RNA eXpress

User Guide 1.4

Sam Forster, Alex Finkel, Jodee Gould and Paul Hertzog

Last Updated: 14/02/2013

Intended Audience

This guide is designed primarily for biologists who wish to analyze transcriptome based next generation sequencing data using existing algorithms. It describes usage of both the graphical user interface and command line interface, including detailing the parameters, both required and optional, to configure runs. Those interested in developing algorithms for implementation within the RNA eXpress framework should refer to the Developer Guide available on the RNA eXpress website.

Table of Contents

Introduction

RNA eXpress allows you to easily identify novel genomic features that are expressed in whole transcriptome next generation sequencing datasets.

The features that can be detected using RNA eXpress include:

- Exons
- Transcripts
- Introns
- micro-RNA
- Long non-coding RNA
- Untranslated regions
- Transcription start sites

Bioinformaticians can also use the tools provided in RNA eXpress to create their own algorithms to target any kind of feature. Anyone interested in doing this is encouraged to read the Developer Guide.

You can run RNA eXpress in two ways:

- Through a Graphical User Interface (GUI)
- On the command line

This guide will focus on using the GUI to configure and run RNA eXpress. It will explain how to install and run the program, and the meaning of the parameters. However, we encourage the adventurous to go straight to http://www.rnaexpress.org/downloads.php and follow the instructions. The default settings should be good enough for most purposes – all you need to do is double-click rnaexpress.bat (rnaexpress.sh on Mac or Linux) to run the program, then tell it the name of an output folder, load up your BAM file and click 'Run'.

Acquisition and Installation

Program Acquisition

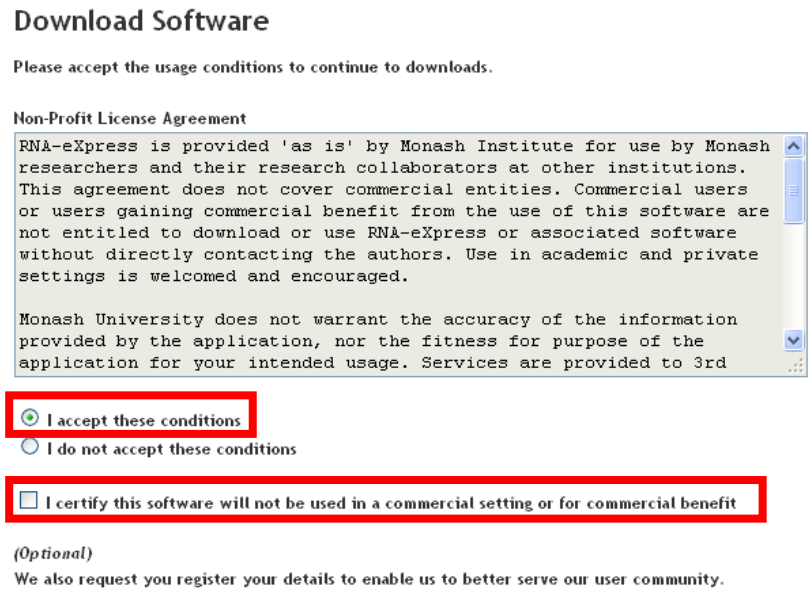
To acquire the program, source code, test data and/or documentation please go to the RNA eXpress website:

http://www.rnaexpress.org/downloads.php

The RNA eXpress program is distributed as a ZIP file containing the program file, the licence information and this user manual. You can run the program from any directory on your computer, but you must unzip the directory first (see "Installation" for more information, or Google "how to unzip directory [your operating system]").

Supported OS: Versions are available for all the major operating systems from the Downloads page on the website. In order to run the program, you will need to have Java installed. Java installation instructions are included in the install section of this manual.

License: Prior to gaining access to the RNA eXpress program you must accept the terms of use. This includes certifying the software will not be used in a commercial setting. If you wish to use this software in such a setting please contact us directly for permission. We also request you provide your contact details so we may contact you to update you on the program however this is optional. Upon acceptance of these terms you will receive access to the download page. After downloading you may install the program on multiple computers however we request you do not re-distribute the RNA eXpress software without the inclusion of the License agreement.



Installing the Program

The RNA eXpress program is available as a pre-compiled .jar that is directly executable on all major operating systems. The Mac version is distributed as an .app package for ease of use. Under either condition no specific installation is required for RNA eXpress however the Java runtime environment is required. You may also wish to create a shortcut on your desktop or start menu to enable simple program starting.

Pre-requisite Install

Prior to installing RNA eXpress you must have the Java runtime environment (JRE) installed. In most cases this will already be installed on your computer. The java version can be determined by visiting the Java website version test at:

http://www.java.com/en/download/installed.jsp?detect=jre&try=1

Alternatively, if you are comfortable with the command line you can type the following into your terminal or command window.

Java –version

If Java is installed and configured correctly you will receive a java version message such as:

java version "1.6.0_23"
Java™ SE Runtime Environment (build 1.6.0_23-b05)
Java HotSpot™ Client VM (build 19.0-b09, mixed mode, sharing)

A command not found error indicates Java is not installed on your system or the path is not correctly configured. Prior to installing RNA eXpress you should download the Java Runtime Environment from the Java website and follow the install instructions

http://www.java.com/en/

Windows

The windows distribution is available in .zip or .tar.gz archive format. You will require a compression utility such as Winzip or 7Zip to access these files. We recommend extracting to C:\Program Files\RNAeXpress. Extracting the archive will result in the RNAeXpress folder containing the following files:

rnaexpress-windows-1.2.jar
rnaexpress.bat
rnaexpress-userguide.pdf
license folder

RNA eXpress can be run by double clicking the jar file or running the .bat file. The .bat file will pass extra parameters to the JVM to increase the memory allocated. It may also be beneficial to create a desktop link to enable each access to the program for all users.

Desktop Shortcut Icon

1. First browse to the directory that you saved RNA eXpress in. Then right-click the RNA eXpress.bat file and click “Create shortcut”.

2. Now give it a name such as “RNA eXpress” and move it to your Desktop.

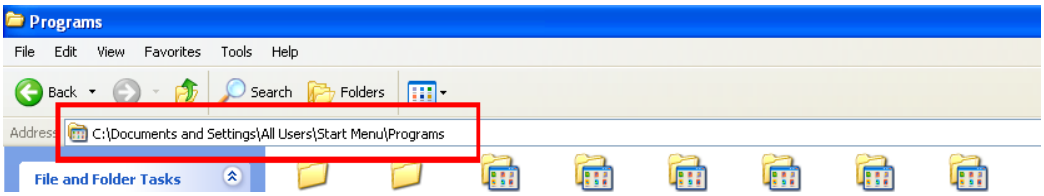
Start Menu Icon

1. To create a start menu icon open Windows Explorer or My Computer.

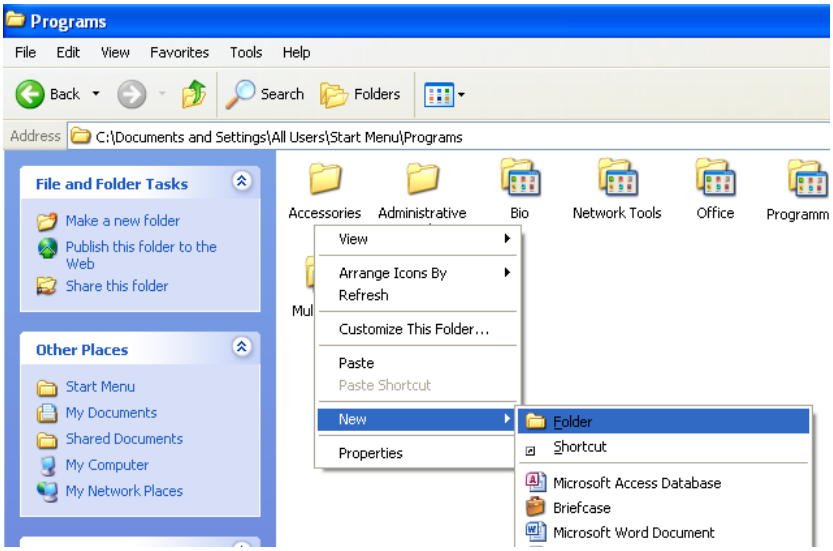
2. Navigate to the Program Start Menu Folder

   For Windows XP add start menu shortcuts to:
   C:\Documents and Settings\All Users\Start Menu\Programs\

   For Windows 7 add start menu shortcuts to:
   C:\ProgramData\Microsoft\Windows\Start Menu\Programs\



3. Right click and select “New Folder”



4. Name this new folder RNAeXpress This is the name that will be displayed in your start menu.

5. In this folder create a shortcut to the rnaexpress.bat file located in the install location as described previously.

Linux

The most appropriate installation location under Linux will vary depending on the distribution you are using. Consult your system administrator to determine where RNA eXpress should be installed.

Once you have determined your installation location unzip to this location. For example using gunzip.

```
#  gunzip –c </your/install/location/>
```

If the location selected is not in your path it can be added using

```
#  PATH=$PATH:/your/install/location
#  export PATH
```

This can be added to /etc/rc.local or equivalent to ensure it runs on every startup.

Alternatively you can create a symbolic link from somewhere that is in your path. For example:

```
#  ln –fs /your/install/location/rnaexpress/rnaexpress.sh /usr/bin/rnaexpress
```

Mac

Please Note: The Mac Version of RNA-eXpress is offered as Beta due to specificities in Apple Java versions.
Apple supplies a custom version of Java. As such updates must be performed through the Apple software update feature. RNAeXpress requires Java version 1.6 or above please ensure this is installed.

The Mac distributable is supplied as a zip or tar.gz archive file. Extracting the archive will result in the RNAeXpress folder containing the following files:

rnaexpress-mac-1.0.app
rnaexpress-userguide.pdf
license folder

Double clicking the rnaexpress app will run the program.

If you encounter problems with the Mac version please provide feedback through the Forum or Contact us directly through the Contact Us (http://rnaexpress.org/contactus.php) section of the website.

Using RNA eXpress

Control of the RNA eXpress workflow occurs through specification of a configuration file. This may be done manually using a text editor or using the provided graphical interface. Once the configuration file is completed the run can be commenced. The flexible nature of the workflow allows you to start and complete analysis runs at any of the stages.

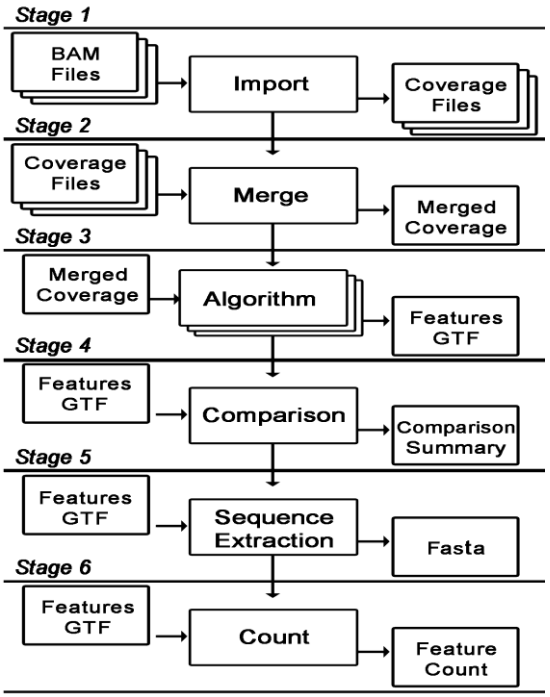
If in doubt, use the default settings for everything except the location of your BAM files and output folder.

Workflow

RNA eXpress may be executed between any of six stages. The user interface will assist you in working out which stages you wish to run. However, they are also described in detail here.

Stage 1: Import

Stage 1 involves import of one or multiple BAM files. These files should be sorted but indexes will be generated automatically if they are not present. All BAM files within the selected folder will be processed as a single experimental grouping. The output from this process will be WIG and BED coverage files for each chromosome and strand of each sample. Quality filtering, scaling and low value cut-offs are also applied during this import stage.



Stage 2: Merge

The Merge stage additively combines BED and WIG files for each chromosome and strand across samples. This results in a single merged BED and WIG files for each chromosome and strand.

