RAPID Supplemental Material

Release 1.0

Sivarajan Karunanithi, Martin Simon, Marcel Schulz

Feb 18, 2019

Contents

Understanding the role of small RNA (sRNA) in diverse biological processes requires detailed attention to strand specificity, length distribution, and nucleotide soft-clipping. No integrated computational solution exists to investigate novel sRNA data in an unbiased way. We developed a generic eukaryotic sRNA analysis tool which captures information inherent in the dataset and automatically produces numerous visualizations, as user-friendly HTML reports, covering multiple categories required for sRNA analysis. Our tool also facilitates an automated comparison of multiple datasets, with different normalization techniques. For ease of use, our tool also integrates an automated differential expression analysis using DESeq2.

While our tool can be used for generic sRNA analysis, they are tailored to address the needs of eukaryotic siRNA analysis.

CHAPTER 1

Features of RAPID

Read Alignment, Analysis and Differential Pipeline (RAPID) is a set of tools for the alignment, and analysis of genomic regions with small RNA clusters derived from small RNA sequencing data. RAPID currently consists of four modules, whose functionalities are described below.

- **rapidStats**: Prime module which performs alignment, calculates the basic statistics and writes them to a file
- **rapidNorm**: Normalizes the statistics of given samples and writes the normalized values to a file, enabling us to compare genes/samples, with one normalization technique (KnockDown Corrected Scaling; KDCS) dedicated for siRNA knockdown analysis.
- **rapidVis**: Generates insightful graphs of the basic statistics and comparison
- **rapidDiff**: Differential analysis of the given samples using DESeq2

CHAPTER 2

Installation

We strongly recommend using the latest version of our tool, using a conda environment, as this sorts out all the dependency issues.

2.1 Conda

RAPID is available as a recipe in the bioconda channel. If bioconda is not in your channel list (You can see it, with the command "conda info"), you can add it

```
conda config --add channels bioconda
conda config --add channels conda-forge
```

You can search for rapid using the following command:

```
conda search rapid
```

An example command to use RAPID as a conda environment :

```
conda create --name <environment_name> rapid
```

This command creates an environment for RAPID's latest version. We advise to use conda environment based approach, as this would not disturb your existing installations, and use only the compatible versions of dependencies.

If you wish to test the installation, download the testData folder from the git repository RAPID.

Please refer to **TroubleShooting_FAQs** section, if you encounter issues.

First activate the desired conda environment

```
source activate <environment_name>
```

Move inside the testData folder, and Now, simply run

```
bash runTest.sh
```

Upon successful completion, there should be two folders TestRapid created by **rapidStats**, and TestCompare from **rapidNorm** in the testData folder. You should see outputs as shown in the SampleOutput folder. The output descriptions can be found in the **Visualization** section.

If you encounter any issues, which is not addressed in the **TroubleShooting_FAQs** section, please report to us.

To move out of the environment, type

```
source deactivate <environment_name>
```

2.2 Manual

RAPID does not require any compilation.

- You need to download (or clone) the git repository RAPID.
- Extract the RAPID/bin/ files in your preferred location.
- Ensure to give execute permissions to all files in RAPID/bin/ and then add the installed location to PATH variable.

RAPID makes use of the following tools, and requires them to be in your PATH.

- Bedtools2
- Bowtie2 (version 2.1.0 or higher)
- R version 3.2 or higher
- Samtools (version 0.1.19 or higher)
- pandoc
- **R Packages required:**
  - DESeq2, gplots and RColorBrewer (if you are using rapidDiff)
  - ggplot2, scales, rmarkdown, knitr, viridis

2.3 Simple Test

After installation you can try running RAPID using the provided script runTest.sh (You will have to uncomment respective lines) in the testData folder. Ensure to add the rapid in the PATH variable or provide the scripts with an environment variable for rapid. Now, simply run

```
bash runTest.sh
```

Upon successful completion, there should be two folders TestRapid created by **rapidStats**, and TestCompare from **rapidNorm** in the testData folder. You should see outputs as shown in the SampleOutput folder. The output descriptions can be found in the **Visualization** section.

CHAPTER 3

Basic Usage

3.1 rapidStats

Basic statistics calculation like analyzing read counts, distribution of reads on the two DNA strands and listing small-RNA modifications stratified by the defined regions are done using this script.

3.1.1 *Input*

- Trimmed sequence file (FASTQ) or an alignment file (BAM/SAM)
- BED file containing the localization and names of genes/regions to be quantified

We generate the alignments with bowtie2, if FASTQ files are provided as input. A two step alignment can also be performed, if necessary. i.e. First, to remove the sequences aligning to contaminants, and then aligning the rest of the sequences against the reference genome. To facilitate these alignments, bowtie2 index files should be provided against the respective input parameters along with the FASTQ file. We then subject the aligned files to quantify the read counts for the regions provided in the BED file. This quantification step provides an output file containing the read counts of various read lengths, modification, strandedness, etc.

3.1.2 *Sample script*

If using a previously aligned BAM file:

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.bam -ft=BAM --remove=no -
↪a=file.bed -r=/rapidPath/
```

If using a fastq file, and wish to quantify multiple BED files. Results will be stored in separate folders with each annotation file's name:

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.fq -a=file.bed,file2.bed -i=/
↪path_to_index -r=/rapidPath/
```

If using a fastq file, and wish to perform a two-step alignment:

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.fq -a=file.bed -i=/path_to_
↪index --contamin=yes --indexco=/path_to_contaminants_index -r=/rapidPath/
```

The different parameters we provide currently are listed below.

short	long params	explanation
-h	–help	show the help on screen
-o	–out	path to the output directory, directory will be created if non-existent
-f	–file	path to the read fastq/BAM/SAM file
-ft	–filetype	BAM/SAM/fq : Mention either BAM/SAM or FASTQ. Default FASTQ
-a	–annot	bed file with regions that should be annotated with read alignments (Multiple Bed files should be separated by commas)
-r	–rapid	set location of the rapid installation bin folder (e.g. /home/software/RAPID/bin/) if not in PATH
-i	–index	set location of the bowtie2 index for alignment
-p	–proc	An INTEGER for number of processors; for bowtie's use (default: 4)
-m	–multi	An INTEGER for number of alignments to report. '-k' param of bowtie2 (default: 100)
	–contamin=yes	use a double alignment step first aligning to a contamination file (default no)
	–indexco	set location of the contamination bowtie2 index for alignment (only with contamin=yes)
	–remove=yes	remove unecessary intermediate files (default yes)

3.1.3 *Bed file format* (Do not provide a header, its shown here only for clarity)

chromosome	start	end	geneName	type	strand (Gene Direction)
chr1	1234	1368	geneA	region	+
chr2	1234	1368	geneB	region	-
chr2	1432	1568	geneB	region	-
chr3	1234	1368	geneC	background	-

The column *type* in the Bed file says whether a gene has to be treated as background (knockdown) or not during normalizations.

3.2 rapidNorm

Normalization module aims to facilitate the comparison of genes across various samples, and vice versa. As sequencing depth differs across samples, the read counts have to be normalized. RAPID facilitates two kinds of normalization. (i) DESeq2 based, and (ii) a variant of Total Count Scaling (TCS) method to account for the knockdown associated smallRNAs inherent in sequencing. For a detailed description of the normalization strategy, please have a look at the bioarXiv.

By default, RAPID uses the modified TCS based normalization method. However, in order to provide flexibility with the choice of normalization, we have also incorporated the DESeq2 based normalization.

3.2.1 *Input*

- BED file containing the localization and names of genes/regions to be compared. Care should be taken to include only the gene/regions which were quantified in **rapidStats**
- Config file containing the location of **rapidStats** output folders

3.2.2 Sample script:

If normalizing using the TCS based normalization:

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/
```

If normalizing using the DESeq2 based normalization:

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/ -d=T
```

If normalizing using the TCS based scaling, while considering only reads of length 23bp, and 25bp:

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/ -l=23,25
```

short	long params	explanation
-h	–help	output help
-o	–out	path to the output directory, directory will be created if non-existent
-c	–conf	the config file that defines which rapidStats analysis folders should be used
-a	–annot	bed file with regions that should be used for the comparison, this must be a subset of the regions that was used for rapidStats calls
-r	–rapid	set location of the rapid installation bin folder (e.g. /home/software/RAPID/bin/) or put into PATH variable
-d	–deseq	LOGICAL value. Use only TRUE or FALSE. Set this to TRUE, if you wish to use DESeq2 based normalization. Default is FALSE, which does a total count based scaling.
-l	–re-strictlength	An INTEGER of Read Lengths to be considered. If not provided, all reads will be used. (Multiple thread lengths should be separated by commas)”

The config file is a simple **tab-delimited** file that has three columns, the path to the folder produced by **rapidStats**, the name of the experiment, and list of regions need to be corrected in TCS based normalization. Each line is one dataset that should be included in the Normalization. Later these normalized statistics can be used to make comparison plots using **rapidVis**.

3.2.3 *Config file format*

location	name	background
/Control1/	Ctrl1	none
/Control2/	Ctrl2	none
/Condition1/	Cond1	*geneA,geneB*
/Condition2/	Cond2	none

geneA,geneB - Gene names provided as background should be same as provided in the **rapidStats** *bed file*.

3.3 rapidVis

The visualization module of RAPID creates informative plots from the output of **rapidStats**, and **rapidNorm**.

3.3.1 *Input*

- Path of the output folder from **rapidStats**, and **rapidNorm**
- BED file containing the localization and names of genes/regions need to be visualized. Care should be taken to include only the gene/regions which were quantified in **rapidStats**

3.3.2 Sample script:

If you want to plot rapidStats output:

```
./rapidVis.sh -t=stats -o=/path_to_output_directory_rapidStats/ -a=regions.bed -r=<
↪$rapid>
```

If you want to plot rapidNorm output:

```
./rapidVis.sh -t=compare -o=/path_to_output_directory_rapidNorm/ -r=<$rapid>
```

short	long params	explanation
-h	–help	output help
-o	–out	outputFolder_of_rapidStats.sh or rapidNorm.sh (Where Statistics and other files are located)
-t	–type	stats OR compare - use **stats** to visualize **rapidStats** or use **compare** to visualize results of **rapidNorm**
-a	–annot	bed file with regions that should be visualised (Not required for **compare**). Caution: Include only the gene/regions which were quantified in **rapidStats**
-r	–rapid	set location of the rapid installation bin folder (e.g. /home/software/RAPID/bin/) or put into PATH variable

3.4 rapidDiff

This module of RAPID implements DESeq2 software and generate basic graphs to highlight the differentially expressed gene/region among the samples.

3.4.1 *Input*

- Path of the output folder from **rapidStats**
- Config file describing the DESeq2 analysis setup

3.4.2 Sample script:

Generic Format:

```
./rapidDiff.sh --out=complete/path/outputDirectory/ --conf=data.config
```

If a different q-value cut-off is required:

```
./rapidDiff.sh --out=complete/path/outputDirectory/ --conf=data.config --alpha=0.01
```

If only reads of length 23bp, and 25bp should be considered: :: ./rapidDiff.sh –out=complete/path/outputDirectory/ –conf=data.config –alpha=0.01 -l=23,25

short	long params	explanation
-h	–help	output help
-o	–out	path to the output directory, directory will be created if non-existent
-c	–conf	the config file that defines which rapidStats analysis folders should be used for extracting the raw counts of gene/regions analyzed
-a	–alpha	qValue (adjusted p-value) cut-off to highlight in MA-Plot. Default is 0.05
-n	–nVal	Top 'n' values to be shown as heatmap. The top 'n' values are chosen in ascending order of qValue
-r	–rapid	set location of the rapid installation bin folder (e.g. /home/software/RAPID/bin/) or put into PATH variable
-l	–re-strictlength	An INTEGER of Read Lengths to be considered (Default: All). Separate multiple values by commas.

3.4.3 *Config file format*

sampleName	location	condition
Control1	Ctrl1	untreated
Condition1	Cond1	treated

This config file is a simple **tab-delimited** file that has three columns, with the **same** headers as mentioned in the above format.

sampleName tells the name to be used in the analysis output. *location* tells the location of rapidStats analysis folders should be used for extracting the raw counts of gene/regions analyzed (**USE ONLY ABSOLUTE PATH**) *condition* tells whether the sample is *untreated* or *treated* sample. For example, Use *treated* for drug treated cancerous samples; and *untreated* for cancer samples.

CHAPTER 4

Output Description

One of the strengths of RAPID is that a number of useful files with statistics and plots are automatically created, which can be used for additional analysis.

4.1 Statistics

An output folder is created, for each annotation BED file supplied in **rapidStats** analysis, with the following files:

- Statistics.dat - A tab-separated file that contains a number of statistics for each region including read counts, number of read modifications and coverage on DNA strands
- TotalReads.dat : Lists the total number of reads mapped to the genome (given by parameter -i and excluding reads that may have mapped to the contamination file)
- Other associated files used for calculation and reporting. * alignedReads.sub.compact has the compact information of aligned reads. If intermediate files are not removed, aligned BAM files will be present.

4.2 Normalization

In each folder created by rapidNorm analysis exist the following files:

- NormalizedValues.dat - A tab-separated file that contains the actual and normalized values for each region/sample provided in the config file.
- Other associated files used for calculation and reporting.

4.3 Visualization

RapidVis output description when ran in two different modes.

- stats

FolderName.html - An automatically generated main HTML file which is an ensemble of individual gene/region's HTML files that contain different plots analyzing read counts, distribution of reads on the two DNA strands and listing smallRNA modifications stratified by the defined regions.

- compare

  FolderName.html - An automatically generated HTML file consisting of various plots like read lengths, antisense ratio, etc. in different scales, compared across all the samples.

More description about each plot can be found in UseCases.

4.4 Differential Analysis

In each folder created by rapidDiff analysis exist the following files:

- DiffExp_Statistics.csv - A CSV file containing the normal counts retrieved for each sample and the DESeq2 statistics obtained
- DiffExp_Plots.pdf - A PDF file containing MA-Plot, Heatmap of top 'n' q-values, PCA plot of the samples analysed

CHAPTER 5

Use Cases

Small RNA transcriptomics is gaining a lot of interest over the past decade. There is a lot of smallRNA (sRNA) computational analysis tools available. However, they focus mainly on predicting novel miRNAs, piRNAs, etc. and annotating them. This leads to complete ignorance of the other sRNA information inherent in the dataset. There is no integrated computational solution that can investigate novel sRNA data in an unbiased way, to the best of our knowledge. Hence, we developed a generic eukaryotic sRNA analysis offline tool, Read Alignment,Analysis, and Differential PIpeline (RAPID). RAPID quantifies the basic alignment statistics with respect to read length, strand bias, non-templated nucleotides, nucleotide content, etc. for an user-defined set of genes or regions. Once the basic statistics is performed for multiple sRNA datasets, our tools aids the user with versatile functionalities, ranging from general quantitative analysis to visual comparison of multiple sRNA datasets.

