Once the raw read quality has been assessed and determined to be sufficient, or the data has been filtered and trimmed to acceptable standards, the reads can be aligned to a reference. This process is an extremely active field of research and novel aligners are frequently published. There is, sadly, no ‘silver bullet’ and the choice of aligners will be dependent on the reference to be used (genome or transcriptome), the data type (short vs. longer reads) and the available computational power, among other factors. Most recent aligners use either BWT (Burrows-Wheeler transformation; (Burrows & Wheeler, 1994) or MEM (Maximum Exact Matches; (Khan *et al*, 2009) based approaches to perform alignment. Older generation alignment algorithms relied on a spliced-seed approach (Li & Homer, 2010). The numerous implementations of these different strategies all come with a myriad of options that may significantly affect the alignment outcome. Selecting the most accurate aligner and determining the optimal parameter set for a project can often represent a small project in itself. At the time of writing this guide there was no guideline available as to which aligner is most appropriate for a given situation (read length, type of reference, *etc.*). Hence, in the following, we exemplify using aligners that we have incorporated in our processing pipeline based on internal benchmarking for our most common experimental setup: tree genome / transcriptome, Illumina HiSeq 2500, 101bp PE sequencing. The aligner of choice varies based on the type of reference available for the project: For genome based alignment of RNA-Seq data we use STAR, a MEM based aligner - it actually uses MMP (maximum mappable prefix, a variation of MEM); for alignment of RNA-Seq data to a reference transcriptome (Dobin *et al*, 2013) we use either bowtie (version 1, BWT FM-index based, Langmead *et al*, 2009) or the BWT FM-index or MEM implementations of BWA (Li & Durbin, 2009, 2010).

a) Alignment to the genome

First, the genome needs to be indexed. This is performed using the following command:

```
STAR --runMode genomeGenerate --genomeDir indices/genome --genomeFastaFiles genome.fa --runThreadN 8 --sjdbOverhang 99 --sjdbGTFfile genome.gff3
```

where several parameters have to be set to match your environment: the “indices/genome” parameter specifies the output directory, “genome.fa” specifies the genome FASTA file file path and “genome.gff3” the file path of the gene annotation file, such as can typically be retrieved from Ensembl (in gtf format) or UCSC (in gff3 format). We also provide an additional option that would need to be edited depending on your sequencing read length (--sjdbOverhang 99); we selected 99 as most our reads are 101bp long.

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

Note: STAR can become resource-greedy when creating the genome, in particular in the case of a draft genome containing a large number of scaffolds. The parameters `--genomeChrBinNbits` and `--limitGenomeGenerateRAM`, among others, can help mitigate the memory requirements of during index creation. Further details are available in the documentation or in the STAR mailing list archive.

Once the genome index is built, we can align our sample reads to it. This is achieved as follows:

```
STAR --genomeDir indices/genome --readFilesIn read-sortmerna-trimmomatic_1.fq.gz read-sortmerna-trimmomatic_2.fq.gz --runThreadN 8 --alignIntronMax INTRONMAX --outSAMstrandField intronMotif --sjdbGTFfile genome.gff3 --readFilesCommand zcat --outFileNamePrefix results/read-sortmerna-trimmomatic-STAR --outQSconversionAdd -31 --outReadsUnmapped Fastx
```

where there are a number of additional parameters: `INTRONMAX` is important to specify so that STAR does not try to align split reads across a distance greater than `INTRONMAX` bp, *i.e.* reads that span an exon-exon junction (EEJ) only need to span at most the longest intron in your genome. The parameter “`results/sample-sortmerna-trimmomatic-STAR`” sets the path and prefix to where the results will be written (note that from now on, as the reads have been combined into a single result file, we refer to our exemplary data as “sample”). We provide a few additional parameters that may require adjustment based on your data: our sample files are gzipped so we inform STAR how to read it (`--readFilesCommand zcat`). As our files were generated using the Illumina v1.5 FASTQ format, we convert them into Sanger FASTQ (`outQSconversionAdd -31`) and finally we specify that STAR should output unmapped reads separately (`--outReadsUnmapped Fastx`).

Note: STAR can utilize shared memory; *i.e.* if the alignments are performed on a single machine, the index can be loaded once in memory and accessed by all the alignments processes. This saves time (the genome is read only once into memory) and resources (there is only one copy in memory at all times). Consult the STAR documentation for details of the `--genomeLoad` option.

