Molecular Profiling of RNA Tumors Using High-Throughput RNA Sequencing: From Raw Data to Systems Level Analyses

Molecular Profiling of RNA Tumors Using High-Throughput RNA Sequencing: From Raw Data to Systems Level Analyses

Willian A. da Silveira, E. Starr Hazard, Dongjun Chung, and Gary Hardiman

Abstract

RNAseq is a powerful technique enabling global profiles of transcriptomes in healthy and diseased states. In this chapter we review pipelines to analyze the data generated by sequencing RNA, from raw data to a system level analysis. We first give an overview of workflow to generate mapped reads from FASTQ files, including quality control of FASTQ, filtering and trimming of reads, and alignment of reads to a genome.

Then, we compare and contrast three popular options to determine differentially expressed (DE) transcripts (The Tuxedo Pipeline, DESeq2, and Limma/voom). Finally, we examine four tool sets to extrapolate biological meaning from the list of DE genes (Genecards, The Human Protein Atlas, GSEA, and ToppGene). We emphasize the need to ask a concise scientific question and to clearly under stand the strengths and limitations of the methods.

Key words: High-throughput sequencing (HTS), RNAsequencing (RNAseq), FASTQ, Tuxedo pipeline, HTSeq, DESeq 2, Limma/Voom, Gene Set Enrichment Analysis/GSEA, TOPPGENE

1 Introduction

Classical techniques in genetics and molecular biology remain the gold standard when one wants to detect the presence and sequence of a gene (i.e., using polymerase chain reaction (PCR) and Sanger sequencing techniques), its mRNA expression level (i.e., using quantitative PCR (qPCR)), and the corresponding protein levels (i.e., using western blots) [1]. The human genome encodes approximately 25,000 genes, with thousands of them expressed in multiple combinations in diverse cellular contexts. Furthermore multiple isoforms exist for the same gene with many possible downstream post-translation modifications [2]. The emergence and application of high throughput approaches in the past decade, the so-called “omics” fields have ignited a revolution in biological research. Classical genomics techniques only allow investigation of a small number of genes and proteins at the same time, where as “omics” approaches (including high throughput RNA sequencing) enable investigation of the entire mRNA content at the same time [1, 2].

Transcriptomics is the study of the transcriptome, i.e., the complete set of RNA transcripts that are produced by the genome, under specific circumstances or in a specific cell, using highthroughput methods, such as RNAseq [3]. Comparison of transcriptomes allows the identification of genes that are differentially expressed in distinct cell populations, for example in healthy or tumor tissues, or in response to therapeutic regimes. In this chapter we discuss the methods to analyze and interpret data from human tumor samples generated by RNAseq technology. We cover the key steps and the progression from the “FASTQ” files generated by the sequencing instrument through a list of differentially expressed genes, and system level analyses.

Although the methods and workflows described in this chapter are best suited to a Linux environment, there are options to run many of these programs in Windows or Mac OS X as well, e.g., using Cygwin (https://www.cygwin.com/) in Windows. A basic knowledge of the command line environment, file structures and rudimentary coding skills are assumed. More details about these

programs and the statistical models underlying them used can be found in the references and/or in the links of the websites in the Materials section. As noted by Mayer-Scho¨nberger and Cukier, “the data can reveal secrets to those with the humility, the willingness, and the tools to listen” [4].

2 Materials

2.1 Computing Infrastructure

There are many challenges associated with selecting and implementing the right set of tools. Bioinformatics analyses are complex, multistep processes composed of multiple software applications. Ideally, many of the programs used for RNAseq analysis are designed to show the optimal performance in a high-performance computing environment. However, these programs can still run in a sufficiently powerful and well-configured laptop or desktop machine, e.g., an Intel Core i7 processor with storage RAID 0 configured 2 parallel 1-TB hard disk drives. We recommend at least 4GB of RAM, preferably 8GB.

