

**ALEXA-Seq**  
([www.AlexaPlatform.org](http://www.AlexaPlatform.org))

## **User Manual (v.1.17)**

**15 Feb 2012**

# Table of Contents

Table of Contents .....	2
Authors .....	3
Citation .....	3
License .....	3
Acknowledgements .....	3
Affiliations .....	3
Introduction .....	4
Before Starting .....	4
System requirements .....	5
Warnings .....	5
Example data .....	5
Dependencies .....	6
Obtaining ALEXA-Seq code .....	8
Using the ALEXA-Seq Virtual Machine .....	9
Installation of ALEXA-Seq code .....	10
Script and Module Locations .....	10
Configuration .....	10
1. Analysis configuration .....	11
2. Project configuration .....	11
3. Installation of ALEXA-Seq annotation database .....	12
A. Download and install an existing database (recommended) .....	12
B. Request the database .....	12
C. Create new ALEXA-Seq database .....	12
Analysis .....	13
Command creation .....	13
0.) Install the Ensembl API and BioPerl .....	13
1.) Create target directories .....	13
2.) Import raw read data .....	13
3.) Basic data statistics .....	14
4.) Generate read fasta files .....	14
5.) Mapping .....	14
6.) Read assignment (aka 'parsing') .....	15
7.) Read assignment summary and statistics .....	15
8.) Expression calculations .....	15
9.) Summarize expression statistics .....	16
10.) Regenerate expression values .....	16
11.) Create expression matrix files for each feature .....	16
12.) Create custom UCSC track files to visualize data .....	16
13.) Calculate Differential Expression (DE) of all features .....	16
14.) Calculate Alternative Expression (AE) of all features .....	16
15.) Populate the ALEXA-seq data viewer .....	16
Result file locations .....	17
Using the ALEXA-Seq data viewer .....	19
1.) Populate ALEXA-Seq data viewer .....	19
2.) Index all gene records using the Xapian-Omega utility .....	19

## Authors

ALEXA-Seq is the work of Malachi Griffith and Marco A. Marra.

## Citation

Malachi Griffith, Obi L. Griffith, Ryan D. Morin, Michelle J. Tang, Ying-Chen Hou, Trevor J. Pugh, Rodrigo Goya, Jill Mwenifumbo, Suganthi Chittaranjan, Adrian Ally, Jennifer K. Asano, Susanna Y. Chan, Haiyan I. Li, Helen McDonald, Kevin Teague, Yongjun Zhao, Thomas Zeng, Allen Delaney, Martin Hirst, Gregg B. Morin, Steven J. M. Jones, Isabella T. Tai, Marco A. Marra\*. *Alternative expression analysis by RNA sequencing*. Pending publication.

## License

ALEXA-Seq is open source and available for public use under the terms of the GNU General Public License, version 3. For details refer to:

<http://www.gnu.org/licenses/gpl.txt>

## Acknowledgements

We are grateful for funding provided by the following organizations: the University of British Columbia, Faculty of Graduate Studies and Faculty of Medicine, the Michael Smith Foundation for Health Research, the Natural Sciences and Engineering Research Council, Genome British Columbia, the National Cancer Institute of Canada and the Terry Fox Foundation.

## Affiliations

British Columbia Cancer Agency - Genome Sciences Centre

University of British Columbia - Faculty of Medicine – Department of Medical Genetics

## Introduction

This manual provides instructions to assist in the completion of an ALEXA-Seq analysis (Alternative Expression Analysis by massively parallel RNA sequencing). The purpose of the ALEXA-Seq method is to comprehensively profile the expression, differential expression and alternative expression of a transcriptome and compare transcript specific expression events between two or more conditions of interest. Novel transcript discovery is also facilitated. These analyses require as input, paired-end RNA-Seq (aka WTSS) data from a high throughput sequencing device (such as an Illumina GAI). For examples of the output of the analysis please refer to our website. Briefly, the output consists of expression, differential expression and alternative expression values for transcripts as well as their component exons, junctions, boundaries, introns, etc.