Stage 3: Algorithm

The algorithm stage will vary depending on the algorithm selected. All algorithms require as a minimum the merged BED and WIG files for processing and will produce a GTF file representing the detected features. To improve performance the RNA eXpress framework includes threading support for this step. A number of algorithms also require additional files and parameters. Please see the algorithm description section of this manual or the website for more information.

Stage 4: Comparison

To assess the overlap between your annotated data and available online annotations such as NCBI or Ensembl RNA eXpress provides the ability to compare the GTF output from the algorithm stage with these public GTF files. A summary comparison file and an annotated version of your GTF file is produced as a result of this comparison. When using the graphical user interface a histogram illustrating feature size distribution and pie charts showing comparison statistics will also be displayed.

Stage 5: Sequence Extraction

Where features identified are of interest for downstream analysis RNA-eXpress can automatically extract either the sequence for the exact region or a defined surrounding region. Sequence extraction can be configured to behave differently depending on the type of feature identified and will produce separate FASTA files for each sequence.

Stage 6: Count

The final stage of analysis provided by RNA eXpress is the ability to count reads that map to locations within the GTF file. This step uses the original BAM files provided and produces either single file or multiple tab delimited count files suitable for use in downstream differential expression analysis.

Configuration File

The configuration file is a text file with comments denoted with an # character. By convention a .ini extension is applied to complete configuration files while a .tp extension is applied to partially complete files. When using extensions other than .ini you will receive a runtime warning message however it will not affect the run in any way.

The following parameters may be included in a configuration file. The Key used in the configuration file, a description and the default values are provided in the table below. The key is used as follows:

```
STARTPOINT=1
```

KEY	DESCRIPTION	VALUE(S)
STARTPOINT	The position in the workflow to start program execution: 1. Start With BAM File 2. Start By Merging Wig Files 3. Start With Peak Counting on Merged Wigs 4. Start With Gtf Comparison 5. Start With Counting	1 – 5
ENDPOINT	The position in the workflow to end program execution. 1. After Generating Wig Files From Bams 2. After Merging Wig Files, resulting in merged wig file 3. After Peak Counting, resulting in a single gtf 4. After Gtf Comparison, results in comparison statistic and merged gtf file 5. After Counting	1 – 5
OUTPUTFOLDER	The output folder where all output files will be located	/output/
PROCESSORS	Number of processors to use during execution. This will determine how many threads are executed at the threaded stages.	1

BAMFOLDER	The folder where BAM files and BAM indexes are located	/bamfiles/
WIGOUTPUTNAME	The folder where WIG Files are located or should be produced. This should be under the output folder	/wig
JUNCTIONSOUTPUTNAME	The folder where JUNCTION Files are located or should be produced. This should be under the output folder	
WIGCUTOFF	The cutoff value for total number of required reads across samples to be included as input to algorithm (Merged Wig File)	1
WIGQUALCUTOFF	The minimum quality value of reads to be included in the WIG file counts	5
WIGQUALSCALE	A “true” or “false” instruction as to whether or not to scale coverage in the wig file according to mapping quality.	True
COMPRESSWIGS	Specify whether to produce standard WIG format or a compressed memory efficient WIGC. RNA-eXpress will treat these files independently	true
IGNORESTRAND	Specify whether to ignore the strand information. This option is relevant if the RNA-Seq protocol is not stranded and results in strand information being recorded as 0. Analysis of non-stranded data without selecting false will result in features being incorrectly distributed across both strands.	false
MERGEDWIGOUTPUTNAME	The folder where merged WIG files will be stored. This will be located under the OUTPUTFOLDER directory.	mergedWIG
MERGEDJUNCTIONOUTPUTNAME	The folder where merged JUNCTION files will be stored. This will be located under the OUTPUTFOLDER directory.	mergedJunction
ALGORITHM	The ALGORITHM to be applied for example lncRNA	See list of algorithms
ALGORITHMPARAMETERS	Multiple ALGORITHMPARAMETER values can be provided to match the parameters required for the algorithm selected. The order of these parameters should also match the algorithm.	

GTFOUTPUTNAME	The name of the GTF produced through this analysis	Output.gtf
FOREIGNGTF	The foreign GTF, typically from EnsEMBL or RefSeq that represents all known features in the genome of interest.	Foreign.gtf
ANNOTATEBYFOREIGN	Specifies how to compare the foreign GTF to the GTF produced during program execution.	ALL, OVERLAPPING, IDENTICAL, NONE
FOREIGNANNOTATIONWINDOW	Window around endpoints to consider the same feature	0
SEQUENCEINPUTGTFTYPE	Comparison output GTF to use for sequence extraction	ALL, NOVEL, OVERLAPPING
REFERENCESEQUENCEFILE	Location of chromosome reference file	\sequences\chromosomes.fa
SEQUENCEOUTPUTNAME	Sequence output folder	sequences
SEQUENCEOFFSET	Feature type, upstream and downstream length relative to the feature Start and End positions where sequence will be extracted from. Using ALL or * as the feature type will produce sequence for all features in the GTF. Where the strand is unknown both positive and negative strand features will be produced.	ALL 0 0 TSS 1500 500
COUNTINPUTGTFTYPE	Comparison output GTF to use for feature counting	ALL, NOVEL, OVERLAPPING
COUNTOUTPUTFILEFORMAT	The output format of the counts either as a SINGLE output file for all bam file or in MULTIPLE file format, one file representing each BAM file.	SINGLE, MULTI
COUNTOUTPUTFORMAT	Specify whether to count unique transcripts or unique exons	TRANSCRIPT, EXON
COUNTMETHOD	Specify whether to produce raw value counts or RPKM counts	RAW, RPKM
COUNTOUTPUTNAME	Count output folder	counts
DEBUGPRIORITY	Specify the level at which error messages will be written to the Run Output Screen.	1
DEBUGLOGFILE	Specify the debug log file.	/output/run.log

Input Files

BAM Files

The standard input format for RNA-eXpress is the BAM file. RNA-eXpress also requires a BAM index file however where a BAM index file is not provided one will be generated automatically.

Strand Specificity

When processing BAM files, reads will by default be assumed to originate from strand specific sequencing. Where this is not the case the IGNORESTRAND parameter should be set to 'true'. Failure to do this will result in features being predicted on both strands. Where this parameter is set to 'true' downstream files will record the strand as 0 except for the Sequence extraction step which will produce both negative and positive strand sequence.

Paired End Reads

From version 1.4 RNA-eXpress considers both pairs in paired end data for downstream analysis. In should be noted, where data contains paired end reads and the IGNORESTRAND parameter is set to false the second member of a read pair is corrected to match the strand of the first pair. Reads that lack a mapped pair or where both pairs exist on the same strand will not be altered. These changes do not modify the BAM file they just impact the downstream files produced during RNA-eXpress analysis.

Output Files

Output File Types

The output file format will vary depending on the step or combination of steps that have been run. With the exception of custom algorithms all files will conform to one of the following file formats

WIG Files

Wig files adhere to the standard WIG file format with the optional UCSC WIG file track header for display in certain genome browsers such as UCSC. All Wig files produced are in the "variable step" format. More information on the WIG file format can be found on the ENSEMBL website (http://www.ensembl.org/info/website/upload/wig.html). There is also an optional WIGC format which can be used to perform lossless compression of WIG files. This significantly improves the disk read/write performance and is the recommended approach for RNA-eXpress analysis in the normal case where the Wig files will not be used for downstream analysis. To convert between WIGC and WIG format an additional program WigConverter.jar is available from the Documentation page (http://rnaexpress.org/documentation.php).

GTF2 Files

The GTF2 file format is used for input and output at various stages of the workflow. Depending on the algorithm the feature types include exons, peaks, TSS and UTRs. GTF files contain the mandatory gene_id and transcript_id which are used to group related features into genes or transcripts. More information on the GTF format can be found on the UCSC website (http://genome.ucsc.edu/FAQ/FAQformat.html#format4). Other information may also be added to the attributes column after the gene_id and transcript_id attributes. These are included in the following table:

Attribute	Description	Values
exon_number	The exon count numbered from the start of transcript independent of strand.	Numeric. Eg. exon_number "1"
match_type	The type of match that was detected in the comparison step of the analysis	none, close, etc. for full list of matches see appendix (page 30)
match_score	Score representing the quality of the match detected. This value can be useful for ordering features by match quality	Numeric Eg. match_score "2"
source_file	The source file from which this feature originated.	source_file "OutputGTF.gtf"
line_number	The line number in the source file from which this feature originated	Numeric Line_number "1"
best_match	Best matching feature in the format of transcript_id-start position-end position	Best_match "1477-92392742-92392812"

FASTA Files

Where sequences are required FASTA files should be provided in the standard FASTA format as described on the EMBL-EBI site (http://www.ebi.ac.uk/help/formats.html#fasta). Where chromosomes are being imported the chromosome name should be stated first. Any information provided after a space or a pipe '|' symbol will be ignored. Where the chromosome format is numeric, X, Y or contains the full word chromosome the "chr" abbreviation will be used for all outputs. This feature improves compatibility with downstream tools and enables the use of Ensembl chromosomes and sequence data in the analysis. This process will not occur where sequences are provided as contig names. The sequence provided will be assumed to represent the entire chromosome sequence. Where a '.' exists in the chromosome name, the WIG and JUNCTION file names will contain a '_DOT_' rather than a '.'. This characteristic is scheduled for removal in coming releases.

For example:

>1

>chromosome1 Other Information

>1|Other information

>1 Other information

All match features on "chr1".

>NT_0000001

Will match features on contig NT_0000001.

Count Files

The count files are produced as a standard tab delimited format text file format. This format will either include a **single** file for all sample counts or **multiple** files, one for each sample depending on the setting selected for COUNTOUTPUTFORMAT. Counting will be performed either by unique transcript or unique exons depending on settings selected.

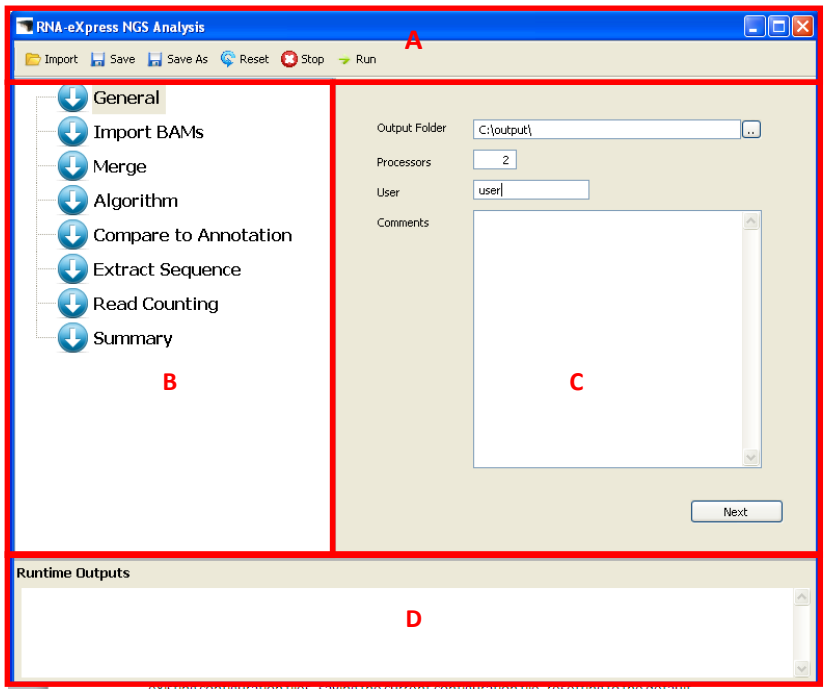
Output File Description

The files produced at every stage of the analysis pipeline are described in the table below.

Completion Stage	Output File Type	Description
Import BAMs	Wig Files	Wig files for every chromosome and every strand from each sample BAM file provided.
	Junction Files	Junction files for every chromosome and every strand from each sample BAM file provided.
Merge	Wig Files	Merged Wig files for every chromosome and every strand across the samples provided.
	Junction Files	Merged Junction files for every chromosome and every strand across the samples provided.
Algorithm	GTF File	The GTF file containing all the features identified by the algorithm
Compare to Annotation	GTF File: All Features	The GTF file containing all the features identified by the algorithm annotated through the Foreign GTF. Depending on the selection options. This GTF file will be produced in the GTF folder
	GTF File: Novel	The GTF file containing those features that were not represented in the Foreign GTF file provided. This GTF file will be produced in the GTF folder
	GTF File: Overlapping	The GTF file containing those features that overlap but differ significantly from those features annotated within the Foreign GTF provided. This GTF file will be produced in the GTF folder
Sequence Extraction	FASTA File	Fasta file containing the relevant regions of sequence corresponding to the features identified. For each feature type defined in the feature extraction process a complete and 'no-duplicates' version of the output is included as separate Fasta files in the sequence output folder. Where a strand is not specified these files will contain BOTH the positive and negative strand sequences.
Counting	Counts File	Single or Multiple count files containing read counts for each sample corresponding to the features identified.

Graphical User Interface

The graphical user interface is composed of four sections that enable management of configuration file generation and run monitoring. These sections are described below.



A. Toolbar: The toolbar provides all the basic workflow management tools including importing existing configuration files, saving the current configuration file, resetting to the default configuration file and running the analysis based on the currently active configuration file.

B. Workflow Overview Panel: The workflow overview panel shows the full workflow currently being configured. The “General” and “Summary” will always be present. Modification to workflow Start position, end position for “Bam Imports” through to “Read Counting” can be performed through right clicking items in this panel. Additionally resetting the entire pipeline equivalent to clicking reset in the toolbar is possible through right clicking in this panel.

C. Current Panel: The current panel enables the entering of parameters required to perform the run. Where possible these parameters will be pre-filled with default values. Basic validation functionality is provided however validation of file locations and associated system specific parameters does not occur to enable analysis and program execution to be performed on different computers.

D. Runtime Outputs Panel: Where run executions are performed using the GUI the command line output is captured and displayed in the Runtime Output Panel. This enables identification of errors and monitoring of run performance.

Toolbar Options

Control of the run and basic parameters is achieved through the toolbar. There are five run controls that are described below.

Import

The import button allows you to import previously saved configuration files into the RNA eXpress user interface. This is equivalent to passing the configuration file name as a command line parameter.

Save

The save button will save the current configuration settings. If the file was imported the changes will automatically be saved over this file. Where no file is associated you will be prompted to select a path to save the file to.

Save As

The save as button will save the current configuration settings to a new location. You will be prompted to select a path to save the file.

Reset

The reset button will return the RNA-eXpress GUI to the default layout. All existing changes will be lost.

Stop

The stop button allows you to stop a currently executing analysis. Clicking the stop button once will result in a termination at the next workflow break point. This will allow you to continue your workflow from this point or any previously completed point. Clicking the stop button again, once a breakpoint stop is already underway will result in instantaneous termination of the process. This is not recommended except under exceptional circumstances. The files resulting from the currently running analysis will be left in a partial complete state and thus not usable for downstream analysis.

Run

The run button will initiate the program execution. This requires a complete, saved configuration file. If a configuration file does not exist you will be prompted to save one. Alternatively existing configuration files, where imported, will be automatically updated with the current changes.

Getting Started

To test program is successfully installed we recommend you work through the following tutorial and compare the output to that provided in the getting started data. This is a small dataset that should take no longer than a few minutes to run on most modern computers. Run times in excess of 10 minutes suggest your computer is not sufficiently provisioned or you need to modify the Java virtual machine to increase available memory to enable running this software efficiently. All new versions of RNA-eXpress are tested internally with this workflow to ensure consistent results are obtained prior to release.

This workflow will assume we wish to perform all the steps from importing BAMs through to read counting. We require the following files to perform this analysis:

File Name	Description
Test.BAM	Test BAM file containing mapped reads. This is a small extract from Thiagarajan et al. 2011 (SRA026710)
Test.BAI	Test BAM index files (This will be created if they are not present)
Foreign.gtf	Relevant section of Ensembl GTF to enable feature comparison
Reference.fa	Sequence file for sequence extraction
Config.ini	RNA eXpress Configuration File

These file can be obtained from the RNA eXpress documentation page in the TestData.zip file (<40MB). (http://www.rnaexpress.org/documentation.php)

Command Line Workflow

To use the provided configuration file exists command line running of the workflow is relatively straight forward.

1. The configuration file assumes the Bam Files are located in C:/bamfiles/ and the output location is C:/output/. This is almost certainly not appropriate for your system configuration so first we must modify the configuration file.
    a. Modify the BAMFOLDER parameter to point to the relevant directory where Test1.bam is located.
    b. Modify the OUTPUTFOLDER parameter to point to the location you want the output to be located in.
    c. Modify the FOREIGNGTF parameter to point to the location where Foreign.gtf is located
2. Run the following command to begin execution of RNA eXpress (under linux run rnaexpress.sh)

```
#  rnaexpress.bat -n data\config.ini
```

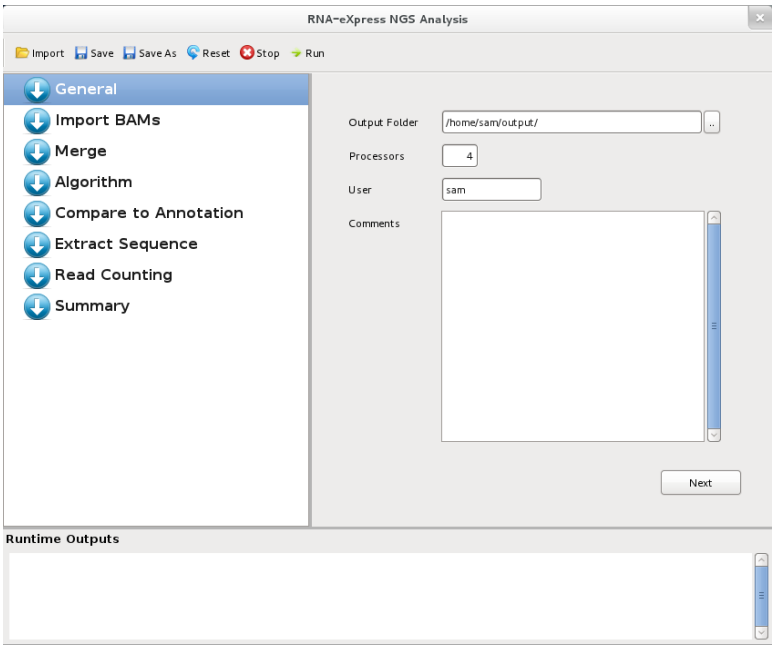
Alternatively if you wish to modify java specific parameters you can run rnaexpress.jar directly with:

```
#  java -jar -<additional parameters eg. -Xmx> rnaexpress.jar -n data/config.ini
```

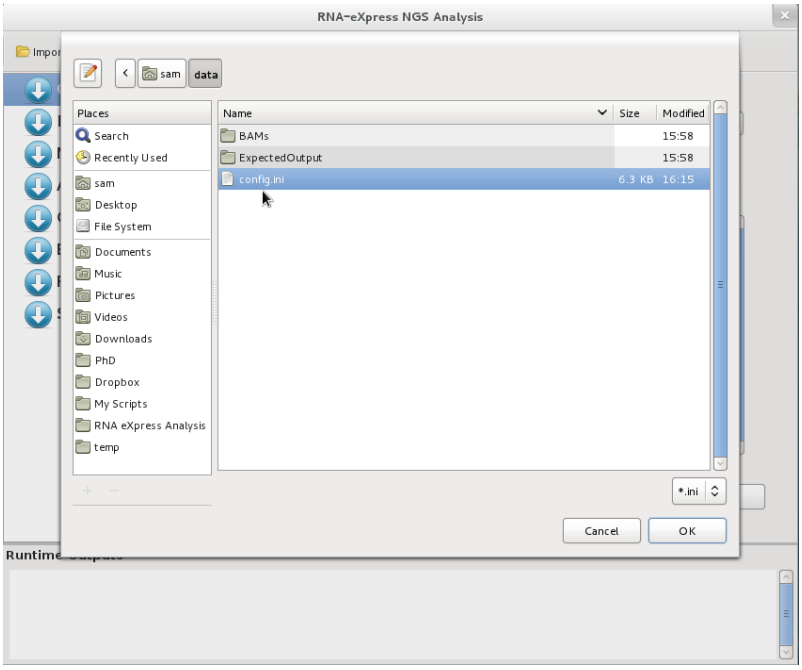
3. At the completion of execution the output directory should be identical to the example output directory provided in the getting started data.

GUI Workflow

1. To start RNA eXpress, select the program icon. If installation was successful you should see a loading screen followed by the workbench view shown below. This provides the fields for creation of the configuration file.



2. To import the provided configuration file select “Import” from the toolbar and select the config.ini file. This will import as many configuration parameters as possible from the provided configuration file. Where parameters cannot be identified they will revert to their default value.



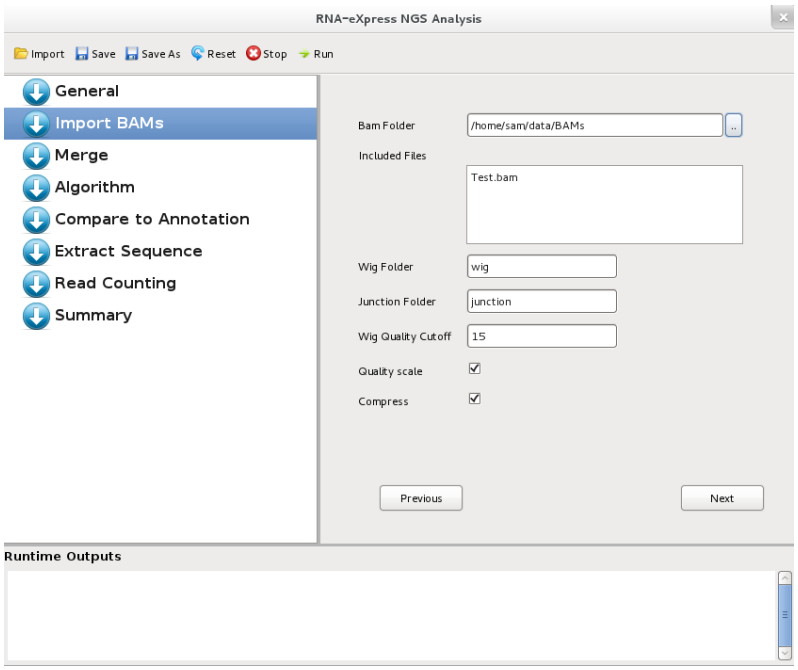
3. To view imported values click on the “General” item displays the general settings parameters as shown on the previous page. You will need to set an empty folder as the output folder - name it something memorable. Other parameters may also be modified if required.

    a. The output folder can be selected by clicking the browse button (“..”) or typed directly into the text field provided.

    b. The number of processors will automatically be set to 1 from the configuration file. To ensure consistency of randomly generated identifiers and allow the “diff” comparison to be performed on the output file it is advised to leave this setting. Under normal circumstances, this can be increased to the number of processors available to facilitate parallel processing.

    c. The user name and comments fields are not required for program execution however they are added as comments, along with the creation date to enable tracking of configuration files.

4. Clicking “Next” on the “General” panel or “Import BAMs” in the overview panel will display the “Import BAMs” parameters.



    a. Selecting the Bam Folder, as with the Output Folder previously will display a selection panel. The BAM files located in the selected folder will be displayed in the Included Files box. Test.bam should be listed here if the correct bam folder has been selected.

    b. Wig Folder and Junction Folder will be created in the output folder to store the WIG and Junction BED files respectively these can be left as default values.

    c. Wig Quality Cutoff is described in the parameter section. To enable comparison of outputs in this test data these values should not be altered in this example.

    d. Enabling Wig Compression will use a custom format that stores the same information as a WIG much more efficiently to use less disk space. If you wish to view your wigs in IGV, disable compression.

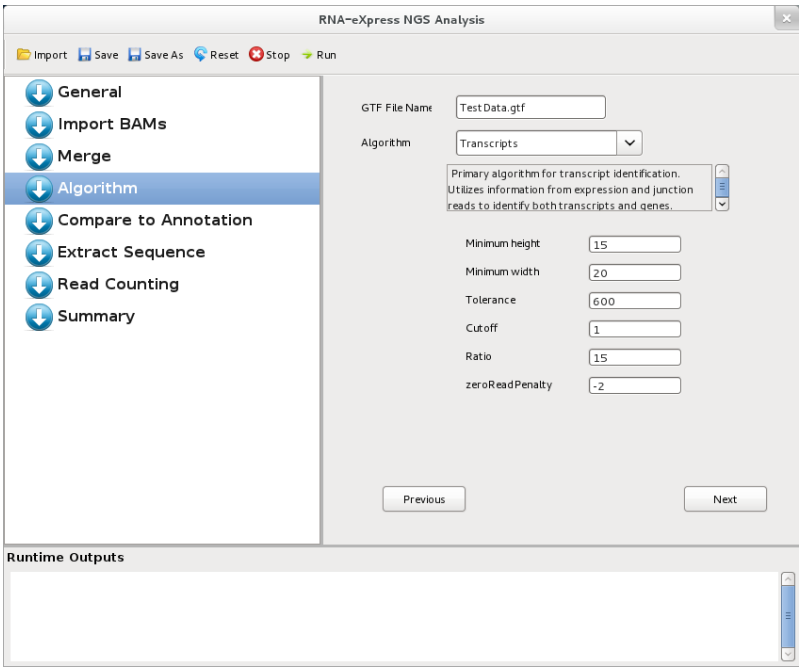
    A command-line program for Wig to Compressed Wig conversion called ‘WigConverter’ is available on the RNA eXpress website (http://www.rnaexpress.org/documentation.php)

5. Clicking “Next” on the “Import BAMs” panel or “Merge” in the overview panel will display the “Merge” parameters.

    a. Merged Wig and Merged Junction will be created in the output folder to store the result of merging with the WIG and JUNCTION files. These folder names can be left as default.

    b. The WigCutoff value is described in the parameter section. This value should be left at 0 for this comparison

6. Clicking “Next” on the “Merge” panel or “Algorithm” in the overview panel will display the “Algorithm” parameters.

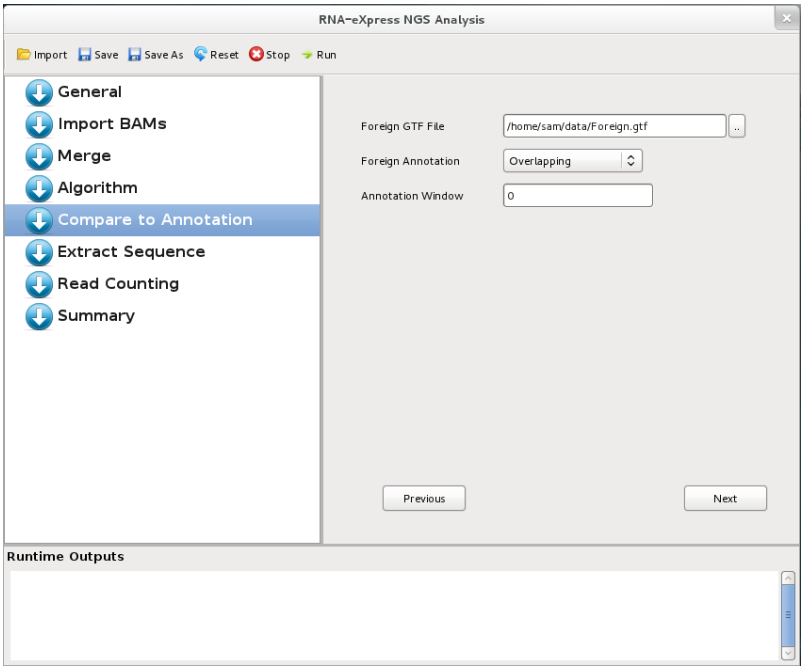


    a. The algorithm to be applied can be selected from the drop down box. All available algorithms in the current program will be displayed. After selection the appropriate algorithm parameters will be displayed. Please refer to the specific algorithm documentation to determine appropriate values. For the purpose of the example data ensure Transcripts is selected and the following parameters are set:

    Minimum Height: 15
    Minimum Width: 20
    Tolerance: 600
    Cutoff: 1
    Ratio: 15
    Zero Read Penalty: -2

    b. The GTF File name is the name of the GTF file that will be created through this analysis. This file will be created in the Output folder. The name should be left the same for comparison to the expected output.

7. Clicking “Next” on the “Algorithm Settings” panel or “Compare to Annotation” in the overview panel will display the “Compare to Annotation” parameters.



This stage will measure the precision and recall in retrieving features from this BAM file, assuming that the ‘foreign’ file constitutes a correct annotation. It also produces a GTF file that includes all of the features in the GTF file produced by RNA-eXpress, and either all or only the overlapping features from the foreign file. In addition separate GTF files are produced that contain only Novel and only Overlapping features respectively. Any of these three files can be selected as input to the Extract Sequence or Read Counting stages.

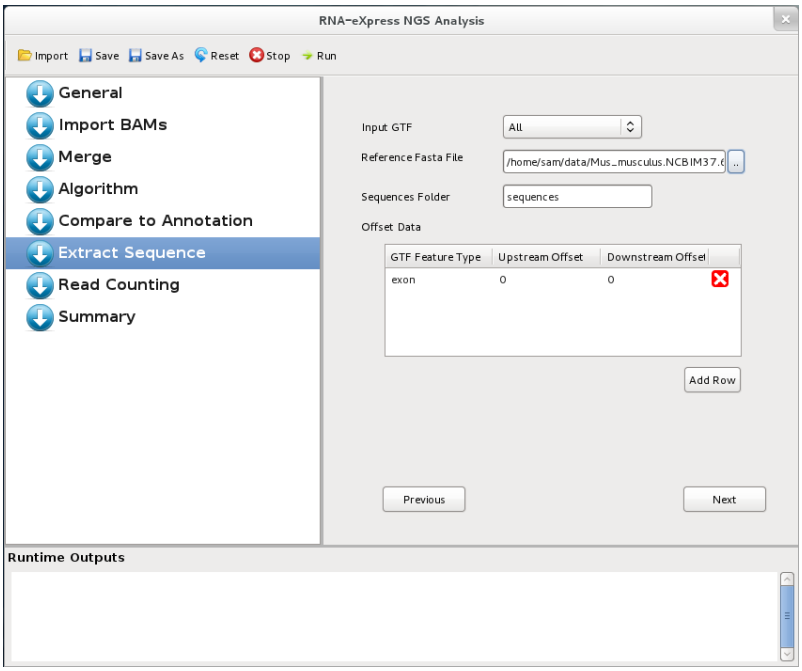
a. Features in the GTF file created by RNA-eXpress will be compared to features in the ‘foreign’ GTF file. If “All” is selected as the annotation method the entire contents of the ‘foreign’ file will be copied to the merged GTF in the output folder, otherwise only those ‘overlapping’ or ‘identical’ to features in RNA-eXpress’s GTF will be included.

b. For the purposes of this analysis we will include only those features “Overlapping” the identified features.

c. The annotation window should remain set at 0 for this analysis.

8. Clicking “Next” on the “Compare to Annotation” panel or “Extract Sequence” in the overview panel will display the “Extract Sequence” parameters.

Add a row for each feature type that you wish to extract by clicking the “Add Row” button under the table - or just double click in an empty row. Delete a row by clicking on the red X.



Bear in mind that only the named feature types will have their sequences extracted. Each Algorithm used by RNA-eXpress may produce multiple kinds of features - you can check the documentation for the Algorithm to see what it will produce. However, most produce ‘exon’.

You can additionally ask for an upstream and downstream offset from each feature

RNA-eXpress is somewhat flexible as to what the headers in the FASTA file can contain, please refer to information on the FASTA file in the file format section of this manual (page 16).

For the purposes of this analysis we will extract the sequence for ALL exon features from the included Reference.fa file into the sequences folder.

a. Select ALL from the InputGTF option to include all features

b. Select the Reference.fa reference for mouse chromosome 15 included in the test data as the Reference Fasta File

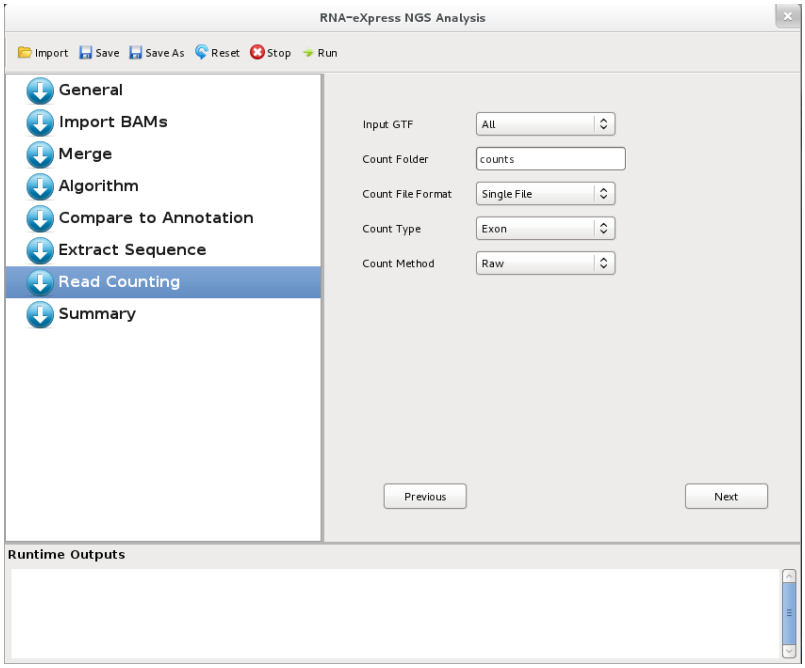
c. Ensure the Sequence Folder name is “sequences”

d. Double click in the Offset data table or click add row. Change the GTF feature type value to “exon”. Ensure the upstream offset and downstream offset values are set to 0. If a mistake is made the row can be deleted by clicking the red cross and a new row added as described above. There is no limit to the number of feature types for which sequence can be extracted.

9. Clicking “Next” on the “Extract Sequence” or “Read Counting” in the overview panel will display the “Read Counting” panel.

   This stage will count the number of reads in the BAM files at each feature in the GTF.

   For the purpose of this analysis we will perform counting for all transcripts with raw counts returned in a single file.



   a. Select the ALL option for Input GTF

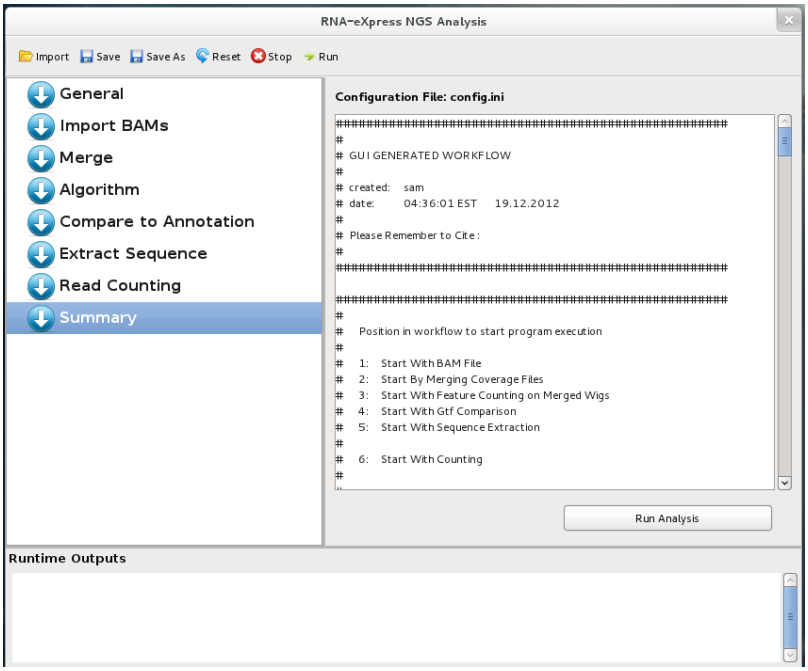
   b. Specify the name of the counts folder. For the purposes of this example the count folder should be name counts

   c. Select “Single File” as the Count File Format

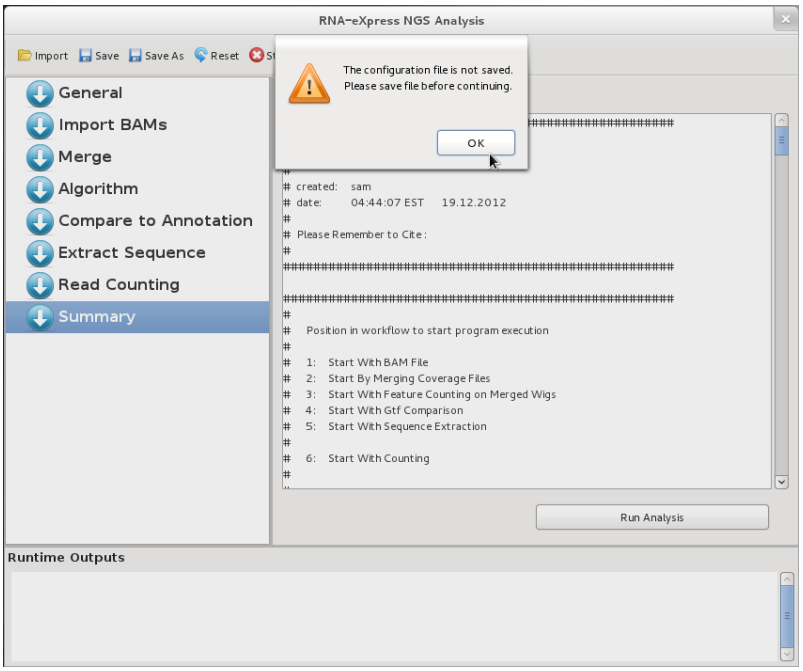
   d. Select “Transcript” as the Count Type

   e. Select Raw as the count method

10. Finally clicking “Next” on the “Read Counting” panel or “Summary” in the overview panel will display the “Summary” panel. Modifications made directly to the configuration file at this stage will only be stored after saving the configuration file. This can be done by clicking “Save” in the toolbar. Alternatively prior to running you will be prompted to save the configuration file. This is the option we will use.

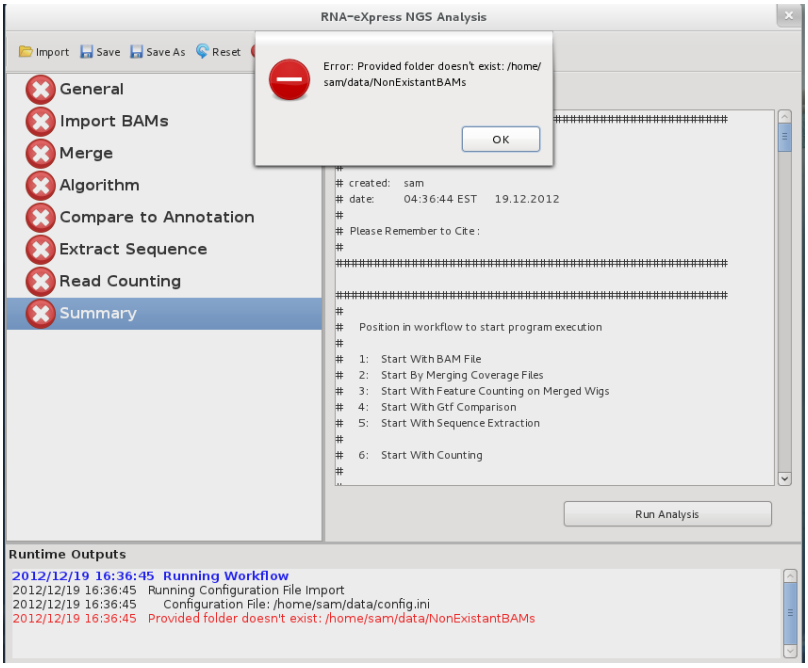


11. Click “Run Analysis” and you will be prompted to save.

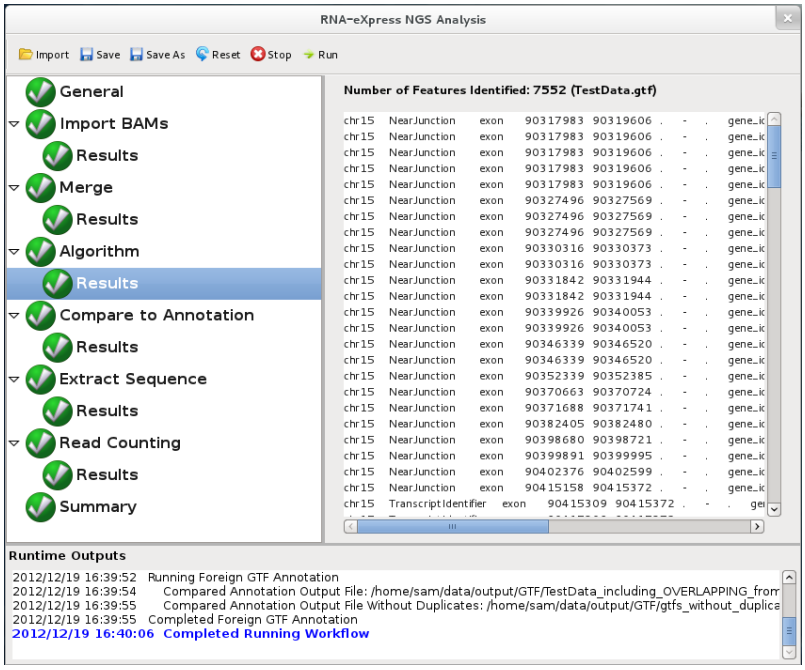


12. Save file as config.ini in your desired location.

13. Run will now start automatically. While the run is progressing you can modify the configuration settings however this will not affect the run in any way. In the vast majority of cases where an error is present in the configuration file a fast fail mechanism will ensure an error occurs within the first few seconds of execution. You will be notified of any error in a pop-up box. Detailed error messages can be found in the Runtime Outputs console and the log file.

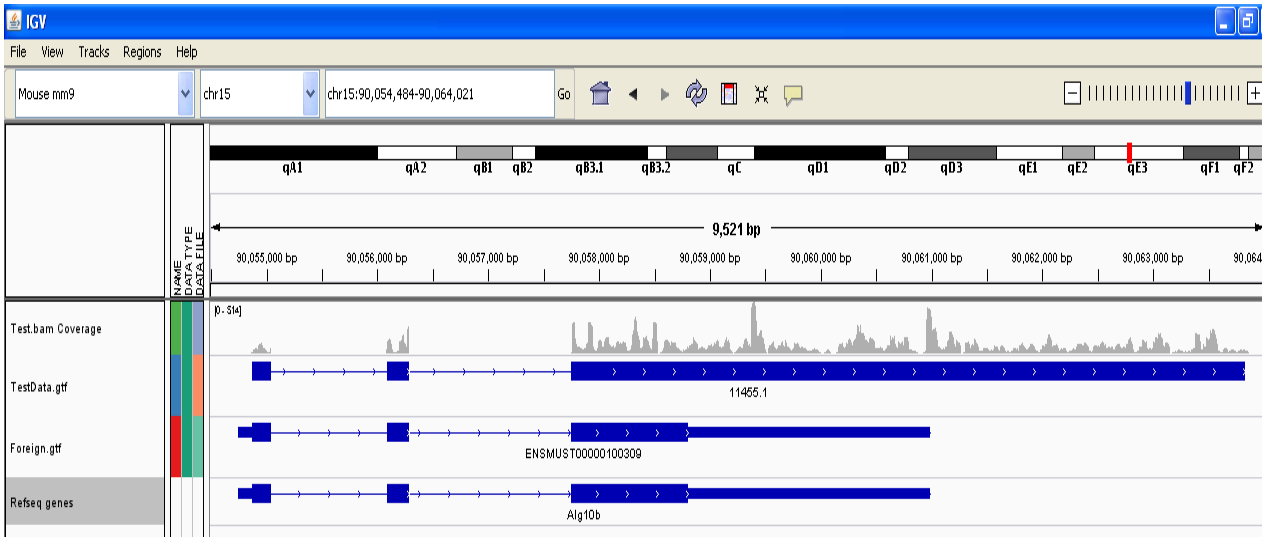


14. Running the test data and configuration should not result in any errors. During execution secondary menus will appear under each stage as it is completed successfully (indicated by a green tick). Clicking this results tab will allow you to view the first few lines of the output files to ensure no errors have occurred. If you notice an error the run can be stopped before commencing the next phase by pressing 'Stop'. If the run must be terminated immediately you may press the 'Stop' button twice. This action is not recommended except in extreme circumstances.

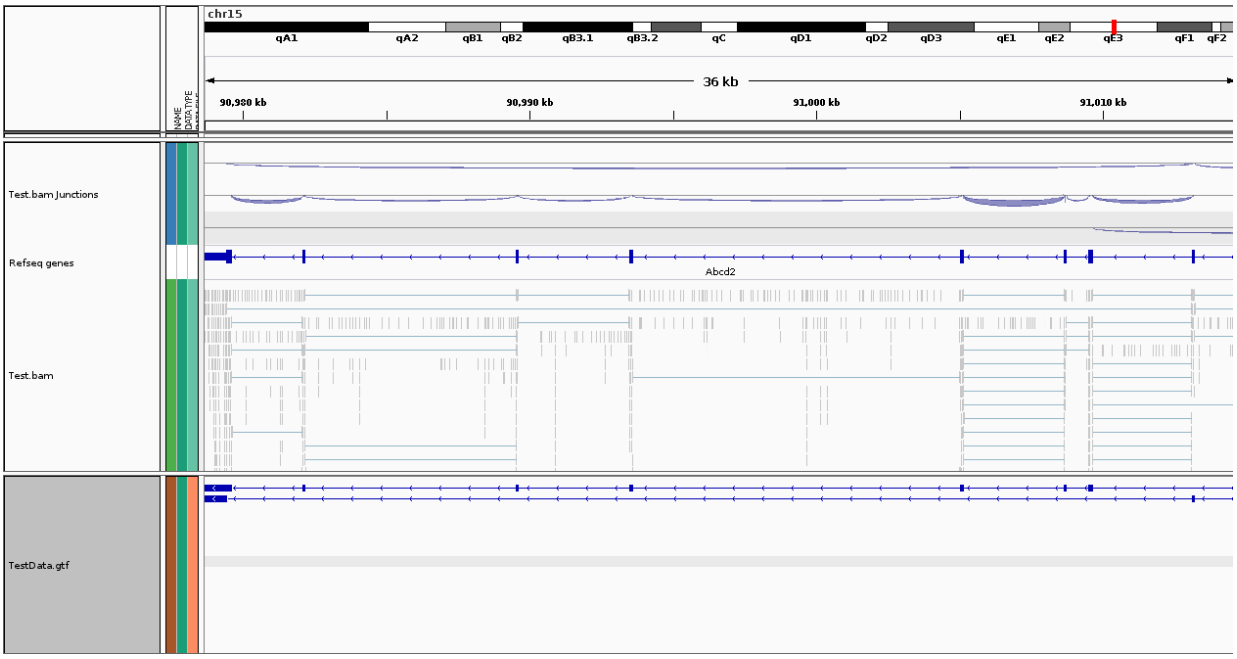


15. Some results sections also contain specific graphical outputs that are not produced during command line running. These images can be saved directly through the user interface by right clicking on the image and selecting Save As.

16. At the successful completion of execution the output directory should be identical to the example output directory provided in the getting started data. Comparing files with the diff command should only result in differences in the run log. Alternatively the Compare to Annotation function can be used to compare the Expected GTF with the GTF produced during this run. The results of the annotation may also be viewed using a genome browser such as IGV. For example Thiagarajan et al. 2011 identified Alg10b with an extended 3'UTR in this dataset. If we examine the dataset at this location we can see RNA eXpress has correctly annotated this region.



17. Equally other potentially novel splicing events are observed within the data. For example Novel splice variants are predicted in Abcd2. Navigate with IGV to position chr15:90,978,646-91,014,721 and two splice variants are clearly predicted



Appendix

Match Types

Key Value	Description	Visual
CLOSE	Both ends of the this feature are very close to the other feature	****** ******
LCS	The left end of this feature is very close to the other feature, but the right end is smaller (this feature is smaller than the other).	*** ******
LCB	The left end of this feature is very close to the other feature, but the right end is bigger (this feature is bigger than the other).	****** ***
RCS	The right end of this feature is very close to the other feature, but the left end is bigger (this feature is smaller than the other).	*** ******
RCB	The right end of this feature is very close to the other feature, but the left end is smaller (this feature is bigger than the other).	****** ***
LEFT_OVERLAP	The left side of this feature overlaps the other feature.	****** ******
RIGHT_OVERLAP	The right side of this feature overlaps the other feature.	****** ******
SMALLER	This feature is wholly contained within the other feature, but neither end is close to the other feature.	***** **********
BIGGER	This feature encompasses the other feature, but neither end is close.	********** *****
LEFT_CLOSE_NONE	The left end of this feature is close to the other feature's right end.	********* *******
RIGHT_CLOSE_NONE	The right end of this feature is close to the other feature's left end.	********* *******
NONE	These two features do not overlap in any way	**********

Algorithms

The following pages provide detailed descriptions of each algorithm. Information about each algorithm is also available on the algorithm page of the website.

http://rnaexpress.org/algorithm.php

Transcripts

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

This algorithm attempts to locate and identify genes and transcripts by using junction reads to show where exons begin and end in an expressed region, and which exons are joined into transcripts.

This information is delivered in the output gtf file by giving each feature a significant gene id, transcript id and exon number.

Additionally, highly expressed feature's with no junction reads will be added. If they occur in the region of a gene, they will be given a unique transcript id on that gene. Otherwise, they will be given a unique gene id.

Our approach is greedy – each potential transcript indicated by the junction reads will be extended as far as possible. For example, a ten-exon gene in which some junction reads indicate that exon three can join directly to exon five will result in two transcripts, one with 9 exons and one with 10. This does not indicate that both transcripts necessarily do extend this far, only that the potential for them to do so exists. Therefore, many individual feature's will be duplicated, with the only difference being their transcript id – however, the combination of exon number and transcript id is guaranteed to be unique.

Recommended Use Case

The Transcripts algorithm is our recommended algorithm for most purposes, including differential expression analysis, and identification of novel transcripts.

Requires

This algorithm has a dependency on the following algorithms:

- FeaturesWithJunctions
- TLA

Parameters

Name	Type	Description
Minimum Height	Positive Integer Must be greater than zero	The minimum expression level that, on its own, indicates the presence of a feature. We recommend using the lowest possible number of reads here for which you could still possibly be interested in the feature, e.g. 5 * number of BAM files
Minimum width	Positive Integer Must be greater than zero	The minimum width to attain before creating a feature. We recommend using the lowest possible width that is consistent with the read length of the sequencer and fragment size used to generate the library, as any 'feature' narrower than this would seem to imply an artefact, e.g. 15.
Tolerance	Positive Integer	This number indicates how willing the algorithm is to ignore a temporary drop below either cut-off. This can be important to reduce sensitivity to localized read bias For a detailed account of how this works, see 'Tolerance' in the documentation for 'TLA'. We recommend using a tolerance between 500 and 2000.
Low cut-off	Positive Integer Must be greater than zero	The minimum expression level that, when additional evidence exists (junction reads), indicates the presence of a feature. We recommend using the smallest level sufficient to ignore random noise, e.g. 1 * number of BAM files.
Cut-off ratio	Positive Integer	Optional. An adaptive low cut-off and tolerance can be applied. The adaptive cut-off will be set to the greater of (current expression level) / (cut-off ratio) and the supplied low cut-off until the feature ends. The tolerance will be temporarily increased by the same ratio. We recommend using 5.
Zero Read Penalty	Negative Integer	The penalty applied to the cutoff for regions where no reads are found. We achieved best results with this parameter set at either -1 or -2.
Minimun Junction Size	Positive Integer or 0	Specifies the minimum size junction considered for transcript annotation.

Detail

Initially, ‘FeaturesNearJunctions’ is called to find large features that occur near junction reads.

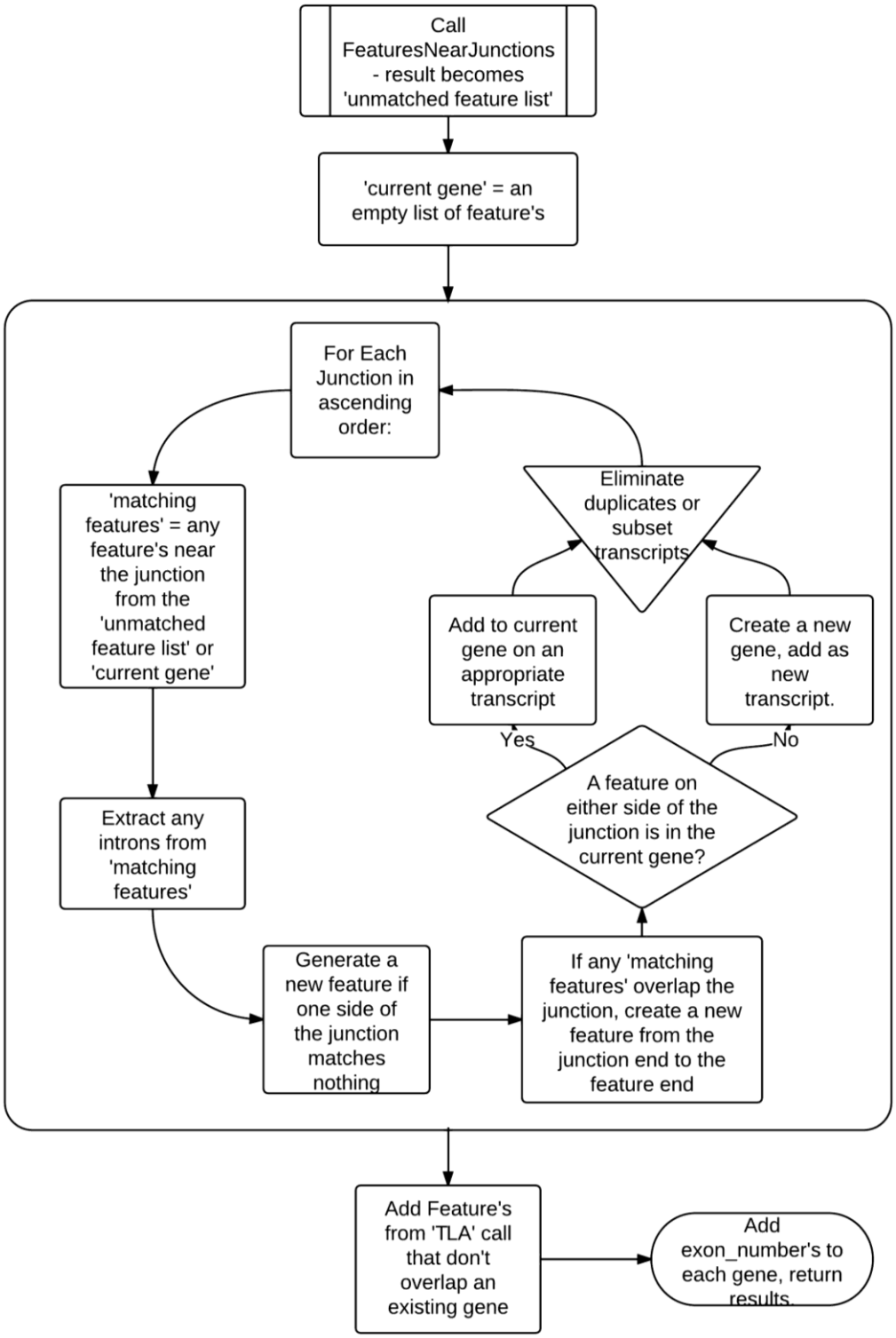
Then, each junction is examined. Based on the features that it overlaps, either

1. Features are joined as exons of a gene
2. New features are created
3. New transcripts are created

For example, if there is a feature at either end, it is assumed that they belong to the same gene. If there is a feature at one end, but the other end ends halfway through a feature, it is assumed that a new transcript must exist.

The following flow chart indicates the program structure in broad terms. More detail is provided on how the program deals with each scenario on the following pages.

Transcripts Flowchart



Extracting introns

If a junction read is wholly contained within an expressed feature, this is assumed to indicate that an intron exists at that location. The feature will be split into a feature at either end of the intron.

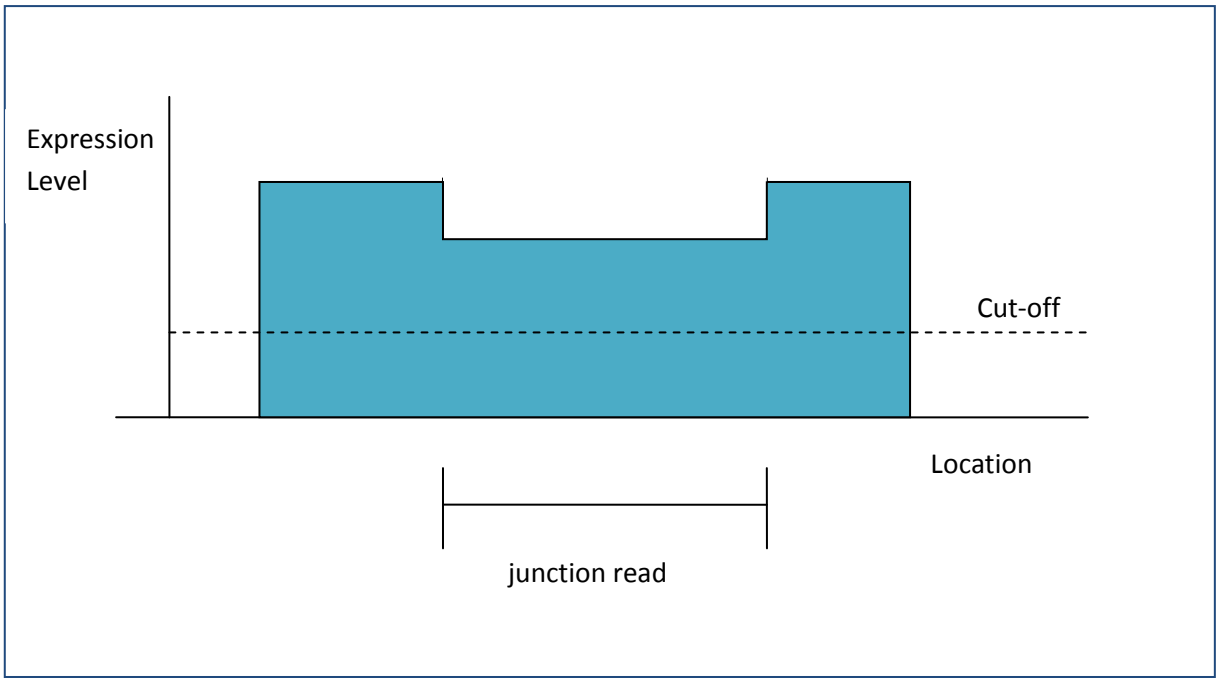
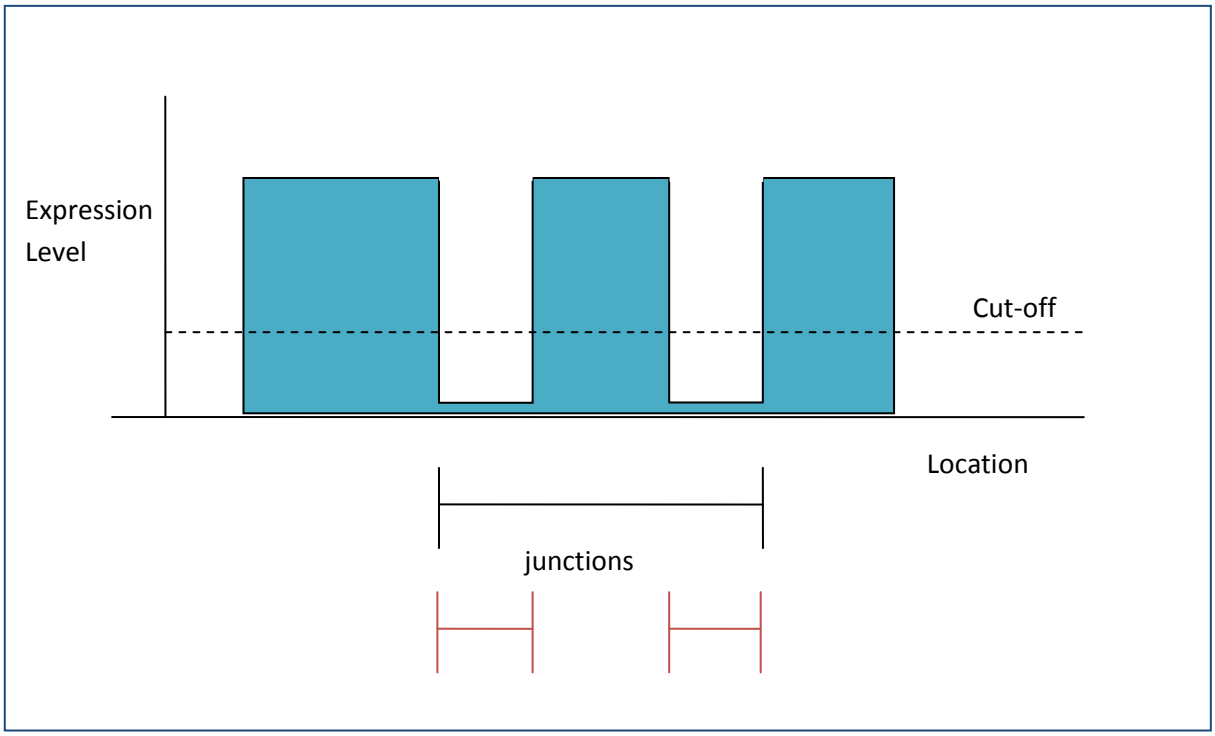


Figure 1 Example coverage where junction reads indicate the presence of two exons joined by a highly-expressed intron

Removing intronic features

Conversely, if a feature is wholly contained within a junction read, it is labeled as occupying an intronic region, but not removed as it must be part of another exon (at this stage, only feature's on either end of junctions exist – highly expressed feature's that aren't near junctions will be added in a final pass).