2.2 Software Tools and Genome Build(s)

SRA Toolkit 2.9.0 (https://github.com/ncbi/sra-tools/wiki/Downloads, http://www.ncbi.nlm.nih.gov/books/NBK158900/), Windows/Linux/MAC OS X.

FastQC v0.11.7 (http://www.bioinformatics.babraham.ac.uk/projects/FASTQc/), Windows/Linux/MAC OS X.

FASTX-Toolkit (http://hannonlab.cshl.edu/fastx_toolkit/), Linux/MAC OS X/Web based.

Cutadapt (https://cutadapt.readthedocs.io), Python Language. Windows/Linux/MAC OS X.

Bowtie1 for reads between 35 and 50 bp (http://bowtie-bio.sourceforge.net/manual.shtml). Linux/MAC OS X.

Bowtie2 for reads greater than 50 bp (http://bowtie-bio.sourceforge.net/bowtie2/manual.shtml). Linux/MAC OS X.

SAMtools (http://samtools.sourceforge.net/) . Windows/Linux/MAC OS X. Samtools also have an “R” compatible version “Rsamtools,” available from the Bioconductor website.

Human Reference Genome Sequence, hg38 (http://hgdownload.cse.ucsc.edu/downloads.html).

TopHat (https://ccb.jhu.edu/software/tophat/manual.shtml)). Linux/MAC OS X.

Cufflinks package (Cufflinks, CuffMerge, CuffMerge, Cuffdiff)(http://cole-trapnell-lab.github.io/cufflinks/tools/). Linux/MAC OS X.

CummeRBound (https://bioconductor.org/packages/release/bioc/html/cummeRbund.html). R Language. Windows/ Linux/MAC OS X.

HTSeq (http://www-huber.embl.de/HTSeq/). Python Language. Windows/Linux/MAC OS X.

Comprehensive gene annotation (GRCh38.p7, ".GTF file").
(http://www.gencodegenes.org/releases/25.html).

DESeq2 (https://bioconductor.org/packages/release/bioc/html/DESeq2.html). R Language.
Windows/Linux/MAC OS X.

Limma/Voom (https://bioconductor.org/packages/release/bioc/html/limma.html). R Language.
Windows/Linux/MACOS X.

Venny (https://www.stefanjol.nl/venny) Web tool.

ToppFun (https://toppgene.cchmc.org/) Web tool.

GSEA tool and Website (http://software.broadinstitute.org/gsea/index.jsp). Web tool and a program compatible with Windows/Linux/MAC OS X.

3 Methods

A schematic of data analysis pipelines is presented in Fig. 1. To initiate the analyses, it is necessary to have the FASTQ files that contain the information of sequenced reads and the quality score for each nucleotide. These FASTQ files can be obtained directly from the sequencing machine, as in the case depicted in the flowchart, or from the Sequence Read Archive (SRA) repository (https://www.ncbi.nlm.nih.gov/sra), where it is required to convert the downloaded ".SRA" file to FASTQ using the SRA Tool kit. Then the quality of the FASTQ file can be checked using the FastQC program and the results from it can be used as parameters in the "FASTX" Tool Kit and/or in Cutadapt to perform trimming and filtering of the reads. Then, the FASTQ file is now ready for the alignment of reads to the genome, which can be attained using the TopHat program with a reference genome, or any other aligner tool. At this point our pipeline bifurcates, the list of DE genes can be obtained using the Tuxedo pipeline (comprised by Cufflinks, Cuffmerge and Cuffdiff) using FPKM (Fragments Per Kilobase of transcript per Million mapped reads). Alternatively, the list of DE genes can also be obtained using the counts of the reads coming from the HTseq program and then processed by DESEQ2 or Limma/Voom. The biological meaning of the DE gene list can be analyzed gene by gene using "The Human Protein Atlas" and "Genecards" websites or at a system level using the GSEA and/or ToppGene Suite. We discuss each step more in detail below.

