Anaquin : User's Manual

A practical guide to analyzing sequins in next-generation sequencing data.

Ted Wong & Tim Mercer, Garvan Institute of Medical Research

anaquin@garvan.org.au

Version 0.9 | August 2016

1 | INTRODUCTION **4**
1.1 | BACKGROUND 4
1.2 | CITATION 4
1.3 | IMPLEMENTATION 4
1.4 | LICENSE AND AVAILABILITY 4
1.5 | COMMUNITY 4
1.6 | CONTACT 5

2 | ANAQUIN **6**
2.1 | INTRODUCTION 6
2.2 | GENERAL USAGE 6
2.3 | SUMMARY OF AVAILABLE TOOLS 6
2.3.1 | RnaQuin Tools *6*
2.3.2 | RNAQuin R-Functions *6*
4 | SUPPORTED INPUT FORMATS 6
3.4 | ANAQUIN CSV OUTPUT 7

3 | INSTALL **8**
3.1 | DOWNLOAD 8
3.2 | COMPILING FROM SOURCE 8
3.3 | INSTALL FOR R 8
3.4 | INDEX *IN SILICO* CHROMOSOME/GENOME WITH YOUR REFERENCE GENOME 8
3.5.1 | Example of co-index with human genome *9*
3.6 | THIRD PARTY TOOLS AND INSTALLATION 9

4 | RESOURCE FILES **10**
4.1 | CHROMOSOME SEQUENCES 10
4.2 | MIXTURE FILES 10
4.3 | ANNOTATION FILES 10
4.4 | LIBRARIES 11

5 | TRANSCRIPTOMICS **12**
5.1 | INTRODUCTION 12
5.2 | DESIGN 12
5.3 | PREREQUISITES 12
5.3.1 | Data and Resource Files *12*
5.3.2 | Third-Party Tools *13*
5.4 | WORKFLOW - GENE EXPRESSION (SINGLE SAMPLE) 14
5.4.1 | Splice-aware alignment of libraries (TopHat2) *14*
5.4.2 | Assess Alignment (Anaquin) *15*
5.4.2.1 | Example Output (`RnaAlign_summary.stats`) 15
5.4.3 | Subsampling (Anaquin) *16*
5.4.4 | Assemble Transcript Models (Cufflinks) *16*
5.4.5 | Assess Assembly (Anaquin) *17*
5.4.4.1 | Plot Assembly Sensitivity curve (Anaquin, R) 17
5.4.6 | Assess Gene Expression (Anaquin) *18*
5.4.6.1 | Quantify Gene Expression (Anaquin) 18
5.4.6.2 | Plot Gene Expression Expression Curve (Anaquin, R) 19
5.4.6.3 | Assess Multiple Replicates (Anaquin, R) 20
5.4.6.4 | Example output from Multiple Replicates (Anaquin, R) 21
5.4.7 | Assess Isoform Expression (Anaquin) *22*
5.4.7.1 | Example output from Multiple Replicates (Anaquin, R) ..... **Error! Bookmark not defined.**
5.5 | WORKFLOW - DIFFERENTIAL GENE EXPRESSION (MULTIPLE SAMPLES) 23

5.5.1 | Differential analysis (Cuffdiff) *24*
5.5.2 | Assess fold-changes in gene expression (Anaquin) *24*
5.5.2.1 | Example output from RnaFoldChange 24
5.5.2.2 | Plot observed fold-change (Anaquin, R) 25
5.5.4 | Assess fold-changes in isoform expression (Anaquin) *26*
5.5.4.1 | Plot observed fold-change (Anaquin, R) 27
5.6 | WORKFLOW - BIOCONDUCTOR (DESEQ2) 28
5.6.1 | Make A Count Table *28*
5.6.2 | Identify Differential Gene Expression (DESeq2 in R) *29*
5.6.3 | Quantify Differential Gene Expression (Anaquin, in R) *29*
5.6.3.1 | Example summary statistics from RnaFoldChange 29
5.6.3.2 | Plot observed fold-change (Anaquin, R) 30
5.6.3.3 | Plot ROC curve (Anaquin, R) 32
5.6.3.4 | Plot LODR curve (Anaquin, R) 33
5.7 | WORKFLOW - ALIGNMENT-FREE (KALLISTO) 34
5.7.1 | Quantify Transcript Abundance (Kallisto) 34
5.7.2 | Quantify Kallisto Quantification (Anaquin) 35

APPENDIX A – COMMAND LINE USAGE 37
A.1 | RNAALIGN 37
A.2 | RNAASSEMBLY 40
A.3 | RNAEXPRESSION 43
A.4 | RNAFOLDCHANGE 46
A.5 | RNASUBSAMPLE 49

APPENDIX B – R USAGE 52
B.1 | OPEN & LOAD R SCRIPTS 52
B.1.1 | Use RStudio 52
B.1.2 | Use command-line 52
B.1.3 | Install third-party R-packages 52
B.2 | CREATE ANAQUIN DATASET 53
B.3 | MODIFYING R SCRIPTS ..... **ERROR! BOOKMARK NOT DEFINED.**
B.4 | PLOTROC 60
B.4 | PLOTLODR 62
B.5 | PLOTLINEAR 64
B.6 | PLOTLOGISTIC 66
B.7 | CREATEANAQUINDATA ..... **ERROR! BOOKMARK NOT DEFINED.**

APPENDIX C – SIMPLE TEXT INPUT FORMATS 68
C.1 RNAEXPRESSION SIMPLE FORMAT 68
C.2 RNAFOLDCHANGE FORMAT 68

APPENDIX D – STATISTICAL TERMS AND DEFINITIONS 70
D.1 GLOSSARY 70
D.2 PIECEWISE SEGMENTATION 70

APPENDIX E – VISUALISATION WITH IGV ..... ERROR! BOOKMARK NOT DEFINED.
E.1 LOAD IN SILICO CHROMOSOME ..... **ERROR! BOOKMARK NOT DEFINED.**
E.2 VISUALIZE ANNOTATIONS ..... **ERROR! BOOKMARK NOT DEFINED.**
E.3 VISUALIZE ALIGNMENTS ..... **ERROR! BOOKMARK NOT DEFINED.**

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. Deveson, I et al., (2016). Representing genetic variation with synthetic DNA standards. *Nature Methods*.
2. Hardwick, S et al., (2016). Spliced synthetic genes as internal controls in RNA sequencing experiments. *Nature Methods*.

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 Creative Commons License.
See - https://en.wikipedia.org/wiki/Creative_Commons_license for details.

The GitHub source code is available - www.github.com/student-t/Sequins.

1.5 | Community

We have an active Slack channel, sequins.slack.com , where you can chat directly with our team. Please email us at anaquin@garvan.org.au for an invitation.

You can also post your questions on support.bioconductor.org, www.biostars.org and 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 you answers. If you still have questions, or new ideas, feature requests or bug fixes, please email us at anaquin@garvan.org.au

2 | Anaquin

2.1 | Introduction

Anaquin is a software toolkit designed for the analysis of synthetic sequin controls in NGS experiments. Anaquin 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:

```
anaquin <tool> <options>
```

Where:
`<tool>` Name of the tool to be used
`<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:

```
anaquin VarAlign -rbed reference.bed -o output -ufiles aligned.bam
```

Where:
`VarAlign` - is the name of the tool being called
`reference.bed` - is a reference annotation for sequins in BED format (specified with `-rbed`),
`output` - is the output directory results are written to (specified with `-o`).
`aligned.bam` - is the user generated BAM file from a third-party software (specified with `-ufiles`),

Many command options are tool-specific. Using the `-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 **Appendix A**.

2.3 | Summary of Available Tools

2.3.1 | RnaQuin Tools

RnaAlign	Measure the spliced read alignments from sequins to the *in silico* chromosome
RnaAssembly	Compare assembled transcript models to sequin annotations in the *in silico* chromosome
RnaExpression	Quantitative analysis of sequin expression
RnaFoldChange	Assess fold-changes in gene expression between multiple samples
RnaSubsample	Calibrate sequence coverage of sequins across multiple replicates

2.3.2 | RNAQuin R-Functions

plotAssembly	Generate an assembly plot between input concentration and assembly sensitivity
PlotLinear	Generate a scatter plot between expected abundance (eg: input concentration) against measured abundance (eg: FPKM)
plotROC	Generate a receiver operating characteristic (ROC) plot between true positive rate (TPR) against false positive rate (FPR) at various threshold settings (such as fold-change)
plotLODR	Generate a Limit-of-Detection Ratio (LODR) plot between measured abundance and p-value probability

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, edgeR, DESeq2, TopHat2, STAR, GATK, VarScan amongst others) and, where possible, standard data formats (such as SAM, BAM, BED, VCF, GTF etc.)

However, supported software 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 D**.

If you have a script for processing data, using alternative unsupported software, or perform a novel analysis, please post it at - groups.google.com/d/forum/garvansequins.

2.5 | Anaquin CSV output

Anaquin generates multiple tab-delimited CSV 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.csv` 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/getsoftware
```

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 anaquin@garvan.org.au. 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. Catch (https://github.com/philsquared/Catch)
3. KLib (https://github.com/attractivechaos/klib)
4. Boost (http://www.boost.org/users/download)

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 currently supported. To test your installation, run version check:

```
$ anaquin -v
v1.0
```

While your version might differ, the `-v` option should always report the version number.

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, start R and enter:

```
> source('https://bioconductor.org/biocLite.R')
> biocLite('Anaquin')
```

To view documentation for the version of this package installed in your system, start R and enter:

```
> browseVignettes('Anaquin')
```

You can also download our R vignette on the Bioconductor website here:

```
www.bioconductor.org
```

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, 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/chromosomes/
```

It is important to ensure that your chromosome/genome version are 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 perform the following command:

```
$ wget s3.amazonaws.com/sequins/chromosomes/CRN007_v001.fa
```

This 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 CRN007_v001.fa > hg38_CRN007.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_CRN007.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_CRN007.fa hg38_CRN007
```

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. Below we have listed some of the most popular tools that we have tested with Anaquin. This list is not definitive and many of the tools can be replaced by an alternative tool not listed here. 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.

SAMtools	Sorting, converting and indexing alignment files	http://www.htslib.org/
Bowtie2	Short read aligner	http://bowtie-bio.sourceforge.net/bowtie2
TopHat2	Spliced read alignment for RNA-Seq	https://ccb.jhu.edu/software/tophat
Cufflinks2	Isoform assembly and quantification	http://cole-trapnell-lab.github.io/cufflinks
R	Statistical programming environment	https://cran.r-project.org/
Kallisto	Alignment-free k-mer quantification	https://pachterlab.github.io/kallisto
IGV	Genome browser	https://www.broadinstitute.org/igv
Trim Galore	Remove adaptor contamination from libraries.	http://www.bioinformatics.babraham.ac.uk/projects/trim_galore/
DESeq2	Differential gene expression analyzer in R	https://bioconductor.org/packages/release/bioc/html/DESeq2.html

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.

In this section, we will describe the number of resource files. These files can be downloaded from:

```
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.

Please contact us (contact@sequin.xyz) if you have any problems with compatibility.

4.1 | Chromosome sequences

The *in silico* chromosome or genome sequences 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.

The sequence files (FASTA) for *in silico* chromosomes and genomes are available to download from:

```
www.sequin.xyz/downloads/chromosomes
```

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 (CSV) 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. Mixture file can be downloaded from:

```
www.sequin.xyz/downloads/mixtures
```

Mixture files are typically accompanied with a README text file that describes the features of the mixture to users.

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 A file (`MRN027_v001.csv`) for RnaQuin:

```
$ wget s3.amazonaws.com/sequins/mixtures/MRN027_v001.csv
$ head MRN027_v001.csv
ID         Length    MXA (attomol/ul)
R1_11_1     703       161.1328125
R1_11_2     785       80.56640625
R1_13_1     1940      5156.25
R1_13_2     698       2578.125
```

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.

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.v24.annotation.gtf` is the file name):

```
$ wget s3.amazonaws.com/sequins/annotations/ARN020_v001.gtf
$ cat gencode.v24.annotation.gtf ARN020_v001.gtf > gencode_v24.ARN020_v001.gtf
```

4.4 | Libraries

A range of simulated and experimental read libraries are 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/libraries
```

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

The following data and files are used in this workflow:

1. **LRN087** to **LRN092** – Example libraries for use in the workflow. These libraries use mixtures of RNA sequins (Mixture A or B) added to total RNA extracted from either K562 or GM12878 human cell lines. The metadata files attached to each library provides technical details on sample preparation, library construction and sequencing technical parameters.

```
$ wget s3.amazonaws.com/sequins/libraries/LRN087.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN087.2.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN088.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN088.2.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN089.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN089.2.fq.gz
```

```
$ wget s3.amazonaws.com/sequins/libraries/LRN090.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN090.2.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN091.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN091.2.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN092.1.fq.gz
$ wget s3.amazonaws.com/sequins/libraries/LRN092.2.fq.gz
```

2. **CRN007.fa** - *In silico* chromosome that should be co-indexed with hg38 for alignment of libraries

```
$ wget s3.amazonaws.com/sequins/chromosomes/CRN007_v001.fa
```

3. **ARN020.gtf** – Gene annotations on *in silico* chromosome

```
$ wget s3.amazonaws.com/sequins/annotations/ARN020_v001.gtf
```

4. **MRN027.csv** – Mixture file for RNA sequins, Mixture A

```
$ wget s3.amazonaws.com/sequins/mixtures/MRN027_v001.csv
```

5. **MRN028.csv** – Mixture file for RNA sequins, Mixture B

```
$ wget s3.amazonaws.com/sequins/mixtures/MRN028_v001.csv
```

6. **MRN029.csv** – Mixture file for RNA sequins, Mixture A&B

```
$ wget s3.amazonaws.com/sequins/mixtures/MRN029_v001.csv
```

7. **ARN024.index** – K-mer index for the RNA sequins

```
$ wget s3.amazonaws.com/sequins/annotations/ARN024_v001.index
```

8. **ARN025.index** – Nucleotide sequence for the RNA sequins

```
$ wget s3.amazonaws.com/sequins/annotations/ARN025_v001.fa
```

9. **hg38.fa -** The most recent human genome build (GRCh38) can be downloaded by:

```
$ 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**.

10. **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
```

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.v23.annotation.gtf` is the GENCODE annotation file name):

```
$ cat gencode.v24.anntoation.gtf ARN020_v001.gtf > gencode_v24.ARN020_v001.gtf
```

5.3.2 | Third-Party Tools

RNA sequencing analysis is complex, and there are 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 by 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:

Bowtie2	http://bowtie-bio.sourceforge.net/bowtie2
TopHat2	https://ccb.jhu.edu/software/tophat
SAMtools	http://www.htslib.org
Cufflinks	http://cole-trapnell-lab.github.io/cufflinks
R	https://cran.r-project.org
IGV	https://www.broadinstitute.org/igv
DESeq2	https://bioconductor.org/packages/release/bioc/html/DESeq2.html
Kallisto	https://pachterlab.github.io/kallisto
Trim Galore	http://www.bioinformatics.babraham.ac.uk/projects/trim_galore

5.4 | Workflow - Gene Expression (Single Sample)

Sequins can be used to assess the performance of RNA-Seq, and measure the limitations in gene assembly and gene expression. Sequins are added directly to users' RNA samples prior to library preparation and sequencing. Here we describe the analysis of single RNA sample (from the K562 human cell-type) that has spiked-in RNA mixture A. In this workflow, we will perform the following steps:

1. Splice-aware alignment of reads using *Tophat2*
2. Assess spliced-alignment performance using *RnaAlign*
3. Subsampling the reads using *RnaSubsample*
4. Assemble alignments into transcript models using *Cufflinks*
5. Assess assembled transcript models using *RnaAssembly*
6. Estimate gene/isoform expression using *Cufflinks*
7. Quantify gene/isoform expression measurements using *RnaExpression*

5.4.1 | Splice-aware alignment of libraries (TopHat2)

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

```
$ trim_galore --paired LRN087.1.fq LRN087.2.fq
```

The usage will generate `LRN087.1_val_1.fq` and `LRN087.2_val_2.fq` in the working directory.

2 | 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_CRN007.fa hg38_CRN007
```

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

3 | 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_CRN007 LRN087.1_val_1.fq LRN087.2_val_2.fq
```

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_v24.ARN020_v001.gtf -o A1 hg38_CRN007 LRN087.1_val_1.fq \
      LRN087.2_val_2.fq
```

COMMENT | User's can simultaneously assess the *de novo* and guided assembly 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.

5.4.2 | Assess Alignment (Anaquin)

1 | 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 5.4.2 -rgtf gencode_v24.ARN020_v001.gtf \
      -ufiles A1/accepted_hits.bam
```

Where:
`RnaAlign` is name of the tool
`5.4.2` is the output directory specified by `-o`
`gencode_v24.ARN020_v001.gtf` is the combined annotation file specified by `-rgtf`
`accepted_hits.bam` is the user input alignment file given by `-ufiles`

`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 5.4.2.1** for an example of the output file, including a description and interpretation of statistics.
2. `RnaAlign_sequins.csv` – provides number of reads, exon sensitivity, intron sensitivity and precision for each individual sequins.

5.4.2.1 | 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_v24.ARN020_v001.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)

       *Intron level
        Sensitivity: 0.73
```

```
        Precision:   0.88

       *Base level
        Sensitivity: 0.73
        Precision:   0.95

-------Comparison of alignments to reference annotation (Genome)

       *Intron level
        Sensitivity: 0.42
        Precision:   0.66

       *Base level
        Sensitivity: 0.51
        Precision:   0.14
```

5.4.3 | Subsampling (Anaquin)

1 | 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 5.4.3 -method 0.02 -ufiles A1/accepted_hits.bam |
       samtools view -bS - > A1/subsampled.bam
```

CRITICAL | `RnaSubsample` provides 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.

Where:
`RnaSubsample` is name of the tool
`5.4.3` is the output directory specified by `-o`
`0.02` is the fraction (2%) to subsample specified by `-method`
`A1/accepted_hits.bam` is the user generated alignment file in **Section 5.4.1**.

`RnaSubsample` will generate the following file in the output directory:
1 | `RnaSubample_summary.stats` - summary statistics reporting the 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.

2 | In the above example, we have subsampled alignments to the *in silico* chromosome to 2% 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 2% RNA sequin spike-in amount. This can be easily performed by repeating the above command for each replicate library to be calibrated.

5.4.4 | Assemble Transcript Models (Cufflinks)

1 | 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 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.

2 | 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:

```
$ cufflinks -G gencode_v24.ARN020_v001.gtf -o A1/G A1/subsampled.bam
```

5.4.5 | Assess Assembly (Anaquin)

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

```
$ anaquin RnaAssembly -m MRN027_v001.csv -o 5.4.5 \
      -rgtf gencode_v24.ARN020_v001.gtf -ufiles A1/D/transcripts.gtf
```

Where:
`RnaAssembly` is name of the tool
`5.4.5` is the output directory specified by `-o`
`gencode_v24.ARN020_v001.gtf` is the combined annotation (`-rgtf`)
`MRN027_v001.csv` is the mixture A (`-m`)
`transcripts.gtf` is the user generated transcriptome file given by `-ufiles`

`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.csv` - 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.

COMMENT: We can similarly quantify the guided assembled transcriptome.

5.4.5.1 | Plot Assembly Sensitivity curve (Anaquin, R)

1 | 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:

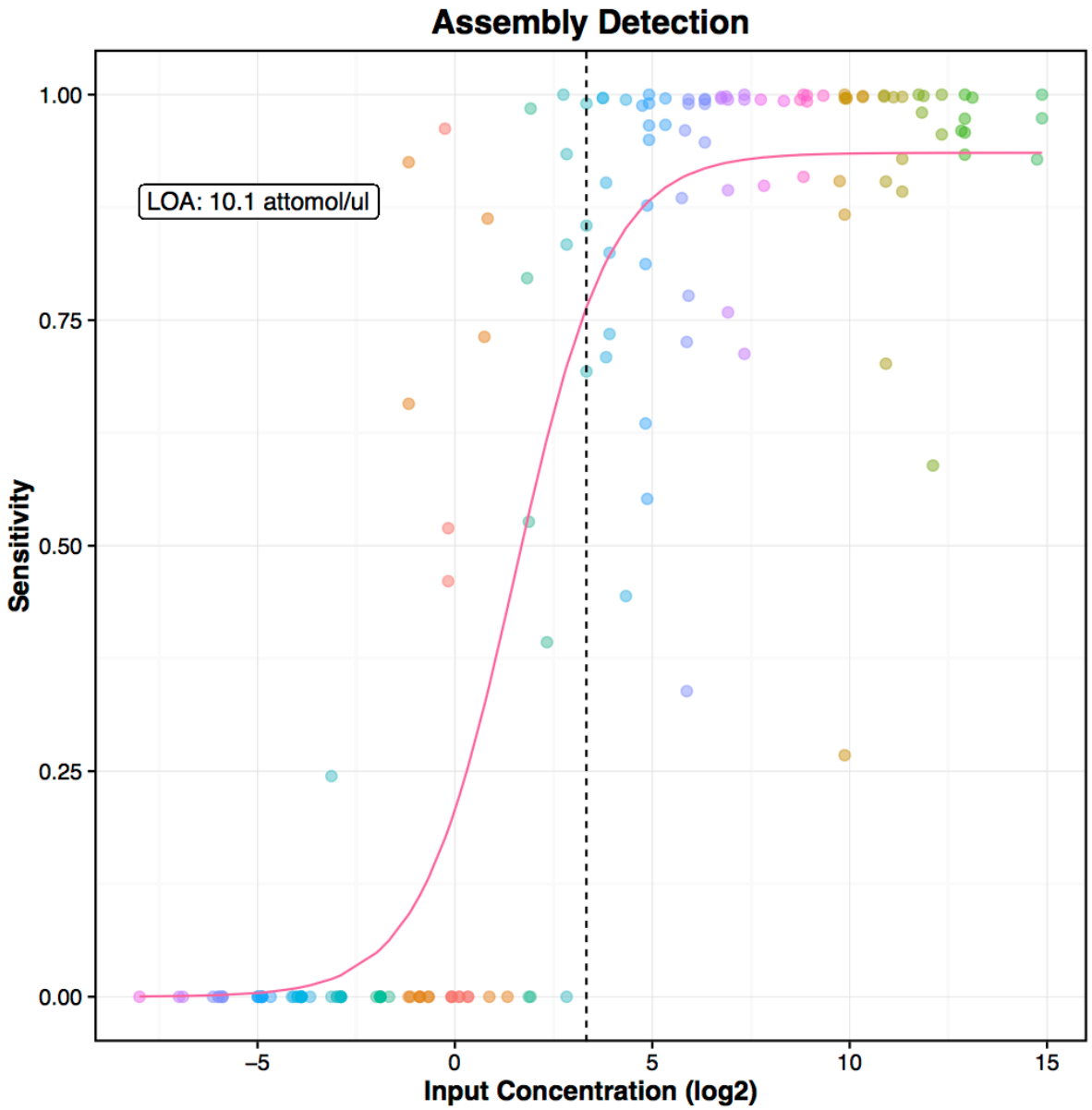

Figure 5.4.4.1 Scatter-plot illustrates the assembly sensitivity of each sequin, relative to input concentration. The assembly of isoforms exhibits an expression-dependent sigmoidal relationship with input concentration. Fitting is performed by non-linear least square fitting on sigmoid function. The minimal concentration required to assemble isoforms according to a user-specific assembly threshold is indicated (for example, dashed line corresponds to 0.70 sensitivity).

2 | The graph is generated by the `PlotLogistic` Anaquin 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
```

5.4.6 | Assess Gene Expression (Anaquin)

5.4.6.1 | Quantify Gene 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).

1 | Recall our guided assembly in **Section 5.4.4**. To quantify gene expression for a single replicate, perform the following command:

```
$ anaquin RnaExpression -o 5.4.6.1 -m MRN027_v001.csv -method gene \
      -ufiles A1/G/transcripts.gtf
```

Where:
`RnaExpression` is the name of the tool
`5.4.6.1` is the output directory specified by `-o`
`MRN027_v001.csv` is the reference mixture file A (`-m`)
`gene` is the metric (gene expression) to quantify (`-method`)
`A1/G/ transcripts.gtf` is the user generated and guided transcriptome file by Cufflinks (`-ufiles`)

`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_sequins.csv` – statistics for estimated expression at each individual sequin gene.
3. `RnaExpression_linear.R` – R script to plot the expression estimated for each sequin gene relative to expected input concentration.

COMMENT | `RnaExpression` requires the gene/transcript ids in the user-generate transcriptome file (eg. `transcripts.gtf`) to match the gene/transcript ids in the mixture file (eg. `MRN027_v001.csv`)
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 ids are 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**.

5.4.6.2 | Plot Gene Expression Expression Curve (Anaquin, R)

1 | To plot the measured expression of each sequin gene relative to expected input concentration in a single replicate, load the `RnaExpression_linear.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:

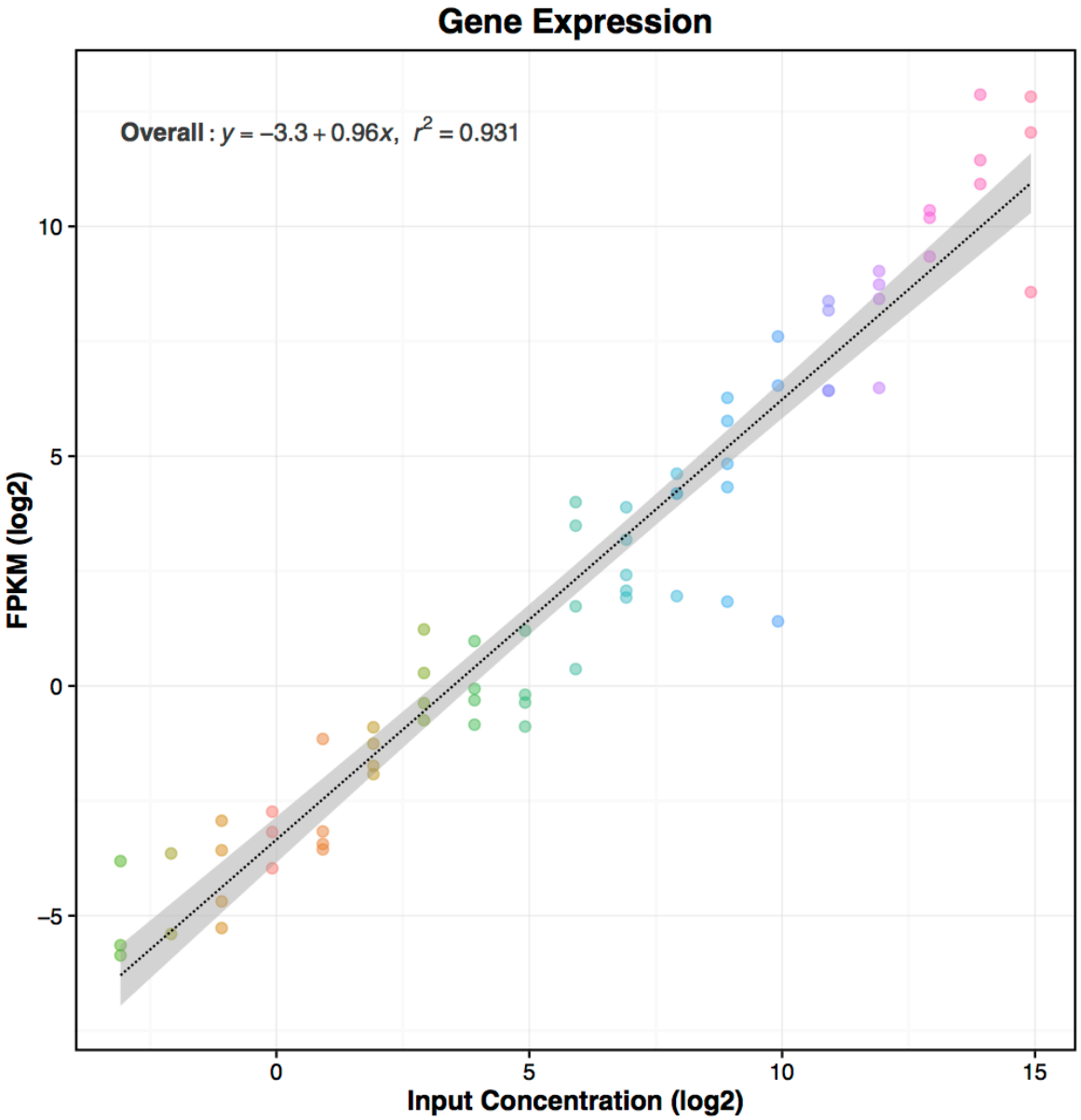

Figure 5.4.6.2 Scatter-plot illustrates the observed abundance (in FPKM) relative to the expected abundance (in attomoles/ul). Dashed blue line (with shadow depicts the 95% confidence interval) shows linear regression model.

5.4.6.3 | Assess Multiple Replicates (Anaquin, R)

1 | 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 `-ufiles` (which must be repeated for each replicate file input).

Each individual expression replicate is given by `-ufiles`. Perform the following command:

```
$ anaquin RnaExpression -o 5.4.6.3 -method gene -m MRN027_v001.csv \
      -ufiles A1/G/transcripts.gtf \
      -ufiles A2/G/transcripts.gtf \
      -ufiles A3/G/transcripts.gtf
```

This will generate the same output results (`RnaExpression_summary.stats`, `RnaExpression_sequins.csv`) and plot (`RnaExpression_linear.R`) as for a single replicate but

with 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.

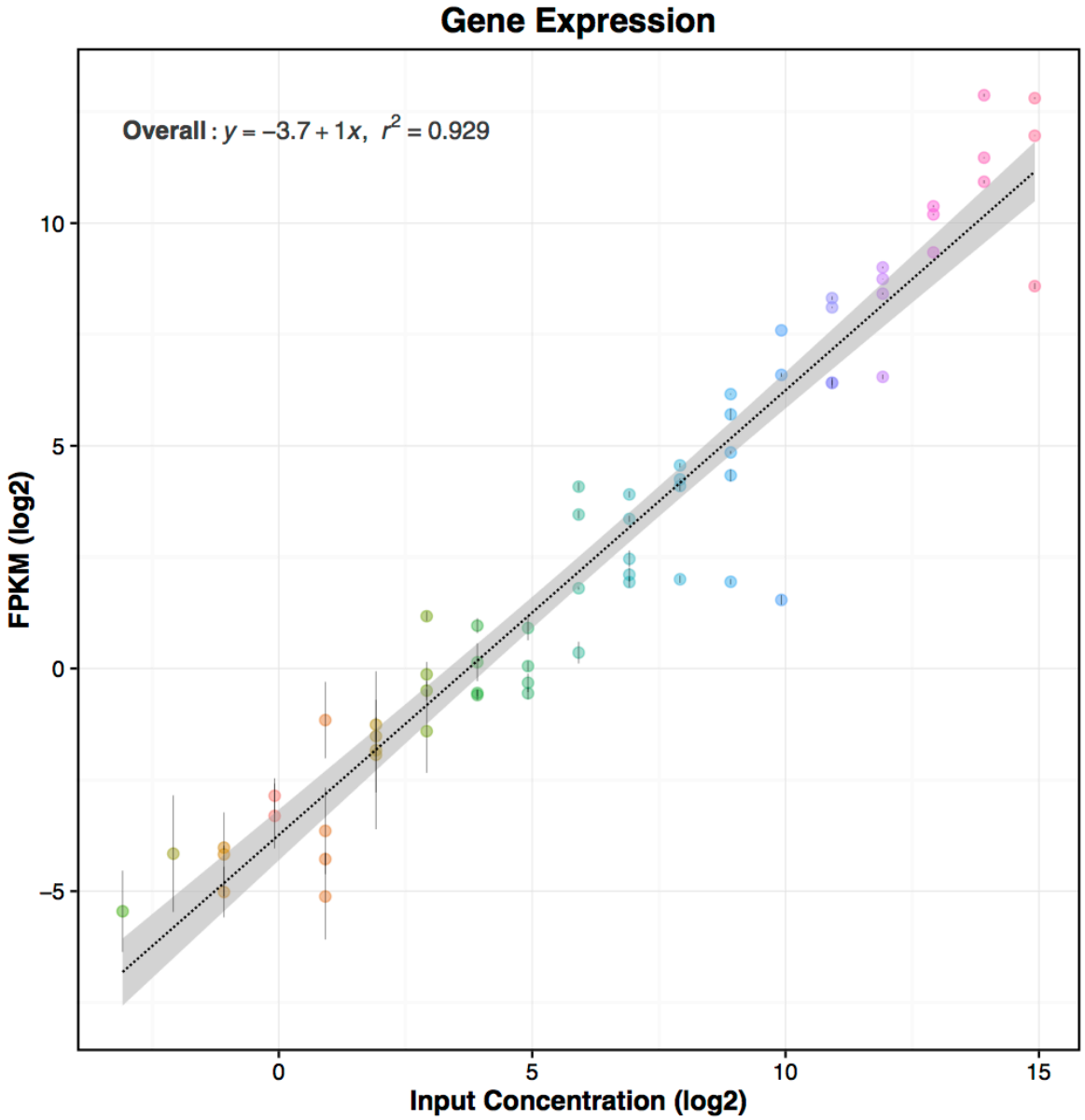

Figure 5.4.6.3 Scatter-plot illustrates the mean observed abundance (in FPKM) for each synthetic gene relative to the expected concentration (in attomoles/ul). Error bars indicate standard deviation, with n = 3.

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

```
-------RnaExpression Output
       Summary for input: A1/G/transcripts.gtf, A2/G/transcripts.gtf,
A3/G/transcripts.gtf
       *Arithmetic average and standard deviation are shown

-------Reference Transcript Annotations

       Synthetic: 78
       Mixture file: MRN027_v001.csv

-------Genes Expressed
```

```
       Synthetic: 69 ± 1
       Detection Sensitivity: 0.118017 (attomol/ul) (R1_33)

       Genome: 199036 ± 0

-------Limit of Quantification (LOQ)

       *Estimated by piecewise segmented regression

       Break LOQ:   - attomol/ul (-)

       *Below LOQ
       Intercept:   -
       Slope:       -
       Correlation: -
       R2:          -
       Genome:      -

       *Above LOQ
       Intercept:   -
       Slope:       -
       Correlation: -
       R2:          -
       Genome:      -

-------Linear regression (log2 scale)

       Slope:       0.97 ± 0.02
       Correlation: 0.96 ± 0.00
       R2:          0.93 ± 0.01
       F-statistic: 896.83 ± 110.59
       P-value:     0.00 ± 0.00
       SSM:         1694.26 ± 30.29, DF: 1 ± 0
       SSE:         129.07 ± 15.30, DF: 67 ± 1
       SST:         1823.34 ± 19.01, DF: 68 ± 1
```

5.4.7 | Assess Isoform Expression (Anaquin)

We can also assess the quantification of individual isoforms, by comparing measured expression relative to expected concentration.

1 | We can reuse the generated `transcripts.gtf` annotation files discussed in **Section 5.4.6**. The following command will quantify the replicates at the isoform level:

```
$ anaquin RnaExpression -o 5.4.7 -method isoform -m MRN027_v001.csv \
       -ufiles A1/G/transcripts.gtf \
       -ufiles A2/G/transcripts.gtf \
       -ufiles A3/G/transcripts.gtf
```

Where:
`RnaExpression`: name of the tool
`5.4.7` is the output directory specified by `-o`
`isoform` specifies the analysis be done at the isoform level (`-method`)
`MRN027_v001.csv` is the reference mixture file A (`-m`)
`transcripts.gtf` are the generated transcriptome files (three replicates) from Cufflinks (`-ufiles`)

The tool generates similar output files already discussed in **Section 5.4.6**.

2 | We can follow the workflow outlined in **Section 5.4.6.2** to plot a scatter plot for sequins between the input concentration and measured FPKM in R.

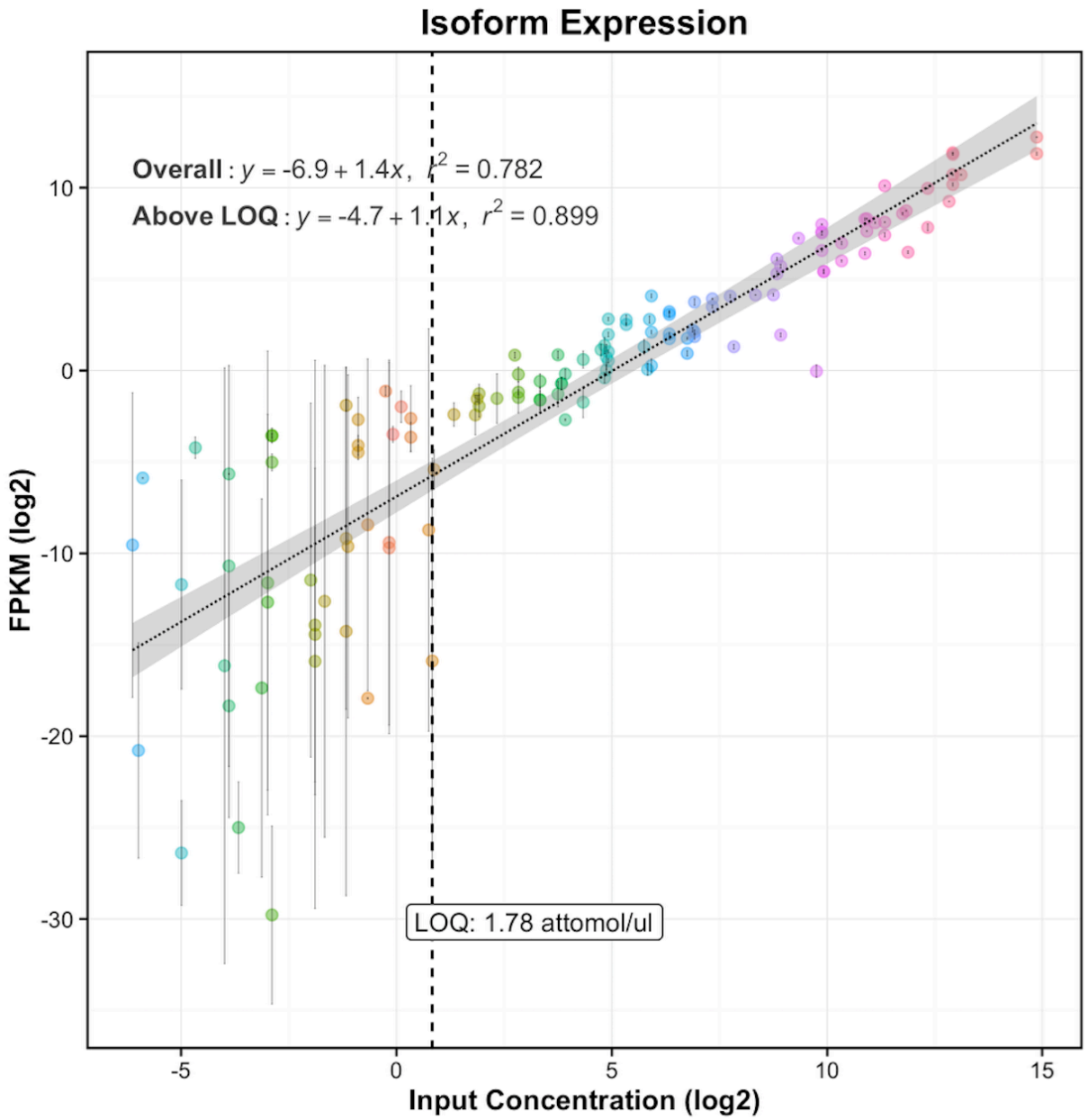

Figure 5.4.7 Scatter-plot illustrates the mean observed abundance (in FPKM) for each synthetic sequin relative to the expected concentration (in attomoles/ul). Error bars indicate standard deviation, with n = 3. The limit-of-quantification (LOQ; indicated by dashed line) is estimated by piecewise-linear segmentation (further details available in **Appendix B**).

5.5 | 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.

5.5.1 | Differential analysis (Cuffdiff)

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

```
$ cuffdiff gencode_v24.ARN020_v001.gtf \
A1/subsampled.bam,A2/subsampled.bam,A3/subsampled.bam \ B1/subsampled.bam,B2/
subsampled.bam,B3/accepted_hits.bam
```

2 | 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.

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

1 | Using gene 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:

```
$ anaquin RnaFoldChange -o 5.5.2 -m MRN029_v001.csv -method gene \
       -ufiles gene_exp.diff
```

Where:
`RnaFoldChange` is name of the tool
`5.5.3` is the specified output directory (`-o`)
`MRN029_v001.csv` is the reference file for Mixture A&B (`-m`)
`gene_exp.diff` is the generated differential gene expression file by Cuffdiff (`-ufiles`)

CRITICAL | Ensure the input mixture annotation file (`-m`) contains sequins concentrations in both mixtures A and B. For example, run the following command:

```
$ head -n2 MRN029_v001.csv

ID         Length    MXA (attomol/ul)     MXB (attomol/ul)
R1_11_1     703       161.1328125          5.035400391
```

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.csv` – 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.

5.5.2.1 | Example output from RnaFoldChange

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

       Summary for input: gene_exp.diff

-------Reference Annotations

       Synthetic: 78 genes
       Mixture file: MRN029_v001.csv

-------Genes Expressed
```

```
       Synthetic: 60 genes
       Genome:    19064 genes

-------Linear regression (log2 scale)

       Slope:       0.939926
       Correlation: 0.887833
       R2:          0.788248
       F-statistic: 212.183
       P-value:     0
       SSM:         321.28, DF: 1
       SSE:         86.3076, DF: 57
       SST:         407.588, DF: 58
```

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

1 | 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:

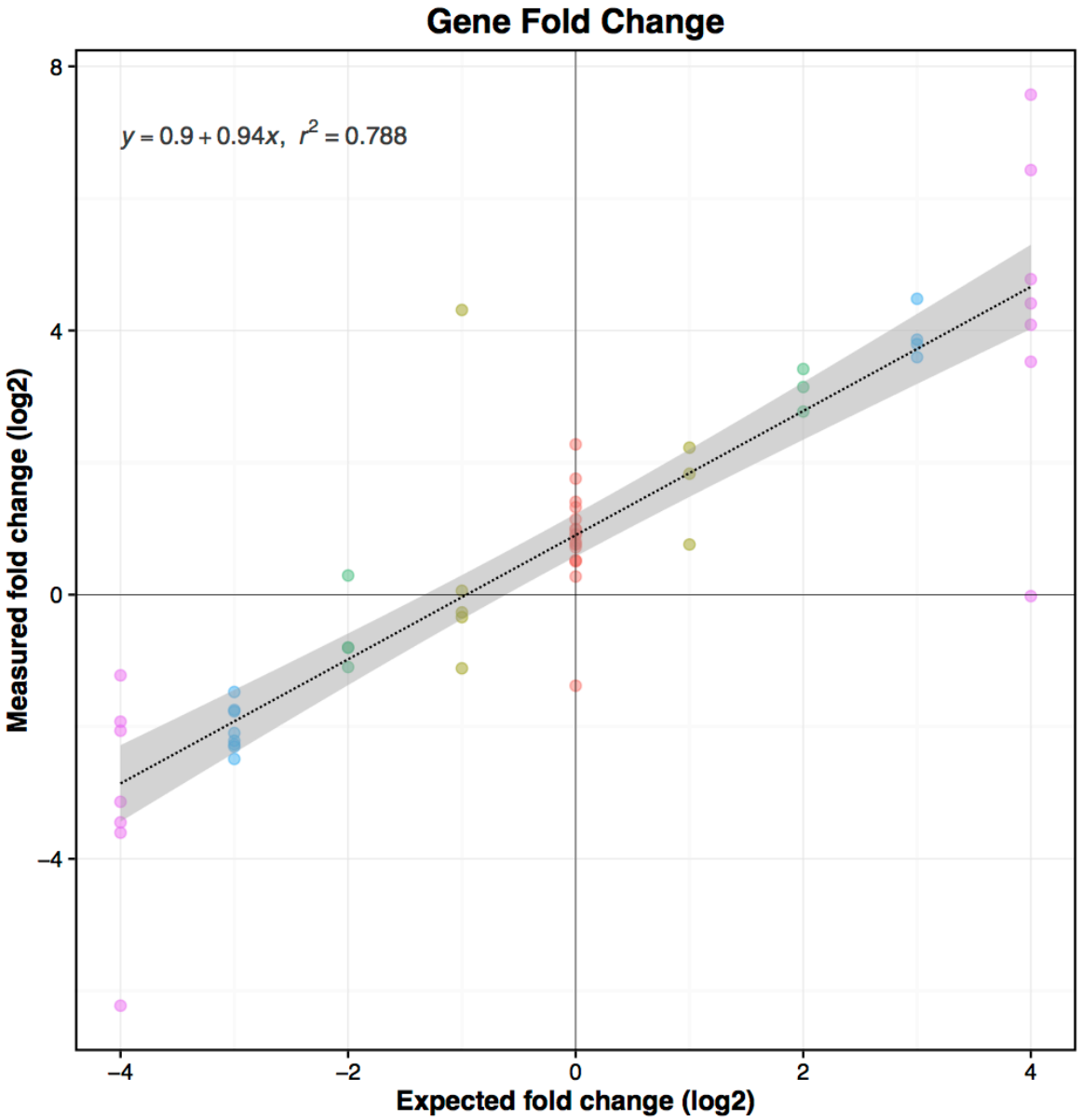

Figure 5.5.2.2: Scatter plot indicates correlation between observed and expected differential log-fold for each sequin gene. Sequin genes correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).

5.5.3 | 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 the isoform analysis.

1 | Perform the following command quantify the isoform differential expression file. Note that we will need to specify isoform in the `-method` option.

```
$ anaquin RnaFoldChange -o 5.5.3 -m MRN029_v001.csv -method isoform \
      -ufiles isoform_exp.diff
```

Where:
`RnaFoldChange` is name of the tool

`5.5.3` is the specified output directory (`-o`)
`MRN029_v001.csv` is the reference file for Mixture A&B (`-m`)
`isoform_exp.diff` is the generated differential isoform expression file by Cuffdiff (`-ufiles`)

The usage will generate identical files already discussed in **Section 5.5.3.**

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

1 | To plot the observed fold-change, we can follow the same workflow outlined in **Section 5.5.3.1**. 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:

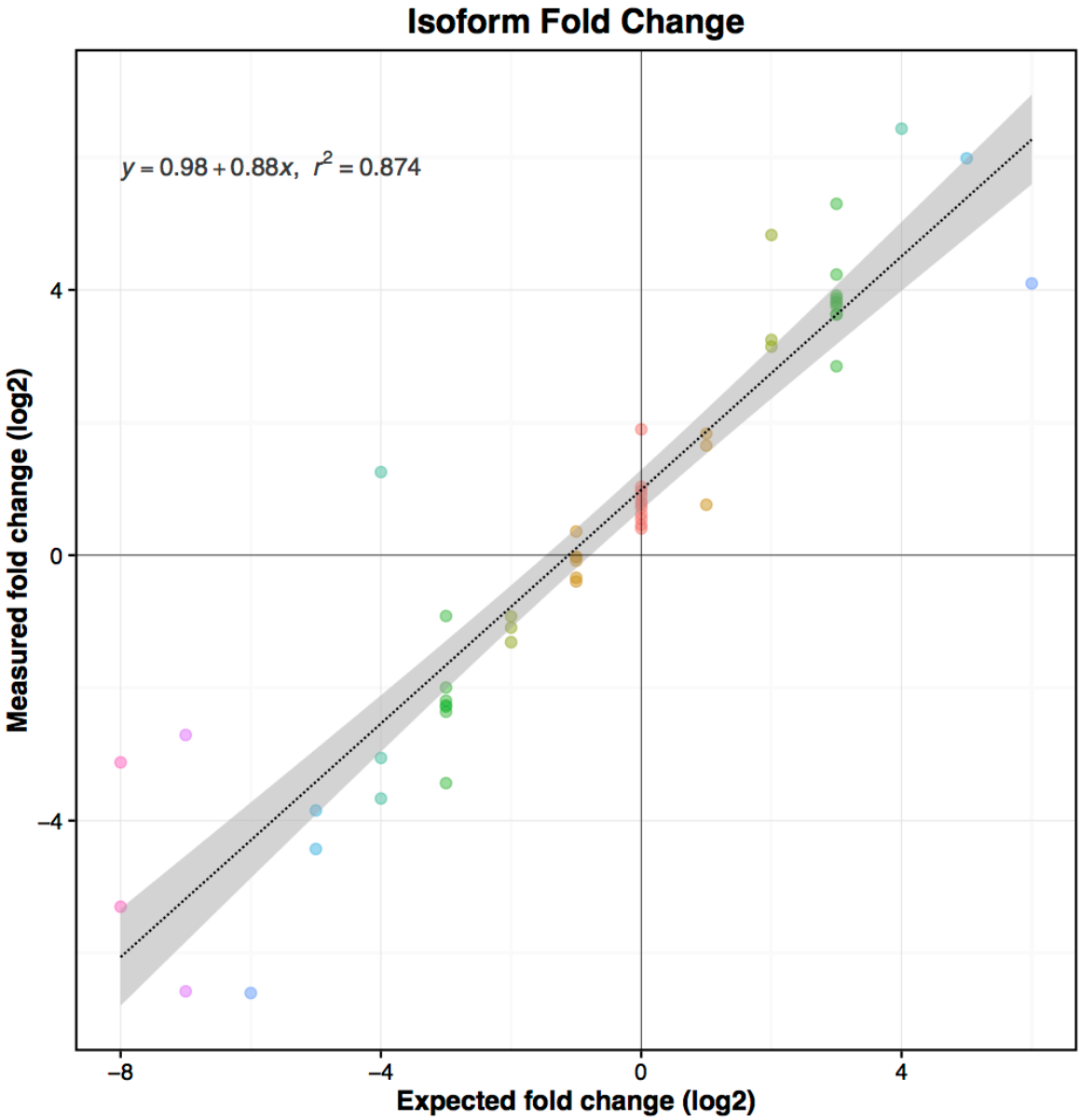

Figure 5.5.3.1: Scatter plot indicates correlation between observed and expected differential log-fold for each sequin isoform. Sequins correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).

5.6 | 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.

5.6.1 | 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_v24.ARN020_v001.gtf` (includes gencode annotation and sequin annotation) file, discussed in **Section 5.3.1**.

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 (please revisit **Section 5.4.1** for detailed description of alignment steps). 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:

```
> 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)
```

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

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

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

```
> bams <- BamFileList(files)
> names(bams) <- c("A1.bam", "A2.bam", "A3.bam", "B1.bam", "B2.bam", "B3.bam")
```

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

```
> model <- makeTxDbFromGFF('gencode_v24.ARN020_v001.gtf', format='gtf')
> genes <- exonsBy(model, by='gene')
```

CRITICAL | The file `gencode_v24.ARN020_v001.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.

5 | We next perform counting with the `summarizeOverlaps` function:

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

6 | 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,

```
Replicate,Sample
A1,K_RMXA
A2,K_RMXA
A3,K_RMXA
```

```
B1,G_RMXB
B2,G_RMXB
B3,G_RMXB
```

Save this file in CSV format as `meta.csv`. The file is also available at:

```
> download.file(url='https://s3.amazonaws.com/sequins/manuscripts/meta.csv', \
      destfile='meta.csv')
```

7 | We then incorporate the above `meta.csv` file with the following command:

```
> meta <- read.csv('meta.csv', row.names=1)
> colData(se) <- DataFrame(meta)
```

8 | We can access the count matrix with:

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

9 | And write the matrix to a CSV file by:

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

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

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

5.6.2 | Identify Differential Gene Expression (DESeq2 in R)

1 | 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:

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

CRITICAL | The CSV file `DESeq2.csv` contains the analysis results and is needed by Anaquin.

5.6.3 | Quantify Differential Gene Expression (Anaquin, in R)

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

```
$ anaquin RnaFoldChange -o 5.6.3 -m MRN029_v001.csv -method gene -ufiles DESeq2.csv
```

Where:
`RnaFoldChange` is name of the tool
`5.6.3` is the specified output directory (`-o`)
`MRN029_v001.csv` is the reference file for Mixture A&B (`-m`)
`DESeq2.csv` is the generated differential gene expression file by DESeq2 (`-ufiles`)

`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.csv` – 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.

5.6.3.1 | Example summary statistics from RnaFoldChange

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

```
       Summary for input: DESeq2.csv

-------Reference Annotations

       Synthetic: 78 genes
       Mixture file: MRN029_v001.csv

-------Genes Expressed

       Synthetic: 78 genes
       Genome:    60554 genes

-------Linear regression (log2 scale)

       Slope:       0.949774
       Correlation: 0.883807
       R2:          0.781115
       F-statistic: 264.077
       P-value:     0
       SSM:         397.481, DF: 1
       SSE:         111.383, DF: 74
       SST:         508.863, DF: 75
```

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

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

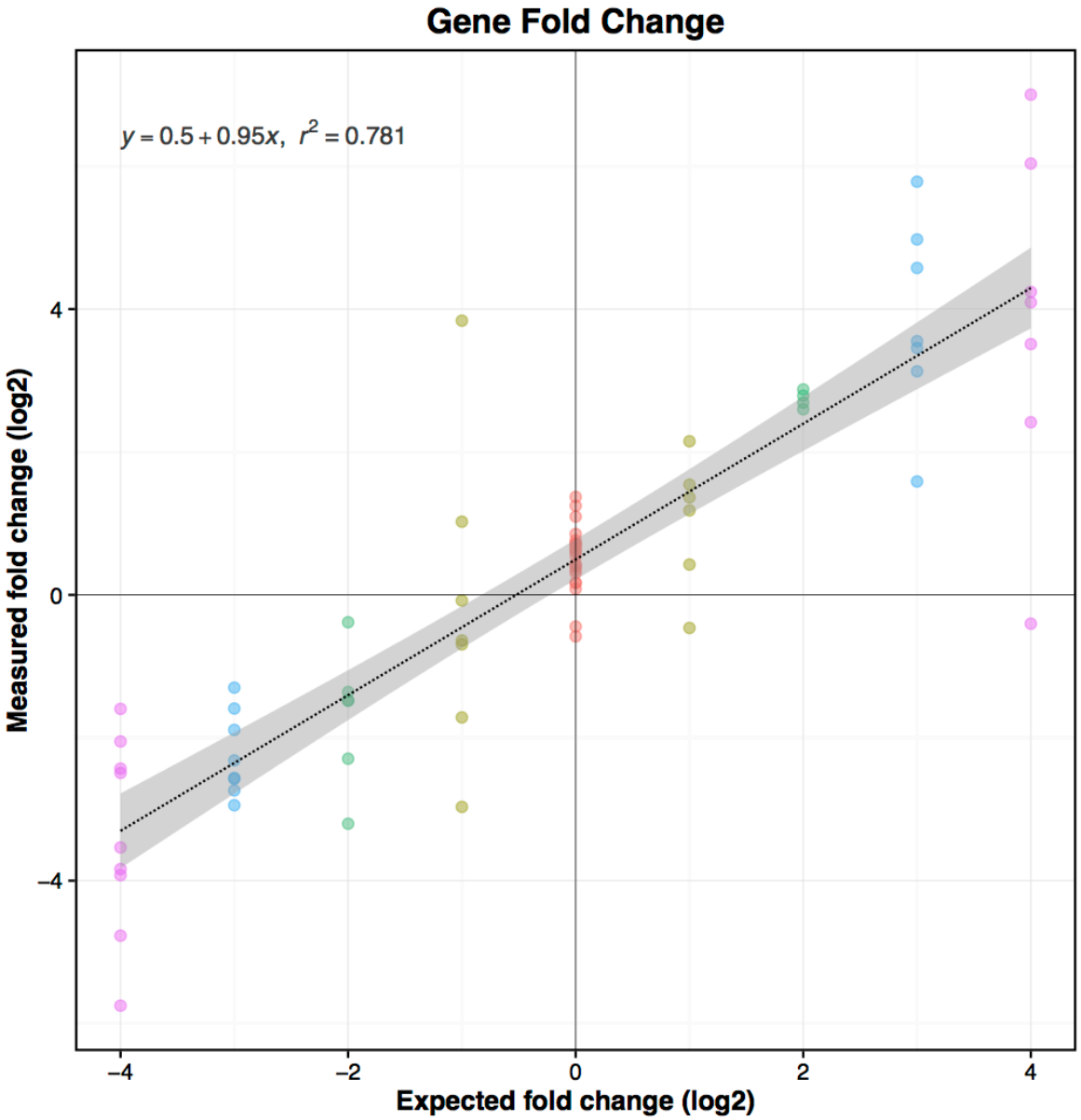

Figure 5.6.3.2.A: Scatter plot indicates correlation between observed and expected differential log-fold for each sequin gene. Sequin genes correspond to points coloured according to level of input concentration. The regression model is shown in top-left corner, with regression line indicated (blue line with shadow being 95% confidence interval).

2 | We can customize the script file `RnaFoldChange_fold.R` for the plot appearance. For example, we can add standard deviation to the plot. Find the following line in the script file:

```
> PlotLinear(data, title=title, xlab=xlab, ylab=ylab, showInter=TRUE, showLOQ=FALSE
```

Add a new option `showSD` (show standard deviation) to the function, update the line to:

```
> PlotLinear(data, title=title, xlab=xlab, ylab=ylab, showInter=TRUE, \
      showLOQ=FALSE, showSD=TRUE)
```

Rerun the script, error bars should now be rendered vertically:

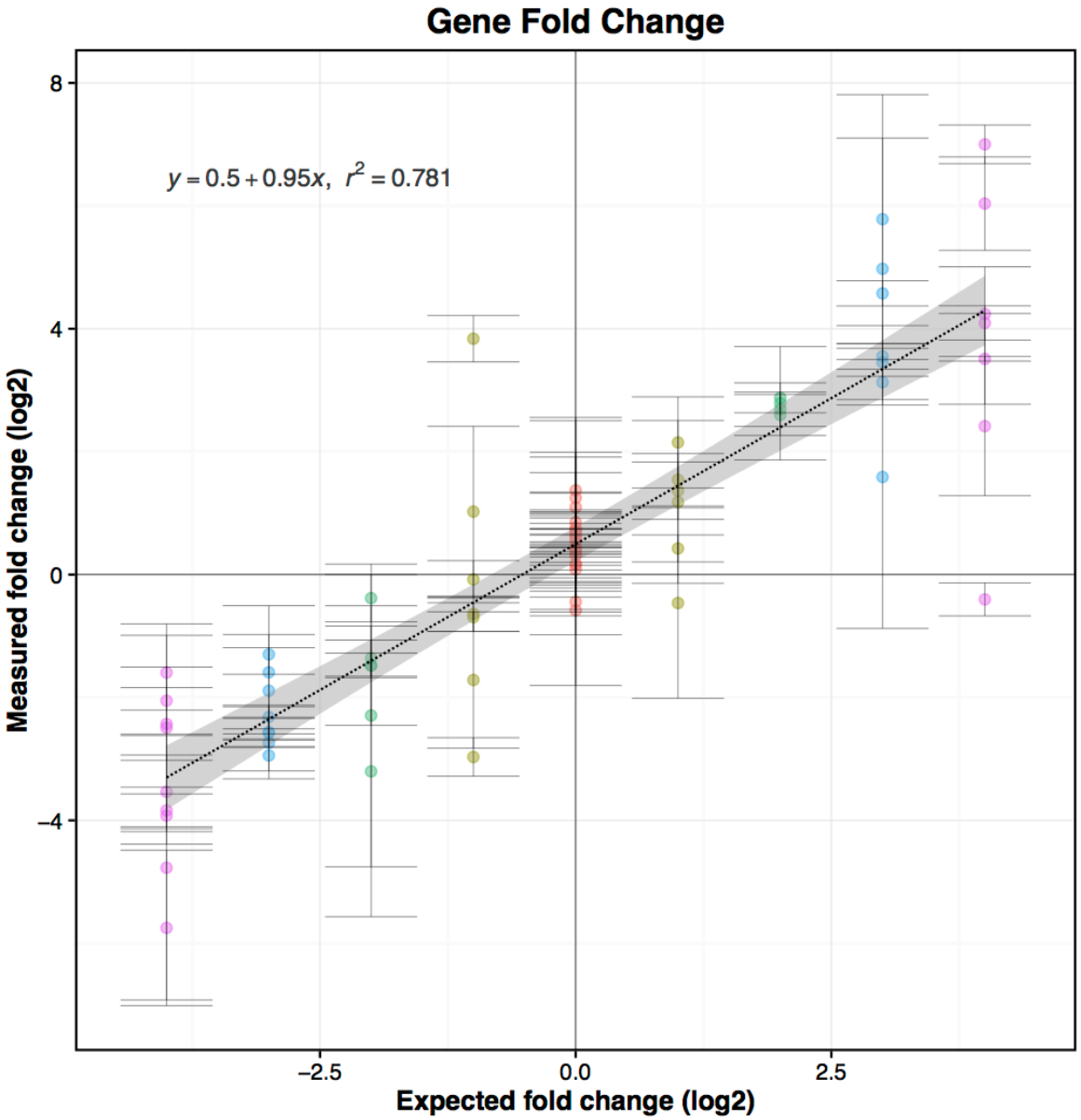

Figure 5.6.3.2.B: Same plot as **Figure 5.6.3.1.A** but standard deviation is now shown vertically.

5.6.3.3 | Plot ROC curve (Anaquin, R)

1 | 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:

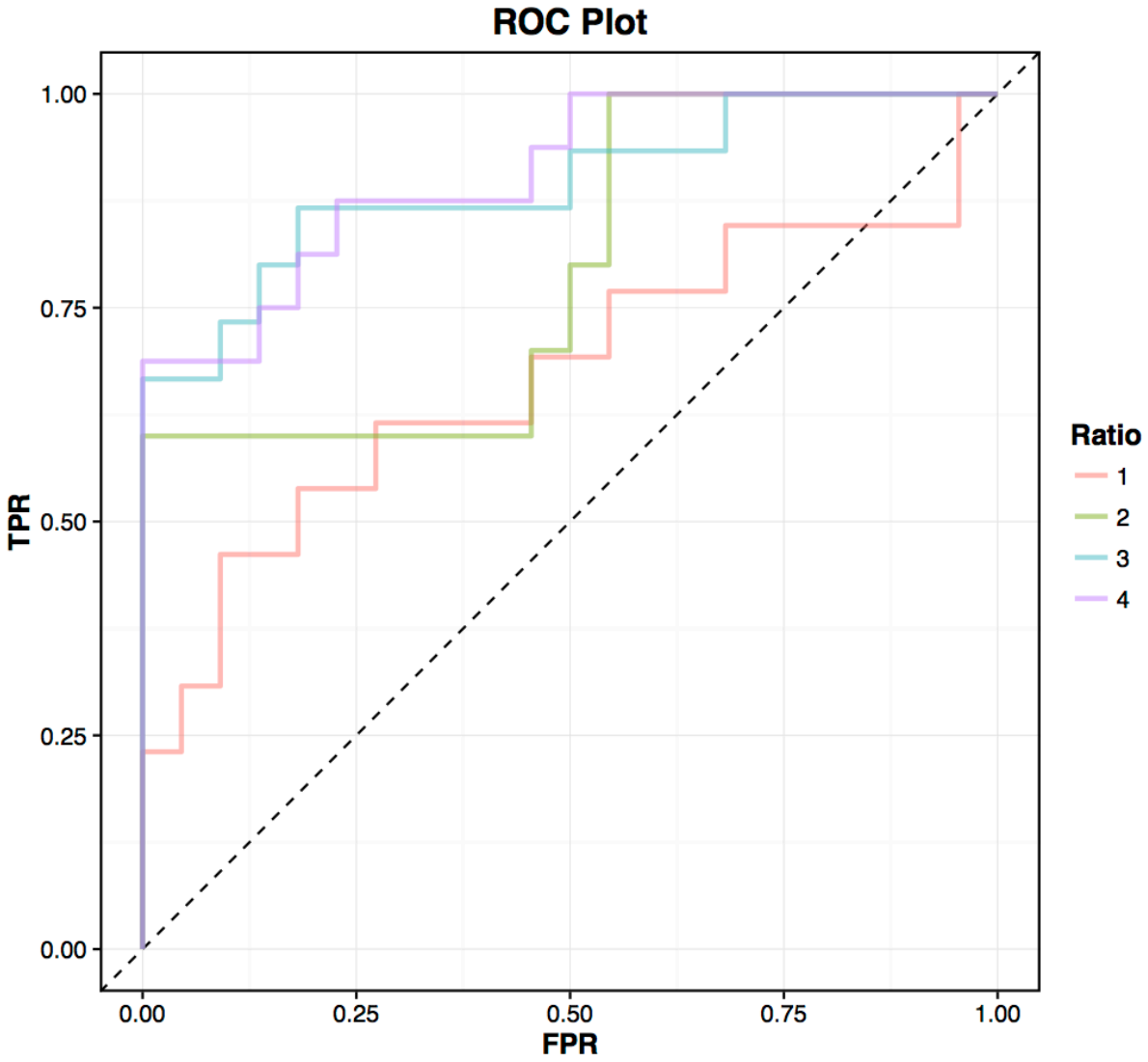

Figure 5.6.3.3: The ROC plot illustrates the performance of identifying variably expressed sequins (true-positives) relative to constantly expressed sequence (false-positives, no fold change), with curve ranked by significance (p-value ascribed by DESeq2). Each sequin group (grouped according to expected log fold change; LFC) is plotted independently.

2 | 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:

Ratio	AUC
1	0.6713
2	0.7955
3	0.8939
4	0.9062

5.6.3.4 | Plot LODR curve (Anaquin, R)

1 | LODR plot indicate the sensitivity with which fold-change differences can be identified relative to the mean gene expression level. This illustrates the impact of expression-dependent bias in differential gene expression analysis.

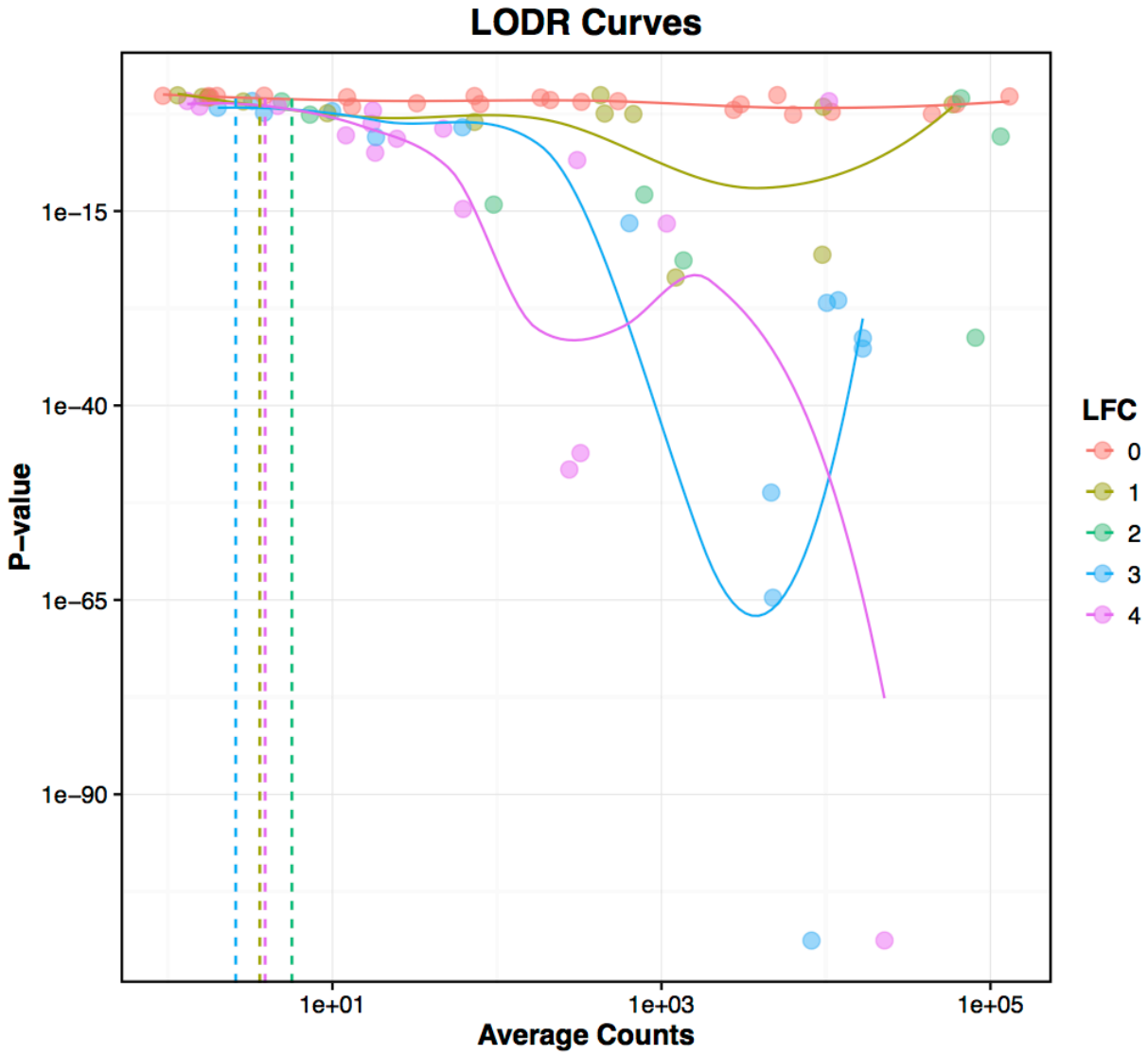

Figure 5.6.3.4: LODR curves for sequin genes indicate the confidence in measured fold-change for gene sequin subsets at different mean expression. The x-axis is the average counts (across replicates) for gene expression, relative to the confidence (p-value) ascribed to DESeq2 to each identified fold-change. The curves are estimated by local regression estimations and are colored by the sequin group, with 90% confidence interval indicated by shadow.

5.7 | Workflow - Alignment-Free (Kallisto)

Alignment-free approaches do not require 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.

5.7.1 | 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 ARN024_v001.index ARN025_v001.fa
```

Where:
`ARN025_v001.fa` is a reference FASTA file of sequin sequences.

`ARN004_v001.index` is the output index name (`-i`).

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 ARN024_v001.index -o A1 LRN087.1_val_1.fq LRN087.2_val_2.fq
```

The command will generate `abundance.tsv` in the `A1` output directory, which gives the estimated abundance for each transcript.

5.7.2 | Quantify Kallisto Quantification (Anaquin)

1 | We can compare the transcript quantification in the `abundance.tsv` file to the expected input transcript concentrations using the following command:

```
$ anaquin RnaExpression -o 5.7.2 -method isoform -m MRN027_v001.csv \
      -ufiles A1/abundance.tsv
```

Where:
`RnaExpression` is name of the tool
`5.7.2` is the output directory specified by `-o`
`isoform` specifies isoform differential analysis by `-method`
`MRN027_v001.csv` is the mixture A (`-m`)
`A1/abundance.tsv` is the converted Kallisto abundance file specified by `-ufiles`

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 5.7.2 -method gene -m MRN027_v001.csv \
      -ufiles A1/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 in the library. Please see **Appendix A.3** for an example of the output file, including a description and interpretation of statistics.
2. `RnaExpression_sequins.csv` – statistics for estimated expression at each individual sequin isoforms.
3. `RnaExpression_linear.R` – R script to plot the expression estimated for each sequin gene relative to expected input concentration.

5.7.2.1 | Plot Isoform Expression Sensitivity curve (Anaquin, R)

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

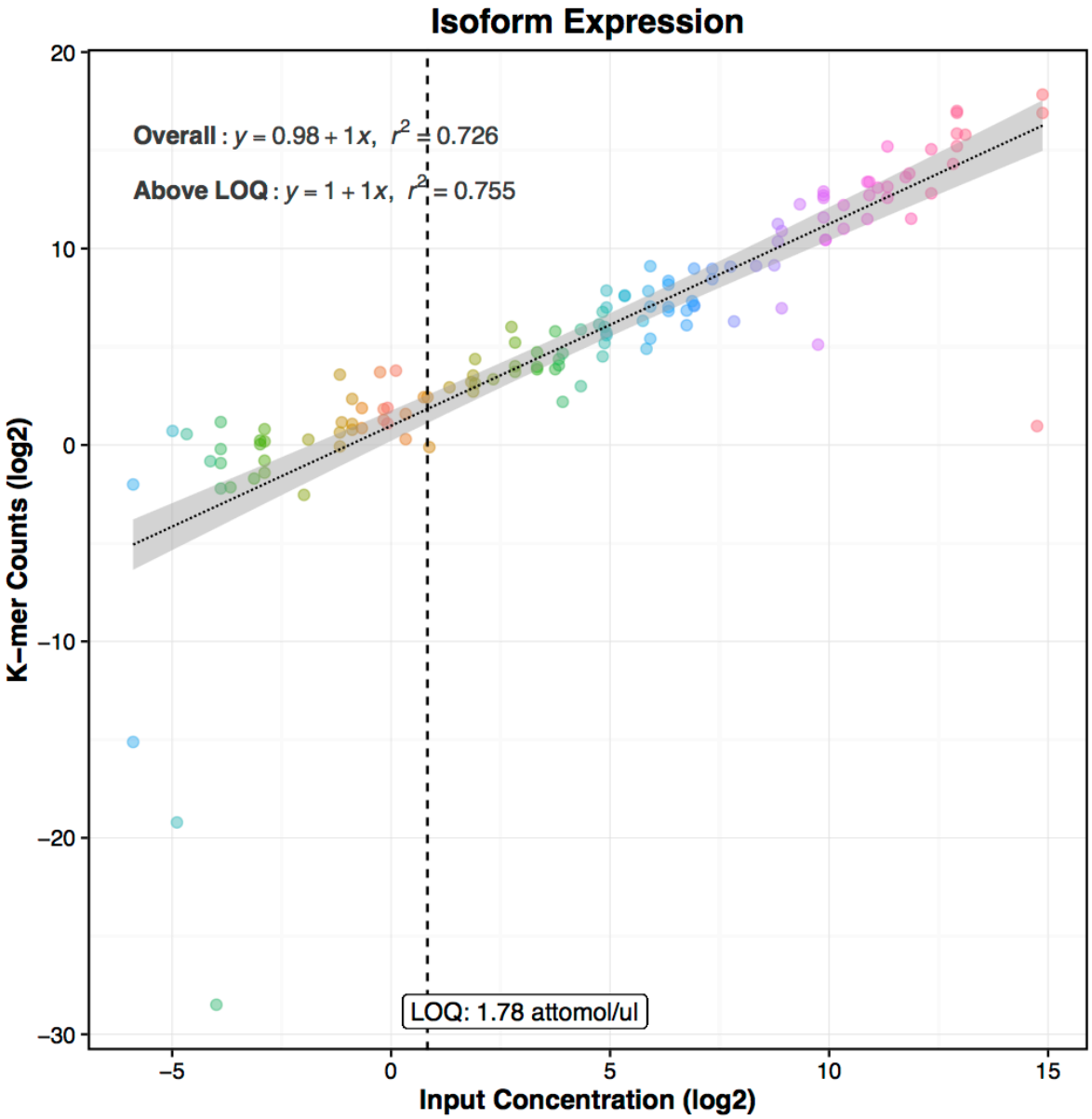

Figure 5.7.2.1 Scatter-plot illustrates the observed abundance (in FPKM) relative to the expected abundance (in attomoles/ul). The limit-of-quantification (LOQ; indicated by dashed line) is estimated by piecewise-linear segmentation (further details available in **Appendix B**). Synthetic genes below the LOQ are poorly measured. Above the LOQ, we fit a regression model (indicated by blue line with shadow depicts the 95% confidence interval).

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 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 –ufiles 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
-ufiles	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 to describe the global alignment profile. Field definitions:

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

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

ID	Name of the sequin gene
Length	Length of the sequin gene
Reads	Number of reads aligned to sequin gene
SnIntron	Sensitivity at the intron level
SnBase	Sensitivity at the base level

Example Output – RnaAlign_summary.stats

```
-------RnaAlign Summary Statistics

       Input alignment file: A1/accepted_hits.bam
       Reference annotation file: gencode_v24.ARN020_v001.gtf

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

       Synthetic: 36552138
       Genome:    11493036
       Dilution:  0.76

-------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: 20575711
       Spliced:     15976427

-------Alignments (Genome)

       Non-spliced: 8611159
       Spliced:     1754969
```

```
-------Comparison of alignments to reference annotation (Synthetic)

       *Intron level
        Sensitivity: 0.99
        Precision:   0.70

       *Base level
        Sensitivity: 0.99
        Precision:   0.98

-------Comparison of alignments to reference annotation (Genome)

       *Intron level
        Sensitivity: 0.29
        Precision:   0.90

       *Base level
        Sensitivity: 0.31
        Precision:   0.30
```

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 CSV format
Generated transcriptome file in GTF format

Usage Example

```
anaquin RnaAssembly –rgtf reference.gtf –ufiles transcripts.gtf
```

Tool Options

Required:

-rgtf	Reference annotation file in GTF format.
-ufiles	Generated transcriptome in GTF format.

Optional**:**

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

Outputs

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

User assembly file	Input sample transcriptome file in GTF format
Reference annotation file	Reference transcriptome annotation file in GTF format
Reference Gene Annotations (Synthetic)	Number of exons, introns, isoforms and genes on the *in silico* chromosome in the reference annotation
Reference Gene Annotations (Genome)	Number of exons, introns, isoforms and genes on the genome in the reference annotation
Synthetic (User Assemblies)	Number of exons, introns, isoforms and genes on the *in silico* chromosome in the user assembly file
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

`RnaAsssembly_sequins.csv` - – provides statistics for each individual sequin. Field definitions:

ID	Name of the sequin
Length	Length of the sequin
InputConcent	Input concentration in attomol/ul
Sn	Sensitivity of the sequin

`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:

'Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation.' Trapnell et. al., *Nature Biotechnology.* 2010 May;28(5):511-5.

'Evaluation of gene structure prediction programs.' Burset et. al., *Genomics*. 1996 Jun 15;34(3):353-67.

Example Output – `RnaAssembly_summary.stats`

```
-------RnaAssembly Summary Statistics

       User assembly file: transcript.gtf
       Reference annotation file: gencode_v24.ARN020_v001.gtf

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

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

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

       Genome: 570957 exons
       Genome: 347410 introns
       Genome: 199169 isoforms
       Genome: 60554 genes

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

       Synthetic: 515 exons
       Synthetic: 416 introns
       Synthetic: 99 isoforms
       Synthetic: 74 genes
```

```
-------User Gene Assemblies (Genome)

       Genome: 1366 exons
       Genome: 835 introns
       Genome: 531 isoforms
       Genome: 524 genes

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

       *Exon level
        Sensitivity: 0.417401
        Specificity: 0.908873

       *Intron
        Sensitivity: 0.454907
        Specificity: 0.988473

       *Base level
        Sensitivity: 0.059783
        Specificity: 0.928484

       *Intron Chain
        Sensitivity: 0.269231
        Specificity: 0.575342

       *Transcript level
        Sensitivity: 0.000000
        Specificity: 0.000000

        Missing exons:   468
        Missing introns: 371

        Novel exons:   16
        Novel introns: 1

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

       *Exon level
        Sensitivity: 0.002264
        Specificity: 0.951111

       *Intron
        Sensitivity: 0.002351
        Specificity: 0.989104

       *Base level
        Sensitivity: 0.001730
        Specificity: 0.959860

       *Intron Chain
        Sensitivity: 0.000534
        Specificity: 0.303333

       *Transcript level
        Sensitivity: 0.000000
        Specificity: 0.000000

        Missing exons:   563469
        Missing introns: 345991

        Novel exons:   18
        Novel introns: 1
```

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` can accept both 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 CSV 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:

```
anaquin RnaExpression -m mixture.csv -ufiles genes.gtf
```

For multiple replicates:

```
anaquin RnaExpression -m mixture.csv -ufiles A1.gtf -ufiles A2.gtf -ufiles A3.gtf
```

Tool Options

Required:

-m	Reference RnaQuin mixture file in CSV format
-method	Whether the analysis should be done at the gene or isoform level. The value must be either "`gene`" or "`isoform`".
-ufiles	Generated expression file in GTF or text format

Optional:

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

Outputs

`RnaExpress_summary.stats` – reports global summary statistics determined from all sequins. Field definitions:

Input file	User generated expression file in GTF format or text format
Reference annotations	Number of genes/isoforms on the *in silico* chromosome in the reference annotation
Mixture file	Reference RnaQuin mixture file in CSV format
Synthetic	Number of genes/isoforms on the *in silico* chromosome in the input expression file
Detection Sensitivity	Sequin with the lowest input that are detected
Genome	Number of genes/isoforms on the genome in the input expression file
LOQ Break	Breakpoint estimated by piecewise segmentation
LOQ Intercept	Intercept before and after the breakpoint
LOQ Slope	Slope before and after the breakpoint
LOQ Correlation	Correlation before and after the breakpoint
LOQ R2	Coefficient of determination before and after the breakpoint
LOQ Genome	Number of genes/isoforms in the human genome expressed below/above the LOQ
Correlation	Pearson's correlation of the linear model
Slope	Regression slope of the linear model
R2	Coefficient of determination of the linear model
F-statistic	Test statistic of the linear model
P-value	P-value probability
SSM	Sum of squares of the linear model
SSE	Sum of squares of residuals
SST	Sum of squares of the total variation

`RnaExpress_sequins.csv` - detailed statistics for each sequin in the reference. Field definitions:

ID	Name of the sequin
Length	Length of the sequin
InputConcent	Input concentration in attomol/ul
Observed	Observed expression (eg: FPKM)

`RnaExpress_express.R` - R-script for building a linear model between expression level (dependent variable) and input concentration (independent variable) on the logarithm scale.

Example Output – RnaExpression_summary.stats

```
-------RnaExpression Output
       Summary for input: A1/transcripts.gtf
       *Arithmetic average and standard deviation are shown

-------Reference Transcript Annotations

       Synthetic: 164 isoforms
       Mixture file: MRN027_v001.csv

-------Isoform Expressed

       Synthetic: 164
       Detection Sensitivity: 0.00393391 (attomol/ul) (R2_38_1)

       Genome: 198485

-------Limit of Quantification (LOQ)

       *Estimated by piecewise segmented regression
```

```
       Break: 3.55 attomol/ul (R1_32_1)

       *Below LOQ
       Intercept:   2.52
       Slope:       0.15
       Correlation: 0.48
       R2:          0.23
       Genome:      193948

       *Above LOQ
       Intercept:   2.07
       Slope:       1.01
       Correlation: 0.95
       R2:          0.91
       Genome:      4537

-------Linear regression (log2 scale)

       Correlation: 0.95
       Slope:       0.73
       R2:          0.90
       F-statistic: 1288.39
       P-value:     0.00
       SSM:         2604.38, DF: 1
       SSE:         305.23, DF: 151
       SST:         2909.61, DF: 152
```

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, area under the curve (AUC) statistics and limit of detection of ratio (LODR) estimates.

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 CSV format (must have two columns, for input concentration defined for mixture A and B respectively)
Generated differential analysis file. Supported formats: Anaquin, Cuffdiff and DESeq2.

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

Usage Example

```
anaquin RnaFoldChange –m mixture.csv –method gene –ufiles diff.txt
```

Additional Information

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

'Assessing technical performance in differential gene expression experiments with external spike-in RNA control ratio mixtures.' Munro et al., *Nature Communications*. 2014 Sep 25;5:5125.

Tool Options

Required:

-m	Reference RnaQuin mixture file in CSV format
-method	Whether the analysis should be done at the gene or isoform level. The value must be either "`gene`" or "`isoform`".
-ufiles	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:

Input file	User generated differential analysis file
Reference (Synthetic)	Number of genes/isoforms detected on the *in silico* chromosome
Mixture	Reference RnaQuin mixture file
Synthetic (Expressed)	Number of genes/isoforms expressed on the *in silico* chromosome
Genome (Expressed)	Number of genes/isoforms expressed on the genome
Slope	Regression slope of the linear model
Correlation	Pearson's correlation of the linear model
R2	Coefficient of determination of the linear model
F-statistic	Test statistic of the linear model
P-value	P-value under the null hypothesis
SSM	Sum of squares of the linear model
SSE	Sum of squares of residuals
SST	Sum of squares of the total variation

`RnaFoldChange_sequins.csv` – provides statistics for individual sequin in the reference. Field definitions:

ID	Name of the sequin
Length	Length of the sequin
Sample1	Expression level for the first sample
Sample2	Expression level for the second sample
ExpLFC	Expected log-fold change
ObsLFC	Observed log-fold change
SD	Standard deviation of the measurement
Pval	Observed p-value probability
Qval	Observed q-value probability
Average	Normalized counts of all samples

`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.

`RnaFoldChange_LODR.R` - R script for building the limit-of-detection-ratio (LODR) curves for the sequin groups.

Example Output – `RnaFoldChange_summary.stats`

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

       Summary for input: DESeq2.txt

-------Reference Annotations

       Synthetic: 78 genes
       Mixture file: MRN029_v001.csv

-------Genes Expressed

       Synthetic: 76 genes
       Genome:    60500 genes

-------Linear regression (log2 scale)
```

```
Slope:       0.64319
Correlation: 0.797842
R2:          0.636551
F-statistic: 129.605
P-value:     0
SSM:         180.632, DF: 1
SSE:         103.134, DF: 74
SST:         283.766, DF: 75
```

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.

For example, if we have two technical samples, and they have the following characteristics:

- Library A with 2 million reads on the *in silico* chromosome (10 million reads in total)
- Library B with 1 million reads on the *in silico* chromosome (10 million reads in total)

Obviously library A has higher sequencing depth than library B. Downstream analysis might be biased unless the sequins are comparable across the samples.

RnaSubsample requires users to specify preferred dilution fraction, typically, we recommend between 1% to 10%. Further details are discussed in **Section 5.4.3**.

Firstly, this tool calculates the number of alignments on the *in silico* chromosome and genome. This is reported as "*Before subsampling*" in the summary statistics. Next, it uses the specified fraction to deduce how much reads on the *in silico* chromosome should be subsampled and is reported as "*After subsampling*" in the summary statistics.

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

```
anaquin RnaSubsample –method 0.01 –ufiles alignment.bam
```

Tool Options

Required:

-method	Dilution fraction as a floating number. For example, 0.01 is 1% and 0.10 is 10% etc.
-ufiles	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:

User generated alignment	User generated alignment file in SAM/BAM format
Synthetic (Before)	Number of alignments mapped to the *in silico* chromosome before subsampling
Genome (Before)	Number of alignments *not* mapped to the *in silico* chromosome before subsampling
Dilution (Before)	Synthetic (Before) / (Synthetic (Before) + Genome (Before))
User Dilution	User specified dilution (specified by `-method`)
Normalization	Calculated normalization factors applied in subsampling *in silico* alignments

Synthetic (After)	Number of alignments mapped to the *in silico* chromosome after subsampling
Genome (After)	Number of alignments *not* mapped to the *in silico* chromosome after subsampling
Dilution (After)	Synthetic (After) / (Synthetic (After) + Genome (After))

Example Output – RnaSubsample_summary.stats

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

       User generated alignment: RnaSubsample_summary.stats

-------User alignments (before subsampling)

       Synthetic: 36552138 reads
       Genome:    11493036 reads
       Dilution:  0.760787

       * Dilution specified by the user:
       Fraction: 0.050000

       * Normalization applied in subsampling:
       Normalization: 0.0165489

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

       Synthetic: 603687 reads
       Genome:    11493036 reads
       Dilution:  0.049905
```

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:

```
$ data <- read.csv('~/Documents/RnaExpression_sequins.csv', 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:

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

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

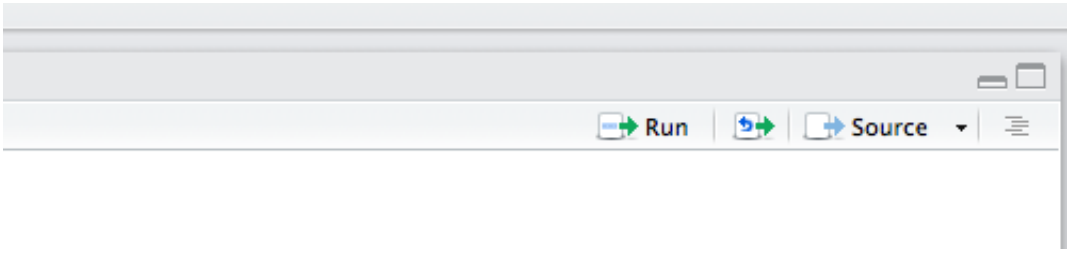

Figure B.1.1: The Source button in RStudio. Open the script file, then click the button will allow RStudio to execute the script.

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):

```
$ 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 | Install third-party R-packages

1 | To install the `ggplot2` package, use the following:

```
> install.packages("ggplot2")
```

And then to load it, use the following:

```
> library("ggplot2")
```

2 | To install the `DESeq2` package, start R and enter:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("DESeq2")
```

```
> library("DESeq2")
```

3 | To install the `qvalue` package, start R and enter:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("qvalue")
```

4 | To install the `GenomicFeatures` package, start R and enter:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("GenomicFeatures")
```

5 | To install the `GenomicAlignments` package, start R and enter:

```
> source("https://bioconductor.org/biocLite.R")
> biocLite("GenomicAlignments")
```

B.1.4 | Modify R-script

The R-script generated by Anaquin is customizable. We should demonstrate how to apply filtering and ranking for a ROC plot. The example comes from **Section 5.6.3**.

1 | Download the dataset by:

```
$ https://s3.amazonaws.com/sequins/manuscripts/5.6.3.zip
$ unzip 5.6.3.zip
```

The script should look like **Script B.1.4**.

```
library(Anaquin)

# Load reference sequins
data <- read.csv('5.6.3/RnaFoldChange_sequins.csv', row.names=1, sep='\t')

# Classify sequins against the negative controls (LFC 0)
data$label <- ifelse(abs(data$ExpLFC) <= 0, 'FP', 'TP')

title <- 'ROC Plot'

# Create Anaquin data set for plotROC
anaquin <- AnaquinData(analysis='plotROC',
                          seqs=row.names(data),
                       input=data$ExpLFC,
                    measured=data$ObsLFC,
                       score=1-data$Pval,
                       label=data$label)

plotROC(anaquin, title=title, refRats=0)
```

Script B.1.4: ROC script for **Section 5.6.3**

The script will generate the following ROC plot (**Figure B.1.4.A**).

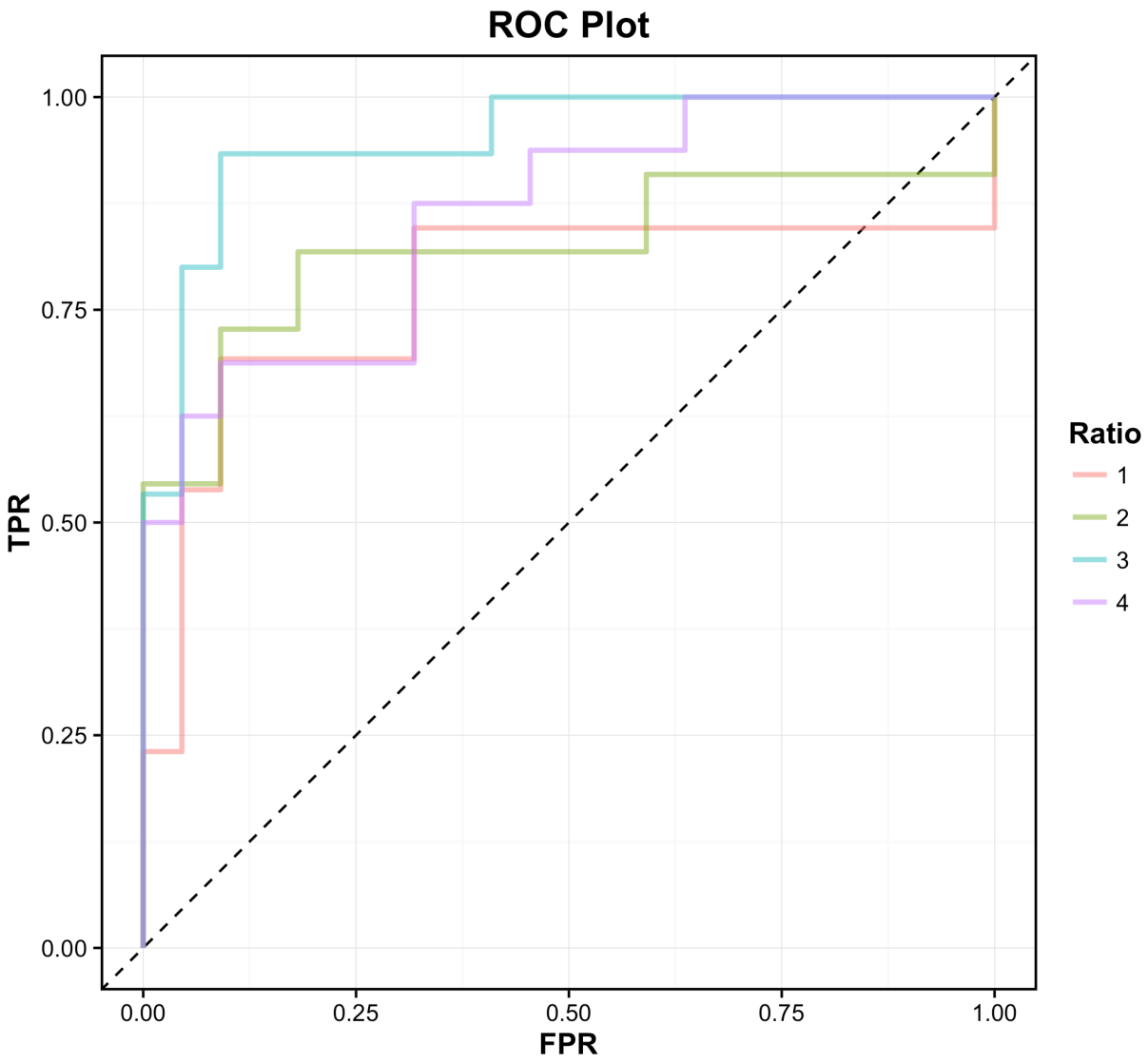

Figure B.1.4.A: ROC plot for **Script B.1.4**.

2 | `data` is an R data frame object holding expected log-fold change (LFC), observed LFC, p-value (and more):

```
> head(data[,c('ExpLFC','ObsLFC','Pval')])

       ExpLFC    ObsLFC         Pval
R1_101     -3 -1.944348 7.371597e-02
R1_102     -4 -2.678297 2.025809e-05
R1_103     -1 -1.066247 1.131863e-11
R1_11      -4 -4.465963 6.532765e-21
R1_12       1 -0.860582 1.771818e-01
R1_13       0 -0.078232 6.112909e-01
```

We can filter out sequins with expected LFC equal to 4 with the following R-code:

```
> data <- data[abs(data$ExpLFC) != 4,]
```

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

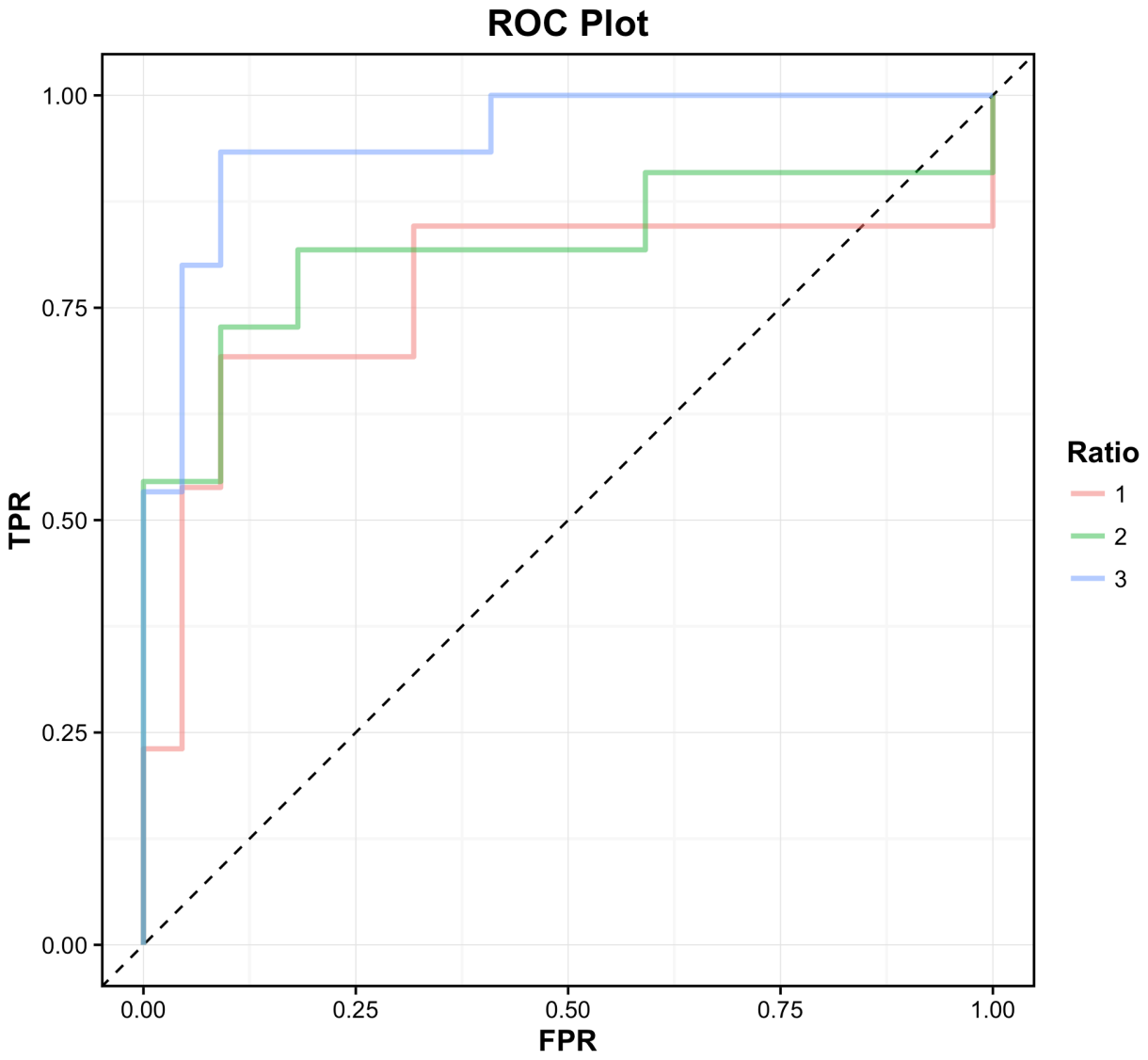

Figure B.1.4.B: ROC plot. LFC 4 is not shown.

