Analysis of EBV Transcription Using High-Throughput RNA Sequencing

Tina O'Grady[1], Melody Baddoo[1,2] and Erik K. Flemington[1,2]

[1]Department of Pathology, Tulane University School of Medicine

[2]Tulane Cancer Center

Corresponding author:

Erik K. Flemington

erik@tulane.edu

Running head: RNA-Seq analysis of EBV

Summary/Abstract

High-throughput sequencing of RNA is used to analyze the transcriptomes of viruses and cells, providing information about transcript structure and abundance. A wide array of programs and pipelines has been created to manage and interpret the abundance of data generated from high-throughput RNA sequencing experiments. This protocol details the use of free and open-source programs to align RNA-Seq reads to a reference genome, visualize read coverage and splice junctions, estimate transcript abundance and evaluate differential expression of transcripts in different conditions. Particular concerns related to EBV and viral transcriptomics are addressed and access to EBV reference files is provided.

Key Words

RNA-Seq, Epstein-Barr Virus, Transcriptomics, Differential expression, STAR, Integrative Genomics Viewer, IGV, RSEM, EBSeq

1. Introduction

High-throughput RNA-sequencing has become a powerful tool in virus research. The parallel sequencing of millions of short reads has allowed transcriptome-wide analysis of cellular and viral gene expression, not to mention discovery of novel genes, splice junctions and isoforms. Translational research makes use of high-throughput RNA sequencing to detect known and novel viruses in clinical samples and to analyze both transcript and transcriptome-level changes as viruses interact with their hosts. In addition, high-throughput RNA sequencing can be combined with other techniques for specialized tasks such as transcription start site annotation, identification of RNA-protein partners and examination of active transcription and translation genome-wide [1-4].

While several high-throughput RNA-sequencing technologies have been developed, Illumina RNA-Seq has become the *de facto* standard. Variations in protocols allow researchers to ask different questions of the transcriptomes under study. One important consideration is the treatment of the RNA before sequencing. Poly(A)-selection extracts mRNA and polyadenylated long noncoding RNA for sequencing. Ribodepletion removes most ribosomal RNA before sequencing, leaving polyadenylated RNAs, RNA species without poly(A) tails such as EBV's EBERs, and some unprocessed mRNA transcripts. At the RNA library preparation step, researchers can choose to use a stranded protocol to preserve information about which DNA strand the sequenced RNA arose from. This approach has revealed significant previously unknown antisense and intergenic transcription in EBV and other herpesviruses [5-8] and is recommended, especially for cases in which the virus is expected to be in a replicative phase. Researchers can also choose between single-end and paired-end sequencing. In single-end sequencing a single read is generated from one end of each RNA fragment. In paired-end sequencing two reads are produced from each RNA fragment: one from each end. Single-end sequencing is useful for expression quantification and with sufficient read length, it can provide meaningful information on structural features such as splice junctions and fusion transcripts. Paired-end sequencing offers more transcript structure information, especially for cases of fusion transcripts, circular RNA and other unannotated variants, but comes at a greater time and financial cost. The read lengths obtainable with RNA-Seq have increased substantially in recent years. High quality 100 bp reads are now readily obtainable on the Illumina platform and this length is sufficient to allow sensitive detection of splice junctions. Finally, a common option to reduce RNA-Seq costs is multiplexing: adding unique "barcode" sequences to samples, allowing them to be sequenced together and informatically separated after sequencing. In general, multiplexing still allows sufficient read depth to study EBV genes that are expressed at relatively low levels.

There are some special considerations when applying RNA-Seq to EBV research. Most plainly, EBV is studied within the context of the human cellular environment. RNA-Seq experiments from EBV-infected

cells mostly produce reads that map to the human genome, with the proportion of EBV reads ranging from up to about 25% (after a highly robust lytic induction in a cell line) down to a few reads (e.g. in a weakly EBV-positive tumor sample with a latent gene expression profile). Generally, higher read counts allow more effective transcriptome analysis. However, analysis of EBV-mapping reads is more straightforward under latency conditions, when the viral transcriptome is substantially simpler and it is easier to discern individual transcripts.