Searching for new features

In this scenario, there is evidence that the gene continues, but for some reason the final exon is very poorly expressed (perhaps due to sequencing bias). In this case, we lower the cut-off and walk along the chromosome until either the expression levels drop below even the new cut-off, or another feature is reached.

If another feature is encountered, we extends the region of the feature to reach the junction, then join it into the existing gene.

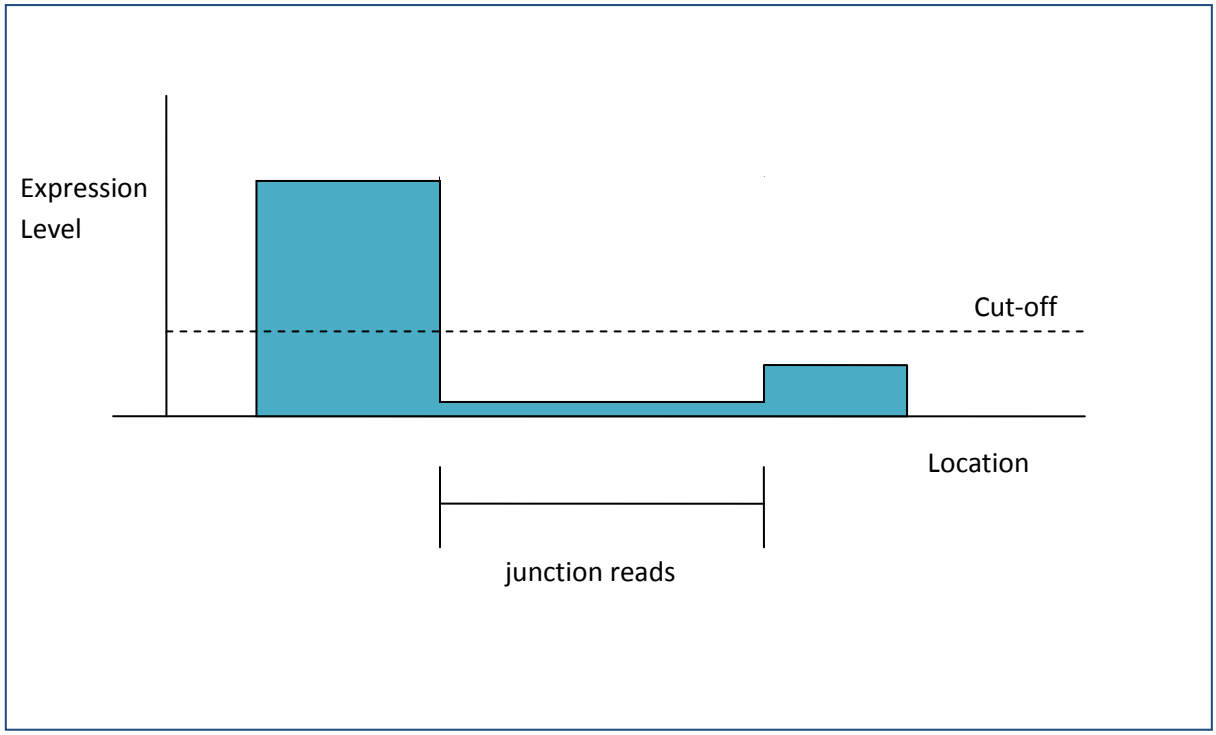


Figure 2a The strongly expressed feature on the left has junction reads spanning to the right,

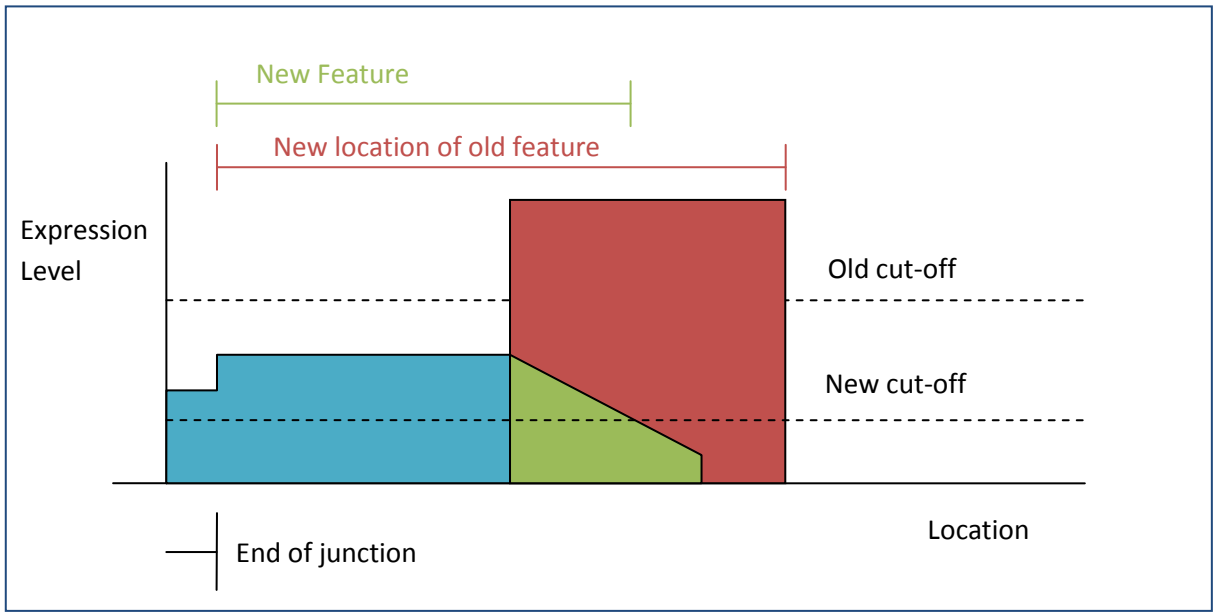


Figure 3b Given the additional evidence for a feature , a new cut-off is set. There are now two possibilities. If a strongly expressed feature is encountered further down the track (red), then the start/end position of that feature is adjusted to meet the junction. If on the other

Adaptive cut-off

When searching for new feature's in this fashion, it might occur that the expression levels become very high (as in the red scenario in the diagram above). In this case, the user has the option of specifying a cut-off ratio, which will cause the cut-off to change to the ratio multiplied by the current expression level. For example, if the cut-off ratio is 5, the cut-off is 3, and the current expression level is 1000, then the cut-off will be set to 200.

Actually, if an adaptive cut-off is used, the program will also annotate the feature once using the original cut-off. This record is kept in a list of 'shadows', for the purpose of excluding any reads that are found in this region as being probable transcriptional noise from the large feature.

Joining exons into transcripts

If expressed features sit on both sides of the gene, this is understood to indicate that those isolated features ought to be joined into exons of the same gene[1].

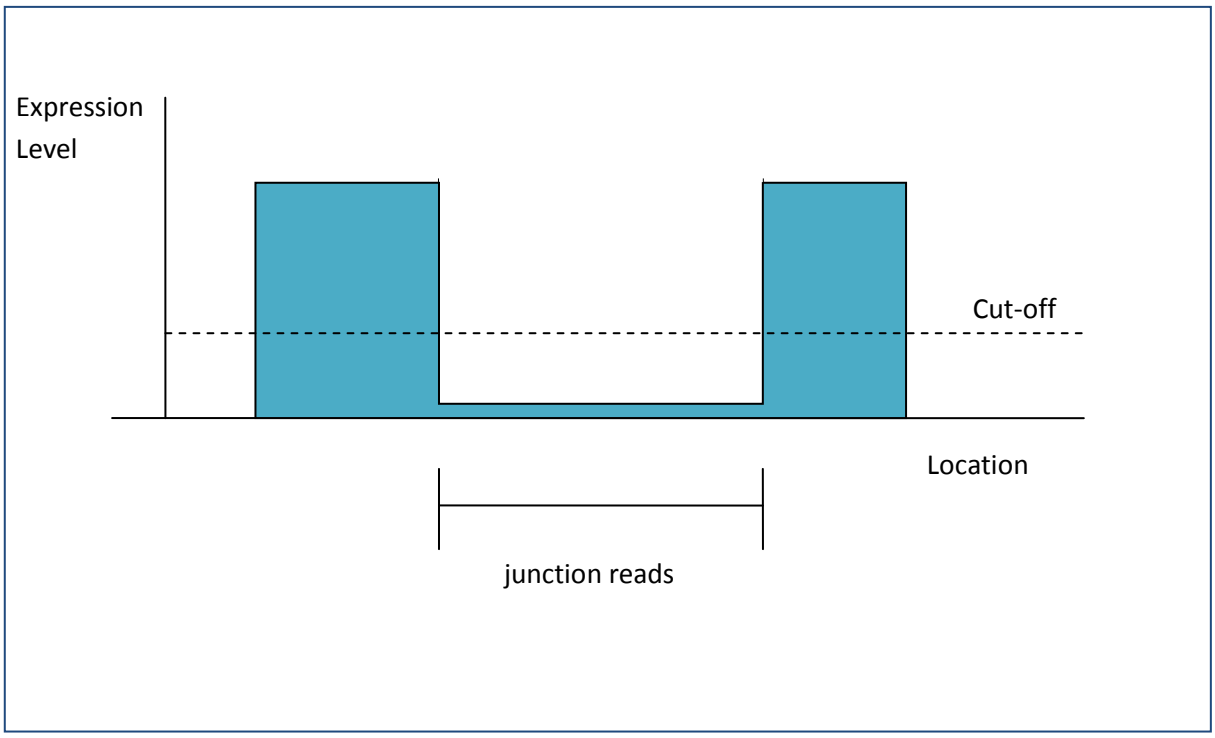


Figure 3 Example coverage where junction reads indicate that two features are connected.

If either feature is already part of a transcript, then the new feature will be added to that transcript. However, if adding the new feature would 'break' the transcript, then that transcript is duplicated, and the new feature is added to the new transcript. A feature 'breaks' a transcript if it overlaps the existing transcript anywhere.

[1] The gene_id used will be that of the 'left-hand' feature, i.e. with a smaller start position.