3 | The script uses the p-value for ranking the ROC plot. But, we can also rank by any other numerical variable. For example, we can convert the probabilities to q-values and use them for ranking. In R, type the following:

```
> library('qvalue')
> qval <- qvalue(data$Pval, fdr.level=0.10)$qvalues
> anaquin <- AnaquinData(analysis='PlotROC',
                         seqs=row.names(data),
                       input=data$ExpLFC,
                    measured=data$ObsLFC,
                       score=1-qval,
                       label=data$label)
> plotROC(anaquin, title=title, refRats=0)
```

Note `score` is now `1-qval`. Run the script again to regenerate new ROC plot.

B.2 | AnaquinData

Every analysis requires an input dataset. The function enables that is `AnaquinData`. `AnaquinData` is a S4 constructor, inputs vary depending on the analysis. We encourage users to use the command-line Anaquin for R-script auto-generation.

The constructor requires the following compulsory inputs:

`analysis`	Analysis type
`seqs`	Sequin names

`analysis` gives the type of the analysis; how a statistical model should build for sequins. The current release supports the following analysis:

`PlotLinear`	Linear model with sequins
`PlotLogistic`	GLM logistic model with sequins
`PlotROC`	ROC analysis with sequins
`PlotLODR`	LODR (LOESS) analysis with sequins

`seqs` gives the sequin names; critical information otherwise no statistical model for sequins is possible. `AnaquinData` will give an error message unless both `analysis` and `seqs` are given.

The constructor also accepts the following optional inputs:

`std`	Standard deviation
`pval`	P-value probability
`qval`	Q-value probability
`ratio`	Expected sequin ratio (eg: allele frequency)
`input`	Input concentration (attomol/ul)
`measured`	Measured variable (eg: FPKM)
`label`	Classified labels (eg: 'TP', 'FP')
`score`	How to rank ROC points?

It is important to check the documentation for individual analysis on what the required inputs. We will illustrate an example for plotting a scatter plot between input concentration against measured FPKM values.

1 | Open R/RStudio. We have prepared a CSV file at:

https://s3.amazonaws.com/sequins/manuscripts/Appendix_B2.csv

Download the data in R:

```
> Library('Anaquin')
> data <-
read.csv(url('https://s3.amazonaws.com/sequins/manuscripts/Appendix_B2.csv'),
row.names=1, sep='\t')
```

Quickly examine the data:

```
> class(data)

[1] "data.frame"

> head(data)

       Length InputConcent   Observed
R1_101    720      15.1062    4.29354
R1_102   1491      15.1062    6.41774
R1_103   1857     966.7970  271.69600
R1_11     786     241.6990   67.32120
R1_12    1583      30.2124    3.19430
R1_13    1941    7734.3800 2628.79000
```

The variable `data` is a data frame. It has three columns; the first column is the nucleotide base length of each sequin, the second column is the input concentration in attomol/ul and the third column is the measured FPKM values.

2 | Check **Apprendix B.2** to **B.6** (documentation for R-functions)

- `plotLinear` (**Apprendix B.5**)
- `plotLogistic` (**Apprendix B.6**)
- `PlotROC` (**Apprendix B.3**)
- `PlotLODR` (**Apprendix B.4**)

Obviously, we will use the `PlotLinear` to build a linear model. **Apprendix B.4** has a section `Data Inputs` (other R-function has the section too):

Data Inputs

names	Sequin names (eg: `R1_1_1`)
input	Input concentration in attomol/ul
measured	Measured abundance (eg: FPKM)

This table states `PlotLinear` requires:

- Sequin names (`names`)
- Input concentration for each sequin (`input`)
- Measured FPKM for each sequin (`measured`)

The following code will create a data set for `PlotLinear`:

```
> anaquin <- AnaquinData(analysis='PlotLinear'
                              seqs=row.names(data),
                             input=log2(data$InputConcent),
                          measured=log2(data$Observed)
```

Where

`AnaquinData` is the S4 constructor
`names` gives the sequin names
`input` gives the input concentration
`measured` gives the measured FPKM

COMMENTS: In this example, the attribute `measured` is FPKM but it can also be any other attributes correlated to the input concentration (eg: k-mer counts).

3 | Next, we can create a linear model with the following code:

```
> PlotLinear(data)
```

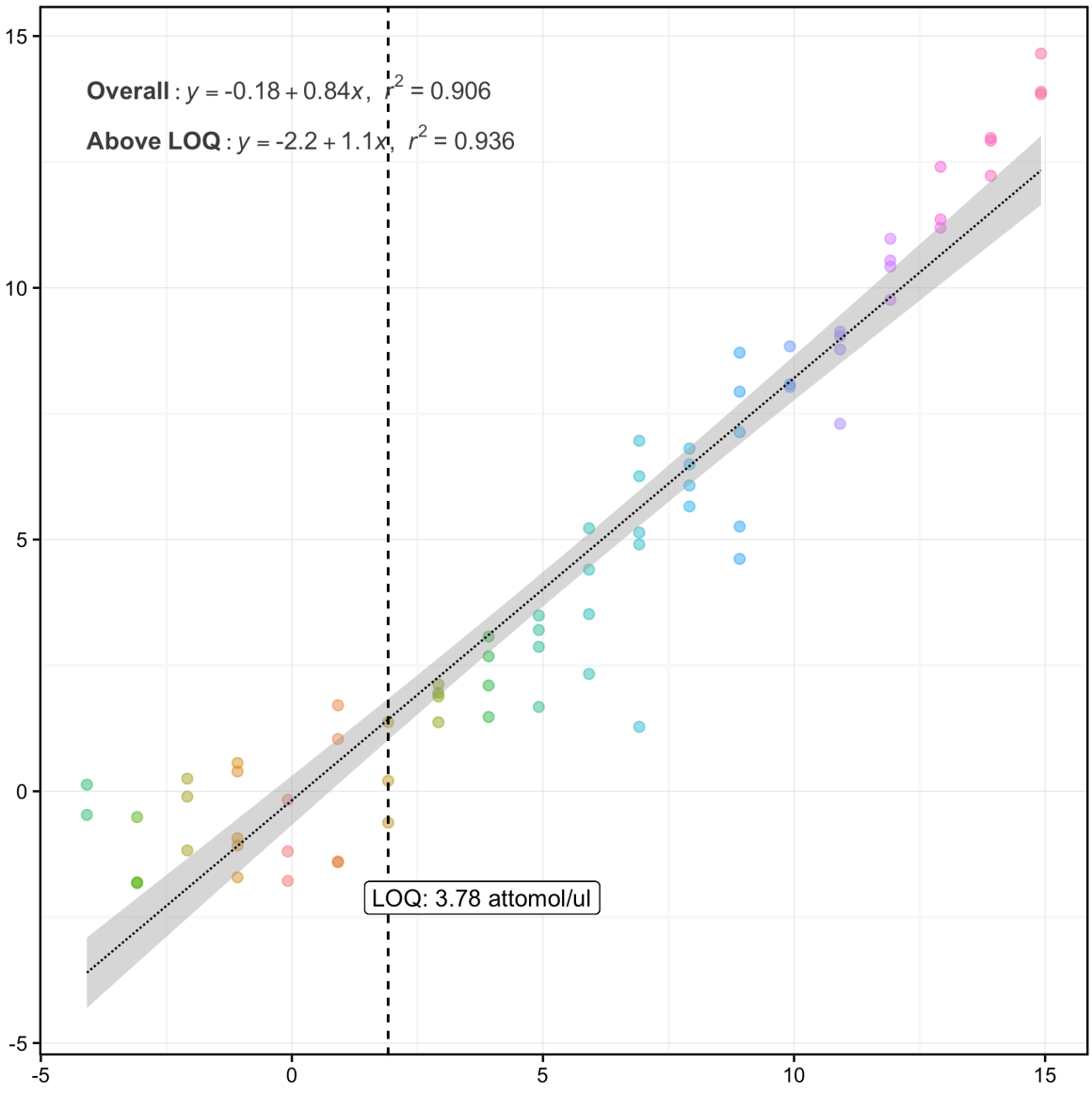

Figure B.2: Scatter plot generated by R-script in **Apprendix B.2**.

B.3 | plotROC

Description

Create a receiver operating characteristic (ROC) plot. The plot models true positive rate (TPR) against false positive rate (FPR) at various threshold settings.

Usage

plotROC(data, refRats, title, legTitle, …)

Arguments

data	Anaquin dataset created by `AnaquinData`. It needs to define information in **Data Inputs**. **Apprendix B.2** has details on how to use the function.
refRats	Reference ratio group
title	Title of the plot. Default to NULL.
legTitle	Title of the legend. Default to 'Ratio'.
…	Reserved for internal testing

Data Inputs

seqs	Sequin names
label	Classified labels ('TP' or 'FP')
score	How the ROC points should be ranked
ratio	Expected ratio; eg: expected log-fold ratio

Details

The ROC plot illustrates the performance of an experiment as the scoring threshold is varied. The curve is created by plotting the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings.
The plot provides diagnose performance 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).

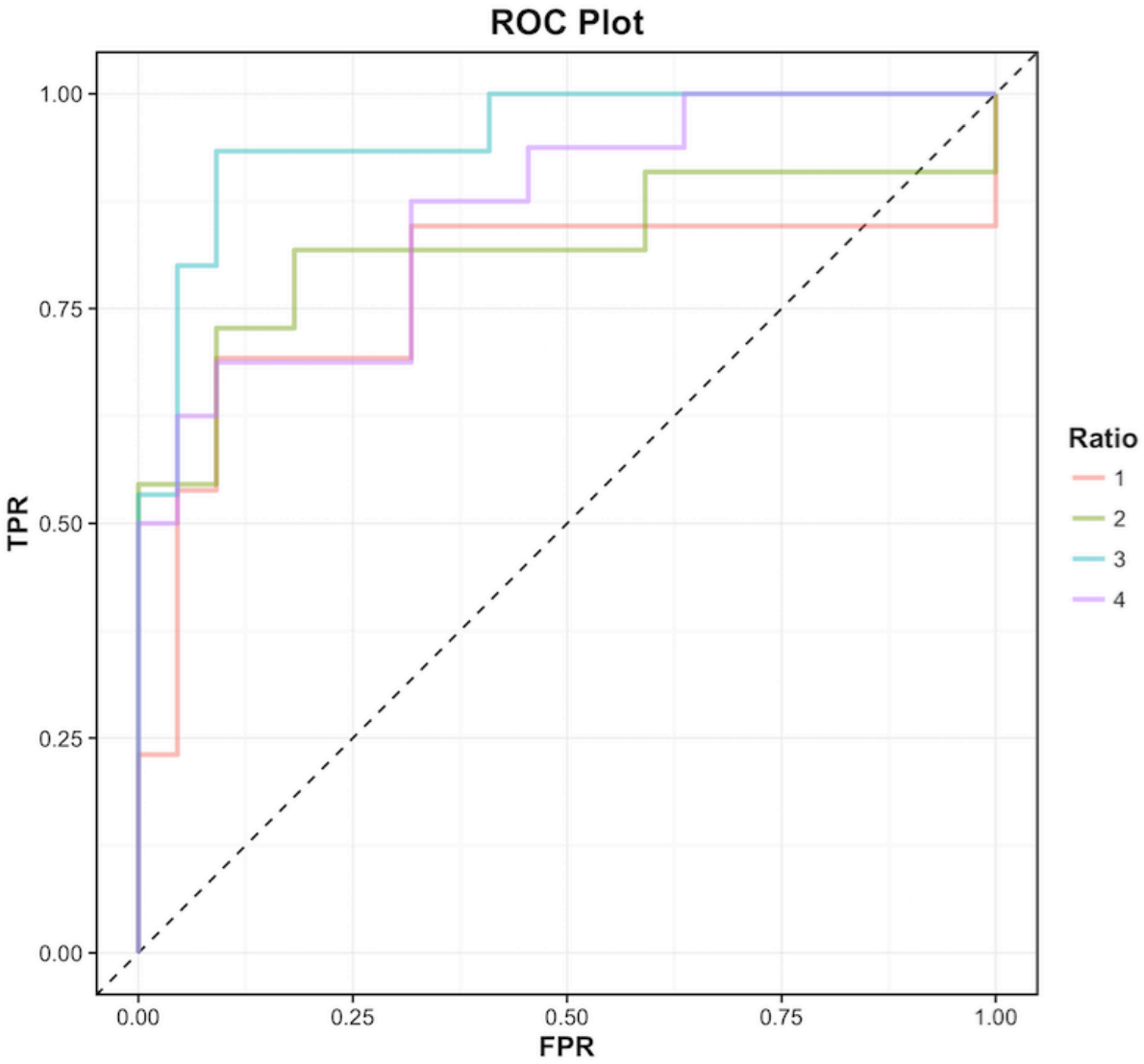

Figure B.3.1: ROC curve measure true / false positive differential identification as ranked by p-value.

B.4 | plotLODR

Create a Limit-of-Detection Ratio (LODR) plot between measured abundance (x-axis) and p-value probability (y-axis).

Usage

plotLODR(data, FDR, title, xlab, ylab, legTitle, showConf, …)

Arguments

data	Anaquin dataset created by `AnaquinData`. It needs to define information in **Data Inputs**. **Apprendix B.2** has details on how to use the function.
FDR	Chosen false-discovery-rate. Default to 0.1.
title	Title of the plot. Default to NULL.
xlab	Label for the x-axis. Default to NULL.
ylab	Label for the y-axis. Default to NULL.
legTitle	Title for the legend. Default to 'Ratio'.
showConf	Show confidence interval? Default to FALSE.
…	Reserved for internal testing

Data Inputs

names	Sequin names
measured	Measured abundance (eg: average counts, DP field in a VCF file etc)
ratio	Expected ratio; eg: expected log-fold ratio or expected allele frequency etc
pval	P-value probability

Details

The LOD plot indicates the confidence in measuring synthetic features (eg: isoforms, SNPs, Indels, etc.) are detected relative to their abundance (eg: allele frequencies, read counts, etc.). The x-axis is the abundance, relative to the confidence (p-value) in the y-axis.

The curves are estimated by local regression estimations, and are colored by the sequin group, with 90% confidence interval indicated.

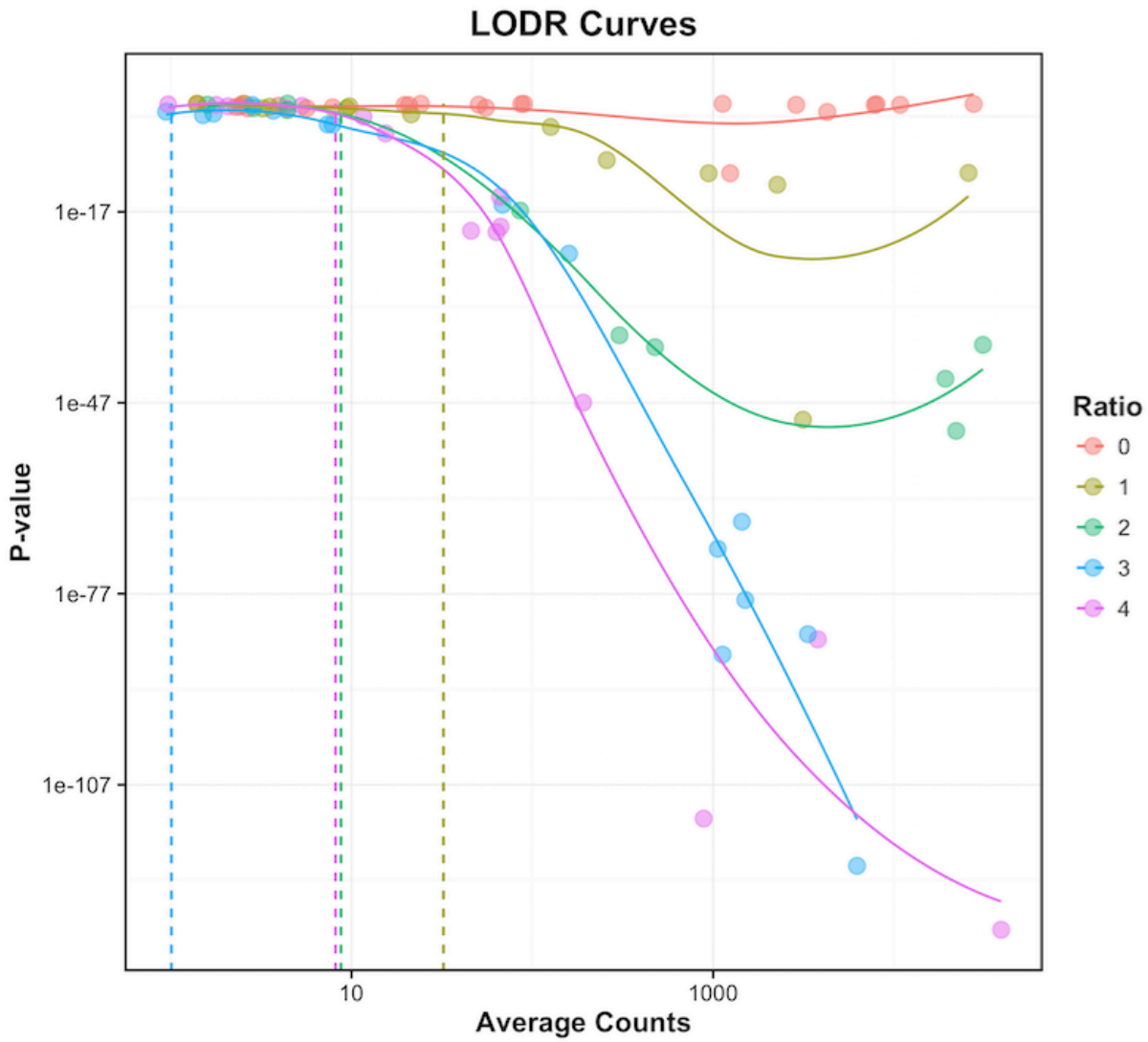

Figure B.4: Limit of detection ratio (LODR) plot indicates the confidence with which synthetic sequins are detected relative to their p-value probabilities.