5.1 Statistics

Assume you are working on a not so well annotated organism, and out of curiosity, you have used your smallRNA transcriptomics dataset to identify novel small RNA expressing regions using any of the existing computational tools. The sRNA prediction tool, probably, would have given you just the basic read count summary of the novel sRNA regions. However, in the field of sRNA transcriptomics, there are various intricate parameters which needs to be paid attention, like read length, strand of origin, soft-clipping of bases, etc. For instance, sRNA with different read length have different downstream functions. They also show difference in function, based on the strand of origin.

rapidStats captures all this essential information, in a single command.

```
rapidStats.sh -o=/path_to_output_directory/ -f=reads.fq -a=file.bed -i=/path_to_index␣
↪-r=/rapidPath/
```

If you have a set of contaminant sequences (known from the organismal knowledge, or from your lab), you can remove the contaminants and map the rest to the genome of interest.

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.fq -a=file.bed -i=/path_to_
↪index --contamin=yes --indexco=/path_to_contaminants_index -r=/rapidPath/
```

Also, many small RNA processing tools, perform alignments and output BAM/SAM files as part of their sRNA annotation pipeline. Depending on their need, they are prone to filter out lot of alignments, reads, etc. And, it can be

meaningful to subject such alignments to quantification. You can simply use the BAM/SAM file produced from the other tool.

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.bam -ft=BAM --remove=no -
↪a=file.bed -r=/rapidPath/
```

What if, you want to quantify different set of small non-coding RNA categories, like siRNAs, and piRNAs. You can simply pass multiple BED files to the annotation parameter, and the respective outputs are stored in subfolders.

```
./rapidStats.sh -o=/path_to_output_directory/ -f=reads.bam -ft=BAM --remove=no -
↪a=file.bed -r=/rapidPath/
```

Sometimes, you may want to restrict your analysis to only certain regions of a gene. For instance, if you want to analyze only two sub-parts of a geneX (namely, positions 1500-2500, and 5000-6500 in chr1). You can simply create the bed file, as shown below.

chromosome	start	end	geneName	type	strand (Gene Direction)
chr1	1500	2500	geneX	region	+
chr1	5000	6500	geneX	region	+
chr3	1234	1368	geneC	background	-

As you can see, both regions have the same *geneName* in the above annotation table. RAPID quantifies those regions and sums up their statistics under the same name to ease up the calculations. Under the column *type*, you can notice two values; region, and background. Region is the default value you need to use, if you don't have any special conditions to be handled during normalisation as described in the section below. A detailed description of all the command-line parameters can be found under the Usage section.

5.2 Normalization

For instance, assume you are performing RNA interference (RNAi) experiments, by introducing exogenous RNA in to the system to trigger the RNAi pathway. Now the resultant sequencing run contains all the introduced exogenous RNA as well. They can add up to millions of reads in the total library size. In order to avoid them from skewing the analysis, you can mention such regions as *background* in the *type* column in the BED file. Such *background* locations are handled during normalization using **rapidNorm**. Another example of a background gene/region could be with the use of RNAi vector constructs (like L4440 in C.elegans). Due to lack of specificity (or any technical inefficiency) non-insert RNA locations will also be transcribed. When the user is aware of such locations, they can be termed as background to RAPID, such that it will be rightfully handled in the analysis.

When you have such different knockdowns, you will like to compare the samples/genes and analyze how their behavior changes in different settings. And, often before performing a different analysis with multiple replicates (which are difficult to produce in many cases), you may appreciate a simple comparison to have an idea of what is going on among the different control vs cases. As sequencing depth differs across samples, the read counts have to be normalized. In order to do that, RAPID facilitates two kinds of normalization. (i) factor-based normalisation from DESeq2 , and (ii) a variant of total count scaling method to account for the knockdown associated smallRNAs inherent in sequencing. This method is called KnockDown Corrected Scaling (KDCS).

Assume read count *R* for a region of interest that we want to compare between samples. *T* is the total number of reads mapping to the genome, and *K* is the number of small RNA reads mapping to the knockdown gene. In KDCS, we compute the normalized read count * :math: ‘hat{R}‘* :

```
:math: `\hat{R} = R \cdot \frac{M}{T-K}`,
```

where *M* is the maximum over all values *(T_1 - K_1),. . . ,(T_n - K_n)* over all *n* samples.

For a detailed description of the normalization strategy, please have a look at the bioarXiv/Manuscript.

Simply add the sample locations you want to compare as described in the config file: and run the following command with all the regions (in the annotation bed file) you previously quantified or only a subset of them which you think is interesting.

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/
```

If you think, using the DESeq2 based normalization is a better choice for your experimental setup, you may well do so.

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/ -d=T
```

Sometimes, you may need to consider only reads of certain read lengths, say, 23bp, and 25bp. Restricting the analysis to certain read length may increase the specificity of your comparative analysis. You can do that, by simply adding the lengths of your interest in the command line.

```
./rapidNorm.sh --out=/path_to_output_directory/ --conf=data.config --annot=regions.
↪bed --rapid=/rapidPath/ -l=23,25
```

For a detailed description of the normalization strategy, please have a look at the bioarXiv/Manuscript.

A detailed description of all the command-line parameters can be found under the Usage section.

5.3 Visualization

To provide a better understanding of the data, **rapidVis** module generates insightful plots from the output of previous rapid modules.

If you want to plot rapidStats output:

```
./rapidVis.sh -t=stats -o=/path_to_output_directory_rapidStats/ -a=file.bed -r=/
↪rapidPath/
```

If you want to plot rapidNorm output:

```
./rapidVis.sh -t=compare -o=/path_to_output_directory_rapidNorm/ -r=/rapidPath/
```

A detailed description of all the command-line parameters can be found under the Usage section.

5.4 Case Studies

To exemplify the use of RAPID, we performed two case studies, which are part of the bioarXiv/Manuscript. Using the Reproducible script, and associated data provided in the GitHub page of RAPID , you can perform all the analysis part of these case studies.

Below you can find few sample output from the case studies.

An example visualisation of the basic statistics produced by RAPID, for the *Paramecium tetraurelia* case study . Visualization of Statistics

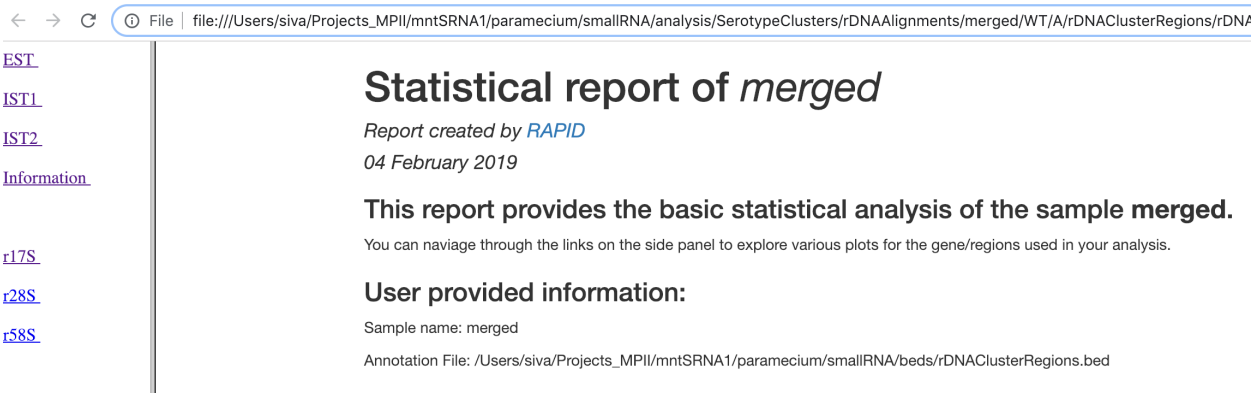
An example visualisation of the normalised comparison plots produced by RAPID, for the *Paramecium tetraurelia* case study . Visualization of Comparisons

5.5 Visualization: Statistical Report

This section describe the plots in the statistical report produced from **rapidVis**. These explanations are merely one of several possible interpretation of each type of plot, and are not conclusive evidences on sRNA mechanism of action.

To exemplify the Visualization abilities of RAPID, we used four small RNA sequencing data sets (unpublished) from the wildtype serotypes (51A, 51B, 51D, and 51H) of textit{Paramecium tetraurelia}. We analyzed only the rDNA cluster producing 17S, 5.8S, 25S ribosomal RNAs. The rDNA cluster sequence can be obtained from GenBank Accession: AF149979.1 ~citep{Preer1999DoesCircle}, with the additional annotation of the 5.8S sequence from GenBank accession: AM072801.1 ~citep{Barth2006IntraspecificSequences}.

FolderName.html - An automatically generated HTML file which is an ensemble of individual gene/region's HTML files that contain different plots analyzing read counts, distribution of reads on the two DNA strands and listing soft-clipped nucleotides stratified by the defined regions.



The left panel contains the list of regions/genes provided as part of the BED file to create the plots. Each region contains the following plots, if applicable.

5.5.1 Read alignment percentage of various read lengths

This plot shows various read lengths (x-axis) utilized in the analysis and their percentage of alignment (y-axis). Read length distribution plot is important to see if there is a predominance of certain length transcripts. As sRNA mechanisms are rather sensitive and specific, different length predominance can indicate different downstream pathways of the sRNA.

EST

IST1

IST2

Information

r17S

r28S

r58S

Statistical report of *EST*

Report created by RAPID

04 February 2019

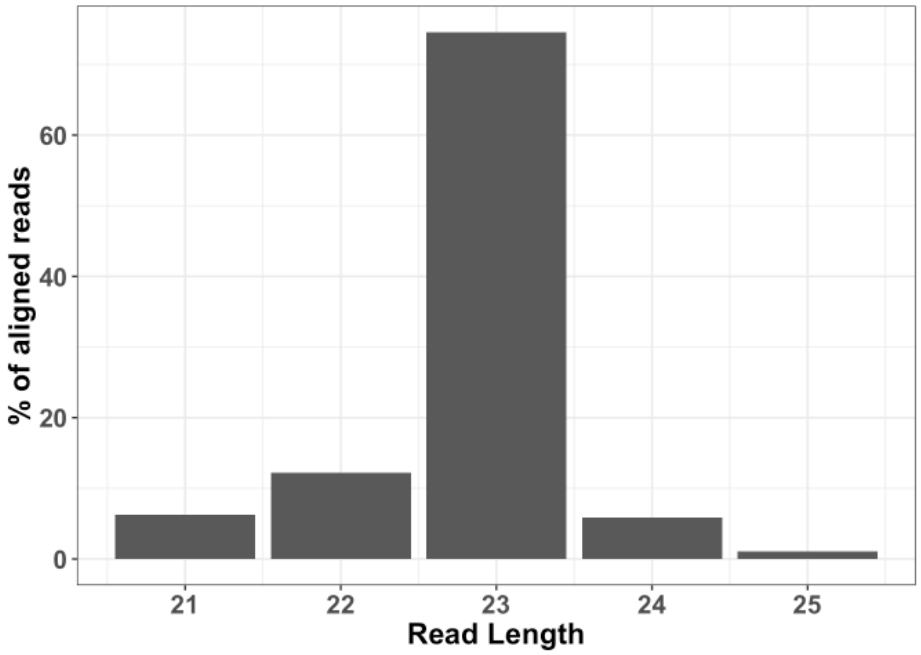
This report provides the basic statistical analysis of **EST**. It consists of various single category plots detailing on the distribution of read length, strand, and base soft-clipping. It also contains double category plots showing the combinations of aforementioned.

Total reads aligned to EST : 33507

Note: If graphs under some category are absent, it means there is not sufficient data with respect to that category.

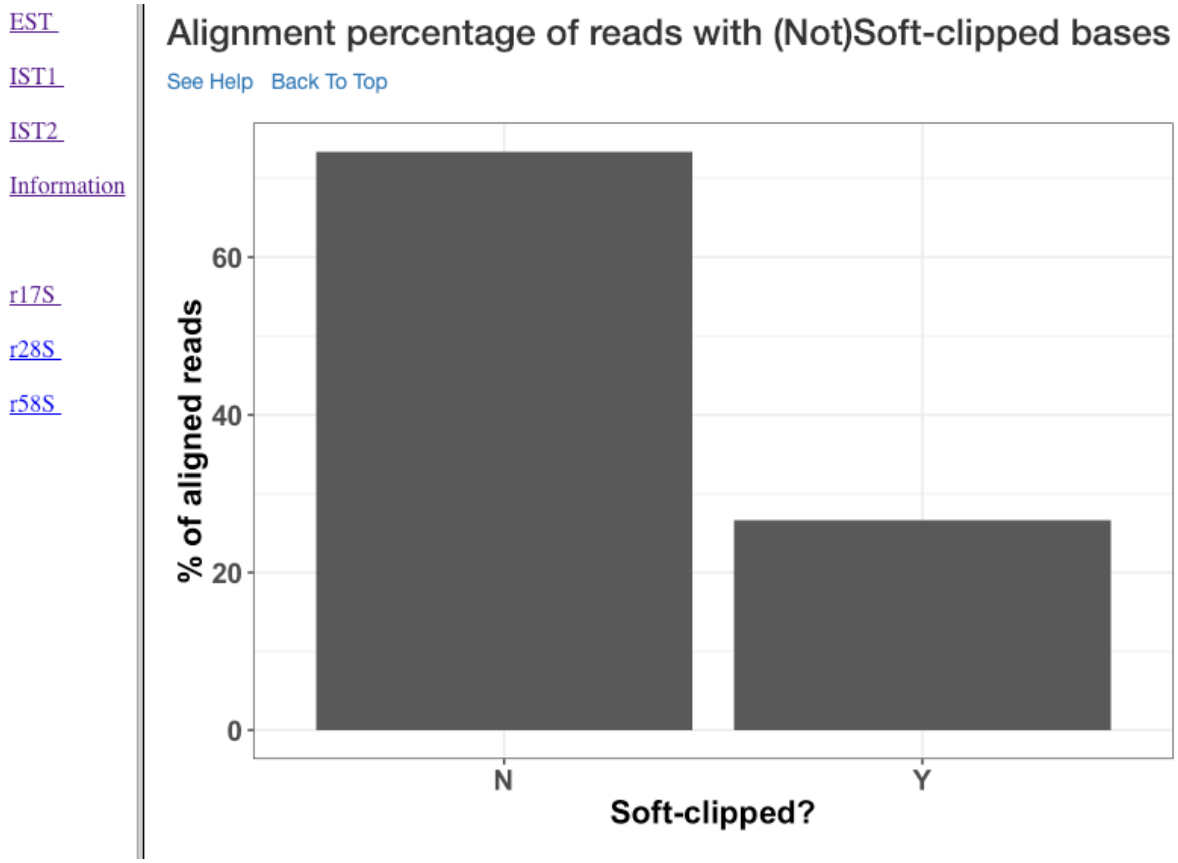
Read alignment percentage of various read lengths

See Help Back To Top



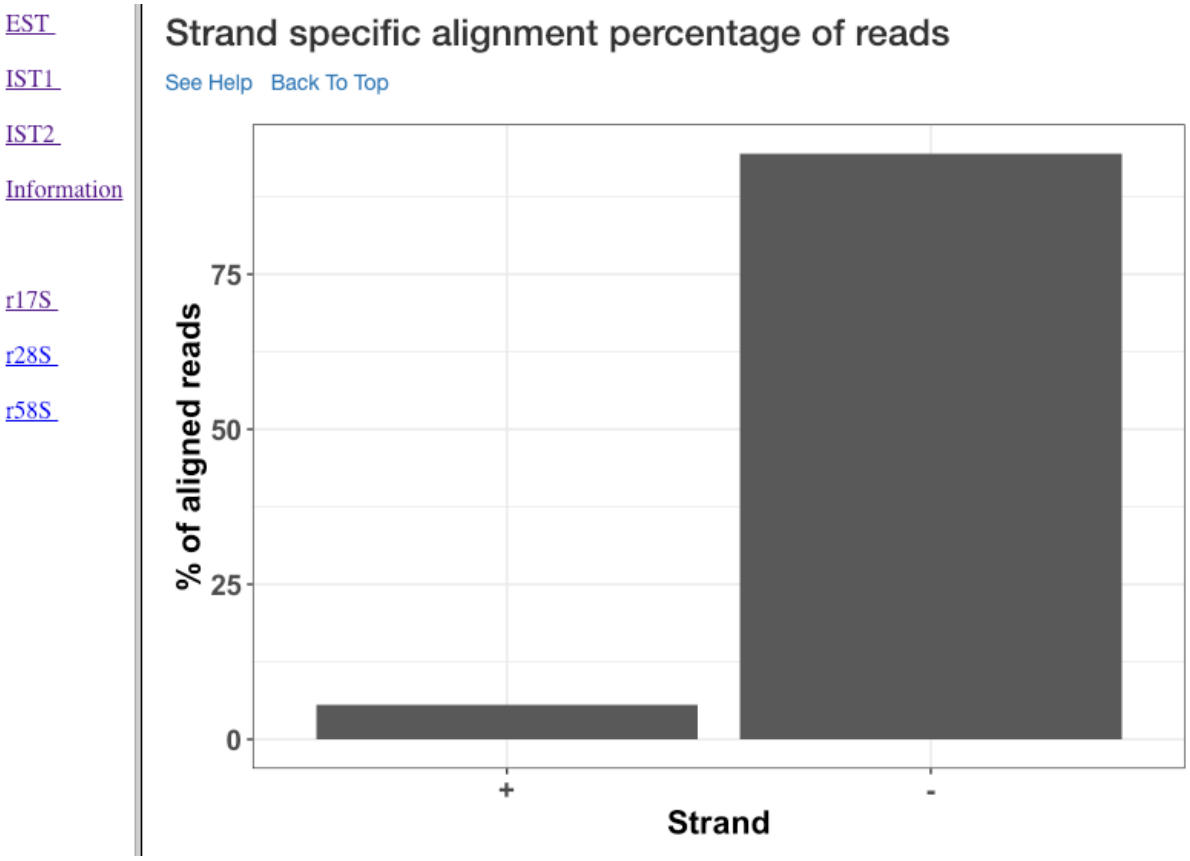
5.5.2 Alignment percentage of reads with (Not)Soft-clipped bases

This plot shows the alignment percentage of reads (y-axis) containing soft-clipped bases (x-axis; Soft-clipping status). Soft-clipping refers to the bases (five-prime or three-prime) in a read that are not part of the alignment. This helps in understanding the percentage of aligned sRNA which has non-specific alignments. In sRNA mechanisms, as it is not uncommon to exclude bases in five or three prime end to achieve base-pairing, this plot can give insights into the mode of base-pairing of the sRNAs in downstream mechanisms.



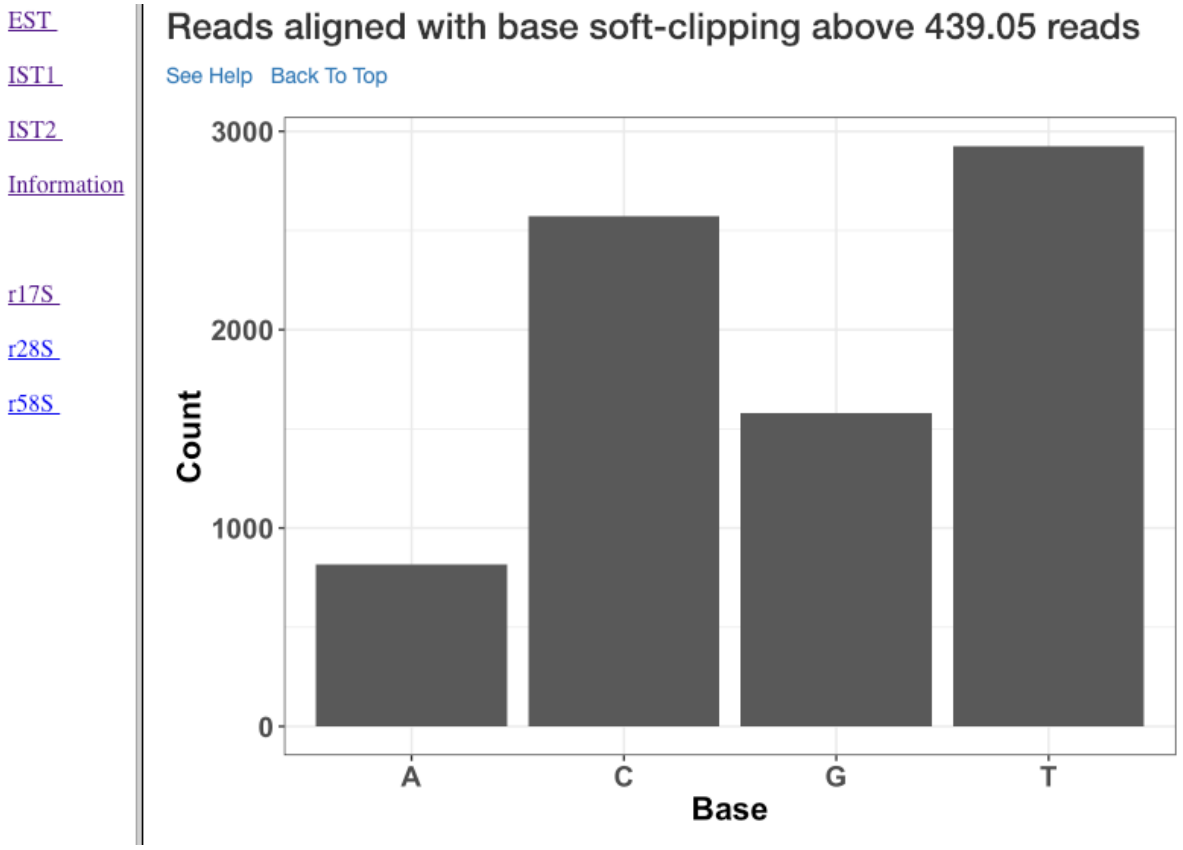
5.5.3 Strand specific alignment percentage of reads

The alignment percentage (y-axis) of reads corresponding to each strand (x-axis) is shown in this plot. sRNA mechanisms are quite specific to length, and their strand of origin. This plot helps in understanding which strand shows a predominance in the library, such that one can hypothesise the role of the analysed small RNAs. For instance, an antisense predominance could indicate a cis-acting mechanism in modulating the target mRNA.



5.5.4 Reads aligned with soft clipping above 'n' reads

This plot shows the soft-clipped bases (x-axis) and the number of reads (y-axis) containing such soft-clipping. We only show bases which have at least 'n' reads; where, 'n' corresponds to 5% of the overall alignment. This plot can help in understanding, if any particular nucleotide is always soft-clipped. It could simply indicate a potential technical inadequacy in trimming adapter, or primer, etc. Also, depending on the organism, biological mechanisms, such as poly-A tailing of small RNAs, may exist. This plot thus gives a chance for the user to see if there is a RNA modification pathway existing, or whether change in conditions are affecting RNA modification frequencies.



5.5.5 Alignment percentage of reads with soft clipping above 'n' reads

This plot (similar to the previous plot) shows the soft-clipped bases (x-axis) and the percentage of reads (y-axis) containing such soft-clipping. We only show bases which have at least 'n' reads; where, 'n' corresponds to 5% of the overall alignment.

EST

IST1

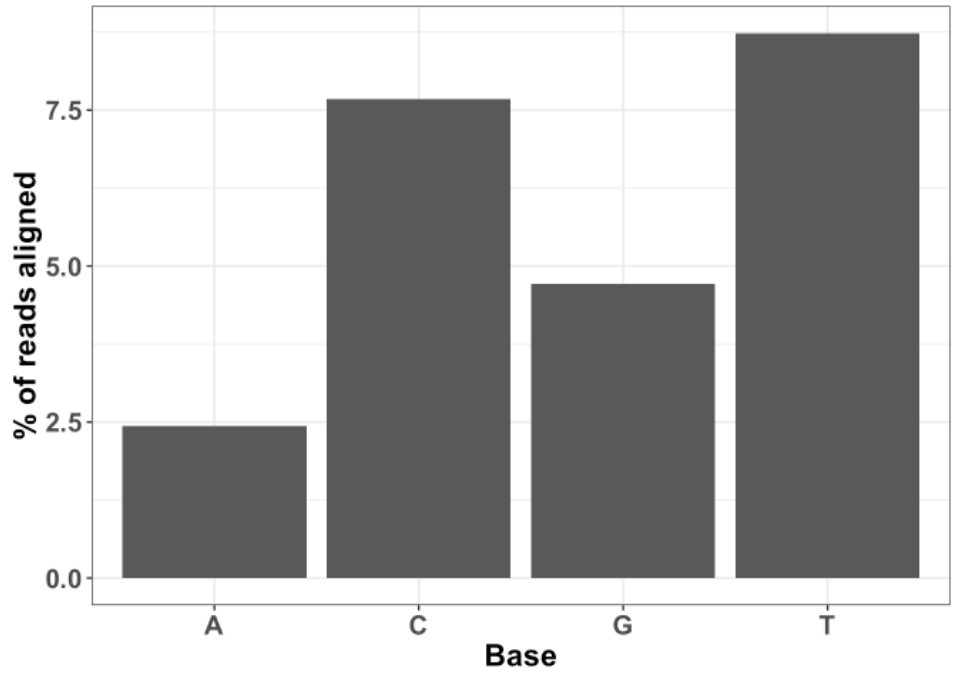
IST2

Information

r17S

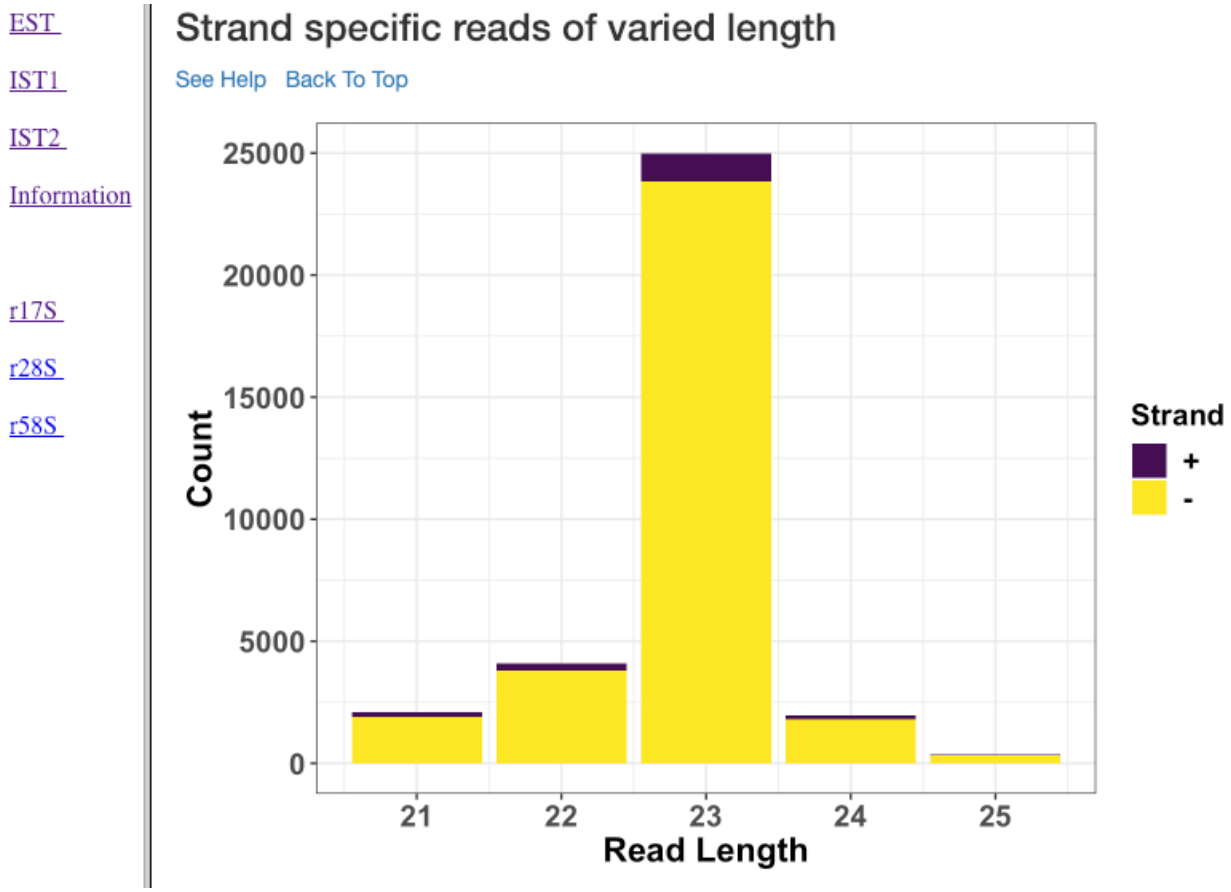
r28S

r58S



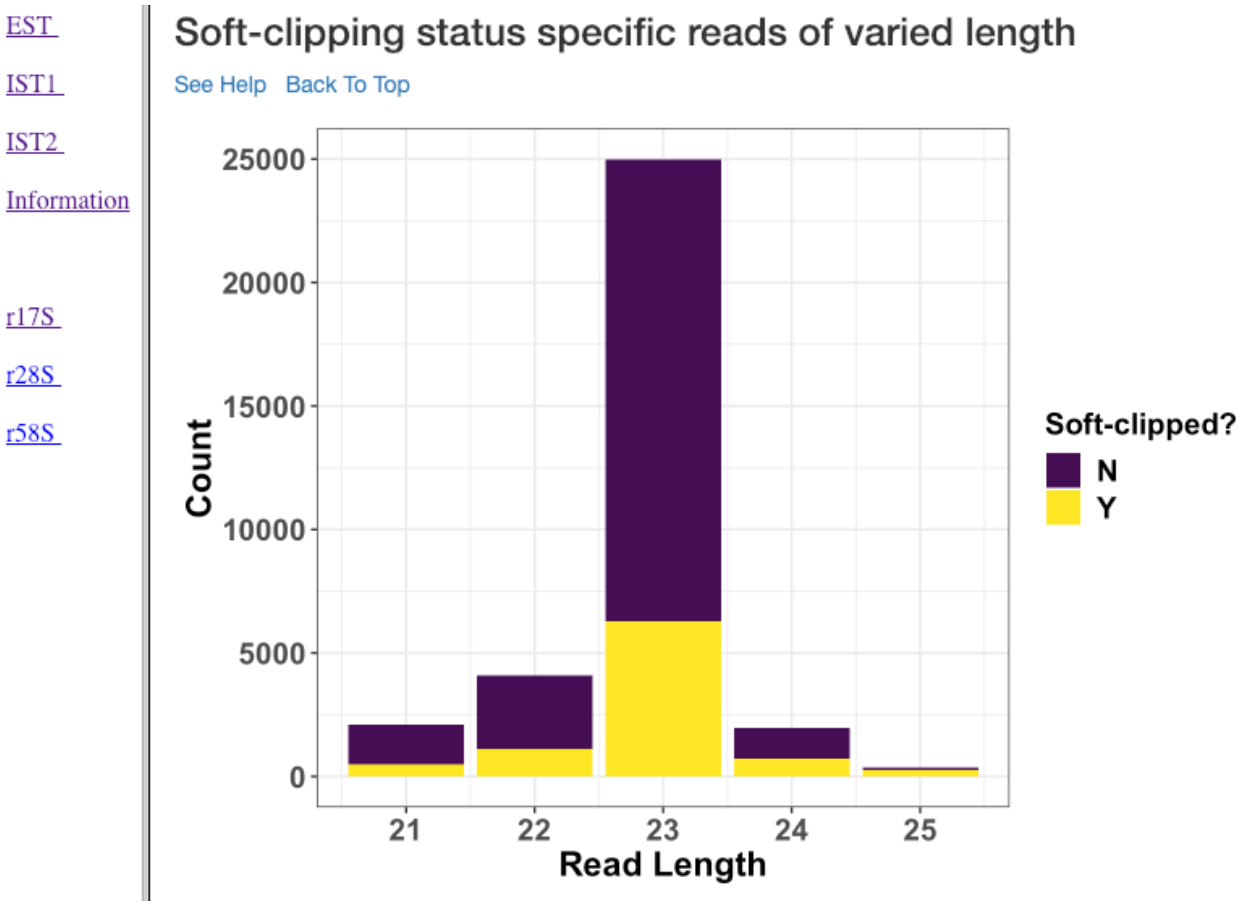
5.5.6 Strand specific reads of varied length

This plot shows various read lengths (x-axis) utilized in the analysis and their read counts (y-axis), specific to each strand. Length, and strand of origin plays an important role in understanding sRNA mechanisms. For instance, an antisense predominance of 23nt bases could indicate a cis-acting mechanism in modulating the target mRNA, and dicer activity.



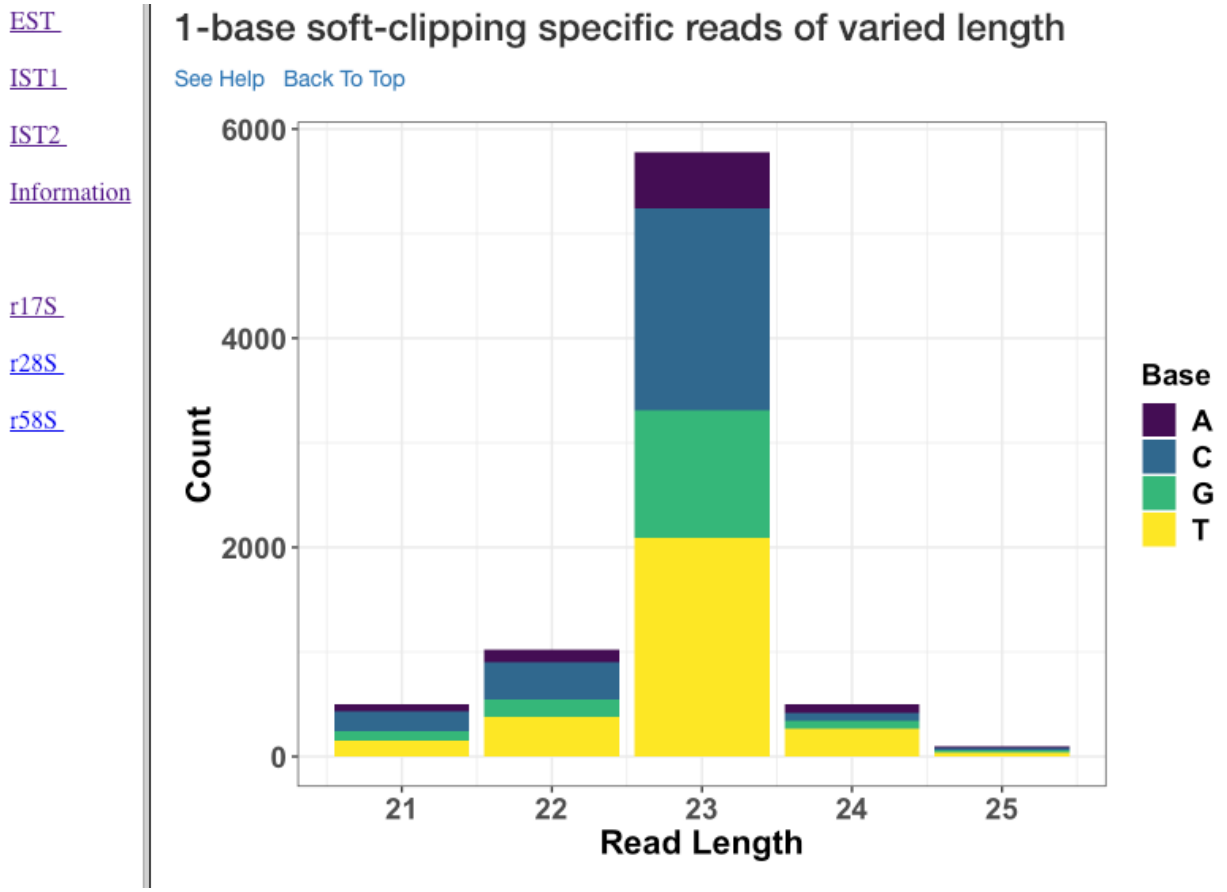
5.5.7 Soft-clipping status specific reads of varied length

Various read lengths (x-axis) utilized in the analysis and their read counts (y-axis), specific to their soft-clipping status is shown in this plot. This plot can further assist in understanding which read lengths are affected by soft-clipping, and if it is important to handle the soft-clipped bases before proceeding to downstream analysis.



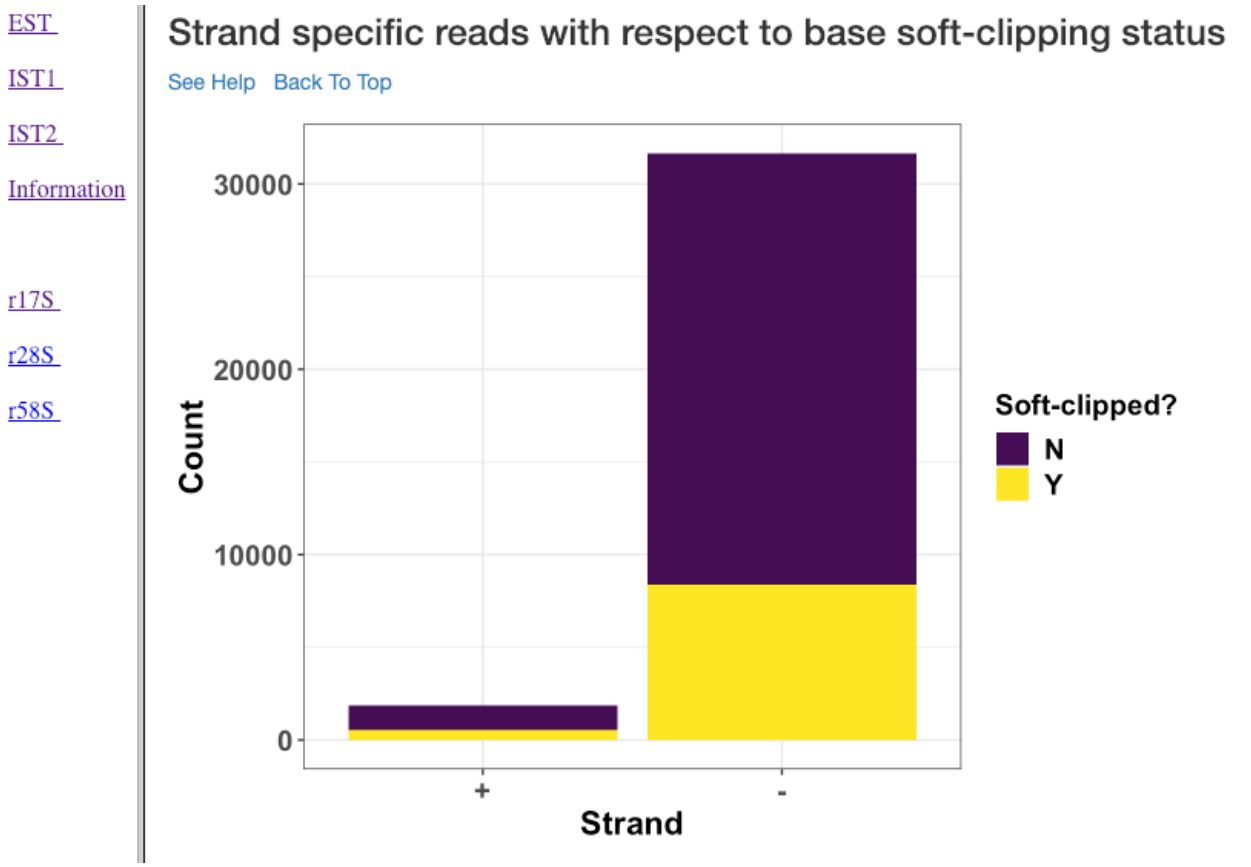
5.5.8 1-base soft-clipping specific reads of varied length

This plot shows various read lengths (x-axis) utilized in the analysis and their read counts (y-axis), with respected to the soft-clipped bases. Only the single bases (A,T,G and C) soft-clipped were considered. This plot can help in understanding, if any particular nucleotide is always soft-clipped. It could further indicate the potential source of the soft-clipped bases. For instance, untrimmed adapter, or primer, etc.



5.5.9 Strand specific reads with respect to base soft-clipping status

This plot shows the strand (x-axis) specific read counts (y-axis) with their base soft-clipping status. This plot can further assist in understanding which strands are affected by soft-clipping, and if it is important to handle the soft-clipped bases before proceeding to downstream analysis.



5.5.10 Strand specific coverage plot

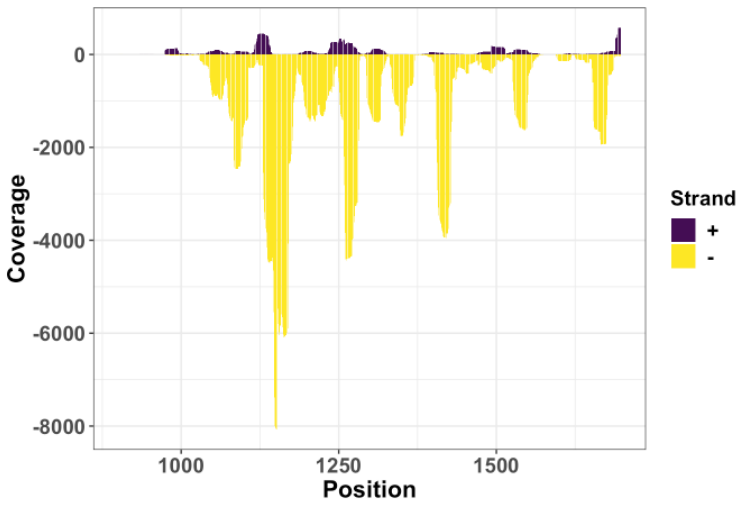
This plot shows the strand specific coverage (1bp resolution). A coverage plot helps in understanding if a particular sub-region in an analysed gene/region has a major predominance. It could also show, if a region appears to be phased giving insights in to the mechanism of action.

EST
IST1
IST2
Information

r17S
r28S
r58S

Strand specific coverage plot

See Help Back To Top



This document was created with R Markdown and the Knit package. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

5.6 Visualization: Comparison Report

This section describe the plots in the comparison report produced from *rapidVis*. The normalized values mentioned below corresponds to the normalization method you choose, while running *rapidNorm*.

The plots are split in to three categories.

- Quality Plots
- Sample based comparison
- Gene based comparison

5.6.1 Clustered heatmap of TPM

This is a heatmap of the TPM of gene/region corresponding to the samples analyzed. The dendograms shown are calculated using the default clustering parameters of heatmap.2 function, which uses a complete linkage method with an euclidean measure.

Comparison plots of samples and gene/regions

Report created by RAPID

04 February 2019

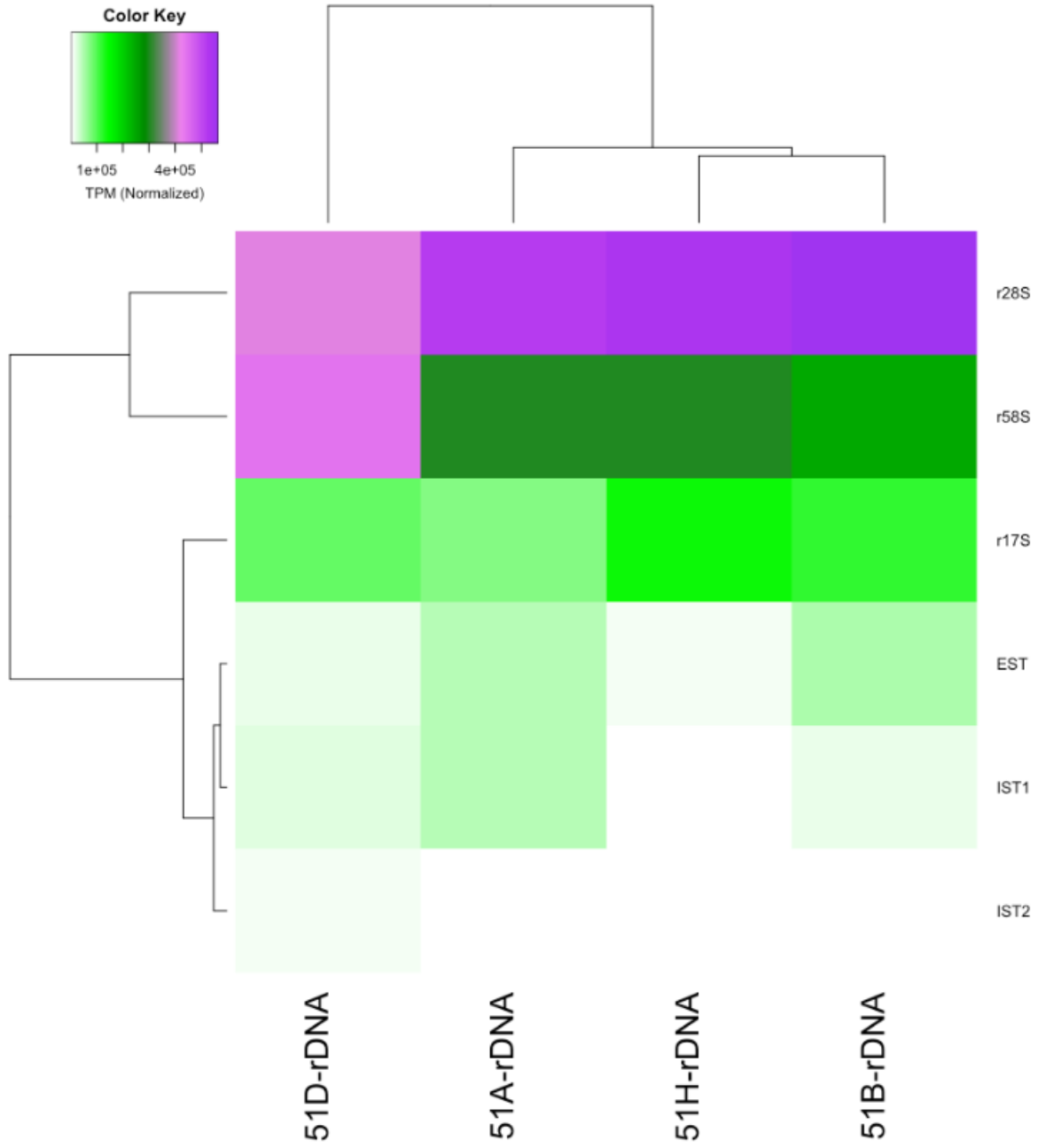
This report provides comparison plots of various samples and gene/region in normal and log2 scale.

Note: If graphs under some category are absent, it means there is not sufficient data with respect to that category.

Quality Plots Sample Comparison Gene/Region Comparison

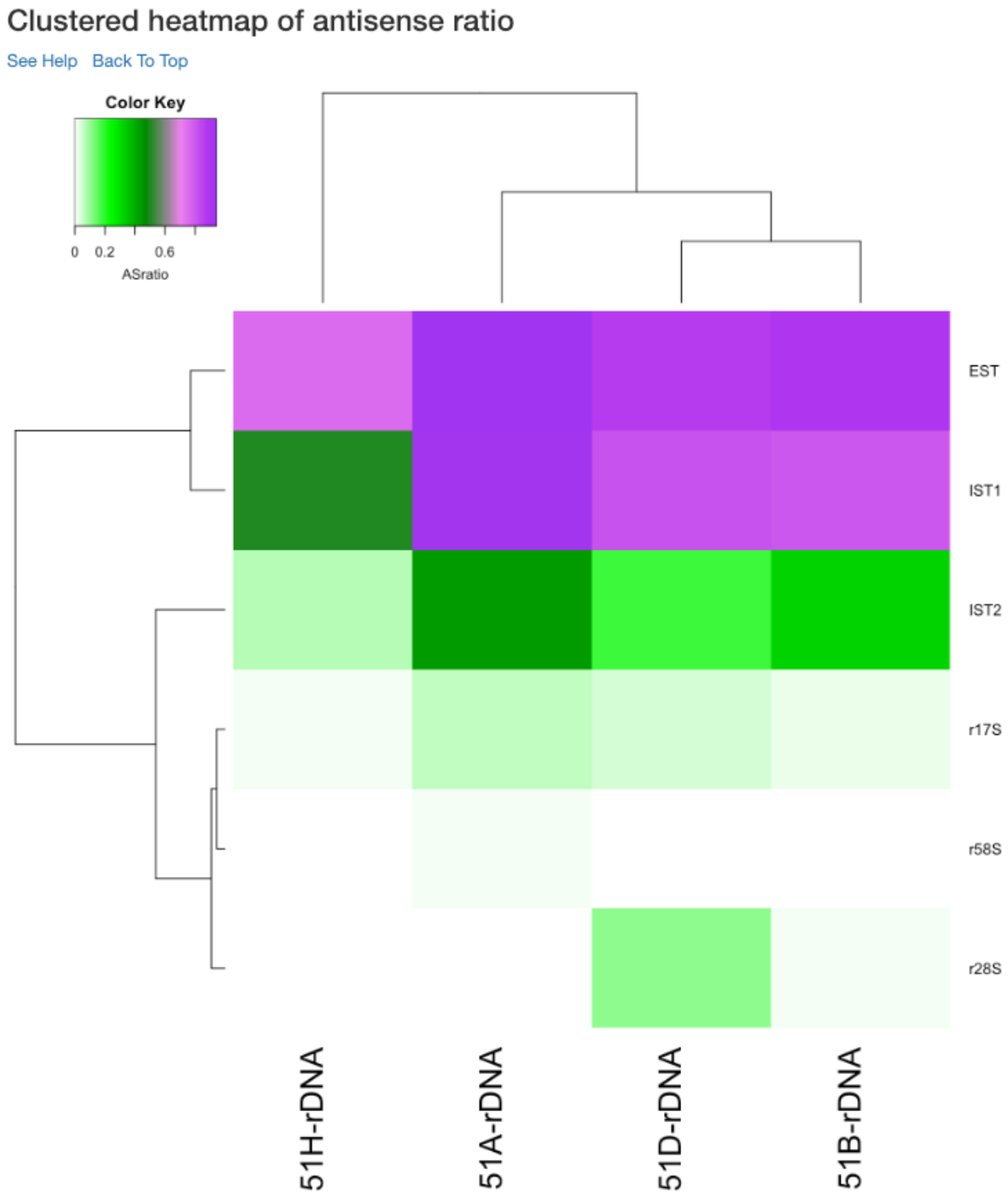
Clustered heatmap of TPM

See Help Back To Top



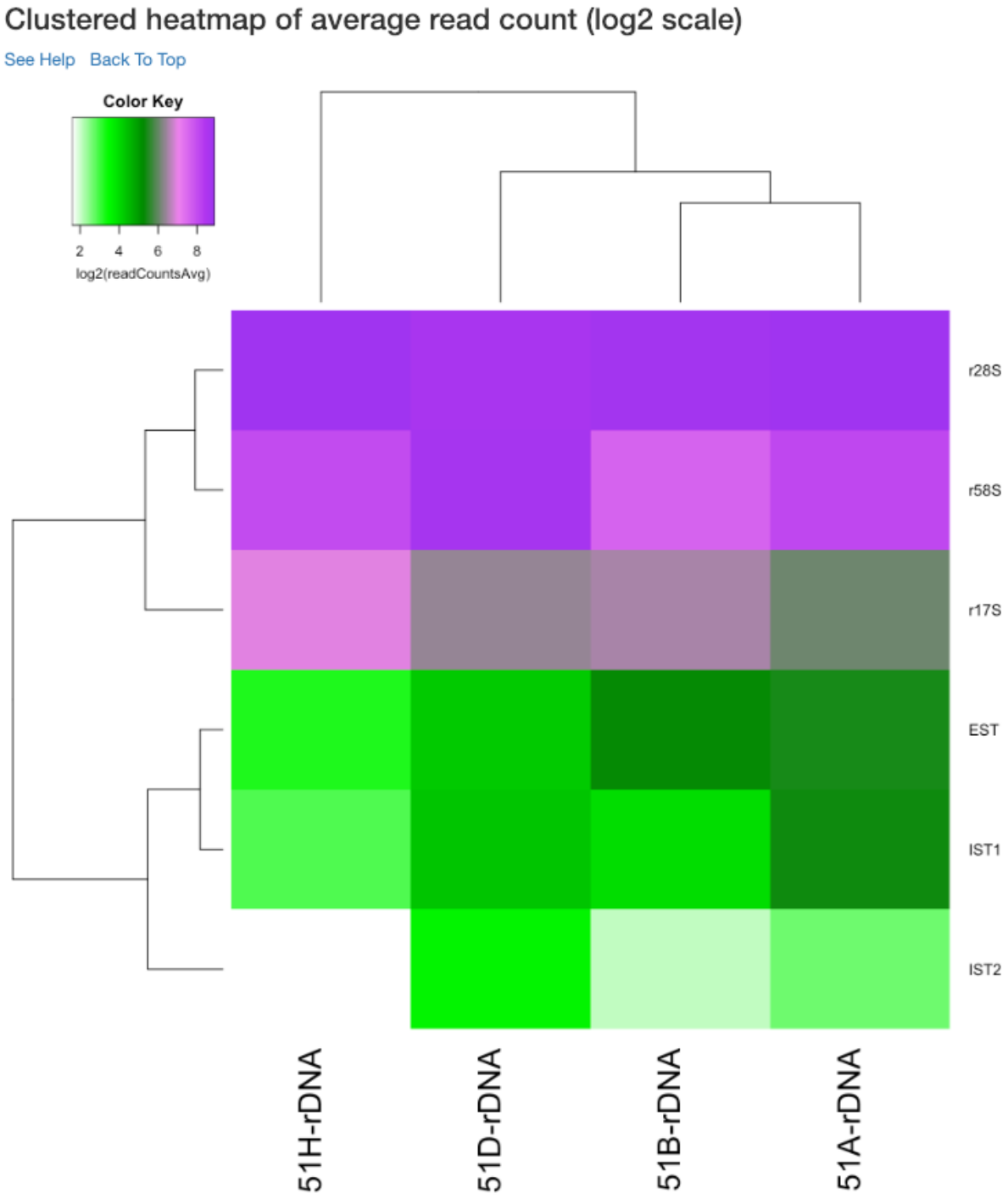
5.6.2 Clustered heatmap of antisense ratio

This is a heatmap of the antisense ratio of gene/region corresponding to the samples analyzed. The dendograms shown are calculated using the default clustering parameters of heatmap.2 function, which uses a complete linkage method with an euclidean measure.



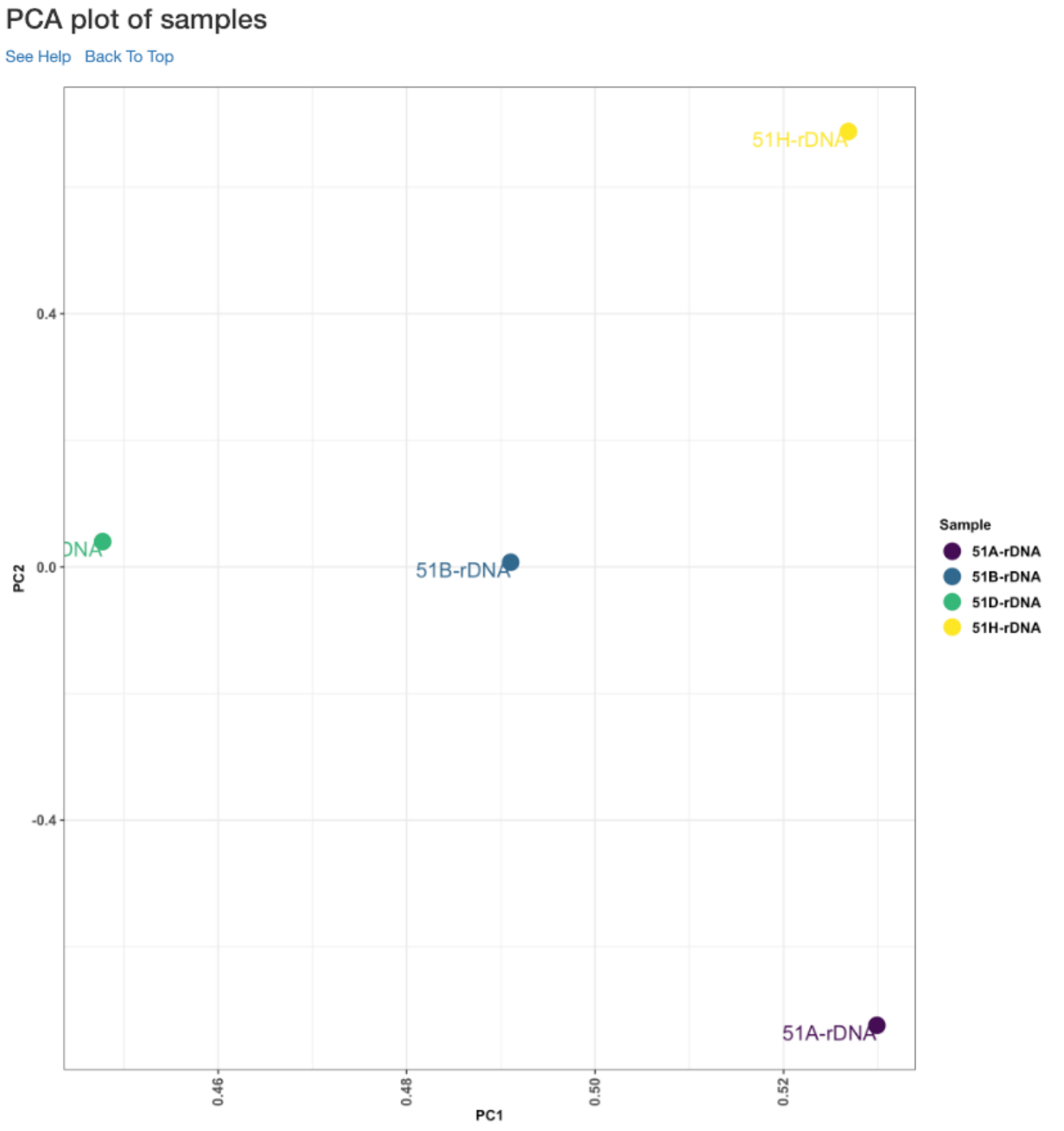
5.6.3 Clustered heatmap of average read count (log2 scale)

This is a heatmap of the average read count (log2) of gene/region corresponding to the samples analyzed. The dendograms shown are calculated using the default clustering parameters of heatmap.2 function, which uses a complete linkage method with an euclidean measure. Average read count is calculated as the reads aligned to a gene divided by the gene length.



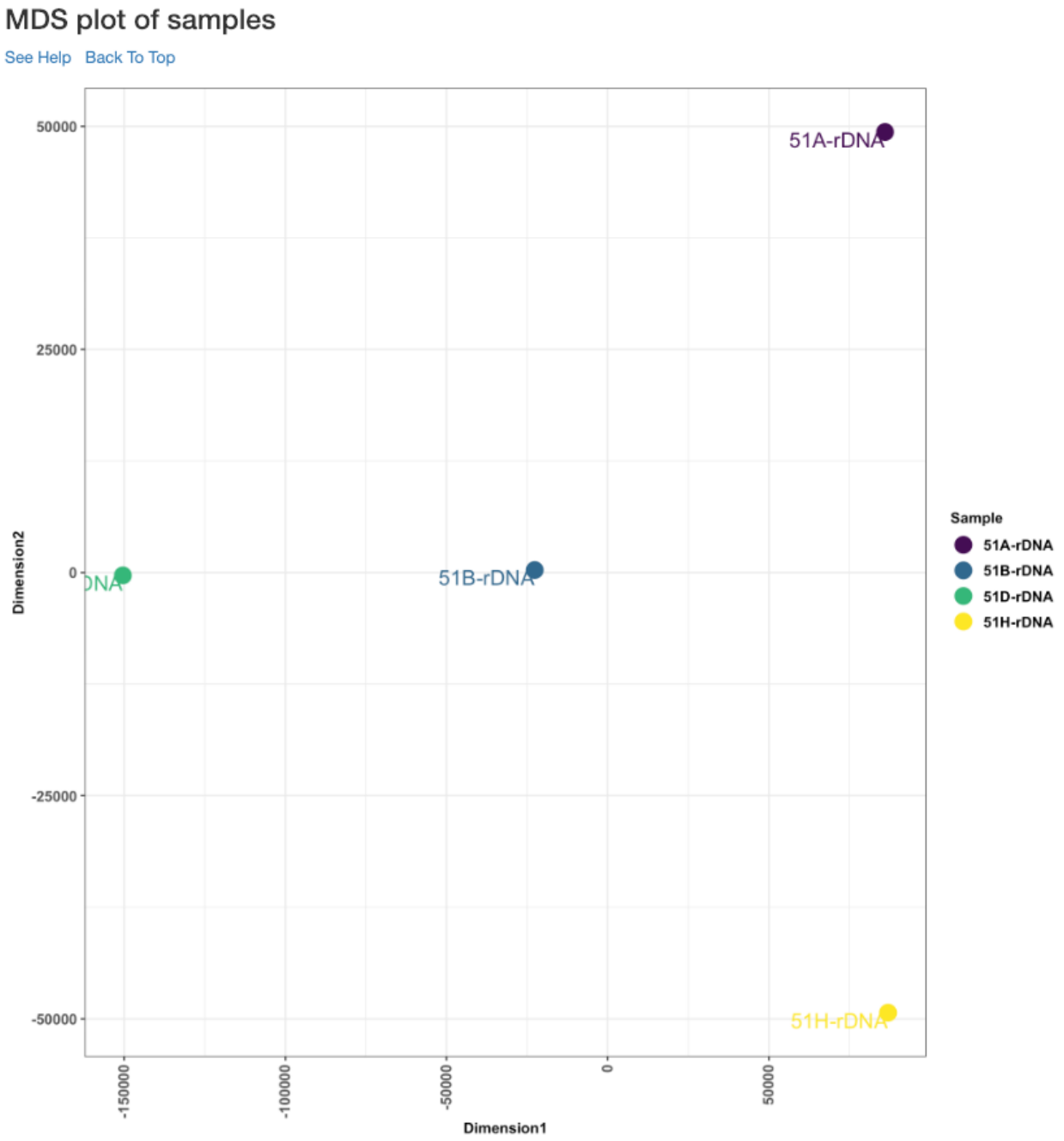
5.6.4 PCA plot of samples

This principle component analysis (PCA) plot shows where your samples fall in the first and second principle components. The principle componenets are calculated using the read counts of each sample. When replicates of a sample are grouped together in this plot, it is an indication of good quality replicates.



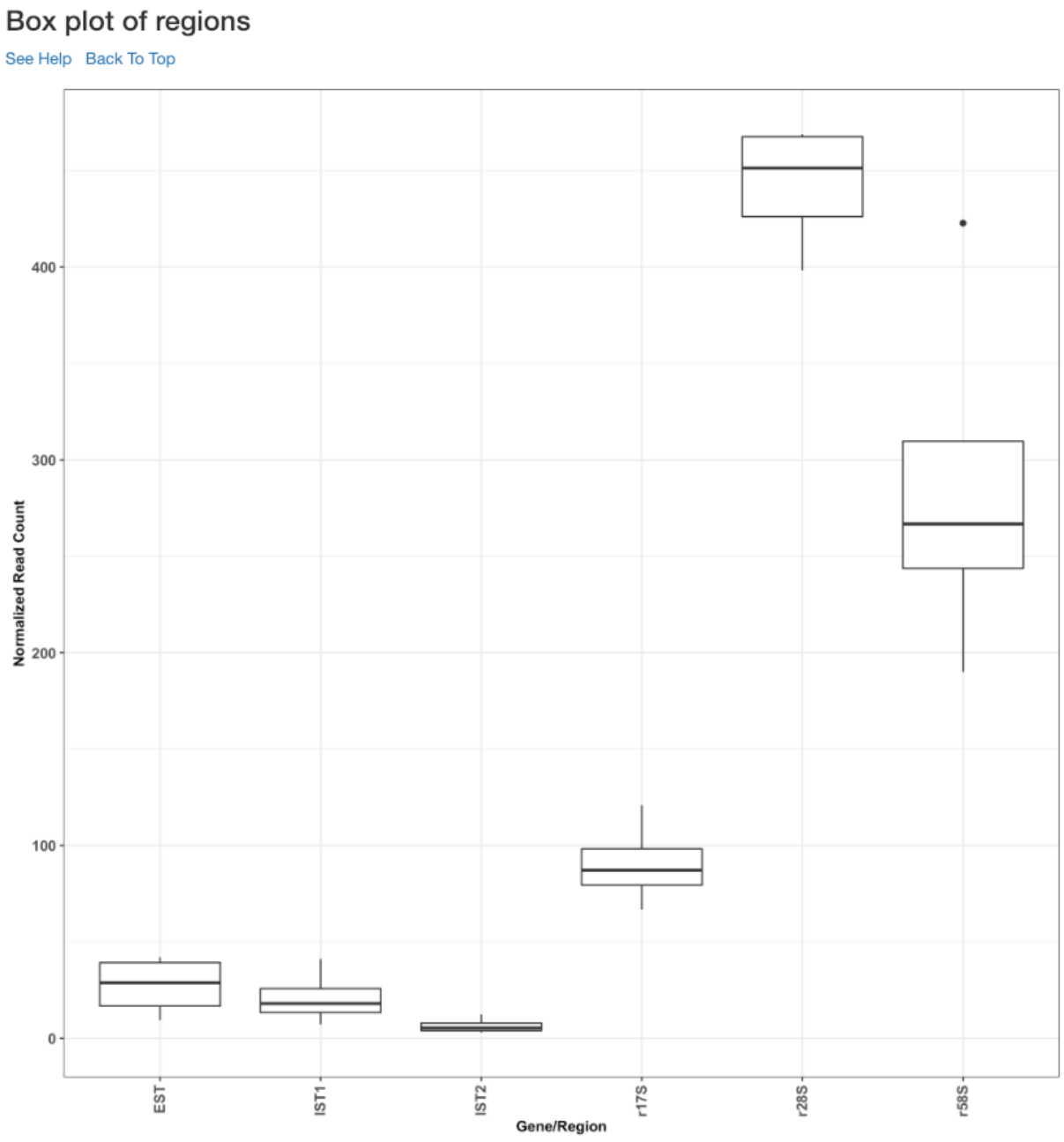
5.6.5 MDS plot of samples

This multi dimensional scaling (MDS) plot shows the proximities of your samples in two dimension. Read counts of each sample is used for performing MDS. When replicates of a sample are grouped together in this plot, it is an indication of good quality replicates.



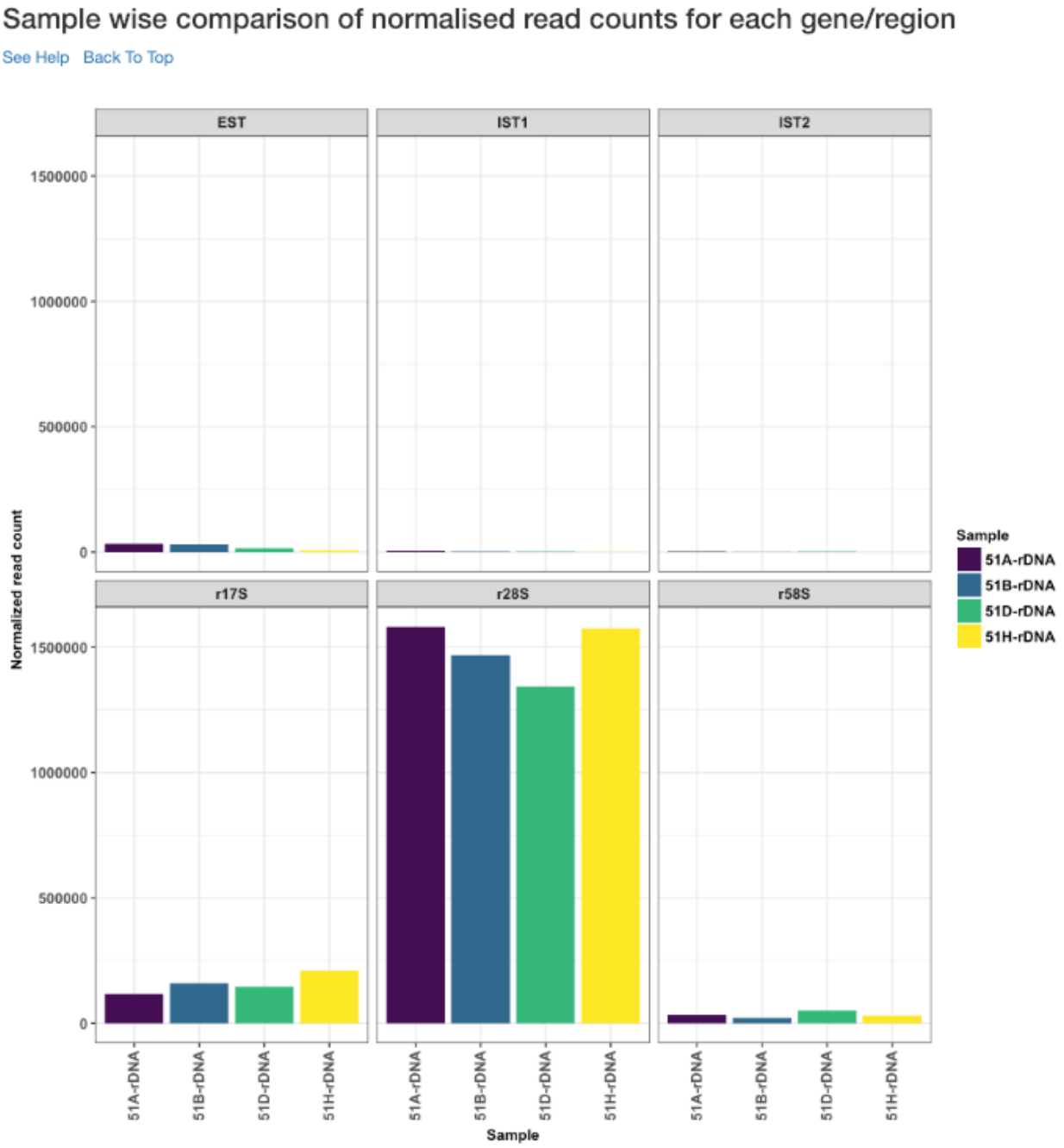
5.6.6 Box plot of read counts

This is a box plot of the normalized read counts of each gene/region across all the samples used in the analysis. This can indicate the variance among samples for a specific gene/region.



5.6.7 Sample wise comparison of normalised read counts for each gene/region

This plot shows the normalised read counts of each sample for each gene/region.

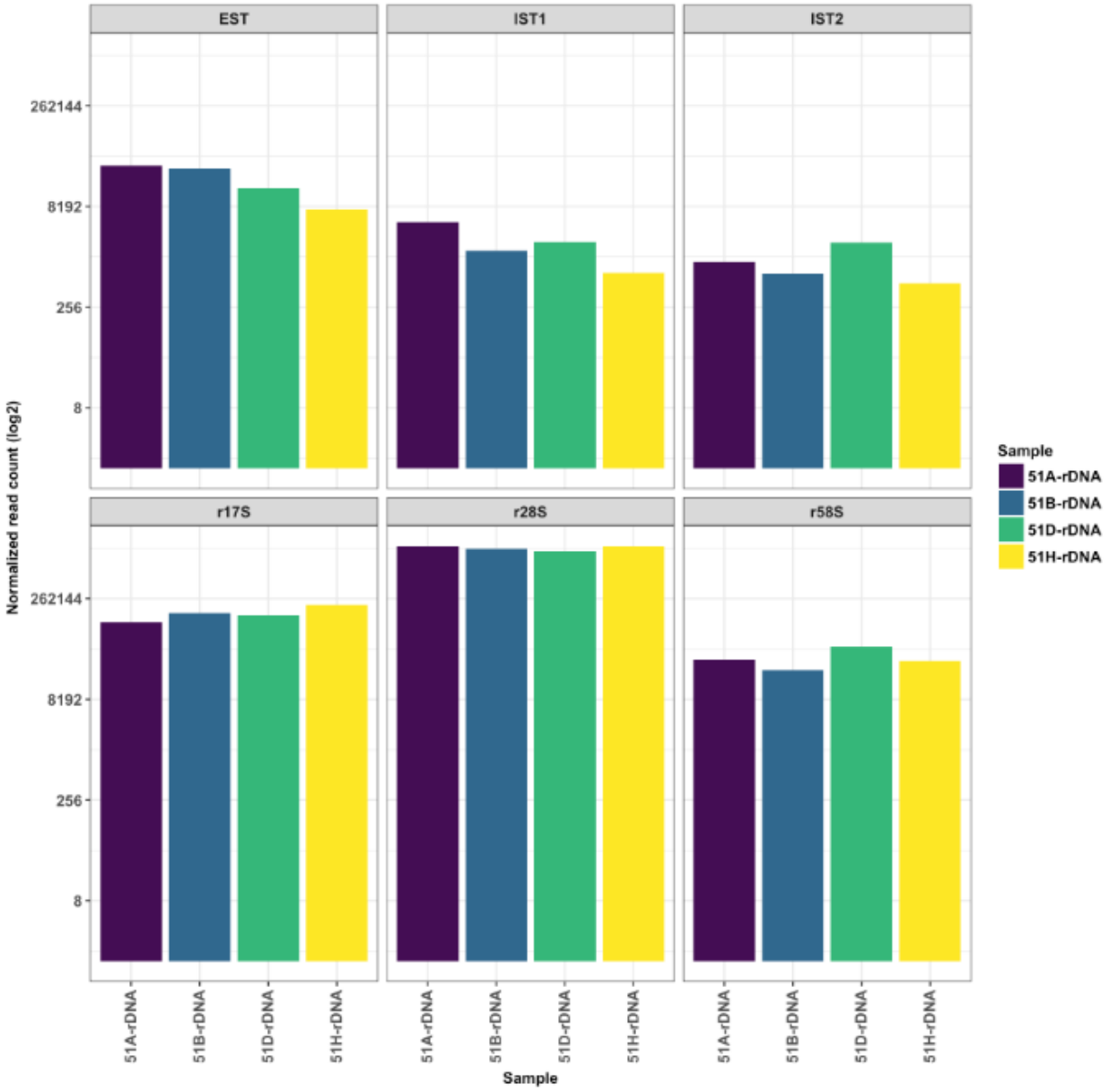


5.6.8 Sample wise comparison of normalised read counts for each gene/region (log2 scale)

Log2 of normalised read counts of each sample for each gene/region is shown in this plot.

Sample wise comparison of normalised read counts for each gene/region (log2 scale)

See Help Back To Top

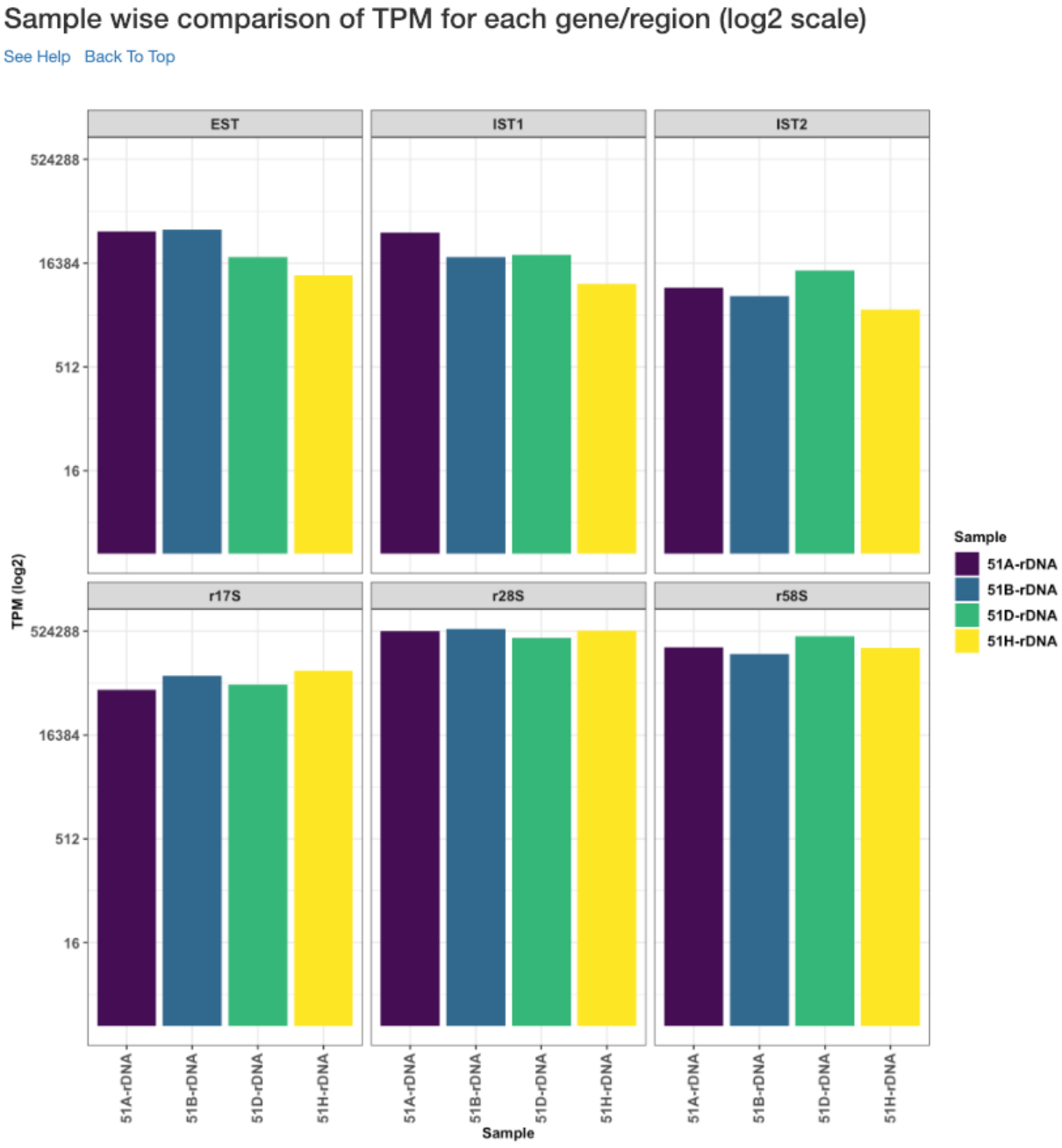


5.6.9 Sample wise comparison of TPM for each gene/region

This plot shows the TPM values of each sample for each gene/region. TPM values are calculated from the read counts, after accounting for read length restrictions, if provided by user.

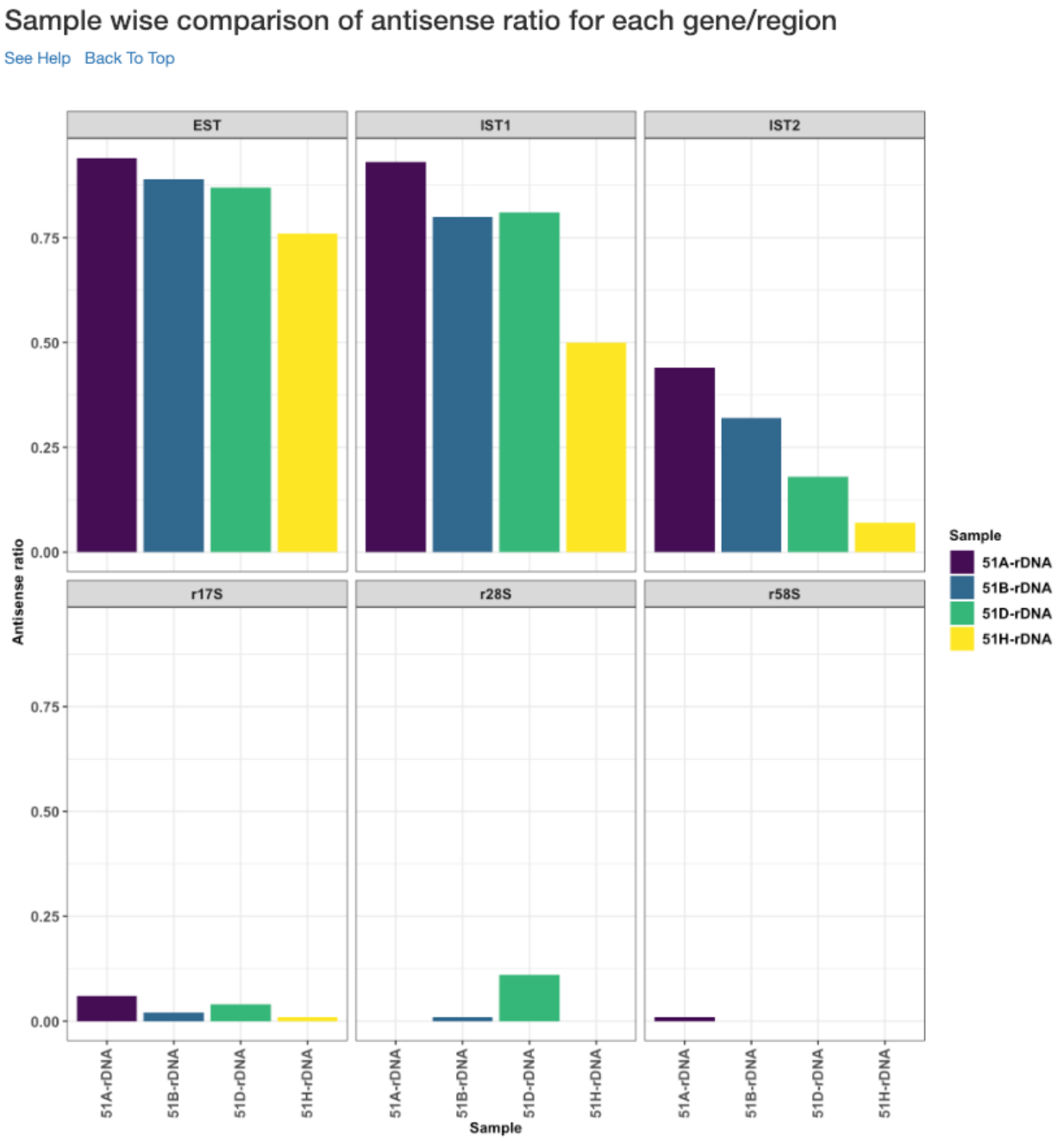
5.6.10 Sample wise comparison of TPM for each gene/region (log2 scale)

Log2 of TPM Values of each sample for each gene/region is shown in this plot. TPM values are calculated from the read counts, after accounting for read length restrictions, if provided by user.



5.6.11 Sample wise comparison of antisense ratio for each gene/region

This plot shows the antisense ratio of each sample is shown for each gene/region.

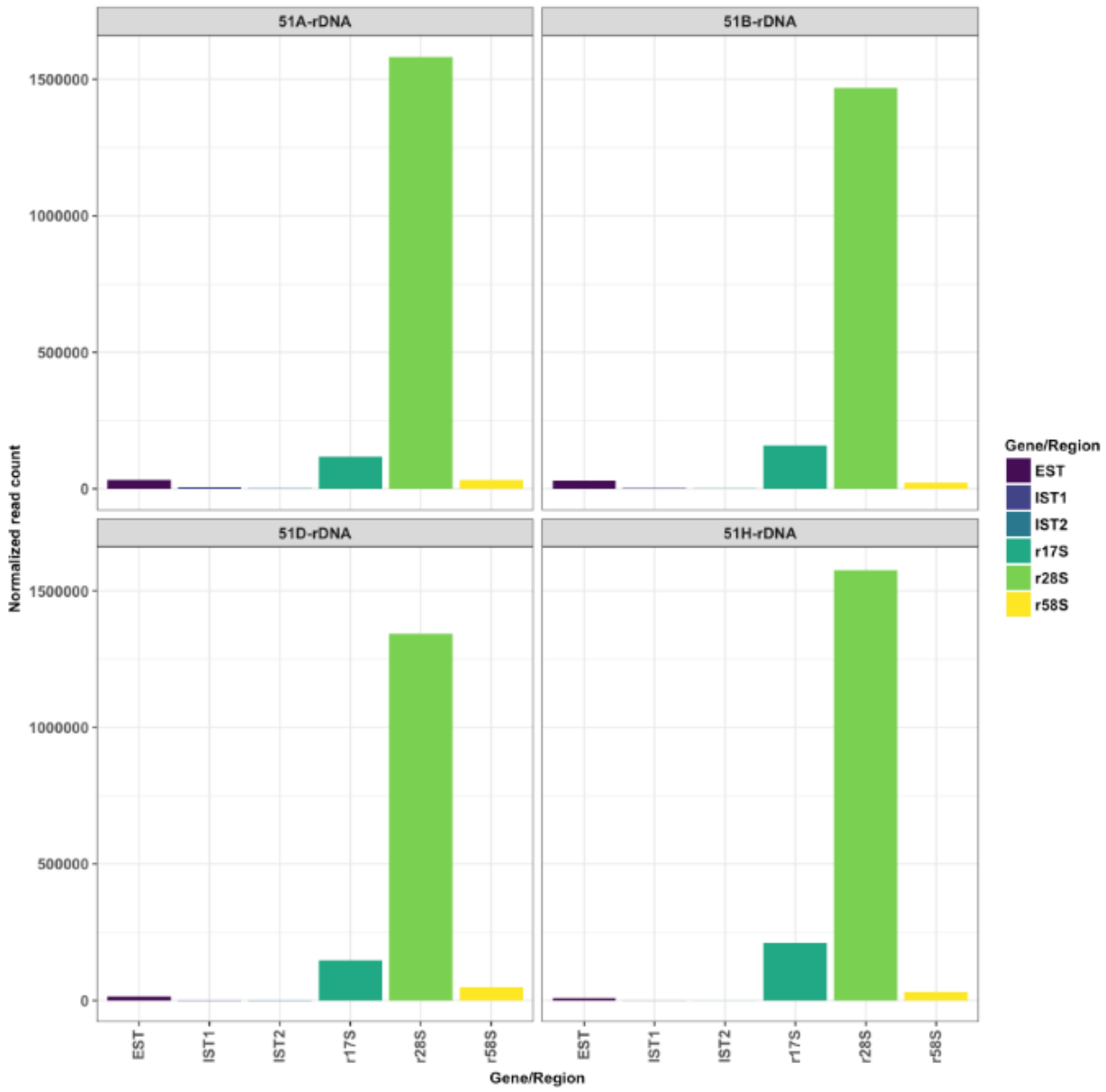


5.6.12 Gene/Region wise comparison of normalised read counts for each sample

This plot shows the gene/region wise normalised read counts for each sample.

Gene/Region wise comparison of normalised read counts for each sample

See Help Back To Top

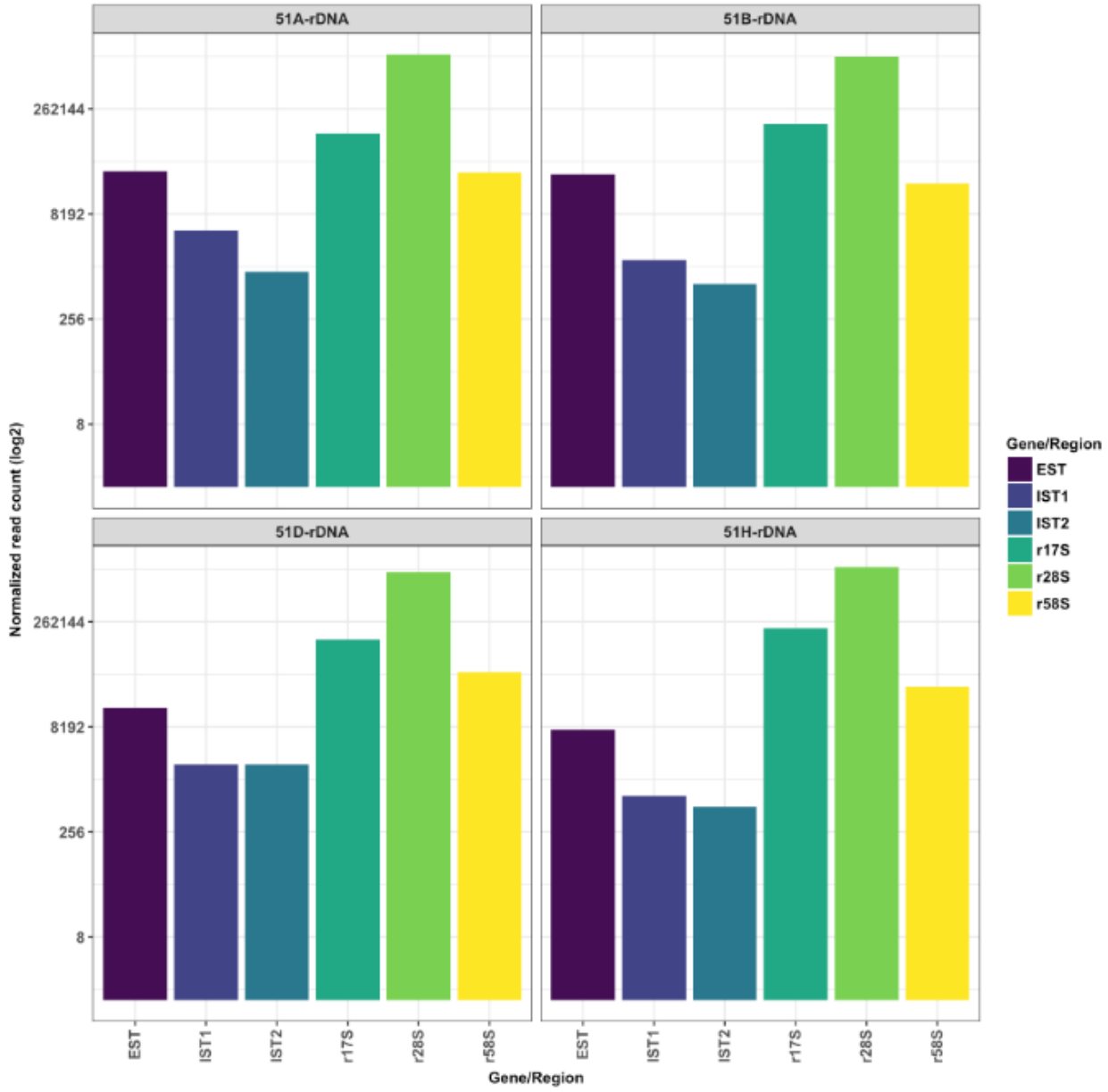


5.6.13 Gene/Region wise comparison of normalised read counts for each sample (log2 scale)

Log2 of gene/region wise normalised read counts for each sample is shown in this plot.

Gene/Region wise comparison of normalised read counts for each sample (log2 scale)

See Help Back To Top

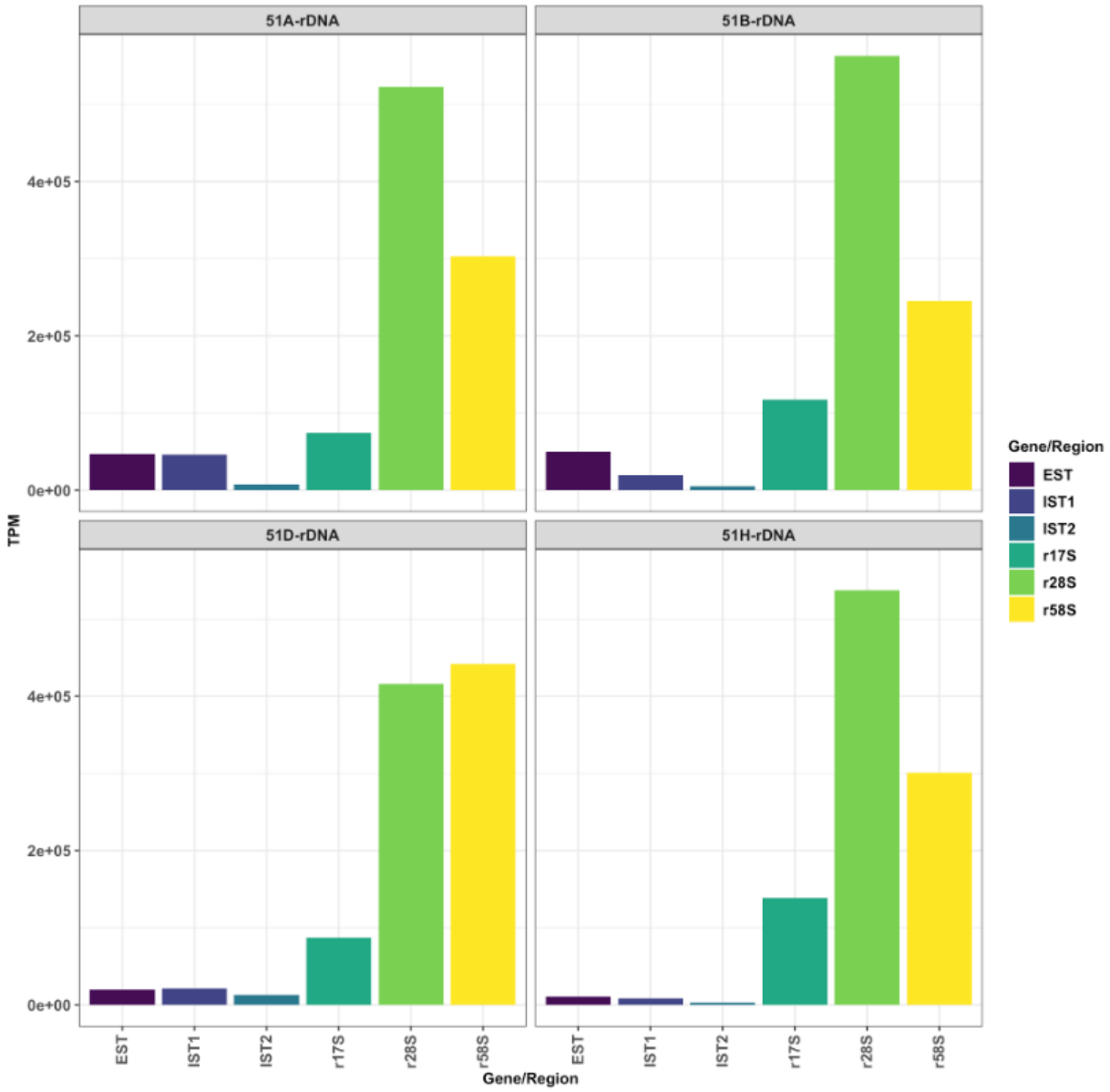


5.6.14 Gene/Region wise comparison of TPM for each sample

This plot shows the gene/region wise TPM for each sample. TPM values are calculated from the read counts, after accounting for read length restrictions, if provided by user.

Gene/Region wise comparsion of TPM for each sample

See Help Back To Top

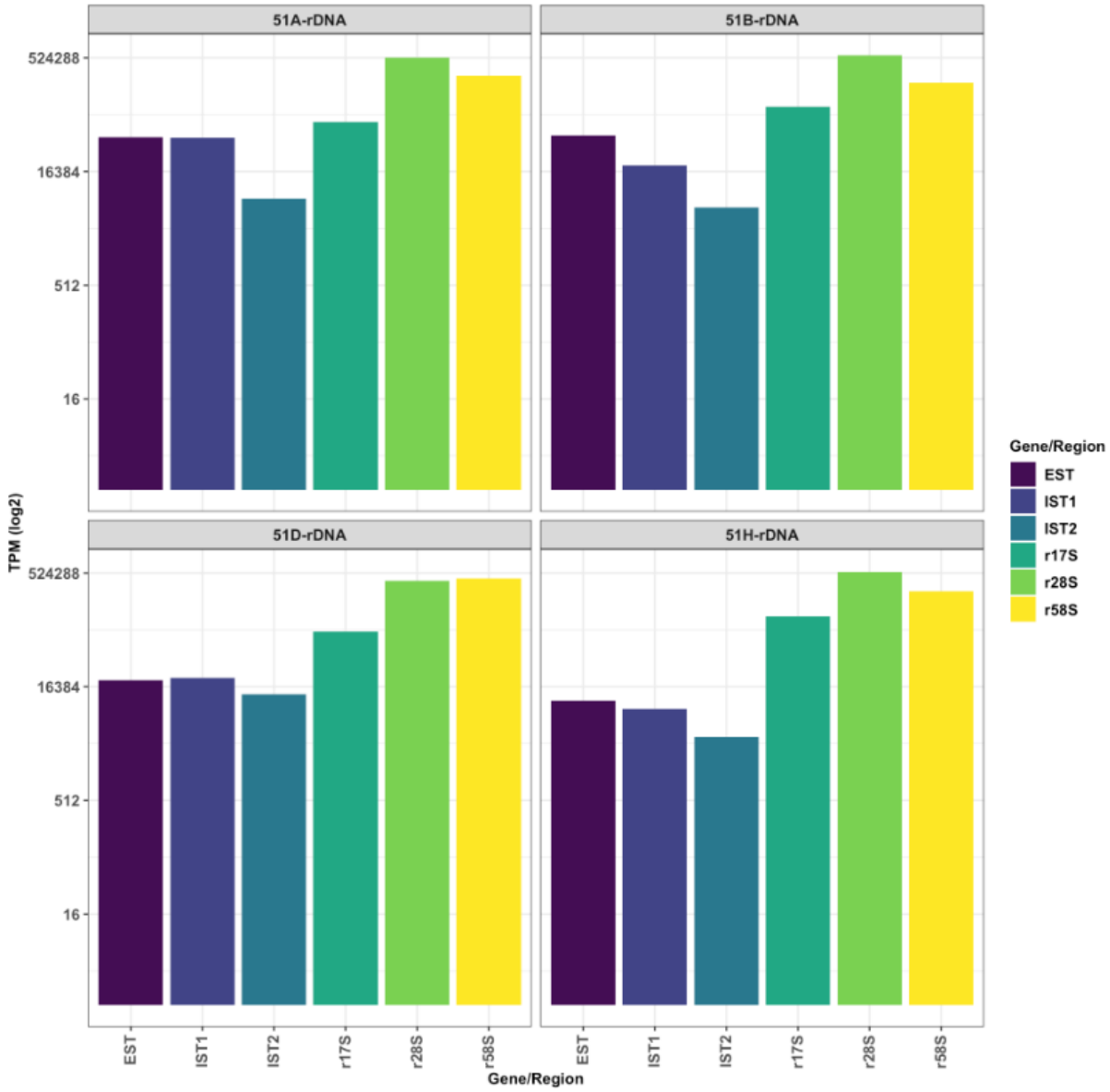


5.6.15 Gene/Region wise comparison of TPM for each sample (log2 scale)

Log2 of gene/region wise TPM for each sample is shown in this plot. TPM values are calculated from the read counts, after accounting for read length restrictions, if provided by user.

Gene/Region wise comparsion of TPM for each sample (log2 scale)

See Help Back To Top

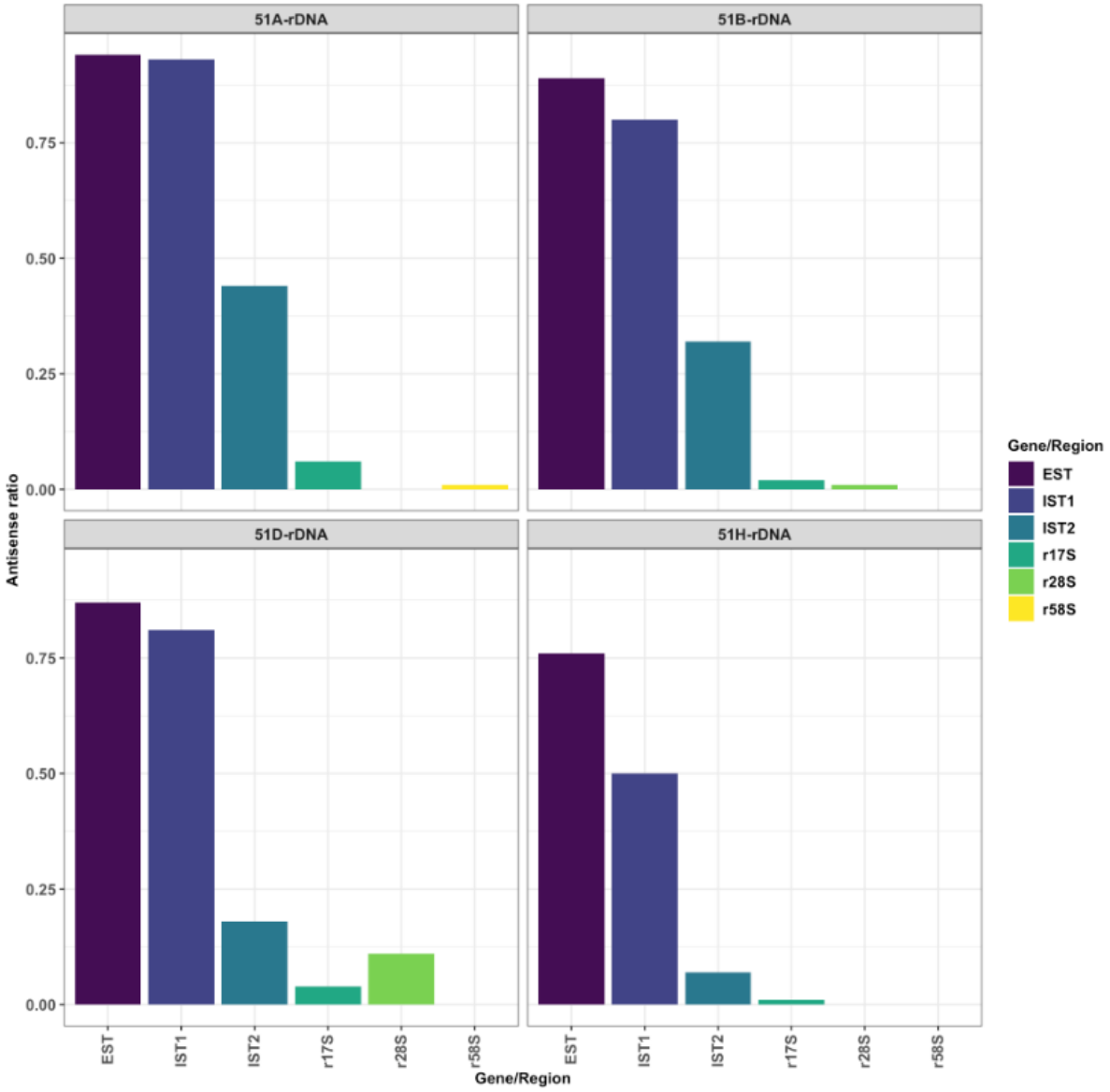


5.6.16 Gene/Region wise comparison of antisense ratio for each sample

Antisense ratio of gene/region for each samples is shown in this plot.

Gene/Region wise comparison of antisense ratio for each sample

See Help Back To Top



This document was created with R Markdown and the Knit package. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

CHAPTER 6

Troubleshooting

This part of the document will address FAQs, known issues, and work arounds. Please do report to us of issues not listed here, and we will try to keep this document up-to-date.

6.1 Conda related

Known issues related to conda based installation/execution.

- Library not loaded (libgfortran.3) *

This issue was encountered in the following scenario:

```
Conda version: 4.6.2
OSX version: 10.14.4

Error:

Abort trap: 6
dyld: Library not loaded: @rpath/libgfortran.3.dylib
Referenced from: /Users/...../rapid08/lib/R/lib/libRblas.dylib
Reason: image not found
```

This seems to be an issue with MacOS, and conda usage after recent updates. For some reason these fortran libraries are getting disabled when you activate the conda environment. The workaround is to change the order of PATH variable to access /usr/local/bin, and /usr/bin folders before the Conda environments bin folder.

For instance, the following PATH variable:

```
/Users/xxx/miniconda2/envs/rapid/bin:/Users/xxx/miniconda2/condabin:/Users/xxx/
↪miniconda2/bin:/usr/local/bin:/usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin
```

Should be changed to:

```
export PATH=/Users/xxx/miniconda2/condabin:/Users/xxx/miniconda2/bin:/usr/local/bin:/
↪usr/bin:/bin:/usr/sbin:/sbin:/opt/X11/bin:/Users/xxx/miniconda2/envs/rapid/bin
```

6.2 R - related

If there are some errors with R modules, Open R by staying in the conda environment, and update the R modules:

```
update.packages(ask = FALSE, checkBuilt = TRUE)
```