STAR returns a number of result files:

- a `sample-sortmerna-trimmomatic-STARAligned.out.sam` SAM file that contains the alignment in SAM format (Li *et al*, 2009).
- two FASTQ files containing the forward and reverse unmapped reads: `sample-sortmerna-trimmomatic-STARUnmapped.out.mate1` and `sample-sortmerna-trimmomatic-STARUnmapped.out.mate2`
- a number of `sample-sortmerna-trimmomatic-STARLog.*` log files
- a number of `sample-sortmerna-trimmomatic-SJ.*` files containing splice junction information.

The SAM file is then converted into the compressed BAM format and is sorted by sequence position (*i.e.* sorted sequentially per chromosome position).

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

```
samtools view -Sb sample-sortmerna-trimmomatic-STARAligned.out.sam | samtools sort -
sample-sortmerna-trimmomatic-STAR
```

The sorted BAM file is then indexed

```
samtools index sample-sortmerna-trimmomatic-STAR.bam
```

Finally, the FASTQ files containing unaligned reads are renamed to “sample-sortmerna-trimmomatic-STAR-Unmapped_1.fq” and “sample-sortmerna-trimmomatic-STAR-Unmapped_2.fq” and are compressed.

The log files, which contain information relating to the processing and splice-junctions, are moved into a log directory.

```
mkdir sample-sortmerna-trimmomatic-STAR_logs
```

```
mv sample-sortmerna-trimmomatic-STARLog.* sample-sortmerna-trimmomatic-SJ.* sample-
sortmerna-trimmomatic-STAR_logs
```

Among the log files, “sample-sortmerna-trimmomatic-STARLog.final.out” and “sample-sortmerna-trimmomatic-STARLog.out.tab” are of particular interest. The first details the alignment rate, percentage of uniquely/multiple aligning reads, rate of mismatches, INDELS identified in the reads, *etc.* The second file describes, in a tabular format, all the EEJs identified by STAR and whether these exist in the provided gff3 file or if they are novel. This is an extremely useful resource that can be used to identify possible new transcript splice variants. One need to keep in mind that transcription, as all biological processes, is a stochastic process and as such, there will be mis-spliced transcripts present at a low frequency in any RNA-Seq sample that has been sequenced to adequate depth. Hence novel identified junctions might represent low-frequency genuine transcription as well as noise.

Note: Among the metrics reported by STAR, one is often misunderstood: “% of reads unmapped: too short” simply means that the read could not be aligned to the genome given the selected parameters, *i.e.* “too short” means that no long enough MMP could be found in the genome.

Note: the newest STAR version (as of version 2.3.1z5, from May 30th 2014) can now output the alignments directly in BAM format.

b) Alignment to the transcriptome

This requires access to, or generation of, a transcript assembly. Although not the focus of this protocol, we detail briefly the process as it does influence downstream data processing choices and hence would impact the later stages of analysis detailed in this protocol.

Numerous tools are available to perform transcript assemblies, among which Trinity (Haas *et al*, 2013) is very popular. To construct a transcriptome from a set of raw FASTQ files in Trinity, we follow their well-detailed protocol at <http://trinityrnaseq.sourceforge.net>, by first *in*

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

silico normalizing the reads to reduce data redundancy (since RNA-Seq data has a very large dynamic range, reads from highly transcribed genes will be massively over-represented and conversely low expressed genes will have very low read coverage). We then use Trinity to reconstruct the transcriptome. Finally, using additional tools (such as Trinotate), we annotate the assembled transcripts as comprehensively as possible. An important additional step that we perform, which is not detailed in the Trinity guidelines, is to BLAST the obtained protein sequences (translated from the assembled transcripts) to the UniRef90 database and to infer from the best hit the likely taxon of origin of every transcript. This is essential as, often, biological material will be “contaminated” by species other than the target organism; e.g. it is common in plant material to also observe the presence of fungal transcripts, as plants are host to a large variety of endophytic fungi.

Having refined the transcriptome, “bowtie” is commonly used for aligning the reads used during assembly back to the assembled transcripts. As previously for the genome, we first need to create an index. This is done using the bowtie-build command as follows:

```
bowtie-build transcriptome.fa index/transcriptome
```

where the “transcriptome.fa” parameter specifies the path of your transcriptome FASTA file and “index/transcriptome” specifies the output directory and prefix name for the constructed index.