3.1 From FASTQ Files to BAM/SAM Files

3.1.1 Raw Data—FASTQ Files

A typical RNAseq data analysis begins with FASTQ files. There is a single FASTQ that corresponds to each RNA sample sequenced. If the study is paired-end sequencing, there are two FASTQ files, each of which corresponds to the left and right reads of each DNA fragment [5]. In the early days of high throughput sequencing, sequenced reads were often stored in the FASTA file format, which is a text file containing a sequence of nucleic acids or amino acids. The FASTQ file is an extension of this file format, i.e., a FASTA that contains both a sequence of nucleic acids and a quality score for that particular sequence. The quality score provides a measure of confidence in the

sequencing data [6]. These files are typically large and easily reaching tens to hundreds of gigabytes in size.

In the FASTQ file, each raw sequence is described with four lines (Fig. 2a). The first line begins with an "@" and contains an identifier for the sequence. The second line contains the raw sequence itself, as found in the FASTA file. The third line starts with a "+" and optionally repeats the content of the first line. The fourth line contains the phred quality score (Q), a measure of the quality of the identification of the base (Fig. 2b) [6]. Phred was originally developed as a quality score for Sanger sequencing data and was adapted in FASTQ. This quality score is calculated by comparing chemical parameters of the given sequencing process with the parameters of a large dataset of known accuracy [6, 7].

In this review, we are using the dataset GSE81167, available from the Gene Expression Omnibus (GEO), for the purpose of illustration. This study evaluated the impact of ZEB1 expression in HCC827 cells, which are human lung cancer cell lines [8]. ZEB1 is one of the principal transcription factors involved in the epithelial-to-mesenchymal transition, a key event in tissue invasion and metastasis [9]. This dataset is available for GEO in the short read archive ".SRA" file format. We used the FASTQ-dump command line program from the SRA Toolkit 2.9.0 to convert files from ".SRA" to ".FASTQ". In most cases, SRA files downloaded from repositories, as from the Gene Expression Omnibus, can be directly converted to SAM files (Sequence Alignment/Map Format), affording savings to both the user's time and central processing unit (CPU) cycles. For didactic reasons, we start from the FASTQ files.

3.1.2 Quality Control of FASTQ Files with FastQC

Although the FASTQ file provides the quality score for each base in every sequence, checking it manually is time-consuming and impractical. FastQC is a program that allows evaluation of the quality of the FASTQ file as a whole. The program uses the FASTQ file as input, and yields three types of FastQC files: A report in the ".html" format, the same report in the compressed ".zip" file, and the "FastQC" folder with the unzipped version of the files [10]. As presented in Fig. 3, the report analyzes a number of items, basic statistics, per-base sequence quality, and other metrics including sequence content, GC content, sequence duplication and the presence of adapter and overrepresented sequences. In our example, even in what can be considered a good report (Fig. Upper Panel), per base sequence content and K-mer content are flagged as potentially problematic. The per-base sequence content suggests sequences with GC-content are over-represented, which might potentially imply contamination from ribosomal RNA (rRNA). K-mer bias occurs when over-represented sequences result in the K-mers derived from these sequences being highly enriched [10]. In our bad report (Fig. 3 Lower Panel), in addition to problems with the quality of the nucleotides sequenced that occur as the read length grows, we have a warning for per tile sequence quality, sequence duplication level and over-represented sequences and a failure for per base sequence content, per sequence GC content and again the K-mer content. Both the warnings and the failures indicate a problem at the library preparation step [10].

3.1.3 Preprocessing—Filtering and Trimming

Once we have the information on sample quality, the next step is to filter out bases and reads with low quality and to extract adaptor, primer, and poly-A tails from the data. To do that, we use the “fastx_trimer” and “fastq_quality_trimmer” programs to filter the adapters, poly-A tails, and PCR primer sequences and “fastq_quality_filter” to filter out low quality bases and/or reads. These programs are a part of the FASTX-Toolkit [11].