This protocol describes analysis steps performed in nearly any RNA-Seq experiment. First, RNA-Seq reads are aligned to a reference genome. Many alignment programs, both open-source and commercial, are available; this protocol uses the fast and sensitive STAR aligner [9]. Output from STAR is used to examine RNA-Seq coverage of the genome as well as to identify splice junctions. The data is visualized on a genome browser, in this case IGV, the Integrative Genomics Viewer [10]. Finally, transcript expression levels are quantified and compared between samples in a statistically meaningful way. This protocol uses the program RSEM [11] and its accompanying EBSeq [12] module for quantification and the evaluation of differential expression.

2. Materials

1. RNA-Seq data: fastq or fasta files acquired from a sequencing facility or downloaded from a data repository
2. Hardware: a computer with X86-64 compatible processors running a 64-bit UNIX/Linux-based operating system (e.g. Mac OS, Ubuntu or Cygwin) with at least 30 GB of RAM and enough storage space to accommodate sequence and reference files. Estimate at least 60 GB of storage for reference/index files and another 50 GB per single-end RNA-Seq sample (*see* **Note 1**).
3. A terminal program to interface with the operating system, (e.g. Terminal in Mac OS) (*see* **Note 2**).

4. STAR aligner: the most recent version of STAR aligner is available at https://github.com/alexdobin/STAR. The site provides the necessary files and instructions for installation.
5. Fasta files for reference genomes: fasta-format files (with file extension .fa) are available for several strains of EBV at https://github.com/flemingtonlab/public/tree/master/annotation (*see* **Notes 3-4**). Fasta-format files for the human genome may be downloaded from Ensembl at http://www.ensembl.org/info/data/ftp/index.html. Fasta-format files for many more genomes can be obtained from the NCBI Nucleotide database at http://www.ncbi.nlm.nih.gov/nuccore.
6. A text editor program (e.g. TextEdit, pre-loaded on most Macs)
7. Integrative Genomics Viewer (IGV): the most recent version of IGV is available, along with instructions for installation, at http://www.broadinstitute.org/software/igv/download.
8. Genome annotation files: bed-format (with file extension .bed) and gtf-format (with file extension .gtf) files are available for several strains of EBV at https://github.com/flemingtonlab/public/tree/master/annotation. Gtf-format files for the human genome may be downloaded from Ensembl at http://www.ensembl.org/info/data/ftp/index.html.
9. Perl: The current version of Perl is available at https://www.perl.org/ (*see* **Note 5**)
10. The Perl script *junctions_to_introns_STAR.pl*, available from https://github.com/flemingtonlab/public/tree/master/code
11. RSEM: the latest version of RSEM can be obtained from http://deweylab.biostat.wisc.edu/rsem/. The site provides the necessary files and instructions for installation.

Methods

3. 1 Creating Genome Index Files for STAR aligner

Aligners require that the genomes first be "indexed" to facilitate quicker and less computationally intensive sequence matching. Genome indexes are generated from fasta files containing each chromosome of the genome of interest and only need to be generated once. This example creates a genome index containing both the human genome and the EBV genome, using a human reference fasta file downloaded from Ensembl and the EBV reference fasta file *chrEBV_Akata_inverted.fa* [13] downloaded from https://github.com/flemingtonlab/public/tree/master/annotation (*see* **Notes 6-7**).

1. Create a new, empty directory to which the STAR genome index will be written (for the example below, the directory must be named *GenomeDirectory*). The path to this directory will need to be entered below, after --genomeDir.
2. Run the following command in Terminal:

   $ STAR --runMode genomeGenerate --genomeDir /PATH/TO/GenomeDirectory --runThreadN 4 --genomeFastaFiles /PATH/TO/chrEBV_Akata_inverted.fa /PATH/TO/Homo_sapiens.GRCh38.dna.primary_assembly.fa

   The runThreadN number, in this case 4, should be equal to or less than the number of processor cores available. Run time for this procedure is about 1 hour using a computer with 12 cores.

STAR will generate multiple files in the genome directory. These files should not be modified.

3. 2 Aligning Sequence Reads to the Reference Genome

In this step the fastq/fasta files containing sequence data are mapped to the reference genome index that was generated in the previous step (*see* **Note 8**). The following command uses STAR to align single-end sequence reads and report their alignments in a SAM format text file and their splice junctions in another text file. Sequence reads that map to fewer than 10 different genomic locations with fewer than 10 mismatches to the genome are reported (*see* **Note 9**).