B.5 | PlotLinear

Plot a linear model between sequins against measured abundance (eg: FPKM).

Usage

plotLinear(data, xlab, ylab, title, showSD, showLOQ, showAxis, …)

Arguments

data	Anaquin dataset created by `AnaquinData`. It needs to define information in **Data Inputs**. **Apprendix B.2** has details on how to use the function.
xlab	Label for the x-axis. Default to empty label.
ylab	Label for the y-axis. Default to empty label.
title	Title for the plot. Default to empty title.
showSD	Show standard deviation bars vertically? Default to FALSE.
showLOQ	Show limit-of-quantification? Default to TRUE.
showAxis	Show x-axis and y-axis? Default to TRUE.
…	Reserved for internal testing

Data Inputs

seqs	Sequin names (eg: `R1_1_1`)
input	Expected abundance. (eg: input concentration)
measured	Measured abundance (eg: FPKM)

Details

The function plots expected abundance on the x-axis, and measured abundance on the y-axis. Generally, the expected abundance is simply the input concentration, although other measures (eg: expected allele frequency) is also possible. The plot builds a linear regression between the two variables, and reports the standard statistics; R2, correlation and regression parameters on the graph.

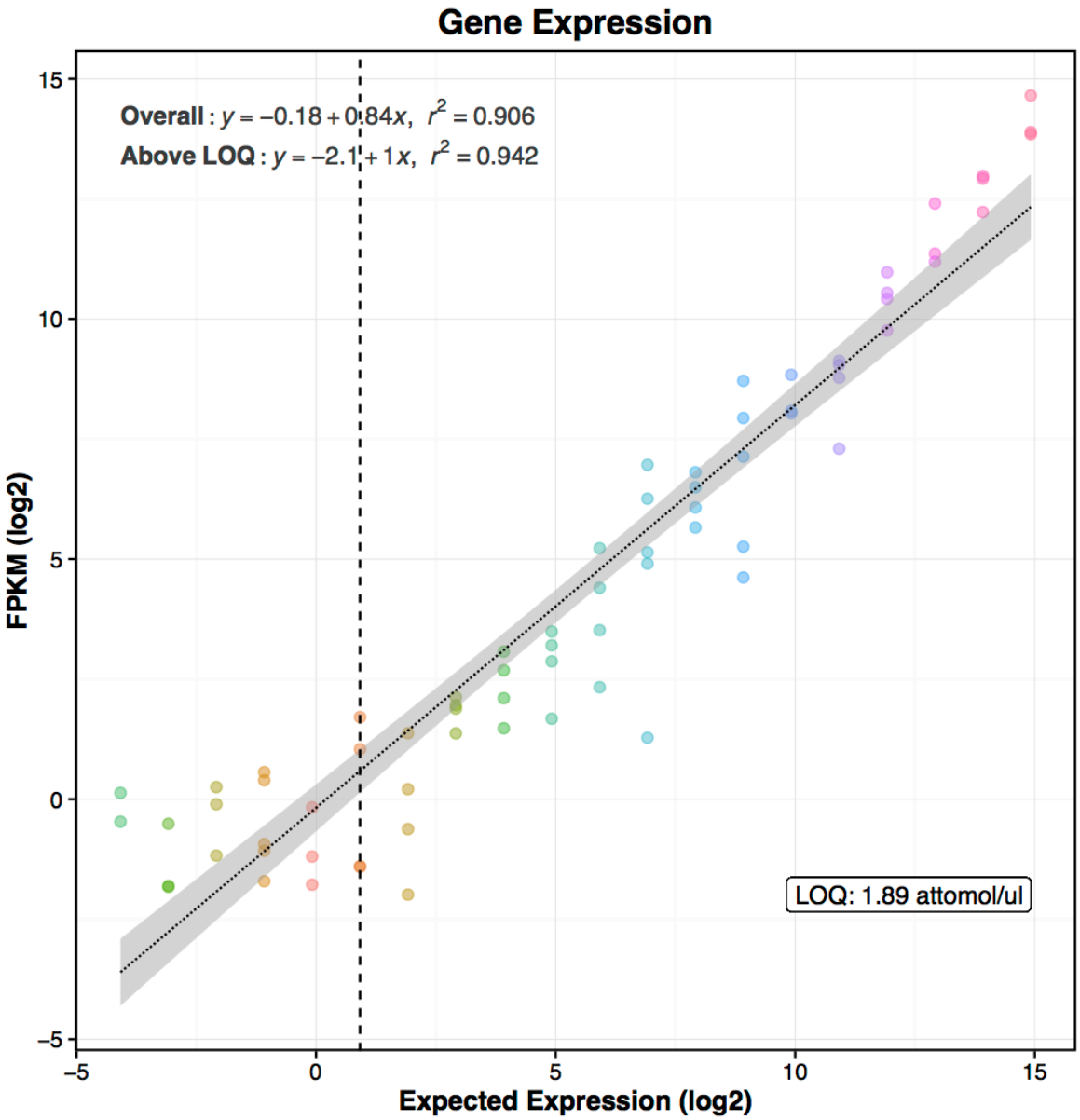

Figure B.5: Sample output for the PlotLinear R-function. The x-axis shows the input concentration for the sequins and the y-axis shows the measured FPKM.

B.6 | PlotLogistic

Plot a GLM logistic model between sequins against measured variable (eg: sensitivity).

Usage

PlotLogistic(data, title, xlab, ylab, showLOA, threshold, …)

Arguments

data	Anaquin dataset created by `AanquinData`. It needs to define information in **Data Inputs**. **Apprendix B.2** has details on how to use the function.
title	Title of the plot
xlab	Label for the x-axis
ylab	Label for the y-axis
showLOA	Show limit-of-assembly?
threshold	Threshold required for limit-of-assembly (LOA)
…	Reserved for internal testing

Data Inputs

seqs	Sequin names
input	Input concentration (in attomole/ul)
measured	Measured variable (eg: sensitivity)

Details

The analysis specifies expected abundance (eg: input concentration) on the x-axis, and measured abundance (eg: sensitivity) on the y-axis. Curve fitting is performed by non-linear least square fitting. LOA is defined as the least abundant sequin reaching sensitivity of `threshold`.

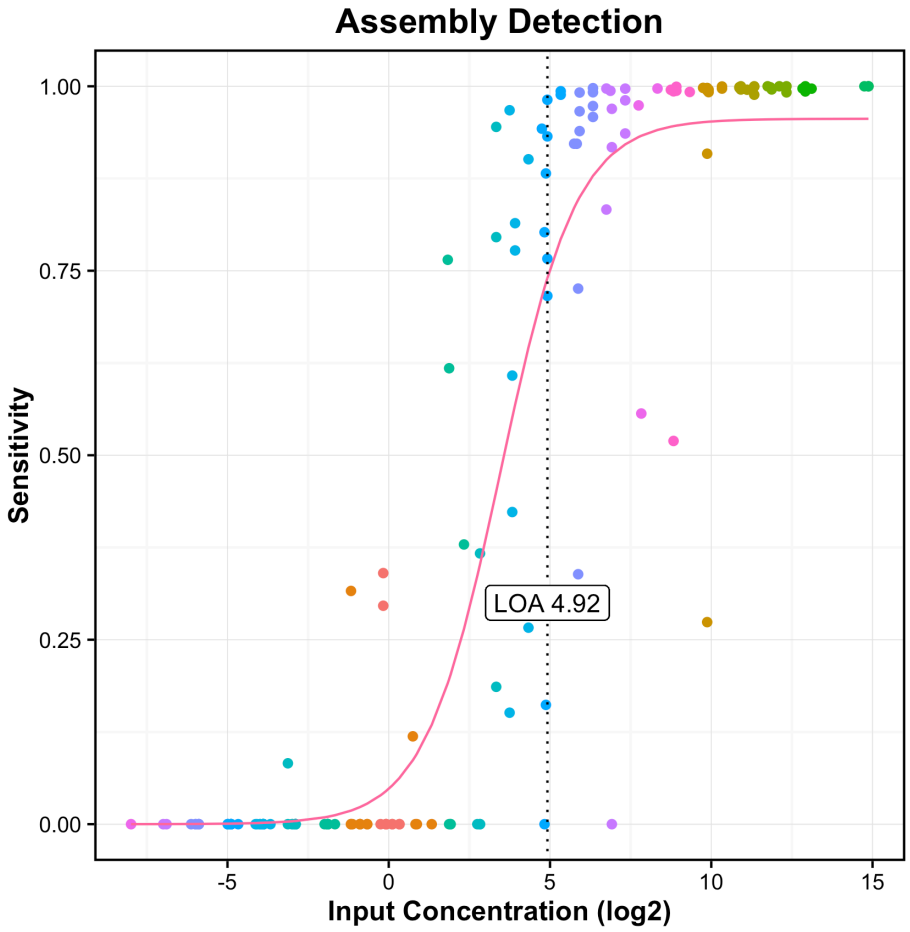

Figure B.6: Sample output for the PlotLogistic R-function. The x-axis shows the input concentration for the sequins and the y-axis shows the measured sensitivity.

Appendix C – Simple Text Input Formats

Whilst Anaquin supports the common file formats (eg: 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:

```
ChrID      GeneID         IsoformID      Abund
```

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.

ChrID	Name of the chromosome
GeneID	Name of the gene
IsoformID	Name of the isoform
Abund	Normalized abundance (eg: FPKM, normalized k-mer counts etc.). Repeat the column for multiple replicates (see below for an example).

Example lines from single replicate:

```
ChrID      GeneID         IsoformID      Abund
chr1   ENSG00000237613.2     -         0.317236
chr1   ENSG00000268020.3     -         0.17778
chrIS      R2_73             -         0.376111
chrIS      R2_71             -         0.252222
```

Example lines from multiple replicates:

```
ChrID      GeneID         IsoformID      Abund1       Abund2
chr1   ENSG00000237613.2     -         0.317236     0.417236
chr1   ENSG00000268020.3     -         0.17778      0.8878
chrIS      R2_73             -         0.376111        -
chrIS      R2_71             -         0.252222     0.252222
```

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`.

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

Example lines from the format:

```
ChrID GeneID  IsoformID Sample1 Sample2 LogFold LogFoldSE PValue  QValue  Average
-      R2_67      -        0.31    0.5    1.61      0.73    0.84    0.96     5.67
-      R1_14     R1_141    1.56    1.56    0        0.02    0.08    0.06     3.83
```

C.3 VarVariant Format

Anaquin supports the common .VCF format, however, other file formats will need to be converted to either .VCF or a simple text format for use with Anaquin.

Files not in the VCF format will need to be converted to the Anaquin VarVariant Format. The Anaquin format is a simple tab-delimited text file that Anaquin can parse.

ChrID	Chromosome name (eg: chr1)
Position	Position of the variant
Ref	DNA sequence for the reference allele
Alt	DNA sequence for the alternative allele
ReadR	Read coverage for the reference allele
ReadV	Read coverage for the alternative allele
Depth	Normalized counts of all samples
QualR	Quality score for the reference allele
QualV	Quality score for the alternative allele
PValue	P-value probability

Example lines from the format:

```
ChrID   Position  Ref  Allele  ReadsR  ReadsV  QualR   QualV   PValue
chr1    373698     T     A       10      1       29      15      0.5
chr2    373704     G     A       227     1       26      16      0.5
chr3    373714     C     A       642     3       27      19      -
chr4    373718     G     A       741     2       25      28      -
```

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.

F-statistic & P-value
These statistics indicates the significance of the model. Under the null hypothesis, the model is not better than a random experiment. A high F-statistic indicates strong evidence against the null hypothesis.

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

SSM & SSE & SST
Sum of squares for the model, residuals and total variation.

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.

Wikipedia (en.wikipedia.org/wiki/Segmented_regression) has the statistical details.

Appendix E - Visualisation with IGV

Users are encouraged to visualize alignments and synthetic features (eg: 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 a copy of the *in silico* chromosome from either our website (www.sequin.xyz/downloads `CRN007_v001.fa`) or directly using with the following command:

```
$ wget s3.amazonaws.com/sequins/chromosomes/CRN007_v001.fa
```

2 | Start a new IGV session and select the `Genomes` menu, select `Load Genome From File.` Load the chromosome file (`CRN007_v001.fa`) from enclosing folder, and then click `Open`.

E.2 Visualize annotations

1 | Download a copy of the *in silico* transcriptome GTF annotation file from our website using the identifier `ARN020` or directly using with the following command:

```
$ wget s3.amazonaws.com/sequins/annotations/ARN020_v001.gtf
```

2 | In the IGV session, select the `File` menu, select `Load From File` and then `Open` to load the annotation file (`ARN020_v001.gtf`) from enclosing folder.

E.3 Visualize alignments

1 | Users can supply their own generated alignment file in BAM format as described in **Section 5.4.1,** or download a sample TopHat2 alignment file using the following command:

```
$ wget s3.amazonaws.com/sequins/igv/accepted_hits.bam
```

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 enclosing 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`).

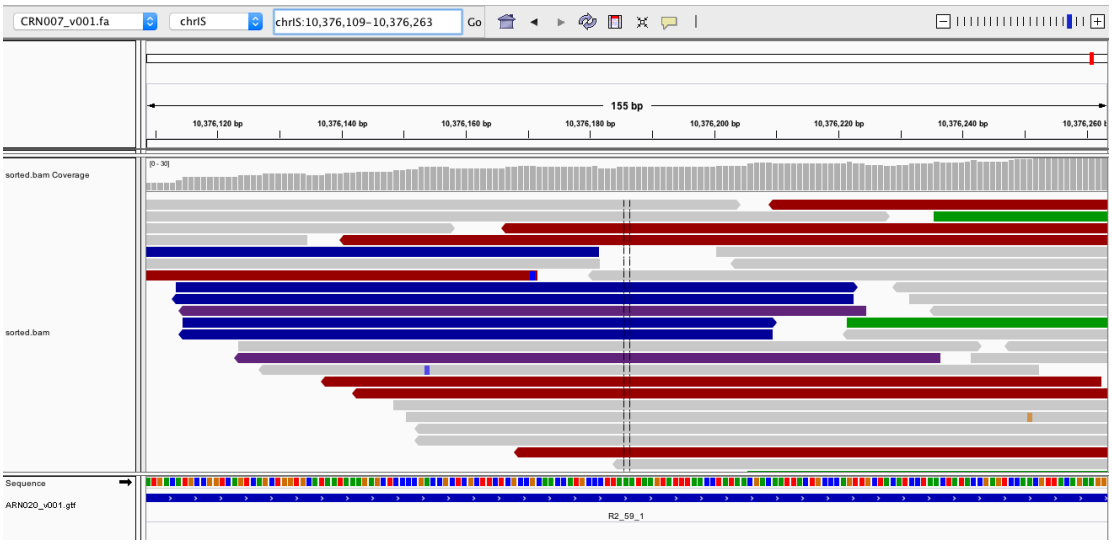

Figure E.4: IGV screenshot showing alignment reads in the middle panel and annotation file in the bottom panel. The region is locus `10376109` to `10376263` on chromosome `chrIS`.