In order to complete an ALEXA-Seq analysis you will need to download the source code and other resources from our website. The website also contains example data, and the results of many ALEXA-Seq analyses performed by us as part of various collaborations. As a department of the BC Cancer Agency, our focus in these collaborations has primarily been the analysis of cancer samples. However, the method described below was designed to work for any comparison between paired or grouped conditions. Although we have primarily analyzed human and mouse data, the method should also work for any species currently annotated by Ensembl. For further details please visit our website: [www.AlexaPlatform.org](http://www.AlexaPlatform.org)

## Before Starting

Before you can run the ALEXA-Seq analysis, you will need to ensure that your system meets certain requirements. We make use of numerous existing bioinformatics tools. Many of these are commonly used, but you should check the list of dependencies below to ensure that all necessary tools are installed. If you have problems that are not covered by this manual, please contact us. Contact information is available on our website: [www.AlexaPlatform.org](http://www.AlexaPlatform.org)

## System requirements

The ALEXA-Seq pipeline is designed to run in a 32- or 64-bit Linux system. We use RHEL 4 and CentOS 5, but any distribution should work fine. Due to the extremely large datasets generated by next-generation sequencing devices you will require considerable CPU and storage resources to perform an ALEXA-Seq analysis. We used about ~20-100 Gb of disk space to process each of the datasets listed in the results section of our website ([www.AlexaPlatform.org](http://www.AlexaPlatform.org)). Most of this is needed only temporarily and the final result should take 5-10 Gb of storage space.

We also had access to a Beowulf style CPU cluster of mostly quad-core nodes. During the mapping stage we typically used at most 100-200 CPUs (~25-50 nodes) and during all other steps of the analysis 20-30 CPUs (5-6 nodes). Although, not technically needed to perform ALEXA-Seq analyses, it is highly recommended to increase the rate of data processing. Since configuration of the CPU cluster will likely differ at your centre, we provide jobs to be submitted to the cluster and leave the actual submission of these batch files to the user. The jobs themselves should not change but the steps involved in submitting those jobs to your cluster may vary from ours.

Finally, in order to create your own ALEXA-Seq data viewer to help visualize and distribute data (optional), you will need access to a web server (we use APACHE2).

## Warnings

You will require at least a basic familiarity with Linux and Perl to perform the ALEXA-Seq analysis. Furthermore, certain aspects of this pipeline are extremely computationally intensive. I have included scripts to assist in the creation of parallel jobs to be run on a cluster of computers. If you do not have access to such computer resources, this analysis may not be practical for large datasets.

## Example data

Example datasets can be downloaded from our website as they become available. [http://www.alexaplatform.org/alex\\_a\\_seq/results.htm](http://www.alexaplatform.org/alex_a_seq/results.htm)

## Dependencies

You will need the following components to be installed. Some of these should be included with your Linux distribution by default. Others will need to be installed. For a detailed walkthrough of the installation procedure, including all dependencies refer to the separate ALEXA-Seq Linux installation manual included with the source code. Installation of the dependencies can be completely avoided by using the ALEXA-Seq Virtual Machine (see 'Obtaining ALEXA-Seq Code' below).

### A.) For the core analysis (everything up to creating the data viewer)

#### Perl

<http://www.perl.org/>

Perl should already be installed with Linux. Both Perl 5.6.x and 5.8.x should work.

#### BLAST

<ftp://ftp.ncbi.nlm.nih.gov/blast/executables/LATEST/>

Download the 32- or 64-bit version as needed.

To ensure compatibility of sequence databases, use the same version as us: 2.2.18. More recent version will probably be fine but if you want to use the latest version you may have to update or re-create the blastable databases. Blast provides a utility for this called 'formatdb'. To create new databases, simply go to the directories containing the database files in fasta format and use 'formatdb' to create an indexed version.

#### BWA

<http://sourceforge.net/projects/bio-bwa/>

Download the latest version. We used version '0.5.8a' to create the databases we provide for download. As with BLAST, newer versions should work with the indexed databases we provide but occasionally the authors release a version that requires your databases to be updated. As described above, fixing this simply requires that you re-index the database using the 'bwa index' command.

#### mdust

<http://compbio.dfci.harvard.edu/tgi/software/>

The version we use in the pipeline is included with the ALEXA-Seq source code package in the 'external\_tools' sub-directory. Unpack and compile this tool. Note the location of the directory containing the 'mdust' binary. This information will be needed when setting up your ALEXA-Seq configuration file.

#### R – The R Project for Statistical Computing

<http://www.r-project.org/>

Once R is installed, begin a session and install the following additional packages: 'RColorBrewer', 'Cairo'

#### Bioconductor (biocLite)

<http://www.bioconductor.org/>

Bioconductor is used for certain data processing and graphing functions. The biocLite

version contains all necessary packages.

### **Berkeley DB Perl Module**

<http://search.cpan.org/dist/BerkeleyDB/BerkeleyDB.pod.P>

Needs to be installed for some components of ALEXA-Seq to work. Ideally it would be installed globally in your system so that Perl knows where to find it. If this is not possible, instructions are provided in INSTALL.txt with the source code for setting the appropriate environment variables.

### **MySQL – Community Server**

<http://dev.mysql.com/downloads/>

MySQL may be included with your Linux distribution. ALEXA-Seq pipeline uses mysql databases to access Ensembl data via the Ensembl API as well to store custom annotation data. We have been using MySQL version 4.1.

### **MySQL DBI/DBD**

<http://search.cpan.org/dist/DBI/>

<http://search.cpan.org/dist/DBD-mysql/>

DBI/DBD may be included with your Linux distribution by default. These are two Perl modules that allow Perl to interact directly with MySQL database.

### **BioPerl**

<http://www.bioperl.org/Core/Latest/index.shtml>

BioPerl is used by the Ensembl API. BioPerl should be installed into the 'ensembl\_api' sub-directory of alexa\_seq. This installation will be handled automatically by the pipeline.

### **Ensembl API**

<http://ensembl.org/info/data/api.html>

Updates should be installed into the 'ensembl\_api' sub-directory of alexa\_seq. This installation will be handled automatically by the pipeline.

## **B.) For creating the data viewer**

### **Cairo**

<http://cairographics.org/>

In order to dynamically produce graphics in R you will need to install both the 'cairo' and 'pixmap' packages for Linux.

### **Xapian-Omega**

<http://xapian.org/>

Used to index results created by the ALEXA-Seq pipeline. This tool will need to be installed somewhere that is accessible by your web server. It will also need permission to run as an executable process on the web server.

## Obtaining ALEXA-Seq code

There are several options for getting the ALEXA-Seq code.

### A.) From our website

The source code package as well as ALEXA-Seq annotation databases can be downloaded from here. The code is available

[http://www.alexaplatform.org/alex\\_seq/downloads.htm](http://www.alexaplatform.org/alex_seq/downloads.htm)

### B.) From SourceForge.net

<https://sourceforge.net/projects/alex-seq/>

### C.) From our public subversion repository

If you have subversion ('svn') installed, you can check out the code directly from our subversion repository. For example, execute the following from a command prompt

```
svn co https://svn.bcgsc.ca/public/ALEXA_Seq/tags/ALEXA_Seq_v1.17 ALEXA_Seq_v1.17
```

## Using the ALEXA-Seq Virtual Machine

Instead of downloading and installing the ALEXA-Seq code and the dependencies it relies on, you can download and run a pre-configured Virtual Machine (VM) that contains the code and dependencies. These downloads are available here:

[http://www.alexaplatform.org/alexa\\_seq/downloads.htm](http://www.alexaplatform.org/alexa_seq/downloads.htm)

### Step-by-step instructions:

- 1.) Download and install one of the player options described in the downloads section of the ALEXA-Seq website.
  - 2.) Download an ALEXA-Seq virtual machine from the website. Pick the 32-bit or 64-bit VM to match your system
  - 3.) Unpack the archive. Use 'tar -zxvf ArchiveName.tar.gz' for Linux or Mac and [7-Zip](#) for Windows.
  - 4.) Start VMware player and select 'Open Virtual Machine' from the 'File' menu.
  - 5.) Browse to where you unpacked the ALEXA-Seq VM, select 'ALEXA-Seq XX-bit.vmx', and select 'Open'.
  - 6.) Select 'Edit virtual machine settings' to modify the number of CPUs and amount of memory used.
  - 7.) When ready, select 'Play virtual machine'. The system will now boot inside a window.
  - 8.) To toggle full screen mode press: 'Ctrl+Alt+Enter'.
  - 9.) The login username and passwords are:  
Username: 'alexa-seq'  
Password: 'alexa-seq' (root password is 'alexa777')
- The same user name and passwords are used for the mysql database installation. There is also a read only mysql user 'viewer' with the password 'viewer'.
- 10.) Once the VM has booted, open the file 'DEMO.txt' on the desktop for a demonstration.

## Installation of ALEXA-Seq code

The following is a general overview of the installation procedure, and assumes all dependencies (p 5-6) are already installed. For a detailed walkthrough of the installation procedure, including all dependencies refer to the ALEXA-Seq Linux installation manual.

Before proceeding, download and unpack the ALEXA code base. For example:

```
mkdir /home/user/ALEXA/  
cp ALEXA_Seq_v.1.17.tar.gz /home/user/ALEXA/  
cd /home/user/ALEXA/  
gunzip ALEXA_Seq_v.1.17.tar.gz  
tar -xvf ALEXA_Seq_v.1.17.tar
```

Unpacking the code is not necessary if you obtained it from our svn repository.

Once the code is installed. Open the INSTALL.txt file and copy the indicated section of environment variables to your .bashrc file (or an equivalent shell parameters file that is automatically loaded on log in). These variables should be edited to reflect your own specific file paths and install directories. Finally, once this is done, log out of your session and log in again to make sure the environment variables take effect.

## Script and Module Locations

The root/reference directory for all scripts is: '~/alexa\_seq' (where ~ is wherever the code was unpacked). Many of the scripts described below make use of methods that we have written and stored in utility modules. The details of these functions are beyond the scope of this document. These Perl modules follow POD (plain old documentation) format and are stored in: '~/alexa\_seq/utilities'. Each script can be run without arguments to display a list of instructions.

## Configuration

Before starting your analysis you will need to track down basic information about your computer system as well as specific info relating to the project. A 'project' relates to a biological question. Many examples of such projects that have already been analyzed by ALEXA-Seq are provided at our website:

[http://www.alexaplatform.org/alexa\\_seq/results.htm](http://www.alexaplatform.org/alexa_seq/results.htm)

For example, a project could consist of a tumour versus normal comparison, or treated versus untreated cell lines, etc. Each project generally has two or more sequence 'libraries' corresponding to these conditions. Each sequence library may itself consist of multiple 'lanes' of paired-end sequence data that were generated as one or more runs of a high-throughput sequencing device. Each project generally also has at least one 'comparison' defined (e.g. tumour vs. normal).

## 1. Analysis configuration

For tidiness, you should have three main directories for ALEXA-Seq processing. Briefly, these contain the code itself, the sequence databases, and the analysis files. For example these directories could be:

```
/home/user/ALEXA/alexa_seq/  
/home/user/ALEXA/sequence_databases/  
/home/user/ALEXA/alexa_seq_analysis/
```

Other folders you may wish to create:

```
/home/user/ALEXA/perl_storables/ (for BerkeleyDB perl storables?)  
/home/user/ALEXA/www/ (for web files in case you can't write directly to your web server)  
/home/user/ALEXA/commands/ (for .commands files for each project)  
/home/user/ALEXA/config_files/ (for config_files for each project)
```

Before performing your first analysis, you will need to create an updated analysis configuration file. This file stores basic system specific parameters, mostly paths to directories or binaries.

Create this file by starting with the example file provided:

```
cp ~/alexa_seq/config_files/examples/ALEXA_Seq_PIPELINE.conf  
~/config_files/ALEXA_Seq_PIPELINE.conf
```

Edit all values in the config files to reflect your own file locations.

The validity of values entered in the configuration files will be tested in the first step of the analysis (by createAnalysisCommands.pl - see below).

## 2. Project configuration

Before starting analysis for a new project (and for the first analysis you run), you will also need to create a project configuration file. This file stores information about the data to be analyzed. Specifically, you must define each 'LANE', 'LIBRARY', and 'COMPARISON'. For example, you will need to determine the paths to your input sequence data. Also included in this file are configuration values that might be project specific (e.g. the species being analyzed, the genome version to use, etc.).

Create this file by starting with the example file provided here:

```
~/alexa_seq/config_files/examples/ALEXA_Seq_<ExampleProject>.conf
```

The validity of values entered in this configuration file will be tested in the first step of the analysis.

### 3. Installation of ALEXA-Seq annotation database

You will need an ALEXA-Seq annotation database to perform the analysis. This database defines all the canonical, alternative and hypothetical sequence features that will be considered in the analysis. The database is specific to a particular species and genome build of that species. You have three options for obtaining these databases. Instructions for each of these options are as follows:

#### A. Download and install an existing database (recommended)

Pre-computed databases are provided here:

[ftp://ftp03.bcgsc.ca/public/ALEXA/alexa\\_seq/](ftp://ftp03.bcgsc.ca/public/ALEXA/alexa_seq/)

Each of these is described here:

[http://www.alexaplatform.org/alexa\\_seq/downloads.htm](http://www.alexaplatform.org/alexa_seq/downloads.htm)

Most of the analyses described on our website used the Human build 'hs\_53\_36o' (NCBI Build 36/UCSC hg18) but databases for several other species and builds are available. Installation of these databases is an automated step of the analysis pipeline.

#### B. Request the database

If your analysis involves a species we have not included as a download, you can request that we add it by contacting us through our website ([www.AlexaPlatform.org](http://www.AlexaPlatform.org)). We currently have ALEXA-Seq annotation databases for: Chicken, Chimp, Fly, Human, Mouse, Rat, Yeast, and Zebrafish.

#### C. Create new ALEXA-Seq database

If you wish to create a complete new annotation database you can use the following tool to create the necessary instructions:

```
~/alexa_seq/createAnnotationCommands.pl
```

You will be asked to supply two configuration files. One was created above. For the other, use the following file as an example:

```
~/alexa_seq/config_files/examples/ALEXA_Seq_dr_57_8c.conf
```

This script will create a new '.commands' file containing detailed instructions for building your own ALEXA-Seq annotation database.

A detailed description of the database schema is provided here:

[http://www.alexaplatform.org/alexa\\_seq/data/ALEXA\\_Seq\\_Schema\\_Description.htm](http://www.alexaplatform.org/alexa_seq/data/ALEXA_Seq_Schema_Description.htm)

# Analysis

## Command creation

Once you have created or updated your system and project configuration files as described above, execute the following to create the analysis commands:

```
cd ~/alexa_seq/  
./createAnalysisCommands.pl  
--alexa_seq_config_file=config_files/examples/ALEXA_Seq_PIPELINE.conf  
--project_config_file=config_files/examples/ALEXA_Seq_<ExampleProject>.conf  
--commands_file=ALEXA_Seq_ExampleProject.commands
```

Open the '.commands' file created by this script and follow the instructions. The tasks that will be performed are briefly described below. *Each step described below corresponds to a step in the .commands file with the same number.* Note that unless explicitly stated each step should be completed before moving on to the next one.

### 0.) Install the EnsEMBL API and BioPerl

The version to use is specified in the configuration file. It will only be installed if not already present.

### 1.) Create target directories

All directories need for the analysis and creation of results files, statistics, figures etc. will be automatically created.

### 2.) Import raw read data

Currently, raw data can be in the 'seq', 'qseq' or 'fastq' formats. This step will concatenate (if necessary) the \*\_seq.txt or \*\_qseq.txt files from a source directory to create a combined raw seq data file. There may be many tile files corresponding to a single lane, or they may already be joined together. These files are expected to be named as follows:

```
s_1_1_0001_qseq.txt.bz2  
i.e. s_<Lane>_<Read-1-or-2>_Tile_<seq-or-qseq>.txt.<gz-or-bz2>
```

Note: compressing seq or qseq files is not required but recommended to save storage space.

The format of sequence data in the raw .seq files is as follows:

```
Lane  Tile  X-coord  Y-Coord  Sequence
```

In this format, the sequence for read1 and read2 is pasted together for paired reads. Bases which could not be resolved are represented by a '.' and will be converted to N's. During this step, the complexity of each read will be determined by 'mdust' and various statistics pertaining to each lane will be summarized (total read counts, low complexity reads, poor quality reads, etc.). During this step, it is also possible to trim reads if there was a problem with the quality of read ends.

A description of qseq and fastq file formats can be found here:

qseq: <http://jumpgate.caltech.edu/wiki/QSeq>

fastq: <http://en.wikipedia.org/wiki/Fastq>

Note that each lane of data can be imported concurrently.

### **3.) Basic data statistics**

Gather basic info about each lane, library and comparison defined in the Project configuration file. Also get number of quality reads, the average read length, and the overall tag redundancy of each library.

#### **3-A.) Gather info about each lane of data**

Imported from the project configuration file.

#### **3-B.) Gather info about each library**

Imported from the project configuration file.

#### **3-C.) Gather info about each comparison**

Imported from the project configuration file.

#### **3-D.) Generate statistics for each library**

This step will determine: the number of quality read counts in each lane, the average read length of the library, and the overall tag redundancy of each library. If the library consists of a mixture of read lengths (not advisable) it may be useful to know the average read length.

### **4.) Generate read fasta files**

One fasta file will be created for each lane of data after filtering out reads that are poor quality (too many ambiguous bases, i.e. N's), low complexity (e.g. polyA reads), or where both reads of a pair are identical (library artifacts).

Note that all of steps 3-4 can be run concurrently.

### **5.) Mapping**

Reads will be mapped to a database of repeat elements, transcripts, known exon junctions and boundaries, hypothetical exons junctions and boundaries, and intronic and intergenic regions of the genome. Where possible, read-pairing information is used to resolve ambiguously mapped reads but if only one read of a pair can be mapped this read will still be retained (i.e. paired mapping is not required). With default parameters reads are mapped with up to 3 mismatches and 1 gap. Sub-string alignments are also allowed. A perfect alignment of 60% of the read length will be allowed (again assuming default parameters).

## **6.) Read assignment (aka 'parsing')**

Based on the alignments, reads will now be assigned to their most probable source. If possible each read is unambiguously assigned to a repeat element, known transcript, novel exon junction, novel exon boundary, intron, or intergenic region. At this stage, the apparent distance between reads of a pair is determined as well as other statistics pertaining to mapping efficiency. Reads that can not be assigned to a repeat or human genome or transcriptome sequence are retained but are marked as 'Unassigned' and excluded from downstream analysis.

## **7.) Read assignment summary and statistics**

The number of reads assigned to each category/read class for each library will be determined. These counts will be determined on a lane-by-lane basis as well as for each library. The presence of potential read position bias (within transcripts) will also be assessed during this step.

Note that all of steps 7-A to 7-E can be run concurrently.

## **8.) Expression calculations**

Using the coordinate information in the ALEXA-Seq annotation database in combination with the mapping results generated above, the expression of 13 types of sequence features will be determined. These features consist of: Genes, transcripts, exon regions, exon junctions (known and novel), exon boundaries (known and novel), introns, active intronic regions, silent intronic regions, intergenic regions, active intergenic regions and silent intergenic regions. Refer to the manuscript methods for further details on the annotation of these features. The result of this analysis for human is an expression value for ~4 million sequence features.

Transcript specific expression for all known Ensembl transcripts is calculated by using only those exon regions and exon junctions that are unique to each transcript.

For each feature, several expression metrics are calculated. These include cumulative coverage, average base coverage (cumulative coverage divided by the length of the sequence feature), the percentage of bases of a feature covered at 1x or greater, etc. See the schema description for a detailed explanation of all expression measures. [http://www.alexaplatform.org/alex\\_seq/data/ALEXA\\_Seq\\_Schema\\_Description.htm](http://www.alexaplatform.org/alex_seq/data/ALEXA_Seq_Schema_Description.htm)

The expression of exon junction and boundary features are systematically lower than those for exon regions. This is due to an inherent mapping disadvantage for these sequences (see manuscript for discussion). For this reason, these values are empirically adjusted to compensate for the disadvantage.

Note that once steps 8-A and 8-B are complete steps 8-C to 8-E can be run concurrently.

### **9.) Summarize expression statistics**

In this step, various expression statistics and graphs are generated. For each library, gene specific expression cutoff values are determined by examination of intronic and intergenic noise levels (see manuscript for details). The number of features of each type that are expressed above background is then determined. The overall sequence coverage of genes is also determined in this step.

Note that steps 9-A to 9-D can be run concurrently.

### **10.) Regenerate expression values**

Using the expression values determined above, the junction/boundary correction factors and gene-by-gene cutoffs values are now used to re-evaluate the expression of all features. To accomplish this, steps 8-9 are repeated.

### **11.) Create expression matrix files for each feature**

This step creates simple expression matrix files for downstream analysis.

Note that you can start step 11 and proceed to the next step without waiting for it to finish.

### **12.) Create custom UCSC track files to visualize data**

Expression values will be used to create custom UCSC track files (a mixture of GFF and wig tracks) to display all features expressed above background and the base level sequence coverage of the genome. Similarly a wig track displaying base level differential expression will also be created in this step for each comparison you define.

### **13.) Calculate Differential Expression (DE) of all features**

Differential expression of all features (Genes, Exons, Junctions, Boundaries, etc.) will be assessed as the log<sub>2</sub> difference in expression level for each comparison you define. Each differential expression value will also be associated with a p-value.

### **14.) Calculate Alternative Expression (AE) of all features**

Alternative expression of all features (Genes, Exons, Junctions, Boundaries, etc.) will be assessed by calculating splicing index (SI) values, reciprocity index (RI), and percent feature contribution (PFC) values for each comparison you define (see manuscript for details).

Note that steps 13 and 14 can be run concurrently.

### **15.) Populate the ALEXA-seq data viewer**

Calculations and organization required to display results in the ALEXA-seq data viewer will be performed. This involves defining library paths, packaging expression matrix files to be made available for download, creating summaries of expression DE and AE, creating library QC summaries, generating candidate gene and peptide lists, creating gene-by-gene summaries, etc.

Note that steps 15 A-G can be run concurrently.

## Result file locations

The following is a brief description of files created during the ALEXA-Seq analysis and their relative storage location (all files are created automatically during the analysis).

In the following descriptions ‘\$analysis\_dir’ refers to the base path where all the results are stored (e.g. /home/user/alexa\_seq\_analysis/). This parameter is set at the beginning of the analysis in the ALEXA-Seq configuration file. Similarly, ‘\$project\_name’ is the name of the project defined in the configuration file and ‘\$library\_id’ is a variable used to denote each of the libraries being processed for that project.

### Batch files

All command files to be executed for a particular project are stored as follows:

[\\$analysis\\_dir/batch\\_jobs/\\$project\\_name/](#)

### Raw sequence data

Raw sequence files imported from the high-throughput sequencing device are stored as follows:

[\\$analysis\\_dir/raw\\_seq\\_data/\\$library\\_id/](#)

### Fasta sequence files

Fasta files containing all reads passing basic quality filters are stored for each lane of each library here:

[\\$analysis\\_dir/fastq\\_seq\\_data/\\$library\\_id/](#)

### Read record files

These files store the read sequences of each pair on a single line along with basic quality metrics for the read and its current assignment status. All reads start as ‘Unassigned’ and if possible are assigned to transcripts, junctions, introns, etc.

[\\$analysis\\_dir/read\\_records/\\$library\\_id/](#)

### Mapping results files

Within the ‘read\_records’ directory, mapping results files are stored in sub-directories. For convenience, of downstream analysis, individual mapping results are provided for each target sequence type (repeats, transcripts, junctions, boundaries, introns, and intergenic regions).

[\\$analysis\\_dir/read\\_records/\\$library\\_id/\\$sequence\\_type/](#)

### Figures and statistics

A large number of statistics, summary files and figures are generated for each lane of data, library and comparison.

Summary of annotations used for the analysis:

[\\$analysis\\_dir/figures\\_and\\_stats/Generic/](#)

Basic lane-by-lane library quality statistics:

[\\$analysis\\_dir/read\\_records/\\$library\\_id/Summary/](#)

Feature expression, expression correlations, read assignments, average coverage values, library statistics, etc.:

[\\$analysis\\_dir/figures\\_and\\_stats/\\$library\\_id/Expression\\_v\\*/](#)

Differential expression results (by sequence feature type):

[\\$analysis\\_dir/figures\\_and\\_stats/DE/\\$project\\_name/\\$feature\\_type/](#)

Alternative expression results (by sequence feature type):

[\\$analysis\\_dir/figures\\_and\\_stats/SI/\\$project\\_name/\\$feature\\_type/](#)

### **Log files**

Log files from the ALEXA-Seq analysis:

[\\$analysis\\_dir/logs/\\$library\\_id/](#)

### **Temporary files**

Working directory for creation of temporary files during data processing:

[\\$analysis\\_dir/temp/](#)

Temporary storage for ALEXA-Seq data viewer files

[\\$analysis\\_dir/temp/\\$project\\_name/](#)

## Using the ALEXA-Seq data viewer

### 1.) Populate ALEXA-Seq data viewer

To help summarize and visualize the expression of known and novel isoforms as well as differential and alternative gene expression between conditions of interest, all expression, differential expression and alternative expression results will be imported into the ALEXA-Seq data viewer (essentially a dynamically generate web interface). This viewer also facilitates data sharing with collaborators, design of validation experiments, and interpretation of the results.

In addition to these results, detailed summaries of the characteristics of each library are also produced.

Candidate gene lists for differential and alternative expression events are automatically generated for each comparison. Similarly, 'matrix' files are generated to facilitate downstream analysis such as clustering, pathway analysis, etc.

Candidate peptide lists to assist in the creation of antibodies that are specific to each condition are also created for each comparison.

Several examples of data sets processed and displayed in the ALEXA-Seq data viewer are available at our website:

[http://www.alexaplatform.org/alexa\\_seq/results.htm](http://www.alexaplatform.org/alexa_seq/results.htm)

### 2.) Index all gene records using the Xapian-Omega utility

For every project, a summary is available for every gene. To allow searching for arbitrary genes, the results will be also indexed with the tool 'Xapian-Omega' in this step.

An example of this search functionality can be found here:

<http://www.bcgsc.ca/xapian-search/omega>