1. In Terminal, use *cd* to move to the desired output directory.
2. Run the following command:

   $ STAR --genomeDir /PATH/TO/GenomeDirectory --readFilesIn /PATH/TO/reads.fastq --runThreadN 4 --outSAMprimaryFlag AllBestScore

   The runThreadN number, in this case 4, should be equal to or less than the number of processor cores available. The argument "--outSAMprimaryFlag AllBestScore" determines output for the reads that map to multiple places in the genome. With this option set, if multiple genomic locations tie for the best score, all of those locations will be reported as primary alignments with the same score. Run time for this procedure, assuming a fastq file 6GB in size is about 6 minutes using a computer with 12 cores.

STAR will generate multiple output files into the current working directory. The file containing the aligned reads is *Aligned_out.sam* and the file containing identified splice junctions is *SJ.out.tab* (*see* **Notes 10-11**).

3.3 Visualizing the Data

The alignment and splice junction files are, to a certain extent, human-readable and may potentially be inspected by opening them in a text editor. Often the alignment file in particular is very large and difficult or impossible (due to RAM limitations) to open in a text editor. An alternative is to open a small portion of the file in Terminal using the command

$ less Aligned_out.sam

The file can then be scrolled through a line at a time using the keyboard arrow keys (*see* **Note 12**). To exit the *less* view and return to terminal, type *q*. Examining these text files in this way is a limited and difficult way to interpret the data however. To better view and analyze alignment data, it can be loaded

onto a genome browser. The steps below outline the use of the Integrative Genomics Viewer (IGV) to view RNA-Seq read coverage across the entire viral genome. Prior to loading alignment data however, the reference genome must be formatted by IGV. The human genome is pre-loaded, but a reference genome must be created for EBV. This example creates a reference genome for the Akata strain of EBV, using the files *chrEBV_Akata_inverted.fa* (genome sequence file) and *chrEBV_Akata_inverted.bed* (genome feature annotation file), both downloaded from https://github.com/flemingtonlab/public/tree/master/annotation. The .genome file created through this process need only be created once.

1. In IGV, from the *Genomes* menu select *Create .genome File*.
2. Enter a user-specified ID for the genome (this will be displayed in the IGV Genomes list) and a descriptive name.
3. Using the fasta file *Browse* button, select *chrEBV_Akata_inverted.fa*.
4. Using the genome annotation *Browse* button, select *chrEBV_Akata_inverted.bed*. Leave the cytoband file box blank.
5. Save the file with a .genome file extension in the directory of your choice. The new genome will be loaded onto the IGV genomes list automatically.

In order to view the alignment file on IGV, it must first be sorted. Sort the alignment file using the following command in Terminal:

$ sort –k4,4n Aligned_out.sam > Aligned_out_sorted.sam

Full alignment files are frequently too large to be loaded onto IGV (because of RAM limitations). To create an alignment file of EBV-mapped reads only, run the following command in Terminal

$ awk ‘ $3==”chrEBV_Akata_inverted” ’ Aligned_out_sorted.sam > Aligned_out_sorted_EBV_only.sam

(*see* **Note 13**).

The sorted alignment file can be loaded onto IGV by selecting the *File* menu, then *Load from File* and browsing to the file. IGV will prompt you to create an index file (necessary for display of the data): reply *OK* to the prompt to allow IGV to create the index file automatically.

For strand specific RNA-Seq, reads in the display are color-coded according to strand (Fig. 1). Above the track displaying the reads IGV displays an automatically generated coverage track that contains a histogram of read depth for each coordinate across the genome. This coverage track combines reads from both strands. To look at each strand separately, create an alignment file for each strand using the following Terminal commands (*see* **Notes 14-16**):

$ awk '$2==0' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only _0_plus.sam

$awk '$2==16' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only _16_minus.sam

Then load each track separately onto the genome browser. Additionally, it is possible to create a standalone coverage track for each strand (or both strands together) using IGV's IGVtools feature with the following steps:

1. Select the *Tools* menu, then *Run igvtoools…*
2. Make sure the *Command* drop-down menu is set to *Count*
3. Use the *Input File* browse button to locate the desired alignment file (e.g. *Aligned_out_sorted_EBV_only _0_plus.sam*)
4. An output filename is automatically generated, with the file extension .tdf. This creates a binary file; if you prefer a human-readable output file, change the file extension to .wig
5. Change the *Window Size* to *1* to get an accurate read depth count for each genomic position
6. Click *Run*

3. 4 Examining Splice junctions

Like the alignment file, the splice junctions output file (*SJ.out.tab*) is human-readable but is easier to visualize on a genome browser. To view the text, use the Terminal *less* command as in step 3.3:

$ less SJ.out.tab

In order to visualize the splice junctions file on IGV, it must first be converted to bed format. To do this, use the Perl script *junctions_to_introns_STAR.pl*:

1. In Terminal, use *cd* to move to the directory that contains the *SJ.out.tab* file.
2. Use the following Terminal command:

   $ perl /PATH/TO/junctions_to_introns_STAR.pl SJ.out.tab

   The output file *SJ.out.tab.bed* will be located in the same directory.
3. Like the alignment file, the full splice junctions file may be too large to load on IGV due to RAM limitations. Create a file of EBV splice junctions by using the Terminal command

   $ awk ' $1==" chrEBV_Akata_inverted" ' SJ.out.tab.bed > SJ.out.tab_EBV_only.bed (*see* **Note 17**)
4. In IGV, load the splice junctions file by selecting *Load from File* from the *File* menu and browsing to *SJ.out.tab_EBV_only.bed*

When loaded on IGV, the *SJ.out.tab_EBV_only.bed* file displays features corresponding to spliced-out introns, with the first base of the feature representing the first base of the intron and the last base of the feature representing the last base of the intron (Fig. 2, *see* **Note 18**). The default view on IGV is *Collapsed*: to view any overlapping introns with greater detail right click on the track name (*SJ.out.tab_EBV_only.bed*, to the left of the display) and select *Expanded*. Mouse over an individual splice junction feature to see its *Score*: this is the number of uniquely mapping RNA-Seq reads that span this junction.

3. 5 Creating Transcriptome Reference Files for RSEM

The RSEM package works well to quantify transcript expression, even for overlapping transcripts that have ambiguous read mappings. It uses STAR to align reads to a reference transcriptome, then estimates the abundance of each transcript using the Expectation-Maximization algorithm. Because abundance estimation involves normalization to total read depth, abundance estimates for EBV genes are better represented when RSEM reference transcriptome files include both viral and cellular genes (*see* **Notes 19-20**). In this step, RSEM prepares transcriptome reference files by using transcript coordinates in a .gtf-format annotation file to extract transcript sequences from a genome fasta file, and indexes those transcripts to serve as reference sequences for alignment. Transcriptome reference files need only be generated once for each transcriptome. To prepare all the necessary reference files for RSEM:

1. Create a directory called, e.g., *human_and_EBV_references* and move all human and EBV genome fasta files and annotation gtf files to it
2. In Terminal, use *cd* to move to the newly created directory
3. Combine the human and EBV gtf-format files into a single file by using the following command

   $ cat Homo_sapiens.GRCh38.81.gtf chrEBV_Akata_inverted_for_EBV.gtf > human_38.81_and_Akata_inverted.gtf
4. Create a new, empty directory to contain the RSEM transcriptome index.
5. In Terminal, use *cd* to move to the new directory
6. Run the following command:

   $ rsem-prepare-reference /PATH/TO/human_and_EBV_references --gtf /PATH/TO/human_38.81_and_Akata_inverted.gtf -p 4 --star human_and_EBV_RSEM_reference

   (*see* **Note 21**)

The final argument is the user-defined name of the reference genome, and should be informative. The -p number, in this case 4, should be equal to or less than the number of processor cores available. The --star argument creates STAR transcriptome index files for the next step. RSEM will generate multiple files

in the transcriptome directory. Run time for this procedure is about 4.5 hours using a computer with 12 cores.

3. 6 Quantifying Transcript Expression

In this step the fastq/fasta files containing sequence data are mapped to the transcriptome reference that was generated in the previous step, and transcript abundance is estimated. The following steps use RSEM and STAR to align single-end sequence reads and produce a text file containing abundance estimates for all transcripts in the transcriptome reference.

1. In Terminal, use *cd* to move to the desired output directory.
2. Run the following command:

   $ rsem-calculate-expression -p 4 --star --no-bam-output /PATH/TO/reads.fastq /PATH/TO/human_and_EBV_RSEM_reference human_and_EBV_RSEM (*see* **Notes 21-23**)

   The -p number, in this case 4, should be equal to or less than the number of processor cores available. The --star argument instructs RSEM to perform the alignment step with STAR, using predetermined parameters optimizing the alignment for expression quantification. The --no-bam-output argument prevents RSEM from producing an alignment file: this argument can be left out of the command if alignment files are desired, though for most purposes this alignment file will not be as useful as that produced in step 3.2 as it presents reads aligned to individual transcripts rather than the genome. Run time for this procedure, assuming a fastq file 6GB in size, is about 15 minutes using a computer with 12 cores.

RSEM will generate three files in the current working directory. Human_EBV_RSEM.genes.results is a tab-delimited text file containing abundance estimates for all genes in the transcriptome file. It can be opened with a text editor or with Microsoft Excel (*see* **Notes 24-25**). The column labeled *expected_count*

contains the number of reads estimated to map to that transcript after taking into account background noise and overlapping genes (*see* **Note 26**). *TPM* (Transcripts Per Million) and *FPKM* (Fragments Per Kilobase of transcript per Million mapped reads) are two different normalized abundance measures that take into account transcript length and overall read depth.

3. 7 Comparing Transcript Expression

EBSeq is an empirical Bayesian differential expression analysis tool that is built into the RSEM package. It is used to find statistically significant differential expression of transcripts between two or more groups of one or more samples each. If EBseq has not yet been used after RSEM installation, use the Terminal command

$ make ebseq

to compile the necessary codes. This need only be done once. The following steps use EBSeq to compare transcript expression in two treatments with three replicates each.

1. In Terminal, use *cd* to move to the desired output directory.
2. Create a matrix of genes and their read counts in the samples to be compared by running the following command:

   $ rsem-generate-data-matrix

   /PATH/TO/first_sample_first_replicate_human_and_EBV_RSEM.genes.results

   /PATH/TO/first_sample_second_replicate_human_and_EBV_RSEM.genes.results

   /PATH/TO/first_sample_third_replicate_human_and_EBV_RSEM.genes.results

   /PATH/TO/second_sample_first_replicate _human_and_EBV_RSEM.genes.results

   /PATH/TO/second_sample_second_replicate _human_and_EBV_RSEM.genes.results

   /PATH/TO/second_sample_third_replicate _human_and_EBV_RSEM.genes.results >

   condition_one_condition_two.counts.matrix

Ensure that input files in this command are grouped in an appropriate order to do comparisons; i.e., files from each group to be compared must be adjacent to each other in the command. RSEM will produce a matrix file of read counts for each transcript, which will be used by EBSeq in the next step.

3. Indicate to EBseq how to group and compare the samples, and run EBSeq's algorithm using the following command:

   $ rsem-run-ebseq /PATH/TO/condition_one_condition_two.counts.matrix 3,3 sample_one_sample_2_ebseq.results

   In this case, 3,3 indicates two samples, each with triplicates. 2,2 would indicate two samples each with duplicates, 3,3,3 would indicate 3 samples each with triplicates, and so on. The order of these numbers must correspond to the order of the samples in the counts.matrix file; i.e. in sample_one_sample_two.counts.matrix produced in the previous step the three sample one replicates are listed, then the three sample two replicates. The final argument is the user-defined name of the output file, which is a text file of the posterior probability of differential expression.

4. Determine which transcripts are statistically significantly differentially expressed by running the following command:

   $rsem-control_fdr /PATH/TO/condition_one_condition_2_ebseq.results 0.05 condition_one_condition_two_DE.txt

   This will produce a text file listing the genes that are statistically significantly differentially expressed at a False Discovery Rate (FDR) of 0.05. The FDR can be altered as desired by replacing the number 0.05 in the command.

Run time for this procedure is about 10 minutes using a computer with 12 cores.

The file produced (in this example named *condition_one_condition_two_DE.txt*) is a tab-delimited text file that can be opened in a Text Editor or Microsoft Excel (*see* **Notes 24-25**). In the above case of two groups being compared this output file contains four columns: *PPEE* is the posterior probability of equal expression, ranging from 0 to 1; *PPDE* is the posterior probability of differential expression, ranging from 0 to 1; *posterior fold change* is the ratio of the posterior mean expression estimate of the gene in sample one to that in sample two; *real fold change* is the ratio of the gene's normalized mean count in sample one to that in sample two. Genes with fold change greater than one are expressed more highly in condition one; genes with fold change less than one are expressed more highly in condition two.

When more than two samples are compared, EBSeq produces a file that indicates the posterior probability that the gene is expressed in any of a number of "patterns". An additional file with the suffix .pattern is the key to these patterns, indicating the samples in which the gene shows differential expression.

4. Notes

1. The first steps of this protocol (up to *Examining Splice Junctions*) may be completed on a computer with 16 GB of RAM. However, quantifying and comparing transcript expression levels using these methods will require up to 30 GB of RAM.
2. For those new to Unix/Linux or command line interfaces, an excellent tutorial is Ian Korf and Keith Bradnam's *Unix & Perl Primer for Biologists*, available at http://korflab.ucdavis.edu/unix_and_Perl/.
3. For the best alignments, it is best to use a reference genome that matches the strain of EBV in the sample. For samples containing the Type I EBV strain (the most common), the Akata genome should provide good results. For samples containing the Type II EBV strain, the AG876 genome can be used.

4. Several files at https://github.com/flemingtonlab/public/tree/master/annotation are available in "inverted" format. This format splits the EBV genome between the BBRF3 and BGLF3 genes (between positions 107954 and 107955) rather than the terminal repeats to allow better detection of LMP2 transcripts, which span the terminal repeats, where EBV genomes are normally split (as in Genbank).
5. Perl may already be installed on your system. A quick way to check is to run the Terminal command

   $ perl -v

   This will report the version of Perl, if any, that is installed.
6. Genome index files that contain multiple genomes, e.g. human and EBV, may be created; however, genome indexes from multiple strains of EBV should be created separately and not combined into a single index.
7. If only the EBV transcriptome is of interest, an index of the EBV genome (only) may be created by leaving out the human genome fasta files from the command. The argument --*genomeSAindexNbases 8* should be added to the command to adjust for the much smaller genome size. The index creation and read alignment steps will be faster and less computationally intensive when using the EBV genome only.
8. This protocol assumes that quality control and filtering steps have already been performed - e.g. by the sequencing center. If not, or if this is unclear, FastQC (http://www.bioinformatics.babraham.ac.uk/projects/fastqc/) is a quick and user friendly program to check data quality.
9. Many STAR parameters can be altered to meet the requirements of different experiments. Some common variations are listed here; many others can be found in the STAR manual at https://github.com/alexdobin/STAR/blob/master/doc/STARmanual.pdf

   For paired-end RNA-Seq data, simply list both read files after the --readFilesIn argument; i.e. --

readFIlesIn /PATH/TO/reads_1.fastq /PATH/TO/reads_2.fastq

To only report alignments for reads that map uniquely to one genomic location, add the argument --*outFilterMultimapNmax 1*. Or change this number to report alignments for reads that map to multiple locations in the genome. The default (as used in step 3.2) is 10: reads aligning to more than 10 genomic locations will not be reported.

Use the argument *–genomeSAsparseD 2* to reduce the RAM usage. This will result in a longer runtime.

10. The names of output files can be specified by adding the argument --*outFileNamePrefix* to the command line statement followed by the desired name, which will be used as a prefix for the standard output names (*Aligned.out*, *Log.out*, etc.). It is good practice to make the file name prefix as informative as reasonably possible; e.g. --*outFileNamePrefix* *sample_one_repeat_one_STAR_chrEBV_Akata_inverted*
11. To check what parameters were used for previous STAR alignments, open the *Log.out* file with a text editor. The default parameters are listed at the top. Scroll down to see the arguments entered on the command line, which defaults were changed and the new values used in the alignment.
12. *Less* has additional functionality that is often useful; for example, patterns may be searched using the format */pattern<enter>*. To learn more about *less*, bring up the manual in Terminal with the command

    $ man less
13. You must ensure that the chromosome name in the command matches the desired chromosome name in the alignment file. In this example, all reads that align to the chromosome called *chrEBV_Akata_inverted* will be extracted and written to the EBV only file. To inspect the names of chromosomes in the alignment file, use

    $less Aligned_out.sam

to open the alignment file, and look in the header lines after *@SQ SN:* for the names of chromosomes.

14. Depending on whether the cDNA protocol used first-strand or second-strand synthesis, the numbers 0 and 16 may be reversed. For example, using the TruSeq stranded protocol, reads corresponding to the plus strand have a FLAG code of 16 and reads corresponding to the minus strand have a FLAG code of 0.
15. Some reads may align to multiple parts of the genome. These commands return only the primary alignments; that is, the location with the best alignment score. In order to return both primary and secondary alignments, use the following commands:

    $ awk '$2==0||$2==256' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_0_plus.sam

    $awk '$2==16||$2==272' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_16_minus.sam
16. For paired-end sequencing, use the following commands:

    $ awk '$2==99||$2==147' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_plus.sam

    $awk '$2==83||$2==163' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_minus.sam

    to create files contain only the best-mapping reds. Use:

    $ awk '$2==99||$2==147' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_plus.sam

    $awk '$2==83||$2==163' Aligned_out_sorted_EBV_only.sam > Aligned_out_sorted_EBV_only_minus.sam
17. Higher numbers of reads spanning a splice junction provide greater support for that splice junction. *junctions_to_introns_STAR.pl* reports all detected splice junctions. If you would like to

specify a minimum read depth to report splice junctions, use a command of the form

$ awk ' $5 >= 5 ' SJ.out.tab_EBV_only.bed > SJ.out.tab_EBV_only_5.bed

This example extracts splice junctions supported by at least 5 uniquely mapping reads.

18. IGV has a built-in option to display splice junctions as visually pleasing arcs rather than the default bed file blocks. To turn on this option, simply rename *SJ.out.tab.bed* to *SJ.out.tab.junctions.bed* and load the new file onto IGV.
19. Note that abundant unannotated transcription has been detected in EBV during reactivation [8]. Annotation files including these new transcripts are not yet complete but will be available at https://github.com/flemingtonlab/public/tree/master/annotation.
20. If annotation files include features that are not transcripts (e.g. annotated repeat regions, promoters, etc.) these should be removed before using the gtf file to create an RSEM reference. Otherwise, RSEM will interpret these features as annotated transcripts and may erroneously assign reads to them.
21. Depending on the installation, the error message */STAR : No such file or directory!* May be encountered. If so, add the argument *--star-path /PATH/TO/directory containing STAR*
22. Many RSEM parameters can be altered to meet the requirements of different experiments. Some common variations are listed here; many others can be found in the RSEM manual at http://deweylab.biostat.wisc.edu/rsem/README.html

    For paired-end RNA-Seq data, add the argument *--paired-end* to the *rsem-calculate-expression* command, and add the second file of reads after the first, e.g.,

    $ rsem-calculate-expression --star -p 4 --no-bam-output --paried-end /PATH/TO/reads_1.fastq /PATH/TO/reads_2.fastq /PATH/TO/human_and_EBV_RSEM_reference human_and_EBV_RSEM

    The last argument in the command (*human_and_EBV_RSEM* in this example) is the name prefix for the ouput files. It is good practice to make the file name prefix as informative as reasonably possible; e.g. *sample_one_repeat_one_RSEM_human_chrEBV_Akata_inverted*

23. An error that may be encountered is *EXITING because of FATAL ERROR: could not open genome file /PATH/genomeParameters.txt*. If this occurs, rerun the command adding the text */a* to the genome directory argument; i.e. */PATH/TO/human_and_EBV_RSEM_reference/a*
24. When using GTF files downloaded from Ensembl, RSEM output may use Ensembl IDs rather than gene names. IDs and gene names may be cross-referenced individually at Ensembl (http://www.ensembl.org) or a complete list for crossreferencing may be downloaded from Ensembl's BioMart (http://www.ensembl.org/biomart).
25. When displaying text files Excel automatically formats cells based on their content. This is sometimes inappropriate: e.g. converting the gene symbol SEPT7 to the date September 7. To avoid this problem, open text files in Excel using the following steps:

    Open Excel

    From the *File* menu, select *Import*

    Select *Text file*, navigate to the desired file and click *Get Data*

    When the Text Import Wizard opens, select *Delimited* and click *Next*

    In the *Delimiters* section, select *Tab*

    The next screen allows selection of the Data Format for each column. Ensure that the column containing gene IDs is set to *Text* and click *Finish.*
26. Note that because the expected count takes into account background noise and overlapping transcripts, its value may not be an integer.

5. Acknowledgements

This work was supported by a US National Institutes of Health Ruth L. Kirschstein National Research Service Award (F31CA180449) to TO, grants R01AI101046 and R01AI106676 to EKF and grant P20GM103518 to Prescott Deininger.

6. References

1. Kurosawa J, Nishiyori H, Hayashizaki Y (2011) Deep cap analysis of gene expression. Methods Mol Biol 687:147-163. doi:10.1007/978-1-60761-944-4_10

2. Zhao J, Ohsumi TK, Kung JT, Ogawa Y, Grau DJ, Sarma K, Song JJ, Kingston RE, Borowsky M, Lee JT (2010) Genome-wide identification of polycomb-associated RNAs by RIP-seq. Mol Cell 40 (6):939-953. doi:10.1016/j.molcel.2010.12.011

S1097-2765(10)00967-6 [pii]

3. Core LJ, Waterfall JJ, Lis JT (2008) Nascent RNA sequencing reveals widespread pausing and divergent initiation at human promoters. Science 322 (5909):1845-1848. doi:10.1126/science.1162228

1162228 [pii]

4. Ingolia NT (2010) Genome-wide translational profiling by ribosome footprinting. Methods Enzymol 470:119-142. doi:10.1016/S0076-6879(10)70006-9

S0076-6879(10)70006-9 [pii]

5. Gatherer D, Seirafian S, Cunningham C, Holton M, Dargan DJ, Baluchova K, Hector RD, Galbraith J, Herzyk P, Wilkinson GW, Davison AJ (2011) High-resolution human cytomegalovirus transcriptome. Proc Natl Acad Sci U S A 108 (49):19755-19760. doi:10.1073/pnas.1115861108

1115861108 [pii]

6. Concha M, Wang X, Cao S, Baddoo M, Fewell C, Lin Z, Hulme W, Hedges D, McBride J, Flemington EK (2012) Identification of new viral genes and transcript isoforms during Epstein-Barr virus reactivation using RNA-Seq. J Virol 86 (3):1458-1467. doi:10.1128/JVI.06537-11

JVI.06537-11 [pii]

7. Arias C, Weisburd B, Stern-Ginossar N, Mercier A, Madrid AS, Bellare P, Holdorf M, Weissman JS, Ganem D (2014) KSHV 2.0: a comprehensive annotation of the Kaposi's sarcoma-associated herpesvirus

genome using next-generation sequencing reveals novel genomic and functional features. PLoS Pathog 10 (1):e1003847. doi:10.1371/journal.ppat.1003847

PPATHOGENS-D-13-01657 [pii]

8. O'Grady T, Cao S, Strong MJ, Concha M, Wang X, Splinter Bondurant S, Adams M, Baddoo M, Srivastav SK, Lin Z, Fewell C, Yin Q, Flemington EK (2014) Global bidirectional transcription of the Epstein-Barr virus genome during reactivation. J Virol 88 (3):1604-1616. doi:10.1128/JVI.02989-13

JVI.02989-13 [pii]

9. Dobin A, Davis CA, Schlesinger F, Drenkow J, Zaleski C, Jha S, Batut P, Chaisson M, Gingeras TR (2013) STAR: ultrafast universal RNA-seq aligner. Bioinformatics 29 (1):15-21. doi:10.1093/bioinformatics/bts635

bts635 [pii]

10. Thorvaldsdottir H, Robinson JT, Mesirov JP (2013) Integrative Genomics Viewer (IGV): high-performance genomics data visualization and exploration. Brief Bioinform 14 (2):178-192. doi:10.1093/bib/bbs017

bbs017 [pii]

11. Li B, Dewey CN (2011) RSEM: accurate transcript quantification from RNA-Seq data with or without a reference genome. BMC Bioinformatics 12:323. doi:10.1186/1471-2105-12-323

1471-2105-12-323 [pii]

12. Leng N, Dawson JA, Thomson JA, Ruotti V, Rissman AI, Smits BM, Haag JD, Gould MN, Stewart RM, Kendziorski C (2013) EBSeq: an empirical Bayes hierarchical model for inference in RNA-seq experiments. Bioinformatics 29 (8):1035-1043. doi:10.1093/bioinformatics/btt087

btt087 [pii]

13. Lin Z, Wang X, Strong MJ, Concha M, Baddoo M, Xu G, Baribault C, Fewell C, Hulme W, Hedges D, Taylor CM, Flemington EK (2013) Whole-genome sequencing of the Akata and Mutu Epstein-Barr virus strains. J Virol 87 (2):1172-1182. doi:10.1128/JVI.02517-12

JVI.02517-12 [pii]

Figure Legends

Fig. 1. SAM-format alignment files displayed on IGV. From top to bottom: RNA-seq coverage and reads from both strands, RNA-seq coverage and reads from the plus strand, RNA-seq coverage and reads from the minus strand, and gene annotation (*chrEBV_Akata_inverted.bed*).

Fig. 2. BED-format splice junction files displayed on IGV. From top to bottom: introns displayed as BED features, introns displayed as splice junction arcs, and gene annotation (*chrEBV_Akata_inverted.bed*).