Once the index is constructed, reads are then aligned to the transcriptome as follows:

```
bowtie -v 3 --best --strata -S -m 100 -X 500 --chunkmbs 256 -p 8 index/transcriptome -1  
<(gzip -dc read-sortmerna-trimmomatic_1.fq.gz) -2 <(gzip -dc read-sortmerna-  
trimmomatic_2.fq.gz) | samtools view -F 0xC -bS - | samtools sort -n - sample-sortmerna-  
trimmomatic-bowtie-namesorted
```

where we specify that reads must be aligned end-to-end (*i.e.* a global alignment) with at most three mismatches (-v 3) assuming a maximal library insert size of 500 bp (the insert size is the fragment size minus twice the read length), and that all valid alignments in the best strata if there are no more than a 100 (-m 100 --best --strata) should be reported in SAM format (-S). The remaining options are for performance enhancement. Note that from now on, as the reads have been combined into a single result file, we refer to our exemplary data as “sample”.

The alignments reported in SAM format are then directly ‘piped’ into the samtools utility to keep only properly paired reads (0xC) and are then converted into BAM format before being sorted by read ID (sort -n option) and saved in the “sample-sortmerna-trimmomatic-bowtie-namesorted.bam” file. The rationale of sorting the reads by names, rather than by position, as was done previously for the genome, is that this sorting is expected by the MMSeg and mmdiff analysis tools we use downstream (Turro *et al*, 2011, Turro *et al*, 2014, respectively).

Note: the reason we allow multi-mapping (allowing a read to have multiple reported valid alignments) even though an individual read can only have originated from a single mRNA

fragment is that we plan to use tools in the downstream analyses that are able to estimate splice variant expression.

Note: The parameters given here are - once again - optimized for Illumina PE 101bp data sequenced on a HiSeq 2500 sequencer.

Note: Sequencing depth influences the number of splice variants reconstructed / observed and the relevance of lowly expressed transcripts is difficult to assess, *i.e.* they could be genuine low expressed transcripts but could alternatively represent pre-mature mRNA or even RNA PolIII / spliceosome stochastic errors.

Note: Whenever possible, alignment to a genome should be performed in preference to transcriptome alignment. This is primarily because alignment rates will be increased and the proportion of incorrectly aligned reads decreased. No transcriptome can be assumed to be complete (conversely no genome either, as there may be differences along individuals) and aligning to the genome will ensure that reads are mapped to unknown exons, revealing novel splice variants. Moreover, mRNA-Seq additionally assays other transcribed sequences such as intronic sequences (*e.g.* from intron retention or sequencing of pre-mature mRNA) or eRNAs (enhancer RNAs). When performing transcriptome-based alignment it is possible that reads that would be perfectly aligned to their correct origin to the genome will be incorrectly and imperfectly aligned to a position in the transcriptome simply because the correct location is not represented and the incorrect alignment selected still returns a valid alignment. Even for tools that only support alignment to the transcriptome (such as most transcript splice variant quantification methods) it may be wise to first align to the genome and to then extract the subset of reads mapping to the transcriptome, using the BEDTools suite (Quinlan & Hall, 2010), for example.

Note: There is a discussion within the community as to whether filtering and trimming the reads may be more detrimental than simply aligning the unprocessed reads to the reference. This is a valid concern, which boils down to: “make sure you have a correct understanding of what you are doing”. As we mention here, we adjust our process based on the initial QC report, and after every each filtering step we validate that the filtering has not been deleterious by performing additional QC assessments. As there is no single gold standard method for processing sequencing data, this process must be performed for every dataset analyzed. The filtering/trimming process was previously important as it would often result in recovery of a significant fraction of reads from some samples. However, as sequencers have improved (101bp PE read libraries with a read average of Phred quality 30 are now routine) and as aligners incorporate additional functions (such as hard clipping sequencing adapters and soft clipping misaligned end of reads), this is becoming less and less the case. This is one reason why these two steps may optionally be skipped depending on the initial QC results.

7. Analysis specific data pre-processing

Read alignment concludes the data pre-processing steps common to the majority of RNA-Seq based experiments. **Table 1** details the typical decrease in the number of read

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

sequences available we observe following the successive data filtering and alignment steps. The results are standardized - for clarity - to a library size of 1M reads. There are subsequently a large number of choices for performing downstream analyses for mRNA-Seq data. Probably the most common downstream analysis options are to identify differential expression between conditions or sequence variants (e.g. Single Nucleotide Polymorphisms (SNP), INDELs (INsertion/DELetion), Copy Number Variants (CNVs)). Some of these analyses, DE analysis for example, require additional data-preparation.

Table 1:

| Step | Input Data | Usable reads | Percentage of the total | Percentage removed from previous step |
|-----------------|--------------------------------|-------------------|-------------------------|---------------------------------------|
| Raw | Raw reads | 1,000,000 | 100 | 0 |
| SortMeRna | Raw reads | 970,000 - 990,000 | 97 - 99 | 1 - 3 |
| Trimomatic | Filtered reads / raw reads | 776,000 - 891,000 | 78 - 89 | 10 - 20 |
| Aligner* (STAR) | Trimmed / Filtered / raw reads | 620,800 - 801,900 | 62 - 81 | 10 - 20# |
| Analysis | Aligned reads | 620,800 - 801,900 | 62 - 81 | 0 |

* The alignment rate depends on the genome quality and completeness and can hence have a large range - the values presented here are from the Norway Spruce, a version 1 draft of the genome.

The values presented here report only uniquely aligning reads. In our example, the rate of non-aligning reads is usually equal to the rate of multi-mapping reads, *i.e.* about 10% for both in the worst cases.

This data preparation varies depending on whether expression at the gene or the transcript level is required. Both approaches are detailed below and refer to the corresponding alignment approach, to the genome or transcriptome, respectively.

a) Data preparation for a DE analyses at the gene level

A typical DE analysis data preparation consists of three steps, the first being to generate a non-redundant annotation (in the following denoted as “features”, which are e.g. protein coding genes), followed by the quantification/summation of the pre-processed reads aligned to each such feature before ultimately a last QC step is performed that assesses whether the observed effects may have biological causes. An example of such a QC is for example to ensure through a clustering or PCA approach that condition’s replicates group together, that conditions appear sufficiently separated, that no obvious confounding factor exists (e.g. sampling date, sequencing date, sequencing flow cell or lane, etc.), ... We refer to this step

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

as the “biological QC”, which opposes the more “technical” QC performed previously that inspects the raw data for technical biases due to sequencing (adapter contamination, base call quality issues, etc.).

i. Creating a non-redundant annotation

One major caveat of estimating gene expression using aligned RNA-Seq reads is that a single read, which originated from a single mRNA molecule, might sometimes align to several features (e.g. transcripts or genes) with alignments of equivalent quality. This, for example, might happen as a result of gene duplication and the presence of repetitive or common domains, for example. To avoid counting unique mRNA fragments multiple times, the stringent approach is to keep only uniquely mapping reads - being aware of potential consequences, see the note below. Not only can “multiple counting” arise from a biological reason, but also from technical artifacts, introduced mostly by poorly formatted gff3/gtf annotation files. To avoid this, it is best practice to adopt a conservative approach by collapsing all existing transcripts of a single gene locus into a “synthetic” transcript containing every exon of that gene. In the case of overlapping exons, the longest genomic interval is kept, *i.e.* an artificial exon is created. This process results in a flattened transcript – a gene structure with a one to one relationship. As this procedure varies from organism to organism, there is, to the best of our knowledge, no tool available for performing this step. The documentation of the R/Bioconductor easyRNASeq package (Delhomme, Padioleau, Furlong, & Steinmetz, 2012 - see paragraph 7.1 of the package vignette) details a way of doing this in R starting from a GTF/GFF3 annotation file. From the “genome.gff3” that was used during the alignment step, we obtain a synthetic-transcript.gff3 file.

Note: a working example of this procedure will be shortly available as part of this protocol companion tutorial.

ii. Counting reads per feature

The second step is to perform the intersection between the aligned position of reads (contained in the alignment BAM file) and the gene coordinates obtained in the previous step, *i.e.* to count the number of reads overlapping a gene. There are two primary caveats here: First the annotation collapsing process detailed above works on a gene-by-gene basis and hence is oblivious to the existence of genes that may overlap another gene encoded on the opposite strand. Second, aligners may return multiple mapping positions for a single read. In the absence of more adequate solution - see the next section on “DE analysis at the transcript level” for an example of what may be done - it is best to ignore multi-mapping reads.