By taking the FASTQ files as input, a sequential use of the two FASTX-Toolkit programs, “fastx_trimer” and “fastq_quality_trimmer”, will return a “cleaned” FASTQ file. FASTX-Toolkit can be used only in the command line, which requires some basic knowledge of Linux. The information acquired with the FastQC program must be used in the parameters of the “fastq_quality_filter”, when necessary.

Alternatively, we can use the program “Cutadapt” [12] for trimming adapter sequences, primers, and poly-A tails from the FASTQ file. Cutadapt accepts the FASTQ as input and operates in any system that runs Python. However the program limits itself to the file cleaning and does not support base quality filtering [12].

3.1.4 Preprocessing — Read Alignment to the Genome

We now have good quality reads and a high level of confidence that the original RNA sequence is represented in the FASTQ file. Is it necessary now to align the reads to our reference genome, in this case the Human Genome, version hg38 [13]. We use, TopHat, a part of the Tuxedo pipeline, described by Trapnell and collaborators in 2012 [14]. TopHat is integrated with the Bowtie and SAMtools programs, uses the FASTQ files as input, uses the selected genome as the reference, and generates a SAM file as output. It runs only in the linux shell [14, 15]. To generate the alignment, firstly TopHat uses the Bowtie program to align the reads to the genome. Bowtie is a fast and efficient short read aligner, but unfortunately cannot align reads with large gaps compared with the reference, which makes it unsuitable to align reads that span introns or fusion genes. After the first round of alignment, TopHat breaks the unmapped reads into smaller parts and run a new alignment round using Bowtie. This is one of the key strengths of TopHat, because it permits the identification of splicing variants and fusion genes [14, 15].

The output generate a folder with a number of files and the most important of them are: the “align_summary.txt” with a summary of the alignment, “accepted_hit.bam” with a list of read alignments in the SAM format (Fig. 4), (“.bam” is a “binary SAM”), “unmapped.bam” with the information on the reads that could not be aligned, “deletion.bed”, “insertion.bed”, and “juntions.bed”, with the information described in its name in the Browser Extensible Data (BED) format.

3.2 From BAM/SAM Files to a List of Differentially Expressed Genes

The need to analyze RNAseq data has given rise to a plethora of methods, with different characteristics and assumptions [16]. Here we describe three widely used tools, the Tuxedo pipeline, DESeq2 and Limma/voom [14, 17–19]. Tuxedo employs FPKM (Fragments Per Kilobase Of Exon Per Million Fragments Mapped) in its analysis pipeline [14], while DESeq2 and Limma/voom accept the

counts derived from HTseq analysis as Input [16, 20]. As both HTseq and Cufflinks accept the ".BAM" files, we can use the "accepted_hit.bam" from the previous step [14, 15] as input for these programs.

3.2.1 The TUXEDO Pipeline

The TUXEDO package consists of Cufflinks, Cuffmerge, and Cuffidff programs, more detailed descriptions of these methods are available in the supplementary data of Trapnell et al. [21]. First, the Cufflinks program that uses the ".bam" files from TopHat to assemble the reads into the most probable transcripts and will give a ".gtf" file with the FPKM for the transcripts. Then Cuffmerge takes the ".gtf" files of all the samples and generates a merged ".gtf" file as output. Cuffdiff estimates the differential expression not only of the genes but also of their isoforms and the promoters used (TSS— transcription start site) [14, 21]. The output is tab-delimited text files, including the ".diff" files with the results and other files with information about the analyses, shown in part in Fig. 5. The pipeline also contains the program CummeRBund that can be optionally used to manage, integrate, and visualize the results produced by the Cufflinks package [14].

The Tuxedo Pipeline was for many years the de facto RNAseq analysis pipeline, due to its attractive ability to provide gene expression, isoform variation, and TSS use. Nonetheless, it can only be executed in the Linux command line, has multiple command steps, and require expertise in this environment. Furthermore, it has also been reported that the Tuxedo pipeline has a lower precision and sensitivity than DESeq 2 and Limma/Voom [16, 17].

3.2.2 HTSeq

HTSeq uses ".bam" files from the samples and a ".gtf" or ".gff" file with the gene models as input. It is scripted in Python and the "HTSeq-count" script is able to count how many aligned reads overlap with the exons of the genes, not considering differential splicing. Reads that align with more than one gene are discarded, excluding fusion genes. The output is a tab-delimited text ".txt" file with two columns: one with the gene name and one with the counts (Fig. 6) [20].

The "count" value is exactly as described. It is a measure of how many reads for that gene exist in the ".bam" files. For the nature of the analysis, the "count" is nonnegative integer valued. A read can be counted as belonging to a gene (+1 in the "count" value) or not [20]. This value is not corrected or normalized in any way by HTSeq and downstream analysis will have to address this issue. At this point, it is interesting to note that the same sample sequenced with different depths will have different count values for each gene simply because of the different quantity of reads generated by the sequencer. This issue needs to be considered when working with samples from different origins and/or from multiple datasets [20, 22]. It is possible to introduce a prefiltering step at this point, i.e., excluding the genes that have zero counts in all samples, to reduce the needed computation time and improve the statistical power as this step can affect the false discovery rate control [23].

3.2.3 DESeq2

The DESeq2 package uses unnormalized count data, such as that provided from HTseq, as input. The program internally corrects for library size, so raw counts needs to be used in order to run the analyses [24]. To run the analyses, it is necessary to create a DESeqDataSet object, using the ".txt" counts files and an user-specified design matrix, i.e., assignment of samples to different treatment groups [24]. At this point, the "DESeq" function is invoked to perform the differential analysis. The output is a table containing the gene symbol, the base 2 log-transformed fold change, the pvalue, and the adjusted p-value (or q-value, a measure of false discovery) (Fig. 7). DESeq2 assumes a negative binomial linear model to describe over-dispersed count data from RNAseq data. It uses an empirical Bayes method for more robust and accurate estimation of parameters for dispersion and fold change, by taking into account small numbers of replicates and low read counts in RNAseq data. Finally, DESeq2 uses the Wald test to estimate significance of differential expression, and the Benjamini—Hochberg correction to control the false discovery rate [17, 24].

3.2.4 Limma/Voom

Limma is one of the well-known R packages used for differential gene expression analysis. It was initially developed for microarray analyses, prior to the emergence of RNAseq, and has been updated to facilitate analysis of RNASeq data [25]. Limma uses linear models to specify the experimental design, empirical Bayes method to moderate the standard errors between genes, and uses the t-test to calculate the differential expression p-value, while providing multiple choices for adjustment of the p-value for multiple testing, including the Benjamini—Hochberg [19, 25]. The use of Limma is well suited for small numbers of samples per group, as few as two, and also powerful when used in multifactor designed tests [16, 25]. It has often reported that DESeq2 and Limma have similar precision and sensitivity in their analysis results [16, 17].

As with DESeq2, Limma accepts the ".txt" count files from HTseq as input but internally transforms the data because Limma assumes the t-distribution for gene expression values as it was originally developed for microarray datasets [25, 26]. In order to address this issue, Limma uses the Voom transformation, i.e., log transformation of counts per million (cpm) with associate precision weights [18]. The Voom transformation is the key step in this analysis, as its log transformation helps gene expression values satisfy the t-distribution assumption of Limma, while using cpm instead of raw count normalizes gene expression values across replicates [18, 25]. Once we have the transformed table, we use the "lmFit" function to fit the data to a linear model, which informs the design of the experiment as in the case of DESeq2. The next step is to call the "eBayes" function that will calculate the moderated t-statistics and log-odds of the differential expression using an empirical Bayes moderation. The output is a table containing the gene symbol, the fold change in log scale with base 2, the p-value, and the adjusted p-value as shown in Fig. 8.

3.3 From a List of Differentially Expressed Genes to Systems Level Analyses

The final output of the Tuxedo pipeline, DESEQ2 and Limma/voom is a list of differentially expressed genes [14, 24, 25]. Depending on the situation and the established cutoff, the length of this list can

vary from tens to thousands genes. In this chapter we consider the analysis comparing two groups of experimental conditions but this can also be easily extended to multiple groups as well. The simplest way to compare the genes from different experimental conditions is using Venn diagrams, which permits to assess what is in common and what is unique in the gene lists of your differential expression analysis [27]. In addition, when only a handful of genes are of interest, deeper interrogation of each gene can be also implemented using Genecards and The Human Protein Atlas [28, 29]. Nonetheless, most of the time, the differentially expressed gene list might be too long for such gene-by-gene analyses and as a result, system level analyses might often be more appropriate, for example, by using GSEA and/or Toppfun [30, 31].

3.3.1 Venn Diagrams — Venny

Area-proportional Venn diagrams are a useful graphic approach to compare different analyses. BioVenn is a convenient web application for the comparison and visualization of biological lists [27]. For example, we can visualize how the gene lists generated by different DE analysis programs described above are related to each other (Fig. 9). On the other hand, Venn diagrams can also be used to compare the results of two different comparisons between groups and/or experimental conditions. As we can see, the way we analyze our data influence the results. If information on one specific gene being differentially expressed in a given condition is needed, it is often necessary to validate it with qPCR [1].

3.3.2 Genecards and the Human Protein Atlas

Once in possession of a list of ranked DE genes, the next task is to make sense of it in a biological context. One possibility is to check the DE gene list on a gene by gene basis, or at least the top ranked genes in the list, for their function and expression in different tissues. For this step there are two web services that can be utilized: Genecards, a repository with information of gene and protein function, expression and known interactions [28] (Fig. 10); and The Human Protein Atlas, with information on RNA and protein levels in different tissue and cancer types, and immunostaining data from images derived from tissue histology [29] (Fig. 11). Although this method is useful to analyze a small list of genes, or the function of the top ranked genes in a list, it is unpractical when we have a list of hundreds or thousands of DE genes. In this case, a system level analysis is required.

3.3.3 Gene Set Enrichment Analysis (GSEA)

One of the key assumptions of the gene set enrichment analysis is that a moderate increase in expression of a large number of genes encoding members of a pathway may dramatically alter the flux through the pathway and may be more important than a huge increase in a single gene from that pathway [30, 32]. GSEA is available as online and desktop versions. Both use the same molecular signature database, comprised of eight collections: The Hallmark Gene sets (H), the Positional gene set (C1), the Curated gene sets (C2), the Motif gene sets (C3), the Computational

gene sets (C4), the Gene Ontology gene sets (C5), The Oncogenic signatures (C6), and the Immunologic signatures (C7) [30, 32]. The desktop version accepts a ".txt" tab-delimited table as input, although the Limma/voom transformed table is preferable, as the program was originally designed to analyze log2 values from microarray data. Count values obtained directly from HTseq are also accepted as input, although in this case it is important to check if the data needs to be normalized [33]. In the basic analyses, the samples in the table are separated into two groups, test group against control group. GSEA determines the enrichment of gene sets using a modified Kolmogorov–Smirnov test. Specifically, it calculates the Enrichment Score (ES) taking into consideration if a gene set is over represented at the top or at bottom of the ranked list, and it estimates the p-values using empirical phenotype-based permutation test procedure, along with their FDR [30, 33]. The output contains the table of gene sets that are positively and negatively correlated with the groups, with ES, p values, and FDR values. In addition, GSEA also provides a graphical representation of the analysis, containing a heatmap comparing the two groups in the context of the gene set and an enrichment plot (Fig. 12). We note that in the GSEA analysis, it is important to provide the complete transcriptome as a whole as input, rather than a previously selected list of genes (e.g., from Limma/voom). The desktop version also accepts a preranked list of genes as input for the analyses. In contrast, the online version only accepts a list of previously selected genes as input, and tests the enrichment of gene sets using the hypergeometric test. Specifically, it evaluates if any of the gene sets are over-represented in the provided gene list, and generates a downloadable table with the statistically significant gene sets, the p-value, and the overlap between the gene sets and the provided gene list (Fig. 13).

3.3.4 ToppGene Suite

ToppGene suite is a web-based service, with four functionalities: ToppFun, ToppGene, ToppNet, and ToppGenet [31]. ToppFun (1), similar to the GSEA web based tool, accepts a gene list as input and provides a downloadable table with the enriched pathways as output. Additionally it also generates tables for 14 annotation categories including GO terms, pathways, protein–protein interaction, microRNAs, and related diseases [31] (Fig. 14). ToppGene (2) takes two gene lists as input, i.e., "training gene set" and "test gene set." In our case, the "test gene set" is the DE gene list while the "training gene set" is a gene list containing geneof our interest. Given these gene lists, ToppGene will run a ToppFun analyses of the "training gene set" to identify the most notable characteristics of this list, and rank the genes in the "test gene set" according to these characteristics. For example, if the "training gene set" is formed by the genes involved in Angiogenesis, the test genes will be ranked based on their relevance to Angiogenesis. Similarly, if the "training gene set" is formed by membrane proteins, the test genes will be ranked based on the degree they are related, or directly interact, with membrane proteins. ToppGene uses all 14 annotation categories to analyze the "training gene set" and rank the "test gene set" based on all of these 14 annotation categories [31].

ToppNet (3) and ToppGenet (4) compare "training gene set" with "test gene set" and construct networks of interactions. Specficially, ToppNet is based only on protein–protein interaction (PPI), while ToppGenet uses both PPI and the genes in the neighborhood in order to take in account possible indirect interaction.

4 Conclusions

In this chapter, we described how to analyze RNAseq from raw data to a list of genes and related systems and pathways. RNAseq has gained popularity with the emergence of high-throughput sequencing (HTS) or next-generation sequencing (NGS). These advances in conjunction with improvements in Proteomics and Metabolomics techniques and related analytical methods have introduced a paradigm shift in biomedical research. This big data landscape was unimaginable just a few short years ago [34].

This postgenomics era has enabled a new way of thinking, as it becomes clearer and clearer that the events occurring in cells and tissues are much more complex than the sum of their parts [35]. The vast amount of data, from diverse origins, and the need for integration can be overwhelming. Thus it is imperative that the researcher have a clear idea of the question that is being asked, the data and concepts that are used to formulate a hypothesis, and the assumptions that statistical methods are based on, and the inherent limitations of these methods.

Acknowledgments

This work was conducted with support from start-up funds from the MUSC COM and an award from SC Epscor to GH. The content is solely the responsibility of the authors and does not necessarily represent the official views of the Medical University of South Carolina.

References

1. Alberts B, Johnson A, Lewis J et al (2015) Analyzing cells, molecules, and systems. In: Molecular biology of the cell, 6th edn. Garland Science, USA, pp 440–528

2. Garrett CT (2015) Molecular biology basics inthe “Omics” Era: genes to proteins. In: Idowu OM, Dumur IC, Garrett TC (eds) Molecular oncology testing for solid tumors: a pragmatic approach. Springer International Publishing, Cham, pp 3–65

3. Wang Z, Gerstein M, Snyder M (2009) RNA-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet 10(1):57–63

4. Mayer-Scho¨nberger V, Cukier K (2013) Big data: a revolution that will transform how we live, work, and think. Harcourt, Houghton Mifflin

5. Sengupta S, Bolin JM, Ruotti V et al (2011) Single read and paired end mRNA-Seq Illumina libraries from 10 nanograms total RNA. J Vis Exp (56):e3340. https://doi.org/10.3791/3340

6. Cock PJ, Fields CJ, Goto N et al (2010) The Sanger FASTQ file format for sequences with quality scores, and the Solexa/Illumina FASTQ variants. Nucleic Acids Res 38 (6):1767–1771

7. Illumina (2011) Quality scores for nest generation sequencing: assessing sequencing accuracy using phred quality scoring. https://www.illumina.com/Documents/products/technotes/technote_Q-Scores.pdf

8. Zhang T, Guo L, Creighton CJ et al (2016) A genetic cell context-dependent role for ZEB1 in lung cancer. Nat Commun 7:12231

9. Lenferink AE (2017) Epithelial-to-Mesenchymal transition (EMT): the good, the bad, and the ugly. In: Wang E (ed) Cancer systems biology. CRC Press, Florida

10. Andrews S (2010) FastQC: a quality control tool for high throughput sequence data. http://www.bioinformatics.babraham.ac.uk/ projects/fastqc

11. Gordon A, Hannon G (2010). Fastx-Toolkit. In: FASTQ/A short-reads preprocessing tools. http://hannonlab.cshl.edu/fastx_toolkit/

12. Martin M (2011) Cutadapt removes adapter sequences from high-throughput sequencing reads. EMBnetjournal 17(1):10–12

13. Speir ML, Zweig AS, Rosenbloom KR et al (2016) The UCSC genome browser database: 2016 update. Nucleic Acids Res 44(D1): D717–D725

14. Trapnell C, Roberts A, Goff L et al (2012) Differential gene and transcript expression analysis of RNA-seq experiments with TopHat and cufflinks. Nat Protoc 7(3):562–578

15. Trapnell C, Pachter L, Salzberg SL (2009) TopHat: discovering splice junctions with RNA-Seq. Bioinformatics 25(9):1105–1111

16. Soneson C, Delorenzi M (2013) A comparison of methods for differential expression analysis of RNA-seq data. BMC Bioinformatics 14:91

17. Love MI, Huber W, Anders S (2014) Moderated estimation of fold change and dispersion for RNA-seq data with DESeq2. Genome Biol 15(12):550

18. Law CW, Chen Y, Shi W et al (2014) Voom: precision weights unlock linear model analysis tools for RNA-seq read counts. Genome Biol 15(2):R29

19. Smyth GK (2005) Limma: linear models for microarray data. In: Gentleman R, Carey VJ, Huber W, Irizarry RA, Dudoit S (eds) Bioinformatics and computational biology solutions using R and bioconductor. Statistics for biology and health. Springer, New York, NY

20. Anders S, Pyl PT, Huber W (2015) HTSeq—a python framework to work with highthroughput sequencing data. Bioinformatics 31(2):166–169

21. Trapnell C, Williams BA, Pertea G et al (2010) Transcript assembly and quantification by RNA-Seq reveals unannotated transcripts and isoform switching during cell differentiation. Nat Biotechnol 28(5):511–515

22. Conesa A, Madrigal P, Tarazona S (2016) A survey of best practices for RNA-seq data analysis. Genome Biol 17(1):181

23. Bourgon R, Gentleman R, Huber W (2010) Independent filtering increases detection power for high-throughput experiments. Proc Natl Acad Sci U S A 107(21):9546–9551

445 24. Love M, Anders S, Huber W (2014) Differential analysis of count data–the DESeq2 package.
446 Genome Biol 15(12):550

447 25. Ritchie ME, Phipson B, Wu D et al (2015) Limma powers differential expression analyses for RNA-
448 sequencing and microarray studies. Nucleic Acids Res 43(7):e47

449 26. Datta S, Nettleton D (2014) Statistical analysis of next generation sequencing data. In: Datta S,
450 Nettleton D (eds) Frontiers in probability and the statistical sciences. Springer International
451 Publishing, Switzerland, pp 1–32

452 27. Hulsen T, De Vlieg J, Alkema W (2008) BioVenn—a web application for the comparison and
453 visualization of biological lists using areaproportional Venn diagrams. BMC Genomics 9:488

454 28. Stelzer G, Dalah I, