# Transcriptomics Resource Files

## 1 | INTRODUCTION

### 1.1 | BACKGROUND

### 1.2 | CITATION

### 1.3 | IMPLEMENTATION

### 1.4 | LICENSE AND AVAILABILITY

### 1.5 | COMMUNITY

### 1.6 | CONTACT

## 2 | ANAQUIN

### 2.1 | INTRODUCTION

### 2.2 | GENERAL USAGE

### 2.3 | SUMMARY OF AVAILABLE TOOLS

#### 2.3.1 | Rnaquin Tools

#### 2.3.2 | Anaquin R-Functions

### 2.4 | SUPPORTED INPUT FORMATS

### 2.5 | ANAQUIN OUTPUT

## 3 | INSTALL

### 3.1 | DOWNLOAD

### 3.2 | COMPILING FROM SOURCE

### 3.3 | INSTALL FOR R

### 3.4 | INDEX IN SILICO CHROMOSOME/GENOME WITH YOUR REFERENCE GENOME

#### 3.5.1 | Example Of Co-Index With Human Genome

### 3.6 | THIRD PARTY TOOLS AND INSTALLATION

## 4 | RESOURCE FILES

### 4.1 | CHROMOSOME SEQUENCES

### 4.2 | MIXTURE FILES

### 4.3 | ANNOTATION FILES

### 4.4 | LIBRARIES

## 5 | TRANSCRIPTOMICS

### 5.1 | INTRODUCTION

### 5.2 | DESIGN

### 5.3 | PREREQUISITES

#### 5.3.1 | Data And Resource Files

#### 5.3.2 | Third-Party Tools

### 5.4 | RNA-SEQ WORKFLOW FOR SEQUINS

### 5.5 | WORKFLOW - ALIGNMENT-FREE (KALLISTO)

#### 5.5.1 | Quantify Transcript Abundance (Kallisto)

#### 5.5.2 | Quantify Kallisto Quantification (Anaquin)

#### 5.5.3 | Splice-Aware Alignment Of Libraries (Tophat2)

### 5.5.4 | Assess Alignment (Anaquin)

#### 5.5.4.1 | Example Output (Rnaalign_Summary.Stats)

### 5.5.5 | Subsampling (Anaquin)

### 5.5.6 | Assemble Transcript Models (Cufflinks)

### 5.5.7 | Assess Assembly (Anaquin)

#### 5.5.7.1 | Plot Assembly Sensitivity Curve (Anaquin, R)

### 5.5.8 | Assess Isoform And Gene Expression (Anaquin)

#### 5.5.8.1 | Quantify Expression (Anaquin)

#### 5.5.8.2 | Plot Isoform/Gene Expression Curve (Anaquin, R)

#### 5.5.8.3 | Assess Multiple Replicates (Anaquin, R)

#### 5.5.8.4 | Example Output From Multiple Replicates (Anaquin, R)
1 | Introduction

1.1 | Background

Next-generation sequencing (NGS) is widely used in biological research and clinical diagnostics. We have developed sets of RNA and DNA standards, termed sequins that are designed to be spiked-in to users’ sample prior to library preparation and therefore undergo concurrent sequencing with the sample. Because sequins have a synthetic sequence, resultant sequenced reads will not align to a natural reference genome, but rather to an accompanying in silico genome. This enables sequins be analyzed in parallel with the sample, and be used as internal quantitative and qualitative controls for most steps in the NGS workflow, including downstream bioinformatics analysis.

We have developed a software toolkit, known as Anaquin that contains many useful tools to analyze sequins, and assess NGS performance. We aim to provide an analytical and statistical framework in which users can more easily analyze sequins.

This documentation aims to be a practical reference for the use and analysis of sequins. Where possible, we have described the use of sequins, and analysis with Anaquin, in the context of real-life experimental scenarios, such as with RNA sequencing, genome sequencing, cancer sequencing and metagenome sequencing experiments. If you do not have your own libraries containing sequins, we have example data you can download example data to familiarize yourself with the analysis of sequins at www.sequin.xyz/downloads

1.2 | Citation

We suggest that users familiarize themselves with the underlying concepts of sequins described in the citations below, and, if you are using sequins in your research or work, please cite the following manuscripts as appropriate:

1.3 | Implementation

Anaquin was implemented in C++ and R programming language. Users will be required to have an installation of both C++ and R. Further details on download and installation are available in Section 3.

The C++ command-line software is performed at the command line, and performs a number of processing and statistical functions, as well as integrates with other bioinformatics tools. Typically, we use this command-line Anaquin software to generate summary statistics and scripts that will then be loaded into R. The analysis of data in R enables graphs to be plotted, statistical functions called, and integration with other Bioconductor packages.

1.4 | License and Availability

Anaquin is freely available under BSD license. See - en.wikipedia.org/wiki/BSD_licenses for details. The GitHub source code is available - www.github.com/student-t/Anaquin.

1.5 | Community

You can post your questions on seqanswers.com, we will answer your questions as we actively monitor the sites.
1.6 | Contact

If you have any questions, please first check the FAQ section or the discussion group for your answers. If you still have questions, or new ideas, feature requests or bug fixes, please email us at sequin@garvan.org.au.
2 | Anaquin

2.1 | Introduction

Anaquin is a software toolkit designed for the analysis of synthetic sequin controls in NGS experiments. The software is designed for data visualization, data manipulation, and statistical analysis that describes the performance of the synthetic sequin controls. In addition, Anaquin performs functions, such as subsampling alignments, to aid the analysis of sequenced libraries that contain sequins.

2.2 | General Usage

The Anaquin toolkit is organised in a hierarchal fashion, with the following general syntax:

\texttt{anaquin \<tool> \<options>}

Where:

- \texttt{<tool>}: Name of the tool to be used
- \texttt{<options>}: Provides the user with the ability to modulate the tools function

To illustrate this usage, consider the following example command to analyze an alignment file:

\texttt{anaquin RnaExpression -method gene -rmix mixture.tsv -usequin transcripts.gtf}

Where:

- RnaExpression - is the name of the tool being called
- mixture.tsv - is a reference mixture for sequins in TSV format (specified with -rmix),
- transcripts.gtf - is the user generated GTF file from a third-party software (specified with -usequin)

Many command options are tool-specific. Using the \texttt{-h} option with any Anaquin tool will report the command line options. Detail on the usage and options for all tools can be found in \texttt{Appendix A}.

2.3 | Summary of Available Tools

2.3.1 | RnaQuin Tools

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RnaAlign</td>
<td>Measure spliced read alignments from sequins to the \textit{in silico} chromosome</td>
</tr>
<tr>
<td>RnaAssembly</td>
<td>Compare assembled transcript models to sequin annotations in the \textit{in silico} chromosome</td>
</tr>
<tr>
<td>RnaExpression</td>
<td>Quantitative analysis of sequin expression</td>
</tr>
<tr>
<td>RnaFoldChange</td>
<td>Assess fold-changes in gene expression between multiple samples</td>
</tr>
<tr>
<td>RnaSubsample</td>
<td>Calibrate sequence coverage of sequins across multiple replicates</td>
</tr>
</tbody>
</table>

2.3.2 | Anaquin R-Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plotLogistic</td>
<td>Create a scatter plot with input concentration on the x-axis, and measured proportion on the y-axis</td>
</tr>
<tr>
<td>plotLinear</td>
<td>Create a scatter plot with expected abundance on the x-axis, and measured abundance on the y-axis</td>
</tr>
<tr>
<td>plotROC</td>
<td>Create a receiver operating characteristic (ROC) plot at various threshold settings</td>
</tr>
</tbody>
</table>

2.4 | Supported Input Formats

The bioinformatics analysis of next-generation sequencing data is a rapidly expanding field of research, and there is a wide range of available tools, with different approaches, advantages and uses. Anaquin is designed to work with popular software (Cufflinks, DESeq2, TopHat2, STAR, GATK, VarScan amongst others) and, where possible, standard data formats (such as SAM, BAM, BED, VCF, GTF etc.)

However, supported softwares are selected due to popularity only, and users should feel free to try alternative software as required. Due to the number and diversity of available Support Software, it is not
possible to support all data formats, and users simply need to pre-process non-standard input files into a format compatible with Anaquin.

Anaquin also supports a range of simple tab-delimited text formats. These are included so users can easily convert other third-party file formats into these simple text formats. Users can omit text columns by using a '-' character. Further details and example simple text formats are provided in Appendix C.

If you have a script for processing data, using alternative unsupported software, or perform a novel analysis, please post it to our community.

2.5 | Anaquin output

Anaquin generates tab-delimited text files that can be easily copied and pasted into Excel, or any other spreadsheet program to perform further analysis. For example, RnaExpression (for quantifying genes expression) generates RnaExpression_sequins.tsv file that can be loaded directly into Excel.
3 | Install

3.1 | Download

Users can download the latest version of the Anaquin software toolkit from our website:

www.sequin.xyz/downloads

The download includes the latest version of all releases. Please be sure to update your version regularly to benefit from the latest bug fixes and improvements. Please see the Version History page for information on the latest changes and full release notes.

Please note the software is under active development and you may be occasionally confronted with bugs we have not yet caught. If this happens to you, please contact us. Please include your command line and error message and, if required, we may request some sample data. You can also open a ticket issue on our GitHub repository - github.com/student-t/Anaquin.

3.2 | Compiling from Source

Anaquin is a C++ 0x11 software. To build the software from source, you will need the following dependencies:

1. Eigen  (http://eigen.tuxfamily.org)
2. Boost   (http://www.boost.org/users/download)
3. HTSlib  (http://www.htslib.org)

Once the dependencies are installed, run make to build the software. The code requires a C++ 0x11 compiler. We have tested the compilation with g++ and xCode, however, other compilers might also work. Windows is not supported. To test your installation, run:

$ anaquin

3.3 | Install for R

Anaquin has also been implemented in R, enabling its use with a number of Bioconductor packages. To install Anaquin for the Bioconductor, follow the instructions at:


3.4 | Index *in silico* chromosome/genome with your reference genome

If sequins were added to the sample prior to sequencing, the read libraries should simultaneously be aligned to the *in silico* chromosome or genome (such as chrIS.fa), and a reference genome (such as hg38).

To align reads to both the consensus reference genome and *in silico* genome. The two sequences should be combined and then used to generate an index for alignment proposes. The generation of the index is often specific to the alignment tool (such as Bowtie, BWA, STAR etc.) being used.

Sequence files for synthetic chromosomes and genomes can be downloaded from here:

www.sequin.xyz/downloads

It is important to ensure that your chromosome/genome version is compatible with the type and version of sequins you have used.

3.5.1 | Example of co-index with human genome

The following example describes how to generate a Bowtie2/Tophat2 index containing both the reference human genome and the *in silico* chromosome. To download the *in silico* chromosome for RnaQuin download our resource bundle:
The synthetic chromosome is specified as chrIS (chromosome in silico) to distinguish it from human or other natural chromosomes. To download the hg38 reference human genome assembly released from the UCSC genome browser, perform the following command line:

```
$ wget hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz
$ gunzip hg38.fa.gz
```

To combine the in silico chromosome with the hg38 genome, perform the following command:

```
$ cat hg38.fa chrIS.v2.fa > hg38_chrIS.fa
```

It is good practice to confirm the presence of both in silico and human chromosomes in the genome. We can verify using the following command:

```
$ grep chr hg38_chrIS.fa
```

This will generate a list of chromosomes identifiers in the fasta file. Ensure the full complement of human chromosomes as well as the in silico chromosome (specified with chrIS or chrEV) are present.

To build an index for Bowtie from the combined sequences perform the following command:

```
$ bowtie2-build hg38_chrIS.fa hg38_chrIS
```

This creates a collection of index files that will be used by Bowtie2/Tophat2 to perform the alignment.

### 3.6 | Third Party Tools and Installation

Anaquin is compatible with many third party tools. Within the guide we have used some of the most popular tools, however, this list is not definitive and many of the tools can be replaced by alternative tool. While we have endeavored to provide a common framework for sequin analysis, users should feel free to experiment and substitute tools as required. Indeed, to incorporate new tools please contact us.
4 | Resource files

We provide a number of resource files to help with the analysis of sequins, and that are required as input files for Anaquin. These files often provide batch-specific information, such as the concentration of sequins in a mixture, or the annotations of sequins on the in silico chromosome/genome.

Users can download these files individually or bundled from the following site:
www.sequin.xyz/downloads

CRITICAL | It is important that the resource files are compatible with the type and version of sequins you have used. You can use the batch number of your sequins (the batch number is printed on the tube label and side of packaging box) to search the site for resource files associated with your sequins.

4.1 | Chromosome sequences

The in silico chromosome or genome sequences (such as chr1S.fa) are required for the alignment of sequins reads. These sequences are provided in FASTA format and should be combined with the reference genome (such as hg38) sequence, before building a combined index for alignment.

CRITICAL | Chromosome sequences are designed for different sequins, and are occasionally updated. Please ensure you use the chromosome or genome version that is compatible with the type and batch of sequins you have used.

4.2 | Mixture files

Mixture file is a text file that specifies the concentration of each sequin within a mixture (typically in attomoles/µL). Mixture files are often required as input to enable Anaquin to perform quantitative analysis.

CRITICAL | Ensure that you use the mixture file corresponding to your sequin batch number (can be found on shipped tube, or packaging).

To provide an example of a mixture file, let us examine the mixture file for RnaQuin (available in the resource bundle in the next section):

```
$ head -n 2 RNAsequins_isoform_mix.v2.tsv
ID    Length  MXA (attomol/µL)  MXB (attomol/µL)
R1_101_1  719   11.32965       0.472075
```

The field definitions for the mixture file are as follows:
1st column – unique sequin name
2nd column – length of sequin
3rd column – input concentration in attomol/µL for Mixture A.
4th column – input concentration in attomol/µL for Mixture B.

4.3 | Annotation files

Sequins typically represent genetic features, such as genes or variation that are encoded in the in silico chromosome or genome. Annotation files provide the location of such features on the in silico chromosome/genome. Standard formats (such as .BED, .GTF and .VCF) are adhered to wherever possible, and these files can be combined with third-party annotation files from the human genome (for example, GENCODE, dbSNP etc.) in compatible formats for analysis.

For example, we can download the latest GENCODE annotation GTF file from:
http://www.gencodegenes.org/releases/current.html

We can combine it with the in silico annotation file by (assume gencode.v25.annotation.gtf is the file name). The sequins_transcripts.gtf annotation file is available in the resource bundle.
4.4 | Libraries

A range of simulated and experimental read libraries is available for download for example usage. These are to be used to complete tutorials and for users to familiarize themselves with the use and analysis of sequins. Each library comes with attached metadata that describes detail on the sample and sequins used, and the technical specification of library preparation and sequencing. Libraries can be downloaded from:

www.sequin.xyz/downloads

Additionally, we provide a neat sequenced library for each batch of sequins we release. Users may wish to check their sequencing and analysis relative to these reference libraries for troubleshooting or benchmarking purposes.
5 | Transcriptomics

5.1 | Introduction

RNA sequencing (RNA-Seq) can measure both gene or isoform expression, and reconstruct novel and complex spliced isoforms. However, the sheer size and complexity of the expressed transcriptome can confound analysis with RNA-seq. The wide dynamic range between high- and low-expressed genes results in only sparse sequence coverage of lowly expressed genes, and this expression-dependent bias results in the poorer assembly, and more variable quantification of low-expressed genes.

The accurate alignment of reads across large intron junctions; repetitive DNA sequences and small exons can also be difficult, preventing the accurate assembly of alternative-spliced isoform structures. Finally, technical artifacts during the RNA-Seq workflow, including RNA extraction, ribosomal RNA depletion, library preparation, sequencing and analysis further bias the measurement of gene and isoform expression.

To assess the impact of these variables, we have developed a set of RNA sequin standards that emulate synthetic genes and isoforms. These gene sequins are added to a user’s RNA sample prior to library preparation, and provide internal controls for the downstream RNA-Seq workflow.

5.2 | Design

We designed each RNA sequin to represent an individual isoform generated by the alternative splicing of a gene loci from the in silico chromosome. A synthetic gene locus is typically represented by multiple alternative sequin isoforms, and modulating the relative abundance of these synthetic isoforms can thereby emulate the biological process of alternative splicing.

The artificial genes are distributed as complex loci across the in silico chromosome, with bidirectional, antisense and overlapping organizations, reflecting the pervasive transcription of the human genome. Furthermore, exons are demarcated on the chromosome by splicing dinucleotide elements (acceptor and donor sites).

Sequins represent a diversity of alternative splicing events, including intron retention, cassette exons, alternative transcription initiation and termination, and non-canonical splicing. Whilst the relative abundance of each synthetic isoform corresponds to the frequency of the alternative splicing event, the combined abundance of each isoform corresponds to the expression of the gene loci within the mixture.

We titrate RNA sequins into mixtures at different concentration to emulate differences in gene expression and alternative splicing. This establishes a reference scale that encompasses a range in gene expression across the human transcriptome. By differing the concentration of RNA sequins between mixtures, we can also provide a dynamic scale for measuring differential gene expression and alternative splicing between samples. By contrast, RNA sequins with invariant concentrations between mixtures provide static scaling factors to enable quantitative normalization between multiple RNA-Seq libraries.

5.3 | Prerequisites

5.3.1 | Data and Resource Files

Download the resource bundle for the workflow:

```bash
$ wget s3.amazonaws.com/sequins/RnaQuinUserManualBundle.zip
```

The following data and files are used in this bundle:

1. RNAsequins_mixA_rep1 to RNAsequins_mixA_rep3 – Example libraries from RNA sequin (v2, Mix A) for use in the workflow.
2. RNAsequins_mixB_rep1 to RNAsequins_mixB_rep3 – Example libraries from RNA sequin (v2, Mix B) for use in the workflow.
2. RNAsequins.v2.gtf – RNA sequins annotations.
3. RNAsequins.v2.fa – FASTA sequence file for all sequins
4. chrIS.v2.fa - *in silico* chromosome that should be co-indexed with hg38 for alignment of libraries
6. RNAsequins_isoform_mix.v2.tsv – Mixture file for RNA sequins (both Mix A and B)
7. hg38.fa - The most recent human genome build (GRCh38) can be downloaded by:
   ```bash
   $ wget hgdownload.cse.ucsc.edu/goldenPath/hg38/bigZips/hg38.fa.gz
   ```
   Bowtie2/Tophat2 requires a pre-built index from the human genome and *in silico* chromosome for alignment. How to build the index is covered in Section 3.5.
8. Human gene annotations - should also be downloaded to inform alignment and analysis. We recommend users download the most recent GENCODE annotation from
   [www.gencodegenes.org/releases/current.html](http://www.gencodegenes.org/releases/current.html)
   We will also need a combined GTF annotation for both the human genome and the *in silico* chromosome. Perform the following command to combine the two annotations into a single file (assume gencode.v25.annotation.gtf is the GENCODE annotation file name):
   ```bash
   $ cat gencode.v25.annotation.gtf RNAsequins.v2.gtf > gencode_chrIS.gtf
   ```

### 5.3.2 | Third-Party Tools

RNA sequencing analysis is complex, and there is a wide range of software tools available to analyze sequencing data. In this workflow, we use a range of popular tools (Tophat2, Cufflinks, etc.), however, users should feel free to use alternatives (STAR, StringTie etc.).

We strongly recommend users familiarize themselves with the usage of third-party tools. Optional parameters can strongly influence read alignment performance. Within the workflow, we have simply used the default parameters for clarity, however, the optimization of parameters by users would likely improve results (indeed, users can use sequins to assess parameter optimization).

Wherever possible, Anaquin accepts standardized file formats (such as BAM, SAM, GTF, VCF, BED etc.) and any third-party tools that conform to these standards can easily be substituted into the workflow. For other tools, users may need to modify the format of their data to conform with the input format typically provided to Anaquin. See Appendix D for example formats.

The following third-party tools users can find in the workflows:

<table>
<thead>
<tr>
<th>Tool</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bowtie2</td>
<td><a href="http://bowtie-bio.sourceforge.net/bowtie2">http://bowtie-bio.sourceforge.net/bowtie2</a></td>
</tr>
<tr>
<td>TopHat2</td>
<td><a href="https://ccb.jhu.edu/software/tophat">https://ccb.jhu.edu/software/tophat</a></td>
</tr>
<tr>
<td>SAMtools</td>
<td><a href="http://www.htslib.org">http://www.htslib.org</a></td>
</tr>
<tr>
<td>Cufflinks</td>
<td><a href="http://cole-trapnell-lab.github.io/cufflinks">http://cole-trapnell-lab.github.io/cufflinks</a></td>
</tr>
<tr>
<td>R</td>
<td><a href="https://cran.r-project.org">https://cran.r-project.org</a></td>
</tr>
<tr>
<td>IGV</td>
<td><a href="https://www.broadinstitute.org/igv">https://www.broadinstitute.org/igv</a></td>
</tr>
<tr>
<td>Kallisto</td>
<td><a href="https://pachterlab.github.io/kallisto">https://pachterlab.github.io/kallisto</a></td>
</tr>
</tbody>
</table>

### 5.4 | RNA-Seq Workflow for Sequins

Sequins can be used to assess the performance of RNA-Seq, and measure the limitations in transcriptome assembly and gene expression. Sequins are added directly to users’ RNA samples prior to library preparation and sequencing. Here we describe the analysis of RNA-Seq samples that has spiked-in RNA mixture A and B. In this workflow, we will perform the following steps:

1. Alignment-free analysis of single sample with *Kallisto*
2. Assess transcript quantification using *Anaquin RNAExpression*
3. Splice-aware alignment of reads using Tophat2
4. Assess spliced-alignment performance using *Anaquin RnaAlign*
5. Subsampling the reads using Anaquin RnaSubsample
6. Assemble alignments into transcript models using Cufflinks
7. Assess assembled transcript models using Anaquin RnaAssembly
8. Estimate gene/isoform expression using Cufflinks
9. Quantify gene/isoform expression measurements using RnaExpression

5.5 | Workflow - Alignment-Free (Kallisto)
Alignment-free approaches do not require a user to align reads to a reference genome, and is computationally efficient. A common approach is the counting of k-mers within transcripts as a measure of gene expression. In this workflow, we describe the use of Kallisto to quantify sequins.

5.5.1 | Quantify Transcript Abundance (Kallisto)
1 | We are first required to build an index of k-mers from a FASTA file of sequin sequences using the following command:

```
$ kallisto index -i RNAsequin_kallisto_index RNAsequins.v2.fa
```

Where:
RNAsequins.v2.fa is a reference FASTA file of sequin sequences.
-i specifies the output index name (RNAsequin_kallisto_index).

COMMENT | In the above example, we have only provided synthetic sequin sequences. Users may also want to supply human gene sequences (downloaded from GENCODE etc.) that are simply added to the input file FASTA file.

2 | We next quantify the transcript abundance using Kallisto by performing the command on each replicate (only one shown):

```
$ kallisto quant -i RNAsequin_kallisto_index -o 5.5.1 \ 
RNAsequins_mixA_rep1.R1.fq.gz RNAsequins_mixA_rep1.R2.fq.gz
```

The command will generate abundance.tsv in the 5.5.1 output directory, which gives the estimated abundance for each transcript.

5.5.2 | Quantify Kallisto Quantification (Anaquin)
3 | We can compare the transcript quantification in the abundance.tsv file to the expected input transcript concentrations using the following command:

```
$ anaquin RnaExpression -o 5.5.2 -rmix RNAsequins_isoform_mix.v2.tsv \ 
-usequin 5.5.1/abundance.tsv
```

Where:
RnaExpression is name of the tool
-o specifies the output directory (5.5.2)
-rmix specifies the reference mixture file (RNAsequins_isoform_mix.v2.tsv)
-usequin specifies the abundance file generated by Kallisto (5.5.1/abundance.tsv)

COMMENTS | We have specified isoform analysis because Kallisto works at the isoform level (it is a transcriptome analysis tool). However, we can also work at the gene level, for example:

```
$ anaquin RnaExpression -o 5.5.2 -rmix RNAsequins_isoform_mix.v2.tsv \ 
-usequin 5.5.1/abundance.tsv
```

Anaquin will aggregate the isoforms expressions into sequin genes.

RnaExpression will generate the following files in the output directory:
1. RnaExpression_summary.stats - provides summary statistics to describe the quantification of isoforms and genes in the library. Please see Appendix A.3 for an example of the output file, including a description and interpretation of statistics.
2. RnaExpression_isoforms.tsv - statistics for estimated expression at each individual sequin isoforms.
3. RnaExpression_genes.tsv – statistics for estimated expression at each individual sequin genes.
4. RnaExpression_isoforms.R – R script to plot the expression estimated for each sequin isoform relative to expected input concentration.
5. RnaExpression_genes.R – R script to plot the expression estimated for each sequin gene relative to expected input concentration.

5.5.2.1 Plot Expression Sensitivity curve (Anaquin, R)

4 | To plot the measured expression of each sequin gene relative to expected input concentration in the single replicate, load the RnaExpression_isoforms.R script (similarly, load the RnaExpression_genes.R script for gene expression) into R (please see Appendix B for instructions) to plot the following graph:

![](image)

Figure 5.5.2.1 Scatter-plot illustrates the observed abundance (in FPKM) relative to the expected abundance (in attomoles/ul) for sequin isoforms. The limit-of-quantification (LOQ; indicated by dashed line) is estimated by piecewise-linear segmentation (further details available in Appendix B). Synthetic genes below the LOQ are poorly measured. Above the LOQ, we fit a regression model (indicated by blue line with shadow depicts the 95% confidence interval).

5.5.3 Splice-aware alignment of libraries (TopHat2)

5 | Libraries should be first trimmed to remove adaptor contamination using (only one replicate shown for example):

```
$ trim_galore --paired RNAsequins_mixA_repl1.R1.fq.gz
   RNAsequins_mixA_repl1.R2.fq.gz
```

The usage will generate RNAsequins_mixA_repl1.R1_val_1.fq.gz and RNAsequins_mixA_repl1.R1_val_2.fq.gz in the working directory.

6 | Library reads are aligned to the combined index (comprising the in silico chromosome sequence chrIS and human genome sequence hg38.fa; see Section 3.5 for details) using a splice-reads aligner. To build an index for TopHat2/Bowtie from the combined sequences perform the following command:

```
$ bowtie2-build hg38_chrIS.fa hg38_chrIS
```

This creates a collection of index files that will be used by Bowtie2/Tophat2 to perform the alignment.
We can then align reads to the combined index using TopHat2 with the following commands (only one replicate shown for example):

```
$ tophat2 -o A1 hg38_chrIS RNAsequins_mixA_repl.R1_val_1.fq.gz \
    RNAsequins_mixA_repl.R2_val_2.fq.gz
```

This will generate an alignment file (`accepted_hits.bam`) in the specified output directory (`-o` A1).

**COMMENT** | The `-G` option supplies TopHat2 with known gene annotations to guide alignment (rather than solely on de-novo alignment). If this option is exercised, a combined annotation (of both human gene and sequin annotations) should be given. For example:

```
$ tophat2 -G gencode_chrIS.gtf -o A1 hg38_chrIS RNAsequins_mixA_repl.R1_val_1.fq.gz RNAsequins_mixA_repl.R2_val_2.fq.gz
```

**COMMENT** | Users can simultaneously assess the de novo and guided assembly of known and unknown transcripts in a transcriptome by selectively removing subsets or even parts of sequin annotations from the supplied reference annotation file.

### 5.5.4 Assess Alignment (Anaquin)

We can assess the output read alignments to the *in silico* chromosome using the RnaAlign tool. Perform the following command (only one replicate shown for example):

```
$ anaquin RnaAlign -o 5.5.4 -rgtf gencode_chrIS.gtf -usequin A1/accepted_hits.bam
```

Where:

- `RnaAlign` is name of the tool
- `-o` specifies the output directory (`5.5.4`)
- `-rgtf` specifies the combined annotation file (**gencode_chrIS.gtf**)
- `-usequin` is the input alignment file generated by Tophat2 (**accepted_hits.bam**)

RnaAlign will generate the following files in the output directory:

1. `RnaAlign_summary.stats` — summary statistics describing the library-wide alignment performance. The file calculates the sensitivity and precision at the exon, intron and nucleotide level. It also gives the detection limit; defined as the lowest abundance (within the sequins) detected in the input alignment file. Please see **Output 5.5.4.1** for an example of the output file, including a description and interpretation of statistics.

2. `RnaAlign_sequins.tsv` — provides number of reads, exon sensitivity, intron sensitivity and precision for each individual sequin.

**COMMENT** | Measures of sensitivity and precision are relative to the supplied reference annotations (**gencode_chrIS.gtf**). However, because human gene annotations are incomplete, and a large fraction of annotated human genes will not present in a sample due to cell-specific expression, these measures are likely to be inaccurate when comparing and are provided for trouble-shooting purposes only.

#### 5.5.4.1 Example Output (RnaAlign_summary.stats)

**Please refer to Appendix C for statistical definitions.**

```
-------RnaAlign Summary Statistics
     Input alignment file: A1/accepted_hits.bam
     Reference annotation file: gencode_chrIS.gtf

-------Number of alignments mapped to the synthetic chromosome and genome

    Synthetic: 68724501
    Genome:   372569
    Dilution:  0.995

-------Reference annotation (Synthetic)

    Synthetic: 869 exons
    Synthetic: 754 introns
```
5.5.5 | Subsampling (Anaquin)

| Subsampling (Anaquin) |

Users may be required to subsample alignments to the *in silico* chromosome to calibrate spike-in amount (where variation in the amount added can occur) and sequencing coverage between multiple samples, or replicates. We can use the *RnaSubsample* tool to perform this alignment subsampling with the following command:

```
$ anaquin RnaSubsample -o 5.5.5 -method 0.10 -usequin A1/accepted_hits.bam \ 
  | samtools view -bS - > A1/subsampled.bam
```

**CRITICAL** | *RnaSubsample* generates SAM outputs to the console so they can be easily piped into a workflow. The above example will sort and compress the SAM output into a BAM format to reduce memory storage, and will generate A1/subsampled.bam in the output directory.

In the above example, we have subsampled alignments to the *in silico* chromosome to 10% of total alignments. If variable amounts of RNA sequins have been added to multiple replicates, we may want to calibrate multiple replicates to have the same 10% RNA sequin spike-in amount. This can be performed by repeating the above command for each replicate library to be calibrated.

Where:

RnaSubsample is name of the tool

- `-o` specifies the output directory (5.5.5)
- `-method` specifies the fraction to subsample (0.10)
- `-usequin` specifies the user generated alignment file (A1/accepted_hits.bam)

*RnaSubsample* will generate the following file in the output directory:

RnaSubsample_summary.stats - summary statistics reporting the coverage (number of reads on the *in silico* chromosome and genome) before and after subsampling. Please see Appendix A.5 for an example of the output file.

5.5.6 | Assemble Transcript Models (Cufflinks)

| Assemble Transcript Models (Cufflinks) |

Transcript models can be de-novo assembled from read alignments. We use *Cufflinks* to assemble sequenced read alignments into transcript models with default parameters. To perform de-novo transcript assembly (which requires only BAM alignment files and does not require previous annotations), perform the following command (only one replicate shown for example):

```
$ cufflinks -o 5.5.5 --no-headline --merge-reads A1/subsampled.bam
```

© 2017 Garvan Institute. All rights reserved.
$ cufflinks -o A1/D A1/subsampled.bam

This will generate a GTF file that comprises the assembled transcripts annotations (transcripts.gtf; both human and synthetic transcript assemblies) for each replicate.

11 | We will also need to run guided assembly for estimating gene/isoform expression. It is important to provide the combined annotation that contains gene annotations on the in silico chromosome and the accompanying genome. Run the following command (only the first replicate is shown) for guided assembly:

```bash
$ cufflinks -G gencode_chrIS.gtf -o A1/G A1/subsampled.bam
```

**5.5.7 | Assess Assembly (Anaquin)**

12 | We can compare the assembled transcript models to known sequin transcript annotations using the RnaAssembly tool. Perform the following command:

```bash
$ anaquin RnaAssembly -rmix RNAsequins_isofrom_mix.v2.tsv -o 5.5.7 -rgtf \
    gencode_chrIS.gtf -usequin A1/D/transcripts.gtf
```

Where:
- **RnaAssembly** is name of the tool
- `-o` specifies the output directory (5.5.7)
- `-rgtf` specifies the combined annotation file (gencode_chrIS.gtf)
- `-rmix` specifies the mixture file (RNAsequins_isofrom_mix.v2.tsv)
- `-usequin` specifies the user-generation transcriptome assembly (transcripts.gtf)

RnaAssembly will generate the following files in the output directory:
1. **RnaAssembly_summary.stats** - summary statistics describing the assembly performance of the library. Please see Appendix A.2 for an example of the output file, including a description and interpretation of statistics.
2. **RnaAssembly_sequins.tsv** - provides assembly statistics for each individual sequin isoform.
3. **RnaAssembly_assembly.R** - R script to plot the assembly sensitivity of each sequin relative to expected input concentration.

**5.5.7.1 | Plot Assembly Sensitivity curve (Anaquin, R)**

13 | To plot the assembly sensitivity (the fraction of each sequin that is correctly assembled) relative to expected input concentration, load the RnaAssembly_assembly.R script into R (please see Appendix B for details on how to load the script and plot graphs in R/R-Studio) to plot the following graph:
Figure 5.5.1 Scatter-plot illustrates the assembly sensitivity of each sequin, relative to input concentration. The assembly of isoforms exhibits an expression-dependent sigmoidal relationship with input concentration. Fitting is performed by non-linear least square fitting on sigmoid function. The minimal concentration required to assemble isoforms according to a user-specific assembly threshold is indicated (for example, dashed line corresponds to 0.70 sensitivity).

14 | The graph is generated by the Anaquin PlotLogistic R-function. Further details about the function can be found in Appendix B.6. User’s can modify the LOA sensitivity threshold (default is 0.70) by modifying the script line in RnaAssembly_assembly.R:

> threshold <- 0.70

The line can be modified to a user determined limit, for example 0.5:

> threshold <- 0.50

5.5.8 | Assess Isoform and Gene Expression (Anaquin)

5.5.8.1 | Quantify Expression (Anaquin)

Cufflinks also estimates gene and isoform expression within a sample. We can use the RnaExpression tool to compare this estimated expression (in FPKM) to the known input concentration of each sequin in the mixture (in attomoles/µL, as specified in a mixture file).

15 | Recall our guided assembly earlier. To quantify gene expression for a single replicate, perform the following command:

```sh
$ anaquin RnaExpression -o 5.5.8.1 -rmix RNAsequins_isoform_mix.v2.tsv \   -usequin A1/G/transcripts.gtf
```

Where:

RnaExpression is the name of the tool
- o specifies the output directory (5.5.8.1)
- rmix specifies the reference mixture file (RNAsequins_isoform_mix.v2.tsv)
- usequin specifies the transcriptome assembly generated by Cufflinks (A1/G/transcripts.gtf)

RnaExpression will generate the following files in the output directory:
1. **RnaExpression_summary.stats** provides summary statistics to describe the quantification of genes in the library. Please see Appendix A.3 for an example of the output file, including a description and interpretation of statistics.

2. **RnaExpression_isoforms.tsv** – statistics for estimated expression at each individual sequin isoform.

3. **RnaExpression_genes.tsv** – statistics for estimated expression at each individual sequin gene.

4. **RnaExpression_isoforms.R** – R script to plot the expression estimated for each sequin isoform relative to expected input concentration.

5. **RnaExpression_genes.R** – R script to plot the expression estimated for each sequin gene relative to expected input concentration.

**COMMENT** | *RnaExpression* requires the gene/transcript names in the user-generate transcriptome file (e.g. transcripts.gtf) to match the gene/transcript names in the mixture file. If users have performed de-novo assembly, the transcript ids between these two files may not match, and *RnaExpression* will not run correctly. In this case, we recommend that sequin transcript names be assigned to the user’s de novo transcript assembly using the *CuffCompare* tool. Please refer to Cufflinks documentation for further usage details.

**COMMENT** | In cases where a user’s does not have the supported GTF file format, they user can convert their gene expression results into a simple text file format compatible for use with *RnaExpression*. Further details on simple text file formats can be found in Appendix C.

### 5.5.8.2 | Plot Isoform/Gene Expression Curve (Anaquin, R)

**16** | To plot the measured expression of each sequin gene relative to expected input concentration in a single replicate, load the **RnaExpression_genes.R** script (plotting isoform expression is similar with the **RnaExpression_isoforms.R** script) into R (please see Appendix B.1 for details on how to load the script and plot graphs in R/R-Studio) to plot the following graph:

![Gene Expression Plot](image_url)

**Figure 5.5.8.2 Gene expression plot for single replicate.** Scatter-plot illustrates the observed abundance (in FPKM) relative to the expected abundance (in attomoles/ul). Dashed blue line (with shadow depicts the 95% confidence interval) shows linear regression model.
5.5.8.3 | Assess Multiple Replicates (Anaquin, R)

Multiple replicates can be used with RnaExpression to enable an estimate of variation. To perform the gene expression analysis with multiple replicates, we require multiple replicates to be provided to RnaExpression using the option –usequin (which must be repeated for each replicate file input).

Each individual expression replicate is given by –usequin. Perform the following command:

```
$ anaquin RnaExpression -o 5.5.8.3 -rmix RNAsequins_isofrom_mix.v2.tsv \ 
  -usequin A1/G/transcripts.gtf -usequin A2/G/transcripts.gtf \ 
  -usequin A3/G/transcripts.gtf
```

This will generate the same output results (RnaExpression_summary.stats, RnaExpression_isofroms.tsv, RnaExpression_genes.tsv) and plot (RnaExpression_isofroms.R and RnaExpression_genes.R) with additional confidence intervals, error bars etc. that are enabled by the analysis of multiple replicates. We have provided an example RnaExpression_summary.stats output with provides confidence intervals shown below.

![Gene Expression](image)

**Figure 5.5.8.3.1** Gene expression plot for multiple replicate. Scatter-plot illustrates the mean observed abundance (in FPKM) for each synthetic gene relative to the expected concentration (in attomoles/ul). Error bars indicate standard deviation, with n = 3.
Figure 5.5.8.3.2 Isoform expression plot for multiple replicate. Scatter-plot illustrates the mean observed abundance (in FPKM) for each synthetic sequin relative to the expected concentration (in attomoles/ul). Error bars indicate standard deviation, with $n = 3$. The limit-of-quantification (LOQ, indicated by dashed line) is estimated by piecewise-linear segmentation (further details available in Appendix B).

5.5.8.4 | Example output from Multiple Replicates (Anaquin, R)

---RnaExpression Output

Summary for input: transcripts.gtf, transcripts.gtf, transcripts.gtf

---Reference Transcript Annotations

Synthetic: 164 isoforms
Synthetic: 78 genes
Mixture file: RNAsequins_isoform_mix.v2.tsv

---Detected Isoforms

Sequin: 106 ± 3
Detection Sensitivity: 0.059 (attomol/ul) (R2_115_2)
Genome: 199242 ± 0

---Linear regression (Isoform expression) (log2 scale)

Slope: 1.43 ± 0.14
Correlation: 0.72 ± 0.02
R2: 0.52 ± 0.03
P-value: 0.00 ± 0.00

---Detected Genes

Sequin: 51 ± 1
Detection Sensitivity: 0.118 (attomol/ul) (R2_115)
Genome: 58205 ± 0

---Linear regression (Gene expression) (log2 scale)

Slope: 1.00 ± 0.01
Correlation: 0.93 ± 0.00
5.6 | Workflow - Differential Gene Expression (Multiple Samples)

The identification of fold-changes in gene expression between two samples is one of the most common applications of RNA-Seq. Accordingly, there are a number of different tools that measure differential gene expression in different ways. Within this workflow, we describe how to analyze the differential abundance of sequins between samples using Cuffdiff.

Transcript based approaches, such as CuffDiff, estimate the expression of each individual isoform, and assess differential expression accordingly. Users can modify the parameters or statistical models that are applied to differential gene expression, and users are advised to familiarize themselves with Cuffdiff documentation for further details.

5.6.1 | Differential analysis (Cuffdiff)

To perform differential gene analysis between two group (A and B), perform the following Cuffdiff command:

```bash
$ cuffdiff gencode_chr18.gtf \
A1/subsampled.bam,A2/subsampled.bam,A3/subsampled.bam \
B1/subsampled.bam,B2/subsampled.bam,B3/accepted_hits.bam
```

Cuffdiff will generate two files: `gene_exp.diff` and `isoform_exp.diff` in the working directory that indicate fold-changes in the expression of genes and isoforms, respectively.

5.6.2 | Assess fold-changes in gene expression (Anaquin)

Using sequins, we have a simple empirical method to assess the performance with which differential gene and isoform expression is identified. We can compare the measured fold changes (and significance) determined with Cuffdiff to the known fold-changes between sequin mixtures by performing the following command:

```bash
$ anaquin RnaFoldChange -o 5.6.2 -rmix RNAsequins_isoform_mix.v2.tsv -usequin \n  gene_exp.diff
```

Where:
- `RnaFoldChange` is name of the tool
- `5.6.2` is the specified output directory (-o)
- `RNAsequins_isoform_mix.v2.tsv` is the reference mixture file (-rmix)
- `gene_exp.diff` is the generated differential gene expression file by Cuffdiff (-usequin)

Note that for each sequin gene (R1_11_1 in this example), there are mixture A (MXA) and mixture B (MXB), both in attomol/ul.

RnaFoldChange will generate the following files in the output directory:
1. `RnaFoldChange_summary.stats` – summary statistics describing the accuracy and confidence for detecting fold changes between sequin mixtures. An example file, including a description and interpretation of statistics, is provided in Appendix A.4.
2. `RnaFoldChange_sequins.tsv` – expected and observed fold change for each individual sequin gene.
3. `RnaFoldChange_fold.R` – R script to plot the observed relative to expected fold change (see below).
4. `RnaFoldChange_ROC.R` – R script to plot ROC curve to assess the correct detection of differential gene expression, with reported p-values used to rank the points on the curve.

5.6.2.1 | Example output from RnaFoldChange

```
-------RnaFoldChange Output

Summary for input: gene_exp.diff
```

R2: 0.87 ± 0.01
P-value: 0.00 ± 0.00
5.6.2.2 | Plot observed fold-change (Anaquin, R)

To plot the observed fold-change in gene expression, load the `RnaFoldChange_fold.R` script into R (please see Appendix B for details on how to load the script and plot graphs in R/R-Studio) to plot the following graph:

```
$ anaquin RnaFoldChange -o 5.6.3 -rmix RNAsequins_isoform_mix.v2.tsv \
    -usequin isoform_exp.diff
```

Figure 5.6.2.2 Gene expression fold-change for single replicate. Scatter plot indicates correlation between observed and expected differential log-fold for each sequin gene. Sequin genes correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).

5.6.3 | Assess fold-changes in isoform expression (Anaquin)

In addition to `gene_exp.diff` (discussed in the previous section), Cuffdiff also generates `isoform_exp.diff` in the same directory. `isoform_exp.diff` reports differential analysis at the isoform level. We can reuse `RnaFoldChange` to quantify this isoform analysis.

Perform the following command quantify the isoform differential expression file.

```
$ anaquin RnaFoldChange -o 5.6.3 -rmix RNAsequins_isoform_mix.v2.tsv \
    -usequin isoform_exp.diff
```
Where:

- RnaFoldChange is name of the tool
- -o specifies the output directory (5.6.3)
- -rmix specifies the reference mixture file for Mixture A and B (RNAsequins_isoform_mix.v2.tsv)
- -usequin is the output differential isoform expression file generated by Cuffdiff (isoform_exp.diff)

The usage will generate identical files already discussed earlier.

5.6.3.1 | Plot observed fold-change (Anaquin, R)

To plot the observed fold-change, we can follow the same workflow outlined in Section 5.6.2.2. Load the RnaFoldChange_fold.R script into R (Appendix B has detailed instructions on how this can be done for R/RStudio) to plot the following graph:

![Isoform Fold Change](image)

Figure 5.6.3.1 Isoform expression fold-change for single replicate. Scatter plot indicates correlation between observed and expected differential log-fold for each sequin isoform. Sequins correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).

5.7 | Workflow - Bioconductor (DESeq2)

Here we provide a working example of a popular R package; DESeq2 to identify differential gene expression. We will illustrate how to provide a proper coordinate annotation file to Bioconductor. We will also show how to construct a counting matrix for genes (including sequin genes). Finally, we will use DESeq2 to fit a generalized linear model for differential gene analysis.

5.7.1 | Make A Count Table

Before we run differential analysis, we must count alignment reads within genomic intervals and in silico regions. Bioconductor requires a coordinate annotation GTF file; this is typically gencode annotation GTF.
file. However, we will also need to provide sequin coordinates within the in silico chromosome, thus we will supply the `gencode_chrIS.gtf` (includes gencode annotation and sequin annotation) file, as discussed in Section 5.3.1.

The first step is to construct a count table. A count table indicates the alignment distribution across the genomic regions. We first generate a count table from our previous read alignment file. We will follow the workflow recommended by Bioconductor:

http://www.bioconductor.org/help/workflows/rnaseqGene

Start an R session, and perform the following commands to indicate where our alignment files are:

```r
> a1 <- 'A1/subsampled.bam'
> a2 <- 'A2/subsampled.bam'
> a3 <- 'A3/subsampled.bam'
> b1 <- 'B1/subsampled.bam'
> b2 <- 'B2/subsampled.bam'
> b3 <- 'B3/subsampled.bam'
> files <- c(a1, a2, a3, b1, b2, b3)
```

We then create a count matrix using the `GenomicAlignments` and `GenomicFeatures` package. (Please note there are other alternative methods, such as `htseq-count`):

```r
> library('GenomicAlignments')
> library('GenomicFeatures')

We indicate our BAM files using the `BamFileList` function:

```r
> bams <- BamFileList(files)
```

We next provide the gene model that will be used for counting reads using the `makeTxDbFromGFF` function:

```r
> model <- makeTxDbFromGFF('gencode_chrIS.gtf', format='gtf')
> genes <- exonsBy(model, by='gene')
```

**CRITICAL** | File `gencode_chrIS.gtf` gives the transcriptome annotations for the human genome and the in silico regions represented by the sequins. We must provide the combined annotation GTF (containing both human and synthetic gene annotations) for counting.

We next perform counting with the `summarizeOverlaps` function:

```r
> se <- summarizeOverlaps(features=genes, reads=bams, mode='Union',
\[singleEnd=FALSE, ignore.strand=TRUE, fragments=TRUE]
```

We then complete the metadata for the experiment. For this workflow, we can create a comma-separated value (CSV) file with a text editor. The CSV file needs six rows, one for each replicate, the columns indicating the library name for each of the replicates. For example,

<table>
<thead>
<tr>
<th>Replicate,Sample</th>
</tr>
</thead>
<tbody>
<tr>
<td>A1,A</td>
</tr>
<tr>
<td>A2,A</td>
</tr>
<tr>
<td>A3,A</td>
</tr>
<tr>
<td>B1,B</td>
</tr>
<tr>
<td>B2,B</td>
</tr>
<tr>
<td>B3,B</td>
</tr>
</tbody>
</table>

Save this file in CSV format as `meta.csv`.

We then incorporate the above `meta.csv` file with the following command:
We can access the count matrix with:

```r
> counts <- assay(se)
> head(counts)
```

And write the matrix to a CSV file by:

```r
> write.csv(counts, 'data.csv')
```

Finally, read the file back and convert it to class representation:

```r
data <- read.csv('data.csv', row.names=1)
```

### 5.7.2 Identify Differential Gene Expression (DESeq2 in R)

DESeq2 is a popular package for differential analysis of count data, such as RNA-Seq alignments. We can create a DESeq2 data set and perform differential analysis using the following commands:

```r
> library('DESeq2')
> se <- DESeqDataSetFromMatrix(data, DataFrame(meta), design=~Sample)
> dds <- DESeqDataSet(se, design=~Sample)
> dds <- DESeq(dds)
> r <- results(dds, contrast=c('Sample', 'B', 'A'))
> write.csv(r, 'DESeq2.csv', quote=FALSE)
```

**CRITICAL** | The CSV file `DESeq2.csv` contains the analysis results that is required by Anaquin in the following steps.

### 5.7.3 Quantify Differential Gene Expression (Anaquin, in R)

We can repeat similar workflow discussed in Section 5.5.3 for quantifying DESeq2 analysis with synthetic sequins. Run the following Anaquin command:

```bash
$ anaquin RnaFoldChange -o 5.7.3 -rmix RNAsequins_isoform_mix.v2.tsv -usequin DESeq2.csv
```

Where:
- RnaFoldChange is name of the tool
- `-o` specifies the output directory (5.7.3)
- `-rmix` specifies the mixture reference file (RNAsequins_isoform_mix.v2.tsv)
- `-usequin` specific the differential gene expression file generated by DESeq2 (DESeq2.csv)

RnaFoldChange will generate the following files in the output directory:
1. **RnaFoldChange_summary.stats** – summary statistics describing the accuracy and confidence for detecting fold changes between sequin mixtures. An example file, including a description and interpretation of statistics, is provided in Appendix A.4.
2. **RnaFoldChange_sequins.tsv** – expected and observed fold change for each individual sequin gene.
3. **RnaFoldChange_fold.R** – R script to plot the observed relative to expected fold change (see below).
4. **RnaFoldChange_ROC.R** – R script to plot ROC curve to assess the correct detection of differential gene expression, with reported p-values used to rank the points on the curve.

#### 5.7.3.1 Example summary statistics from RnaFoldChange

```
---------RnaFoldChange Output
Summary for input: DESeq2.csv
---------Reference Annotations
```
5.7.3.2 | Plot observed fold-change (Anaquin, R)

28 | Load `RnaFoldChange_fold.R` in R (example instructions in Appendix B) to plot a scatter plot between expected fold change against measured fold change on the logarithm scale.

![Gene Fold Change](image).

*Figure 5.7.3.2 Gene expression fold-change for single replicate. Scatter plot indicates correlation between observed and expected differential log-fold for each sequin gene. Sequin genes correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).*

5.7.3.3 | Plot ROC curve (Anaquin, R)

29 | For the ROC analysis, we group together sequin genes that have expected log-fold changes (LFC) of the same magnitude, regardless of direction (i.e. genes with expected LFC of 4 and -4 are grouped together, 3 with -3, 2 with -2 and 1 with -1). We can then use the ROC curve to assess the performance with which significant fold-changes in gene expression are detected:
Figure 5.7.3.3: The ROC plot illustrates the performance of identifying variably expressed sequins (true-positives) relative to constantly expressed sequence (false-positives, no fold change), with curve ranked by significance (p-value ascribed by DESeq2). Each sequin group (grouped according to expected log fold change; LFC) is plotted independently.

In addition to plotting the ROC curves, the AUC values for each ROC curve are provided in the R console. For example, the AUC values for each sequins group (sorted by log-fold change; LFC) above are:

<table>
<thead>
<tr>
<th></th>
<th>AUC</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>0.7089</td>
</tr>
<tr>
<td>3</td>
<td>0.6980</td>
</tr>
<tr>
<td>1</td>
<td>0.6440</td>
</tr>
<tr>
<td>2</td>
<td>0.5964</td>
</tr>
</tbody>
</table>
APPENDIX A – Command Line Usage

A.1 | RnaAlign

Measure the spliced read alignments from sequins to the in silico chromosome

Overview

RnaAlign can be used to assess the alignment of sequins-derived reads to the in silico chromosome and calculates several useful statistics to describe alignment performance, including:

- **Dilution** indicates fraction of reads that align to the in silico chromosome, relative to the accompanying genome
- **Sensitivity** indicates the fraction of annotated regions covered by alignments
- **Precision** indicates the accuracy of alignments

These statistics are calculated at a nucleotide, exon and intron level. A further description of diagnostic statistics is provided in Appendix C.

Support Software

Spliced-read aligner that generates a SAM/BAM alignment file. Common examples include TopHat2 and STAR.

Inputs

| Reference transcriptome annotation file in GTF format |
| Generated alignment file in SAM/BAM format |

Usage Example

anaquin RnaAlign -rgtf reference.gtf -usequin aligned.bam

Additional Information

The runtime is linearly proportional to the number of reads in the input file, with long run times expected for large alignment files.

Tool Options

Required:

| -rgtf | Reference transcriptome annotation file in GTF format |
| -usequin | Generated alignment file in SAM/BAM format |

Optional:

| -o = output | Directory in which the output files are written to |
Outputs

RnaAlign_summary.stats - provides useful statistics to describe the global alignment profile. Field definitions:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input alignment file</td>
<td>Input sample alignment file in SAM/BAM format</td>
</tr>
<tr>
<td>Reference annotation file</td>
<td>Reference annotation file in GTF format</td>
</tr>
<tr>
<td>Synthetic</td>
<td>Number of alignments mapped to the <em>in silico</em> chromosome</td>
</tr>
<tr>
<td>Genome</td>
<td>Number of alignments mapped to the human genome</td>
</tr>
<tr>
<td>Dilution</td>
<td>Proportion of alignments mapped to the <em>in silico</em> chromosome</td>
</tr>
<tr>
<td>Reference annotation file</td>
<td>Number of exons, introns and bases on the <em>in silico</em> chromosome and genome</td>
</tr>
<tr>
<td>Non-spliced</td>
<td>Number of non-spliced reads on the <em>in silico</em> chromosome and genome</td>
</tr>
<tr>
<td>Spliced</td>
<td>Number of spliced reads for on the <em>in silico</em> chromosome and genome</td>
</tr>
<tr>
<td>Total</td>
<td>Spliced + Non-spliced</td>
</tr>
<tr>
<td>Sensitivity</td>
<td>Sensitivity of the alignments at levels: exon, intron and base (in silico</td>
</tr>
<tr>
<td></td>
<td>chromosome and genome</td>
</tr>
<tr>
<td>Precision</td>
<td>Precision of the alignments at levels: intron and base (in silico chromosome and genome)</td>
</tr>
</tbody>
</table>

RnaAlign_sequins.tsv – provides statistics for each individual sequin gene. Field definitions:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the sequin gene</td>
</tr>
<tr>
<td>Length</td>
<td>Length of the sequin gene</td>
</tr>
<tr>
<td>Reads</td>
<td>Number of reads aligned to sequin gene</td>
</tr>
<tr>
<td>SnIntron</td>
<td>Sensitivity at the intron level</td>
</tr>
<tr>
<td>SnBase</td>
<td>Sensitivity at the base level</td>
</tr>
</tbody>
</table>

Example Output – RnaAlign_summary.stats

-------RnaAlign Summary Statistics

Input alignment file: A1/accepted_hits.bam
Reference annotation file: gencode_chrIS.gtf

-------Number of alignments mapped to the synthetic chromosome and genome

Synthetic: 2944809
Genome: 64545551
Dilution: 0.044

-------Reference annotation (Synthetic)

Synthetic: 869 exons
Synthetic: 754 introns
Synthetic: 5490967 bases

-------Reference annotation (Genome)

Genome: 570980 exons
Genome: 347657 introns
Genome: 1758049931 bases

-------Alignments (Synthetic)

Non-spliced: 1760733
Spliced: 1240992

-------Alignments (Genome)

Non-spliced: 55524175
Spliced: 20265528

-------Comparison of alignments to reference annotation (Synthetic)
<table>
<thead>
<tr>
<th>Level</th>
<th>Sensitivity</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>*Intron level</td>
<td>0.73</td>
<td>0.88</td>
</tr>
<tr>
<td>*Base level</td>
<td>0.73</td>
<td>0.95</td>
</tr>
</tbody>
</table>
A.2 | RnaAssembly

Compares assembled transcript models to sequin annotations in the *in silico* chromosome.

Overview

We can use *RnaAssembly* to compare the assembly of spliced isoform assemblies to known synthetic gene annotations in the *in silico* chromosome. We provide quantitative statistics at exon, intron, intron-chain, transcript and nucleotide level using previous definition by Trapnel et al., 2010. The following statistics are provided:

**Sensitivity** - the fraction of annotated features that are detected by assembly. For example, if a transcript has 10 introns, of which 7 are assembled, the sensitivity (at intron level) will be 0.7.

**Precision** – the fraction of correctly assembled features relative to the total number of assembled features (both true and false positive). This provides an indication of assembly accuracy. For example, if 10 introns are identified by assembly, but 4 are erroneous, we would have a precision of 0.6.

A further description of diagnostic statistics is provided in Appendix C.

Support Software

Any software that assembles transcript models from alignments, with common examples including *Cufflinks* and *StringTie*. Assembled transcript models must be provided to *Anaquin* in GTF file format.

Inputs

- Reference annotation file in GTF format
- Reference RnaQuin mixture file in TSV format
- Generated transcriptome file in GTF format

Usage Example

```
anaquin RnaAssembly -rmix mixture.tsv -rgtf reference.gtf \
-usequin transcripts.gtf
```

Tool Options

**Required:**

- `-rmix` Reference mixture file in TSV format
- `-rgtf` Reference annotation file in GTF format.
- `-usequin` Generated transcriptome in GTF format.

**Optional:**

- `-o = output` Directory in which the output files are written to.
- `-mix = A` Mixture to use. “A” or “B”.

Outputs

`RnaAssembly_summary.stats` provides useful statistics to describe the global alignment profile. Field definitions:

<table>
<thead>
<tr>
<th>Field Definition</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>User assembly file</td>
<td>Input sample transcriptome file in GTF format</td>
</tr>
<tr>
<td>Reference annotation file</td>
<td>Reference transcriptome annotation file in GTF format</td>
</tr>
<tr>
<td>Reference Gene Annotations (Synthetic)</td>
<td>Number of exons, introns, isoforms and genes on the <em>in silico</em> chromosome in the reference annotation</td>
</tr>
<tr>
<td>Reference Gene Annotations (Genome)</td>
<td>Number of exons, introns, isoforms and genes on the genome in the reference annotation</td>
</tr>
<tr>
<td>Synthetic (User Assemblies)</td>
<td>Number of exons, introns, isoforms and genes on the <em>in silico</em> chromosome in the user assembly file</td>
</tr>
</tbody>
</table>
Genome (User Assemblies) | Number of exons, introns, isoforms and genes on the genome in the user assembly file
--- | ---
Sensitivity | Sensitivity for exon, intron, base, intron-chain and transcript levels for both the in silico chromosome and the genome
Specificity | Specificity for exon, intron, base, intron-chain and transcript levels for both the in silico chromosome and the genome
Missing | Missing exons and introns for both the in silico chromosome and the genome
Novel | Novel exons and introns for both the in silico chromosome and the genome

RnaAssembly_sequins.tsv – provides statistics for each individual sequin. Field definitions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the sequin</td>
</tr>
<tr>
<td>Length</td>
<td>Length of the sequin</td>
</tr>
<tr>
<td>Input</td>
<td>Input concentration in attomol/ul</td>
</tr>
<tr>
<td>Sn</td>
<td>Sensitivity of the sequin</td>
</tr>
</tbody>
</table>

RnaAssembly_assembly.R - R-script for building a non-linear model between sensitivity (dependent variable) and input concentration (independent variable). Useful for visualizing the abundance dependent bias and assembly limit for a library.

Additional Information

Internally, the tool embeds the CuffDiff (http://cole-trapnell-lab.github.io/cufflinks) software for quantifying a transcriptome GTF file. The results are exactly identical to what CuffDiff reports. For additional detail on the definition and method for comparing transcript models, please refer to:


Example Output – RnaAssembly_summary.stats

-------RnaAssembly Summary Statistics

User assembly file: A1/D/transcripts.gtf
Reference annotation file: gencode_chrIS.gtf

-------Reference Gene Annotations (Synthetic)

Synthetic: 869 exons
Synthetic: 754 introns
Synthetic: 164 isoforms
Synthetic: 78 genes

-------Reference Gene Annotations (Genome)

Genome: 570980 exons
Genome: 347657 introns
Genome: 199169 isoforms
Genome: 60554 genes

-------User Assemblies (Synthetic)

Synthetic: 491 exons
Synthetic: 431 introns
Synthetic: 88 isoforms
Synthetic: 59 genes

-------User Assemblies (Genome)

Genome: 152635 exons
Genome: 108561 introns
Genome: 49049 isoforms
Genome: 40800 genes

-------Comparison of assembly to annotations (Synthetic)

*Exon level
Sensitivity: 0.542002
Specificity: 0.959267

*Intron
Sensitivity: 0.564987
Specificity: 0.988399

*Base level
Sensitivity: 0.589830
Specificity: 0.999147

*Intron Chain
Sensitivity: 0.288462
Specificity: 0.542169

*Transcript level
Sensitivity: 0.000000
Specificity: 0.000000

Missing exons: 343
Missing introns: 276

Novel exons: 0
Novel introns: 1

-------Comparison of assembly to annotations (Genome)

*Exon level
Sensitivity: 0.212510
Specificity: 0.789939

*Intron
Sensitivity: 0.295041
Specificity: 0.944575

*Base level
Sensitivity: 0.296238
Specificity: 0.434351

*Intron Chain
Sensitivity: 0.048323
Specificity: 0.426736

*Transcript level
Sensitivity: 0.000005
Specificity: 0.000005

Missing exons: 314342
Missing introns: 186569

Novel exons: 24068
Novel introns: 1322
A.3 | RnaExpression

Quantitative analysis of sequin expression

Overview

*RnaExpression* can be used for analyzing the gene or isoform expression of RNA sequins within a library. Comparing the measured expression (typically in FPKM) relative to the known input concentrations provides an indication of the quantitative accuracy for measuring gene expression.

Specifically, *RnaExpression* builds a linear model regressing the measured expression (dependent variable) with the input concentration (independent variable; defined by the mixture). Singular Value Decomposition (SVD) is used to estimate the regression parameters, including:

- **Correlation** – provides a measure of quantitative accuracy across a range of input concentrations.
- **Slope** – indicates the quantitative linearity for gene expression measures.
- **Coefficient of determination** (R2) – indicates the amount of variation that can be accounted for by the linear model.

A further description of diagnostic statistics is provided in Appendix C. *RnaExpression* supports either single replicate or multiple replicate libraries. If multiple libraries are provided, *RnaExpression* will report statistics with standard deviation indicated.

Support Software

*RnaExpression* is compatible with many popular gene expression tools, including: *Cufflinks*, *StringTie* and *Kallisto*. In addition, *RnaExpression* can be used in R for data visualization. Users of alternative software may need to modify their results to conform with file formats produced by these popular software tools, before provision to the tool.

Inputs

- Reference RnaQuin mixture file in TSV format
- Generated express file in GTF format or Anaquin format

Details about the Anaquin format can be found in Appendix D.

Usage Example

For single replicate:

```
anquin RnaExpression -rmix mixture.tsv -usequin genes.gtf
```

For multiple replicates:

```
anquin RnaExpression -rmix mixture.tsv -usequin A1.gtf -usequin A2.gtf
```

Tool Options

Required:

- `-rmix` | Reference RnaQuin mixture file in TSV format
- `-usequin` | Generated expression file in GTF or text format

Optional:

- `-o = output` | Directory in which the output files are written to
- `-mix = A` | Mixture to use. “A” or “B”.

© 2017 Garvan Institute. All rights reserved.
Outs

RnaExpression_summary.stats – reports global summary statistics determined from all sequins.

Field definitions:

<table>
<thead>
<tr>
<th>Input file</th>
<th>User generated expression file in GTF format or text format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference annotations</td>
<td>Number of genes/isoforms on the in silico chromosome in the reference annotation</td>
</tr>
<tr>
<td>Mixture file</td>
<td>Reference RnaQuin mixture file in TSV format</td>
</tr>
<tr>
<td>Synthetic</td>
<td>Number of genes/isoforms on the in silico chromosome in the input expression file</td>
</tr>
<tr>
<td>Detection Sensitivity</td>
<td>Sequin with the lowest input that are detected</td>
</tr>
<tr>
<td>Genome</td>
<td>Number of genes/isoforms on the genome in the input expression file</td>
</tr>
<tr>
<td>Correlation</td>
<td>Pearson’s correlation of the linear model</td>
</tr>
<tr>
<td>Slope</td>
<td>Regression slope of the linear model</td>
</tr>
<tr>
<td>R2</td>
<td>Coefficient of determination of the linear model</td>
</tr>
<tr>
<td>P-value</td>
<td>P-value probability</td>
</tr>
</tbody>
</table>

RnaExpression_isoforms.tsv - detailed statistics for each sequin isoform in the reference. Field definitions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Name of the sequin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Length of the sequin</td>
</tr>
<tr>
<td>Input</td>
<td>Input concentration in attomol/ul</td>
</tr>
<tr>
<td>Observed</td>
<td>Observed expression (eg. FPKM)</td>
</tr>
</tbody>
</table>

RnaExpression_genes.tsv - detailed statistics for each sequin gene in the reference. Field definitions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Name of the sequin</th>
</tr>
</thead>
<tbody>
<tr>
<td>Length</td>
<td>Length of the sequin</td>
</tr>
<tr>
<td>Input</td>
<td>Input concentration in attomol/ul</td>
</tr>
<tr>
<td>Observed</td>
<td>Observed expression (eg. FPKM)</td>
</tr>
</tbody>
</table>

RnaExpression_isoforms.R - R-script for plotting isoform expression analysis for sequins between expression level (dependent variable) and input concentration (independent variable) on the logarithm scale.

RnaExpression_genes.R - R-script for plotting gene expression analysis for sequins between expression level (dependent variable) and input concentration (independent variable) on the logarithm scale.

Example Output – RnaExpression_summary.stats

```
-------- RnaExpression Output
Summary for input: transcripts.gtf, transcripts.gtf, transcripts.gtf
-------- Reference Transcript Annotations
Synthetic: 164 isoforms
Synthetic: 78 genes
Mixture file: RNAsequins_isoform_mix.v2.tsv
-------- Detected Isoforms
Sequin: 106 ± 3
Detection Sensitivity: 0.059 (attomol/ul) (R2_115_2)
Genome: 199242 ± 0
-------- Linear regression (Isoform expression) (log2 scale)
Slope: 1.43 ± 0.14
```
Correlation: 0.72 ± 0.02  
R2: 0.52 ± 0.03  
P-value: 0.00 ± 0.00

--------Detected Genes

Sequin: 51 ± 1  
Detection Sensitivity: 0.118 (attomol/ul) (R2_115)  
Genome: 0 ± 0

--------Linear regression (Gene expression) (log2 scale)

Slope: 1.00 ± 0.01  
Correlation: 0.93 ± 0.00  
R2: 0.87 ± 0.01  
P-value: 0.00 ± 0.00
A.4 | RnaFoldChange

Assess fold-changes in gene expression between multiple samples

Overview

*RnaFoldChange* can be used to analyse the differential expression of sequins between different mixtures that have been alternately spiked-in to multiple samples.

The differential expression of sequins is emulated by modulating the relative concentration of sequins between alternative mixtures (such as between Mixture A and B) and provides a known reference scale of differential expression between samples. This scale can be used to assess the measurement of fold-changes in gene expression between RNA-Seq libraries, and estimate diagnostic power and confidence limits.

Specifically, *RnaFoldChange* builds a linear model regressing the measured log-fold (dependent variable; provided by third party tool such as Cufflinks) to the input concentration (independent variable) defined by a mixture. Singular Value Decomposition (SVD) is used to estimate the regression parameters.

*RnaFoldChange* can also be used to assess the diagnostic performance for detecting sequin fold-change between mixtures with receiver operating characteristic (ROC) curves and area under the curve (AUC) statistics.

Support Software

*RnaFoldChange* is compatible with many popular gene expression tools, including: *Cuffcompare*. In addition, *RnaFoldChange* can be used in R and is compatible with DESeq2. Users of alternative software may need to modify their results to conform with file formats produced by these popular software tools, before provision to *RnaExpression*.

Inputs

| Reference RnaQuin mixture in TSV format (must have two columns, for input concentration defined for mixture A and B respectively) |

Details on the *Anaquin* format is available in Appendix D.

Usage Example

```bash
anaquin RnaFoldChange -rmix mixture.tsv -usequin diff.txt
```

Additional Information

For additional information on the use of ROC/LODR plots to assess differential gene expression, please refer to:


Tool Options

**Required:**

- `-rmix` | Reference RnaQuin mixture file in TSV format
- `-usequin` | User generated differential analysis file

**Optional:**

- `-o = output` | Directory in which the output files are written to

Outputs

*RnaFoldChange summary.stats* — provides global summary statistics of sequin fold-change expression. Field definitions:

© 2017 Garvan Institute. All rights reserved.
<table>
<thead>
<tr>
<th>Input file</th>
<th>User generated differential analysis file</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reference (Synthetic)</td>
<td>Number of genes/isoforms detected on the <em>in silico</em> chromosome</td>
</tr>
<tr>
<td>Mixture</td>
<td>Reference RnaQuin mixture file</td>
</tr>
<tr>
<td>Synthetic (Expressed)</td>
<td>Number of genes/isoforms expressed on the <em>in silico</em> chromosome</td>
</tr>
<tr>
<td>Genome (Expressed)</td>
<td>Number of genes/isoforms expressed on the genome</td>
</tr>
<tr>
<td>Slope</td>
<td>Regression slope of the linear model</td>
</tr>
<tr>
<td>Correlation</td>
<td>Pearson’s correlation of the linear model</td>
</tr>
<tr>
<td>R2</td>
<td>Coefficient of determination of the linear model</td>
</tr>
<tr>
<td>P-value</td>
<td>P-value under the null hypothesis</td>
</tr>
</tbody>
</table>

**RnaFoldChange_sequins.tsv** — provides statistics for individual sequin in the reference. Field definitions:

<table>
<thead>
<tr>
<th>Name</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Name of the sequin</td>
</tr>
<tr>
<td>Length</td>
<td>Length of the sequin</td>
</tr>
<tr>
<td>Sample1</td>
<td>Expression level for the first sample</td>
</tr>
<tr>
<td>Sample2</td>
<td>Expression level for the second sample</td>
</tr>
<tr>
<td>ExpLFC</td>
<td>Expected log-fold change</td>
</tr>
<tr>
<td>ObsLFC</td>
<td>Observed log-fold change</td>
</tr>
<tr>
<td>SD</td>
<td>Standard deviation of the measurement</td>
</tr>
<tr>
<td>Pval</td>
<td>Observed p-value probability</td>
</tr>
<tr>
<td>Qval</td>
<td>Observed q-value probability</td>
</tr>
<tr>
<td>Average</td>
<td>Normalized counts of all samples</td>
</tr>
</tbody>
</table>

**RnaFoldChange_fold.R** - R script for building a linear model between measured log-folds (dependent variable) and expected log-folds (independent variable).

**RnaFoldChange_ROC.R** - R script for building the ROC curve (including AUC statistics) for each sequin group.

**Example Output** — **RnaFoldChange_summary.stats**

```plaintext
-------RnaFoldChange Output

       Summary for input: DESeq2.tsv
-------Reference Annotations

       Synthetic: 78 genes
       Mixture file: M.R.14.tsv
-------Genes Expressed

       Synthetic: 78 genes
       Genome: 60554 genes
-------Linear regression (log2 scale)

       Slope: 0.949774
       Correlation: 0.883807
       R2: 0.781115
       P-value: 0
```
A.5 | RnaSubsample

Calibrate sequence coverage of sequins across multiple replicates

Overview

*RnaSubsample* calibrates sequin coverage across multiple RNA-Seq replicates. The tool is useful to ensure the sequencing depth comparable across different libraries, even when varying amounts of RNA sequins have been spiked-in to replicates or samples.

*RnaSubsample* requires users to specify a preferred dilution fraction (we typically recommend between 1% to 10%). The tool then titrates the alignments to the *in silico* chromosome relative to the alignments to the genome according to the user-determine dilution fraction.

Support Software

Any short-reads spliced aligner that generates SAM/BAM outputs. Common examples include *TopHat2* and *STAR*.

Inputs

User generated SAM/BAM alignment file

Usage Example

```
anasquin RnaSubsample -method 0.02 -usequin alignment.bam
```

Tool Options

Required:

- `-method` Dilution fraction as a floating number. For example, 0.01 is 1% and 0.10 is 10% etc.
- `-usequin` User generated SAM/BAM alignment file

Optional:

- `-o = output` Directory in which the output files are written to

Outputs

*RnaSubsample_summary.stats* — reports summary statistics for the subsampling. Field definitions:

<table>
<thead>
<tr>
<th>User generated alignment</th>
<th>User generated alignment file in SAM/BAM format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synthetic (Before)</td>
<td>Number of alignments mapped to the <em>in silico</em> chromosome before subsampling</td>
</tr>
<tr>
<td>Genome (Before)</td>
<td>Number of alignments not mapped to the <em>in silico</em> chromosome before subsampling</td>
</tr>
<tr>
<td>Dilution (Before)</td>
<td>Synthetic (Before) / (Synthetic (Before) + Genome (Before))</td>
</tr>
<tr>
<td>User Dilution</td>
<td>User specified dilution (specified by <code>-method</code>)</td>
</tr>
<tr>
<td>Normalization</td>
<td>Calculated normalization factors applied in subsampling <em>in silico</em> alignments</td>
</tr>
<tr>
<td>Synthetic (After)</td>
<td>Number of alignments mapped to the <em>in silico</em> chromosome after subsampling</td>
</tr>
<tr>
<td>Genome (After)</td>
<td>Number of alignments not mapped to the <em>in silico</em> chromosome after subsampling</td>
</tr>
<tr>
<td>Dilution (After)</td>
<td>Synthetic (After) / (Synthetic (After) + Genome (After))</td>
</tr>
</tbody>
</table>

Example Output — *RnaSubsample_summary.stats*

```
-------RnaSubsample Summary Statistics

    User generated alignment: A1/accepted_hits.bam

-------User alignments (before subsampling)
```
Synthetic: 2944809 reads
Genome: 64545551 reads
Dilution: 0.043633

* Dilution specified by the user:
  Fraction: 0.020000

* Normalization applied in subsampling:
  Normalization: 0.447315

------- User alignments (after subsampling)

Synthetic: 1315370 reads
Genome: 64545551 reads
Dilution: 0.0199719
APPENDIX B – R Usage

B.1 | Open & load R scripts

B.1.1 | Use RStudio

1 | The latest version of the RStudio can be downloaded at: https://www.rstudio.com/products/rstudio/download. Follow the instructions to complete the installation.

2 | In the File menu, click Open File. Navigate and find the script file.

**CRITICAL**: Users may need to change the path encoded in the script file. For example, if the output directory of the script file is: `~/Documents`, we might see something like the following in the script:

```r
$ data <- read.csv('~/Documents/RnaExpression_sequins.tsv', row.names=1, sep='\t')
```

The path would not be valid if we move it to somewhere else, say to `~/Desktop`. We should update the path to:

```r
$ data <- read.csv('~/Desktop/RnaExpression_sequins.tsv', row.names=1, sep='\t')
```

3 | Click the Source button in the top-panel to run the script.

![Source button in RStudio](image)

**Figure B.1.1**: The Source button in RStudio. Open the script file, then click the button will allow RStudio to execute the script.

4 | The graph will be shown on the Plot panel in RStudio.

B.1.2 | Use command-line

1 | The latest version of the R command-line can be downloaded at: https://cran.rstudio.com. Select your operating system and follow the instructions to complete the installation.

2 | Navigate to the parent directory of where the script file is.

3 | Perform the following command to execute the script (where script.R is the file name):

```bash
$ R CMD BATCH script.R
```

4 | R will write outputs to Rplots.pdf. Further information on usage for the BATCH command can be found in the R manual here: https://stat.ethz.ch/R-manual/R-devel/library/utils/html/BATCH.html

B.1.3 | Modify R-script

The R-scripts that generated by Anaquin is entirely customizable. For example, user’s can modify the ranking of values for an ROC plot to illustrate the impact of filtering criteria on diagnostic power (this example comes from Section 5.6.3):

1 | Open a new R-session and load sample data:

```r
library(Anaquin)
#
# Data set generated by DESeq2 and Anaquin. described in Section 5.6.3.3 of
# the user guide.
#
```
Create a ROC plot with the following R-script:

```R
# Sequin names
seqs <- row.names(UserGuideData_5.6.3)

# Expected log-fold
group <- abs(UserGuideData_5.6.3$ExpLFC)

# How the ROC curves are ranked
score <- 1 - UserGuideData_5.6.3$Pval

# Classified labels (TP/FP)
label <- UserGuideData_5.6.3$Label

plotROC(seqs, score, group, label, title='ROC Plot', refGroup=0)
```

**Figure B.1.3.A** The ROC plot illustrates the performance of identifying variably expressed sequins (true-positives) relative to constantly expressed sequence (false-positives, no fold change), with curve ranked by significance (p-value ascribed by DESeq2). Each sequin group (grouped according to expected log fold change; LFC) is plotted independently.

2 | We can filter out sequins with expected LFC equal to 4 with the following R-code:
> UserGuideData_5.6.3 <- UserGuideData_5.6.3[abs(UserGuideData_5.6.3$ExpLFC) != 4,]

Re-run the script and we should see the following R-plot (LFC 4 is now removed):

![Gene Fold Change](image)

**Figure B.1.3.B:** The ROC plot illustrates the performance of identifying variably expressed sequins (true-positives) relative to constantly expressed sequence (false-positives, no fold change), with curve ranked by significance (p-value ascribed by DESeq2). Each sequin group (grouped according to expected log fold change; LFC) is plotted independently.

3 | The script uses the p-value for ranking the ROC curves. But, we can also rank by any other numerical variable. For example, we can adjust the p-value with bonferroni correction and use them for ranking. In R, type the following:

```r
> adjust <- p.adjust(UserGuideData_5.6.3$Pval, method='bonferroni')
> plotROC(seqs, 1-adjust, group, label, title='ROC Plot', refGroup=0)
```

Note the second argument= is now 1-adjust. Run the script again to regenerate new ROC plot.
B.2 | plotROC

Description
The plotROC function creates a receiver operating characteristic (ROC) plot at various threshold settings. The true positive rate (TPR) is plotted on the x-axis and false positive rate (FPR) is plotted on the y-axis. The plotROC function requires a scoring threshold function, and illustrates the performance of the data as the threshold is varied. Common scoring threshold include p-value, sequencing depth and allele frequency, etc.

ROC plot is a useful diagnostic performance tool; it provides tools to select possibly optimal models and to discard suboptimal ones. In particularly, the AUC statistics indicate the performance of the model relatively to a random experiment (AUC 0.5).

Usage
plotROC(seqs, score, group, label, refGroup, title, legTitle)

Arguments

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>seqs</td>
<td>Sequin names</td>
</tr>
<tr>
<td>score</td>
<td>How to rank ROC points</td>
</tr>
<tr>
<td>group</td>
<td>How to group ROC points</td>
</tr>
<tr>
<td>label</td>
<td>True-positive (TP) or false positive (FP)</td>
</tr>
<tr>
<td>refGroup</td>
<td>Reference ratio groups</td>
</tr>
<tr>
<td>title</td>
<td>Title of the plot. (Default to NULL).</td>
</tr>
<tr>
<td>legTitle</td>
<td>Title of the legend. (Default to ‘Ratio’).</td>
</tr>
</tbody>
</table>

Example

library(Anaquin)

# Data set generated by DESeq2 and Anaquin, described in Section 5.6.3.3 of # the user guide.
#
data(UserGuideData_5.6.3.3)

# Sequin names
seqs <- row.names(UserGuideData_5.6.3.3)

# Expected log-fold
ratio <- UserGuideData_5.6.3.3$ExpLFC

# How the ROC curves are ranked
score <- 1-UserGuideData_5.6.3.3$Pval

# Classified labels (TP/FP)
label <- UserGuideData_5.6.3.3$Label

plotROC(seqs, score, group, label, title='ROC Plot', refGroup=0)
Figure B.2: The ROC plot illustrates the performance of identifying variably expressed sequins (true-positives) relative to constantly expressed sequence (false-positives, no fold change), with curve ranked by significance (p-value ascribed by DESeq2). Each sequin group (grouped according to expected log fold change; LFC) is plotted independently.
B.3 | plotLinear

The `plotLinear` function creates a scatter plot with expected abundance on the x-axis, and measured abundance on the y-axis. The expected abundance is typically the input concentration of sequins in the mixture, although other measures (such as expected allele frequency) are also possible. The function builds a linear regression between the two variables, and reports associated statistics (R2, correlation and regression parameters) on the plot.

The `plotLinear` function also estimates limit-of-quantification (LOQ) breakpoint, and reports it on the plot if found. LOQ is defined as the lowest empirical detection limit, a threshold value beyond which stochastic behavior occur. LOQ is estimated by fitting segmented linear regression with two segments on the entire data set, while minimizing the total sum of squares of the differences between the variables.

**USAGE**

```
plotLinear(seqs, x, y, title, xlab, ylab, showSD, showLOQ, showStats, xBreaks, yBreaks, errors, showLinear, showAxis)
```

**ARGUMENTS**

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>seqs</td>
<td>Sequin names</td>
</tr>
<tr>
<td>x</td>
<td>Input concentration on the x-axis</td>
</tr>
<tr>
<td>y</td>
<td>Measurement on the y-axis</td>
</tr>
<tr>
<td>std</td>
<td>Standard deviation. (Default to NULL).</td>
</tr>
<tr>
<td>title</td>
<td>Label of the plot. (Default to NULL).</td>
</tr>
<tr>
<td>xlab</td>
<td>Label for the x-axis. (Default to NULL).</td>
</tr>
<tr>
<td>ylab</td>
<td>Label for the y-axis. (Default to NULL).</td>
</tr>
<tr>
<td>xBreaks</td>
<td>Breaks for the x-axis. (Default to NULL).</td>
</tr>
<tr>
<td>yBreaks</td>
<td>Breaks for the y-axis. (Default to NULL).</td>
</tr>
<tr>
<td>showSD</td>
<td>Display vertical standard deviation bars. (Default to FALSE).</td>
</tr>
<tr>
<td>showLOQ</td>
<td>Display limit of quantification. (Default to TRUE).</td>
</tr>
<tr>
<td>showStats</td>
<td>Display regression statistics. (Default to TRUE).</td>
</tr>
<tr>
<td>errors</td>
<td>How error bars are calculated. “SD” or “Range”.</td>
</tr>
<tr>
<td>showLinear</td>
<td>Display regression line. (Default to TRUE).</td>
</tr>
<tr>
<td>showAxis</td>
<td>Display x-axis and y-axis. (Default to TRUE).</td>
</tr>
</tbody>
</table>

**Example**

```r
library(Anaquin)

# Data set generated by Cufflinks and Anaquin, described in Section 5.4.6.3 of
# the user guide.
#
data(UserGuideData_5.4.6.3)

title <- 'Gene Expression'
xlab <- 'Input Concentration (log2)'
ylab <- 'FPKM (log2)'

# Sequin names
seqs <- row.names(UserGuideData_5.4.6.3)

# Input concentration
```
```r
x <- log2(UserGuideData_5.4.6.3$InputConcent)

# Measured FPKM
y <- log2(UserGuideData_5.4.6.3[,2:4])

plotLinear(seqs, x, y, title=title, xlab=xlab, ylab=ylab, showLOQ=TRUE)
```

**Output**

![Gene Expression](image)

**Figure B4: Sample output for the PlotLinear R-function.** Scatter-plot illustrates the mean observed abundance (in FPKM) for each synthetic sequin relative to the expected concentration (in attomoles/ul). Error bars indicate standard deviation.
**B.4 | plotLogistic**

The `plotLogistic` function creates a scatter plot with input concentration on the x-axis, and measured proportion on the y-axis. Common measured statistics include p-value, percentage and sensitivity. The plot builds a GLM logistic regression model between the two variables.

The function also estimates limit-of-assembly (LOA) breakpoint, and reports it on the plot if found. The LOA breakpoint is an empirical detection limit, and also the abundance whereby the fitted logistic curve exceeds a user-defined threshold.

**Usage**

```
plotLogistic(seqs, x, y, title, xlab, ylab, showLOA, threshold)
```

**Arguments**

<table>
<thead>
<tr>
<th>NAME</th>
<th>DESCRIPTION</th>
</tr>
</thead>
<tbody>
<tr>
<td>seqs</td>
<td>Sequin names</td>
</tr>
<tr>
<td>x</td>
<td>Expected input concentration on the x-axis</td>
</tr>
<tr>
<td>y</td>
<td>Measured proportion on the y-axis</td>
</tr>
<tr>
<td>title</td>
<td>Title of the plot. (Default to NULL).</td>
</tr>
<tr>
<td>xlab</td>
<td>Label for the x-axis. (Default to NULL).</td>
</tr>
<tr>
<td>ylab</td>
<td>Label for the y-axis. (Default to NULL).</td>
</tr>
<tr>
<td>showLOA</td>
<td>Display limit-of-assembly. (Default to TRUE).</td>
</tr>
<tr>
<td>threshold</td>
<td>Threshold required for limit-of-assembly (LOA). (Default to 0.7).</td>
</tr>
</tbody>
</table>

**Example**

```
library(Anaquin)

# Data set generated by Cufflinks and Anaquin, described in Section 5.4.5.1 of
# the user guide.
#
data(UserGuideData_5.4.5.1)

title <- 'Assembly Plot'
xlab <- 'Input Concentration (log2)'
ylab <- 'Sensitivity'

# Sequin names
seqs <- row.names(UserGuideData_5.4.5.1)

# Input concentration
x <- log2(UserGuideData_5.4.5.1$InputConcent)

# Measured sensitivity
y <- UserGuideData_5.4.5.1$Sn

plotLogistic(seqs, x, y, title=title, xlab=xlab, ylab=ylab, showLOA=TRUE)
```
Figure 86: Sample output for the PlotLogistic R-function. Scatter-plot illustrates the assembly sensitivity of each sequin, relative to input concentration. The assembly of isoforms exhibits an expression-dependent sigmoidal relationship with input concentration. Fitting is performed by non-linear least square fitting on sigmoid function. The minimal concentration required to assemble isoforms according to a user-specific assembly threshold is indicated (for example, dashed line corresponds to 0.70 sensitivity).
APPENDIX C – Simple Text Input Formats

Whilst Anaquin supports the common file formats (e.g. SAM, BAM, VCF etc.), there are numerous bioinformatics tools that do not generate these standardized format. In this case, user’s can generate unsupported file formats to a simple text formats that is compatible with Anaquin usage.

Below we have described the structure of a range of simple text formats. User’s simply need to convert their unsupported file to one of these simple text formats for downstream use with Anaquin. The simple text formats are tab-delimited and designed to be simple, easily read and parsed. The ‘-‘ character can be used to when a value is unavailable.

CRITICAL: The first line of the file (apply to all Anaquin formats) must be the header. For example, the simple text format for RnaExpression has the following header:

<table>
<thead>
<tr>
<th>ChrID</th>
<th>GeneID</th>
<th>IsoformID</th>
<th>Abund</th>
</tr>
</thead>
</table>

C.1 RnaExpression Format

In this manual, we have discussed how Anaquin can be used for GTF transcriptome input files. However, there are many bioinformatics tools generate non-GTF outputs. Anaquin is not able to parse those files unless they are converted to a standard format, defined in Section D and reproduced below.

<table>
<thead>
<tr>
<th>ChrID</th>
<th>Name of the chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneID</td>
<td>Name of the gene</td>
</tr>
<tr>
<td>IsoformID</td>
<td>Name of the isoform</td>
</tr>
<tr>
<td>Abund</td>
<td>Normalized abundance (eg. FPKM, normalized k-mer counts etc.). Repeat the column for multiple replicates (see below for an example).</td>
</tr>
</tbody>
</table>

Example lines from single replicate:

<table>
<thead>
<tr>
<th>ChrID</th>
<th>GeneID</th>
<th>IsoformID</th>
<th>Abund</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>ENSG00000237613.2</td>
<td>-</td>
<td>0.317236</td>
</tr>
<tr>
<td>chr1</td>
<td>ENSG00000268020.3</td>
<td>-</td>
<td>0.17778</td>
</tr>
<tr>
<td>chrIS</td>
<td>R2_73</td>
<td>-</td>
<td>0.376111</td>
</tr>
<tr>
<td>chrIS</td>
<td>R2_71</td>
<td>-</td>
<td>0.252222</td>
</tr>
</tbody>
</table>

Example lines from multiple replicates:

<table>
<thead>
<tr>
<th>ChrID</th>
<th>GeneID</th>
<th>IsoformID</th>
<th>Abund1</th>
<th>Abund2</th>
</tr>
</thead>
<tbody>
<tr>
<td>chr1</td>
<td>ENSG00000237613.2</td>
<td>-</td>
<td>0.317236</td>
<td>0.417236</td>
</tr>
<tr>
<td>chr1</td>
<td>ENSG00000268020.3</td>
<td>-</td>
<td>0.17778</td>
<td>0.8878</td>
</tr>
<tr>
<td>chrIS</td>
<td>R2_73</td>
<td>-</td>
<td>0.376111</td>
<td>-</td>
</tr>
<tr>
<td>chrIS</td>
<td>R2_71</td>
<td>-</td>
<td>0.252222</td>
<td>0.252222</td>
</tr>
</tbody>
</table>

C.2 RnaFoldChange Format

There are currently many different formats for describing gene expression values within and between samples. User’s can convert these different formats to the following simple text input format for compatibility with RnaFoldChange.

<table>
<thead>
<tr>
<th>ChrID</th>
<th>Name of the chromosome</th>
</tr>
</thead>
<tbody>
<tr>
<td>GeneID</td>
<td>Name of the gene</td>
</tr>
<tr>
<td>IsoformID</td>
<td>Name of the isoform</td>
</tr>
<tr>
<td>Sample1</td>
<td>Expression level for the first sample</td>
</tr>
<tr>
<td>Sample2</td>
<td>Expression level for the second sample</td>
</tr>
<tr>
<td>LogFold</td>
<td>Measured log fold-change</td>
</tr>
<tr>
<td>LogFoldSE</td>
<td>Standard deviation for the log fold-change</td>
</tr>
<tr>
<td>PValue</td>
<td>P-value</td>
</tr>
<tr>
<td>QValue</td>
<td>P-value adjusted for multiple testing</td>
</tr>
<tr>
<td>Average</td>
<td>Normalized counts of all samples</td>
</tr>
</tbody>
</table>

Example lines from the format:

<table>
<thead>
<tr>
<th>ChrID</th>
<th>GeneID</th>
<th>IsoformID</th>
<th>Sample1</th>
<th>Sample2</th>
<th>LogFold</th>
<th>LogFoldSE</th>
<th>PValue</th>
<th>QValue</th>
<th>Average</th>
</tr>
</thead>
</table>

© 2017 Garvan Institute. All rights reserved.
<table>
<thead>
<tr>
<th></th>
<th>R2_67</th>
<th></th>
<th>0.31</th>
<th>0.5</th>
<th>1.61</th>
<th>0.73</th>
<th>0.84</th>
<th>0.96</th>
<th>5.67</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1_14</td>
<td>R1_141</td>
<td>1.56</td>
<td>1.56</td>
<td>0</td>
<td>0.02</td>
<td>0.08</td>
<td>0.06</td>
<td>3.83</td>
<td></td>
</tr>
</tbody>
</table>
APPENDIX D - Statistical terms and definitions

D.1 Glossary

We briefly cover the metrics below; users are advised to consult a statistic book for further information.

**Correlation**
The Pearson’s correlation is a measure of the linear correlation between input and measured coverage. A perfect experiment would give a value of 1.0, but practically impossible due to human experimental errors and technical biases that may be introduced.

**Limit of Quantification (LOQ)**
This is the attomol/ul limit where the accuracy becomes stochastic, and is estimated by piecewise segmentation. Sequins with expression level below the point can’t be accurately quantified.

**Slope**
This is the linear proportionality of observed compared to expected abundance across the dynamic range of the standards.

**Coefficient of determination (R2)**
The proportion of the variance in the dependent variable that is predictable from the independent variable.

**P-value**
These statistics indicates the significance of the model. Under the null hypothesis, the model is not better than a random experiment.

**Q-value**
P-value probability adjusted for controlling the type I errors in statistical testing when conducting multiple hypothesis.

**Sensitivity**
Sensitivity is defined as: TP / (TP + FN)
TP is number of true positives
FN is number of false positives

**Precision**
Precision is defined by: TP / (TP + FP)
TP is number of true positives
FP is the number of false-positives

**Specificity**
Precision is defined by: TN / (TN + FP)
TN is the number of true-positives
FP is the number of false-positives

**AUC**
Area under the curve (AUC) is the probability that an experiment will rank a randomly chosen true positive (TP) higher than a randomly chosen false positive (FP).

D.2 Piecewise Segmentation

Piecewise segmentation separated by a breakpoint can be used to model limit-of-quantification (LOQ); the level of abundance below which quantification becomes questionable. The method finds all possible breakpoints and fit a linear regression on each of them. The breakpoint that gives the least total deviance is the LOQ.
APPENDIX E - Visualization with IGV

Users are encouraged to visualize alignments and synthetic features (e.g. transcripts, variants, etc). The Integrated Genome Viewer is an easy and popular software to visualize such features. The software is available at:

www.broadinstitute.org/igv

Here we describe how to load the in silico chromosome (for RNA-Seq), alignments and annotations for visual inspection.

E.1 Load in silico chromosome

1 | Download our resource bundle from if you haven’t already done so:

   $ wget s3.amazonaws.com/sequins/RnaQuinUserManualBundle.zip
   $ unzip RnaQuinUserManualBundle.zip

2 | Start a new IGV session and select the Genomes menu, select Load Genome From File. Load the chromosome file (chrIS.v2.fa) in the resource bundle.

E.2 Visualize annotations

1 | In an IGV session, select the File menu, select Load From File and then Open to load the annotation file (sequins_transcripts.gtf) from the resource bundle.

E.3 Visualize alignments

1 | Users can load their own generated alignment file in BAM format to IGV. Assume the file name is accepted_hits.bam from TopHat2.

2 | Before loading into IGV, the alignment file will need to be sorted using the following command:

   $ samtools sort accepted_hits.bam sorted

   This will generate sorted.bam in the working directory.

3 | The sorted alignment file must also be indexed using the following command:

   $ samtools index sorted.bam

4 | Finally, we can load the sorted alignment file into IGV. In the File menu, select Load From File and then select sorted.bam from the working directory.

E.4 Examine sequin regions

1 | We can examine alignments and annotations on select regions on the in silico chromosome. For example, enter chrIS:10,376,109-10,376,263 in the text box (next to the chromosome name) to go to this region of the in silico chromosome. The following screenshot illustrates the alignment reads within the region (aligned within the synthetic exon R2_59_1).
**Figure E4**: IGV screenshot showing alignment reads in the middle panel and annotation file in the bottom panel. The region is locus 10376109 to 10376263 on chromosome chrIS.