A *de-facto* standard for counting is the htseq-count tool supplied as part of the HTSeq python library (Anders *et al*, 2014). This associated webpage (<http://www-huber.embl.de/users/anders/HTSeq/doc/count.html>) illustrates in greater detail the issues discussed above. For non-strand specific reads we suggest running htseq-count as follows:

```
htseq-count -f bam -r pos -m union -s no -t exon -i Parent sample-sortmerna-trimmomatic-STAR.bam synthetic-transcript.gff3 > sample-sortmerna-trimmomatic-STAR-HTSeq.txt
```

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

whereas for stranded data we advise using the following:

```
htseq-count -f bam -r pos -m intersection-nonempty -s reverse -t exon -i Parent sample-  
sortmerna-trimmomatic-STAR.bam synthetic-transcript.gff3 > sample-sortmerna-  
trimmomatic-STAR-HTSeq.txt
```

Note: the Illumina strand specific sequencing process generates reads that are the reverse complement of the mRNA template, hence the “reverse” value given to the htseq-count “-s” argument.

Note: Ignoring multi-mapping reads may introduce biases in the read counts of some genes (such as that of paralogs or of very conserved gene families), but in the context of a conservative first analysis we are of the current opinion that they are best ignored. One should of course assess how many reads are multi-mapping (check for example the STAR output) and possibly extract them from the alignment read file to visualize them using a genome browser so as to understand where they are located and how they may affect any analysis. Based on this, one may, at a later stage, decide to relax the counting parameters to accept multi-mapping reads.

iii. Processed data pre-analysis

Although this is not *per se* a data preparation step, we advise at this stage to conduct a number of analyses to assess the biological soundness of the data, such as examining how well biological replicates correlate, how the samples cluster in a principal component analysis (PCA) and whether the first dimensions of the PCA can likely be explained by the biological factors under consideration. To achieve this, it is important to have first normalized the data. When a sufficient number of replicates per condition are available (at least three) we recommend that the data be normalized using a Variance Stabilizing Transformation (VST) such as that implemented in the R/Bioconductor DESeq2 package (Love *et al*, 2014), otherwise the data should be normalized using other approaches such as those implemented in the edgeR (Robinson *et al*, 2010) or DESeq2 packages, *i.e.* approaches assuming a negative binomial distribution of the data.

Note: a working example of this procedure will be shortly available as part of this protocol companion tutorial.

b) Data preparation for a DE analyses at the transcript level

To quantify transcript splice variant expression, we currently use the MMSeg tool, which is well documented at <https://github.com/eturro/mmseq>. Briefly, the procedure is as follows:

i. Counting reads per feature

This is performed using the bam2hits command:

```
bam2hits transcriptome.fa sample-sortmerna-trimmomatic-bowtie-namesorted.bam >  
sample-sortmerna-trimmomatic-bowtie-namesorted.hits
```

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

where the specified parameters are the same as those used in paragraph 6b

ii. Obtaining the expression estimates

This is performed using the mmseq utility for every sample

```
mmseq sample-sortmerna-trimmomatic-bowtie-namesorted.hits sample-sortmerna-trimmomatic-bowtie-namesorted
```

The most relevant output files being the “sample-sortmerna-trimmomatic-bowtie-namesorted.mmseq” ones that contain expression estimates for every transcript (the log_μ column of this tab separated file).

8. Downstream analyses

In the following, we only briefly introduce DE and SNP/INDELs analyses. As introduced at the mapping stage, we differentiate the DE analysis conducted at the gene level from those conducted at the transcript level. The rationale is that the assumptions that can be made from the data are different. In the case of the gene level analysis, the initial data (the count table) consists of discrete values (integer count values) whereas the data obtained from a transcript level analysis are continuous expression estimates.

a) Calling Variants

There is a *de-facto* established standard, namely the Genome Analysis Toolkit (GATK, McKenna *et al*, 2010) pipeline from the Broad Institute, which comes with extensive documentation on how to perform such analysis. We very briefly introduce these pipeline steps below, while referring the reader to the GATK online documentation, where the GATK developers recently published a best practice workflow for calling variants from RNA-Seq data (<https://www.broadinstitute.org/gatk/guide/article?id=3891>).

The alignments obtained at the previous step (#6) can directly be utilized for variant (SNPs and INDELs) calling using the GATK workflow that includes marking duplicate reads (with Picard tools; <http://broadinstitute.github.io/picard/>), splitting and trimming reads based on the CIGAR strings, realigning around INDELs, performing base quality score recalibration, calling variant and ultimately filtering the variants to generate a VCF (Variant Call Format) file.

Note: The best practice document for RNA-Seq is in an early stage of development. The authors currently suggest performing a 2-pass alignment of the reads where the splice junctions detected by the first pass alignment guide the second pass alignment. This implies the generation of a new refined genome sequence (section 6a) for each sample, which may be computationally expensive depending on your reference assembly.

b) Differential Expression (DE) analysis at the gene level

Based on the comparative analyses of DE tools presented in Sonesson & Delorenzi (2013), we recommend using DESeq (Anders & Huber, 2010) as a first conservative approach. More specifically, we would suggest using the DESeq2 implementation, although this was not

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

included in the aforementioned manuscript. When the number of replicates is between three and five per conditions, we use the standard DESeq2 approach based on the negative binomial distribution. When there are more replicates per conditions, or if a large number of sample in total offset the lack of replication per condition (e.g. 6 conditions with 3 replicates each), we prefer to use a VST and linear model approach by using e.g. the voom (Law *et al*, 2014) + limma R/Bioconductor packages (Smyth, 2005), or the DESeq2 VST implementation + limma. (**see reviewer comment 1**)

c) Differential Expression analysis at the transcript level

From the data preparation (see point 7b), we have now obtained normalized expression estimates that are continuous (and not discrete counts as for the above gene level based approach), using MMSeq. The corresponding companion tool for performing differential expression analysis is mmdiff, a tool that was developed with corresponding assumptions. We refer you to the detailed tool documentation for further details (<https://github.com/eturro/mmseq>).

Note: If you would prefer a DESeq2/edgeR approach to the DE analysis based on the MMSeq results, you can find guidelines at <https://github.com/eturro/mmseq/blob/master/doc/countsDE.md>.

Note: There are obviously numerous alternatives to the MMSeq/mmdiff approach for transcript splice variant expression estimation. We have selected mmseq after having compared it internally with other solutions using various criteria, not only specificity and sensitivity, but also ease of use. Additional online comments from prominent community experts (such as Trinity's author Brian Haas) have further influenced our selection of this tool.

Note: In the early days of RNA-Seq, the observation that technical replicates yielded identical sequencing results was often abused as a justification that replication was not necessary. This is evidently a fallacy, as with any experiment for which the results are to be statistically assessed, at least three biological replicates are required.

Note: Some readers may have noted that we make no reference to RPKM/FPKM. This is not an oversight as we and others (Dillies *et al*, 2013; Sonesson & Delorenzi, 2013) have shown that it is not the most accurate normalization approach for DE analysis. See also an explanation of this by Dr. Lior Pachter as given during a lecture at the Cold Spring Harbor Laboratory (https://www.youtube.com/watch?feature=player_embedded&v=5NiFibnbE8o, ~31 minutes in).

Acknowledgments

The authors would like to thank Kristian Persson Hodén, James Kolpack and David Weston for work on the UPSCb-public pipeline. We thank Titus C. Brown, Brian J. Haas and Manfred Grabherr for insight into assembly, Alexander Dobin for first class support with STAR, Ernest Turro and Ângela Gonçalves for insight into splice variants expression estimation, Simon Anders and Wolfgang Huber for insight into differential expression analysis and Martin

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

Morgan and the bioconductor core team for their endless efforts integrating NGS data manipulation and analyses in R. Finally, we would like to thank Michael Weber for its invitation to write up this protocol, Mike Love for his comments and both for their time.

References

- Anders S & Huber W (2010) Differential expression analysis for sequence count data. *Genome Biol.* **11**: R106 Available at: <http://genomebiology.com/2010/11/10/R106> [Accessed September 20, 2013]
- Anders S, Pyl PT & Huber W (2014) HTSeq A Python framework to work with high-throughput sequencing data Cold Spring Harbor Labs Journals Available at: <http://biorxiv.org/content/early/2014/02/20/002824.abstract> [Accessed February 21, 2014]
- Bolger AM, Lohse M & Usadel B (2014) Trimmomatic: a flexible trimmer for Illumina sequence data. *Bioinformatics* **30**: btu170– Available at: <http://bioinformatics.oxfordjournals.org/content/early/2014/04/27/bioinformatics.btu170> [Accessed July 9, 2014]
- Burrows M & Wheeler DJ (1994) A block-sorting lossless data compression algorithm. Technical Report 124, Digital Equipment Corporation.
- Cock PJA, Fields CJ, Goto N, Heuer ML & Rice PM (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. *Nucleic Acids Res.* **38**: 1767–71 Available at: <http://nar.oxfordjournals.org/content/38/6/1767> [Accessed July 10, 2014]
- Delhomme N, Padiou I, Furlong EE & Steinmetz LM (2012) easyRNASeq: a bioconductor package for processing RNA-Seq data. *Bioinformatics* **28**: 2532–3 Available at: <http://bioinformatics.oxfordjournals.org/content/28/19/2532.abstract> [Accessed July 16, 2014]
- Dillies M-A, Rau A, Aubert J, Hennequet-Antier C, Jeanmougin M, Servant N, Keime C, Marot G, Castel D, Estelle J, Guernec G, Jagla B, Jouneau L, Laloë D, Le Gall C, Schaëffer B, Le Crom S, Guedj M & Jaffrézic F (2013) A comprehensive evaluation of normalization methods for Illumina high-throughput RNA sequencing data analysis. *Brief. Bioinform.* **14**: 671–83 Available at: <http://bib.oxfordjournals.org/content/14/6/671.full> [Accessed February 19, 2014]
- Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M & Gingeras TR (2013) STAR: ultrafast universal RNA-seq aligner. *Bioinformatics* **29**: 15–21 Available at: <http://bioinformatics.oxfordjournals.org/content/29/1/15> [Accessed January 21, 2014]
- Haas BJ, Papanicolaou A, Yassour M, Grabherr M, Blood PD, Bowden J, Couger MB, Eccles D, Li B, Lieber M, MacManes MD, Ott M, Orvis J, Pochet N, Strozzi F, Weeks N, Westerman R, William T, Dewey CN, Henschel R, et al (2013) De novo transcript sequence reconstruction from RNA-seq using the Trinity platform for reference generation and analysis.

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

Nat. Protoc. **8**: 1494–1512 Available at: <http://dx.doi.org/10.1038/nprot.2013.084> [Accessed July 11, 2013]

Khan Z, Bloom JS, Kruglyak L & Singh M (2009) A practical algorithm for finding maximal exact matches in large sequence datasets using sparse suffix arrays. *Bioinformatics* **25**: 1609–16 Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2732316&tool=pmcentrez&rendertype=abstract> [Accessed December 12, 2014]

Kopylova E, Noé L & Touzet H (2012) SortMeRNA: fast and accurate filtering of ribosomal RNAs in metatranscriptomic data. *Bioinformatics* **28**: 3211–7 Available at: <http://bioinformatics.oxfordjournals.org/content/28/24/3211.abstract> [Accessed February 13, 2014]

Langmead B, Trapnell C, Pop M & Salzberg SL (2009) Ultrafast and memory-efficient alignment of short DNA sequences to the human genome. *Genome Biol.* **10**: R25 Available at: <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2690996/> [Accessed January 28, 2013]

Law CW, Chen Y, Shi W & Smyth GK (2014) Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. *Genome Biol.* **15**: R29 Available at: <http://genomebiology.com/2014/15/2/R29> [Accessed February 3, 2014]

Lefrançois P, Euskirchen GM, Auerbach RK, Rozowsky J, Gibson T, Yellman CM, Gerstein M & Snyder M (2009) Efficient yeast ChIP-Seq using multiplex short-read DNA sequencing. *BMC Genomics* **10**: 37 Available at: <http://www.biomedcentral.com/1471-2164/10/37> [Accessed August 17, 2014]

Li H & Durbin R (2009) Fast and accurate short read alignment with Burrows-Wheeler transform. *Bioinformatics* **25**: 1754–1760 Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2705234&tool=pmcentrez&rendertype=abstract> [Accessed July 9, 2014]

Li H & Durbin R (2010) Fast and accurate long-read alignment with Burrows–Wheeler transform. *Bioinformatics* **26**: 589–595 Available at: <http://bioinformatics.oxfordjournals.org/content/26/5/589.abstract>

Li H, Handsaker B, Wysoker A, Fennell T, Ruan J, Homer N, Marth G, Abecasis G, Durbin R & Subgroup 1000 Genome Project Data Processing (2009) The Sequence Alignment/Map format and SAMtools. *Bioinformatics* **25**: 2078–2079 Available at: <http://www.pubmedcentral.nih.gov/articlerender.fcgi?artid=2723002&tool=pmcentrez&rendertype=abstract> [Accessed July 9, 2014]

Li H & Homer N (2010) A survey of sequence alignment algorithms for next-generation sequencing. *Brief. Bioinform.* **11**: 473–483 Available at: <http://bib.oxfordjournals.org/content/11/5/473.abstract> [Accessed July 9, 2014]

Visit <http://www.epigenesys.eu> for other epigenetics and systems biology protocols

- Love MI, Huber W & Anders S (2014) Moderated estimation of fold change and dispersion for RNA-Seq data with DESeq2. *bioRxiv*: DOI: 10.1101/002832 Available at: <http://biorxiv.org/content/early/2014/05/27/002832.abstract> [Accessed July 9, 2014]
- McKenna A, Hanna M, Banks E, Sivachenko A, Cibulskis K, Kernytsky A, Garimella K, Altshuler D, Gabriel S, Daly M & DePristo MA (2010) The Genome Analysis Toolkit: a MapReduce framework for analyzing next-generation DNA sequencing data. *Genome Res.* **20**: 1297–1303 Available at: <http://genome.cshlp.org/content/early/2010/08/04/gr.107524.110.abstract> [Accessed February 28, 2013]
- Morgan M, Anders S, Lawrence M, Aboyoun P, Pagès H & Gentleman R (2009) ShortRead: a bioconductor package for input, quality assessment and exploration of high-throughput sequence data. *Bioinformatics* **25**: 2607–8 Available at: <http://bioinformatics.oxfordjournals.org/content/25/19/2607> [Accessed July 16, 2014]
- Mortazavi A, Williams BA, McCue K, Schaeffer L & Wold B (2008) Mapping and quantifying mammalian transcriptomes by RNA-Seq. *Nat. Methods* **5**: 621–628 Available at: <http://dx.doi.org/10.1038/nmeth.1226> [Accessed July 11, 2014]
- Quinlan AR & Hall IM (2010) BEDTools: a flexible suite of utilities for comparing genomic features. *Bioinformatics* **26**: 841–842 Available at: <http://bioinformatics.oxfordjournals.org/content/26/6/841.short> [Accessed July 9, 2014]
- Robinson MD, McCarthy DJ & Smyth GK (2010) edgeR: a Bioconductor package for differential expression analysis of digital gene expression data. *Bioinformatics* **26**: 139–140 Available at: <http://bioinformatics.oxfordjournals.org/content/btp616v1/.abstract>
- Smyth G (2005) limma: Linear Models for Microarray Data. In *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, Gentleman R Carey VJ Huber W Irizarry RA & Dudoit S (eds) pp 397 – 420. New York: Springer-Verlag Available at: <http://www.citeulike.org/group/1654/article/1419586> [Accessed September 18, 2014]
- Soneson C & Delorenzi M (2013) A comparison of methods for differential expression analysis of RNA-seq data. *BMC Bioinformatics* **14**: 91 Available at: <http://www.biomedcentral.com/1471-2105/14/91> [Accessed March 10, 2013]
- Turro E, Astle WJ & Tavaré S (2014) Flexible analysis of RNA-seq data using mixed effects models. *Bioinformatics* **30**: 180–8 Available at: <http://www.ncbi.nlm.nih.gov/pubmed/24281695> [Accessed September 28, 2014]
- Turro E, Su S-YY, Gonçalves Â, Coin LJM, Richardson S, Lewin A & Goncalves A (2011) Haplotype and isoform specific expression estimation using multi-mapping RNA-seq reads. *Genome Biol.* **12**: R13 Available at: <http://genomebiology.com/2011/12/2/R13> [Accessed September 15, 2014]

Reviewer comments:

Reviewed by Michael Love, Department of Biostatistics, Harvard School of Public Health, Boston, USA.

(Comment 1) In the DESeq and DESeq2 documentation, we do not recommend a VST followed by linear modeling, as a VST only flattens the variance across the range of mean, but does not directly inform the subsequent model of the precision of log counts. I would recommend using one of DESeq2, edgeR or voom + limma for differential expression.

The other comments have been directly answered by the authors in the text.