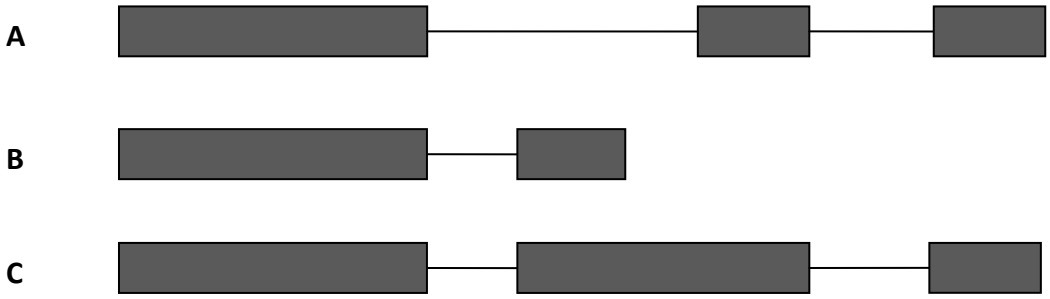


Figure Features that 'break' transcripts

Imagine that we have transcript **A** from figure 4 above. Now we find the junction shown in **B**. Since this new feature overlaps and therefore 'breaks' transcript A, we create it as a new transcript. Furthermore, the end position of the new feature doesn't line up with any end position in transcript **A**, so the entire transcript from that point on may not exist. However, suppose the feature's end *did* match up with the end of the old feature. Then we must preserve any connections that already exist, as we have shown in transcript **C**.

Finally, 'TLA' is called to find any features that do not occur near junction reads. Any features it detects that overlap with an existing exon are ignored, while feature's that overlap an existing gene (but not any exon within it) are added to that gene as a new transcript.

TLA

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

TLA is a Three-Letter Abbreviation that stands for 'The Line Algorithm' (pardon the pun). It looks for features that have an expression level (a 'height') greater than a cut-off value (the 'line'), as well as a width greater than a minimum value.

While originally, this was all that the algorithm did, it now includes a tolerance for temporary drops below the line.

Recommended Use Case

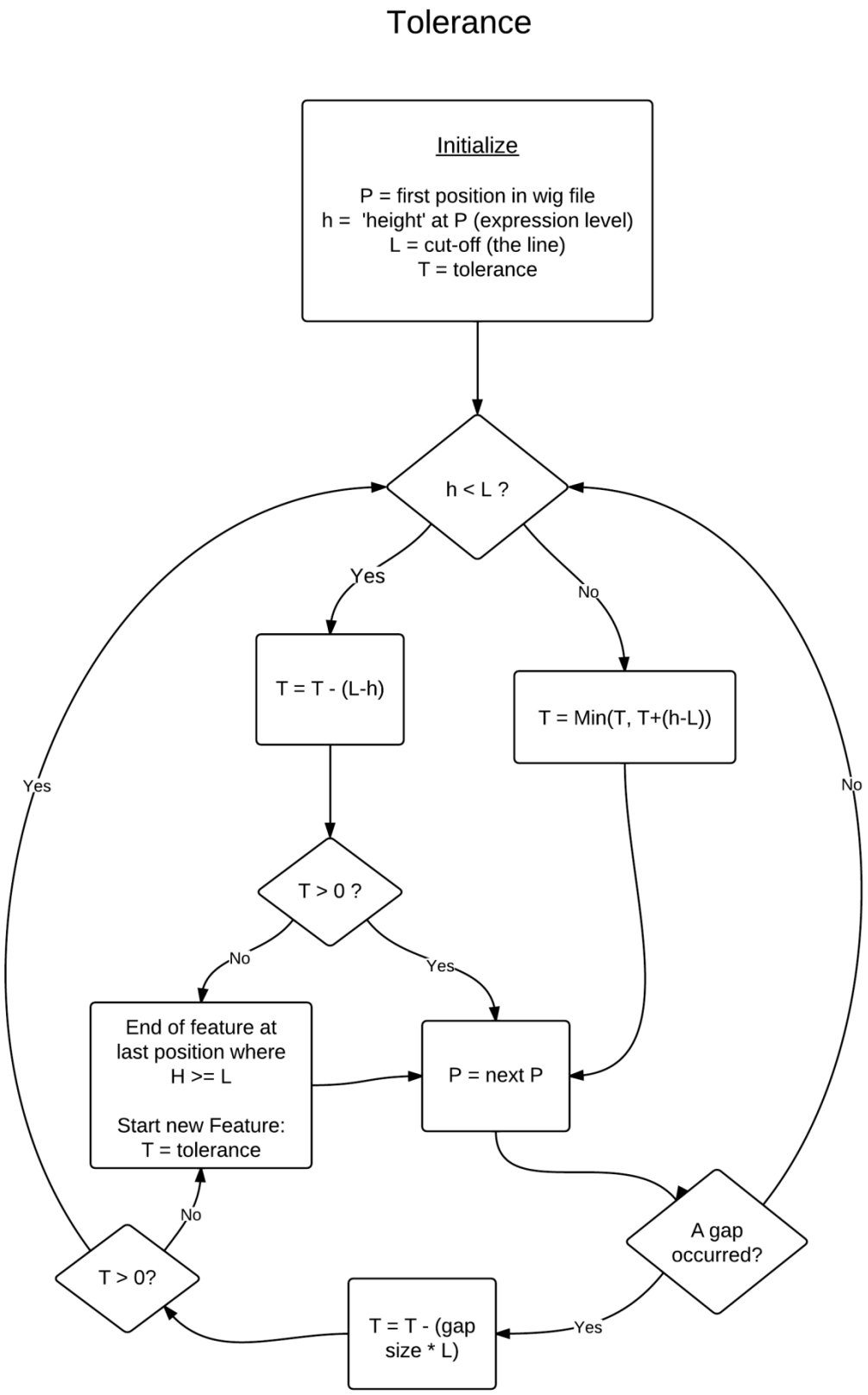
TLA is well-suited to differential expression analysis of highly expressed features. It is also often used in conjunction with other algorithms, or to provide a general overview of the data gathered.

Parameters

Name	Type	Description
The Line	Positive Integer Must be greater than zero	The minimum expression level that, on its own, indicates the presence of a feature. We recommend using the lowest possible number of reads here for which you could still possibly be interested in the feature, e.g. 5 * number of BAM files
Minimum width	Positive Integer Must be greater than zero	The minimum width to attain before creating a feature. We recommend using the lowest possible width that is consistent with the read length of the sequencer, as any 'feature' narrower than this would seem to imply an aberration, e.g. 15.
Tolerance	Positive Integer	This number indicates how willing the algorithm is to ignore a temporary drop below either cut-off. For a detailed account of how this works, see the subsection 'Tolerance' below. We recommend using a tolerance between 500 and 2000.

Tolerance

The implementation of the tolerance feature is depicted by this flowchart:



Essentially, if the current expression level *h* drops below the line, an amount is subtracted from the tolerance equal to the difference between *h* and the line. Conversely, if *h* is greater than the line, the tolerance is similarly increased, though it will never be increased beyond the original parameter.

If the tolerance drops below zero, the current feature will be declared to end at the last position where *h* was above the line, *not* at the current position. Therefore, the tolerance is designed to allow a temporary drop in expression only if the expression levels rise again before the tolerance runs out.

Any gaps in the wig file are assumed to indicate areas of zero expression, though this may not be the case depending on which cutoffs were used in creating the wig file. Therefore, the tolerance will respond to a gap as if to a sequence of values that were zero, i.e. subtracting the line (take zero) from the tolerance once for each point.

FeaturesNearJunctions

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

This algorithm looks between each adjacent pair of junctions. Feature *A* is created by extending from the junction on the left as far to the right as possible, and feature *B* is created by extending from the junction on the right as far to the left as possible. The process used to extend these features is similar to the line algorithm – look for expression above a cutoff, but have a tolerance for temporary drops. If the two features overlap, then a single feature is added to the result that joins the two junctions. Otherwise, *A* and *B* are added to the result as separate features. The effect of this is that all features found neatly line up to junction ends.

Recommended Use Case

This algorithm is primarily intended as a starting point for the Transcripts algorithm, but if otherwise should be used alongside TLA to pick up any expressed features that aren't near junctions.

Parameters

Name	Type	Description
Height	Positive Integer Must be greater than zero	The minimum expression level that, on its own, indicates the presence of a feature. We recommend using the lowest possible number of reads here for which you could still possibly be interested in the feature, e.g. 5 * number of BAM files
Tolerance	Positive Integer	This number indicates how willing the algorithm is to ignore a temporary drop below either cut-off. For a detailed account of how this works, see 'Tolerance' in the documentation for 'TLA'. We recommend using a tolerance between 500 and 2000.
Ratio	Positive Integer	The adaptive cut-off will be set to the greater of (current expression level) / (cut-off ratio) and the supplied low cut-off until the feature ends. The tolerance will be temporarily increased by the same ratio. We recommend using 5.

Introns

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

The Intron algorithm is designed to extract only intronic regions. The Intron algorithm utilizes junction reads to extract potential intronic regions from the dataset. No parameters are required to perform this analysis as all junction reads that pass the filtering will be considered as potential introns.

Recommended Use Case

This algorithm is designed to extract intronic regions from identified transcripts using junction reads

Parameters

Name	Type	Description
Minimum Junction Size	Positive Integer or 0	The minimum junction size to be included.

TSS and UTR

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

TSS and UTR are two separate algorithms that are almost identical in operation.

TSS identifies Transcription Start Sites by searching the 3' region before a gene, using a very low cut-off and high tolerance. See 'TLA' for more detail on how this is performed, as the same process is used.

To determine where the 3' region is, the algorithm first calls 'Transcripts' to identify the location of genes and exons. It then starts looking before any feature with an exon id of 1 (or after if on the negative strand), and creates a one base-pair long feature at the first position at which transcription occurs.

UTR follows a similar process to search for features in the 5' region after a gene. The only difference in the output is that UTR creates a feature spanning from the last exon all the way to the end of the un-transcribed region.

Recommended Use Case

Identification of probable transcription start sites / un-transcribed regions.

Requires

This algorithm has a dependency on the following algorithms:

- Transcripts
    - FeaturesWithJunctions
- TLA

Parameters

Name	Type	Description
High cut-off	Positive Integer Must be greater than zero	The minimum expression level that, on its own, indicates the presence of a feature. We recommend using the lowest possible number of reads here for which you could still possibly be interested in the feature, e.g. 5 * number of BAM files
Minimum width	Positive Integer Must be greater than zero	The minimum width to attain before creating a feature. We recommend using the lowest possible width that is consistent with the read length of the sequencer, as any 'feature' narrower than this would seem to imply an aberration, e.g. 15.
Tolerance	Positive Integer	This number indicates how willing the algorithm is to ignore a temporary drop below either cut-off. For a detailed account of how this works, see 'Tolerance' in the documentation for 'TLA'. We recommend using a tolerance between 500 and 2000.
Low cut-off	Positive Integer Must be greater than zero	The minimum expression level that, when additional evidence exists (junction reads), indicates the presence of a feature. We recommend using the smallest level sufficient to ignore random noise, e.g. 1 * number of BAM files.
Cut-off ratio	Positive Integer	Optional. An adaptive low cut-off and tolerance can be applied. The adaptive cut-off will be set to the greater of (current expression level) / (cut-off ratio) and the supplied low cut-off until the feature ends. The tolerance will be temporarily increased by the same ratio. We recommend using 5.
Minimun Junction Size	Positive Integer or 0	Specifies the minimum size junction considered for transcript annotation.
Gene or Transcript	String	Required..This determines whether the algorithm searches near each transcript, or each gene.

lncRNA

Author: Alex Finkel and Samuel Forster - MIMR-CIIID

Outline

This algorithm is designed to particularly identify long non-coding RNAs by running the TLA algorithm to identify large, un-spliced transcripts before performing sequence analysis to identify only those long features that do not contain significant open reading frames.

Recommended Use Case

This algorithm is designed to allow identification of putative long non-coding RNA

Parameters

Name	Type	Description
Height	Positive Integer Must be greater than zero	The minimum expression level that, on its own, indicates the presence of a feature. We recommend using the lowest possible number of reads here for which you could still possibly be interested in the feature, e.g. 5 * number of BAM files
Length	Positive Integer Must be greater than zero	This number indicates the minimum length feature, in base pairs, that should be considered as a potential lncRNA. We recommend a value greater than 2000
Maximum ORF	Positive Integer Must be greater than zero	This number represents the maximum open reading frame length that can occur before the product is considered a potential protein coding gene. We recommend a value above 50.
FASTA File	Fasta File	A Fasta file of the chromosome to be analyzed should be provided to enable open reading frame analysis to occur. Please refer to the Fasta file documentation in the File Type section of the User Guide for more information
GTF File	Gtf File	A Gtf file of features to be analyzed as lncRNA can optionally be provided to facilitate guided analysis. Please refer to the GTF file documentation in the File Type section of the User Guide for more information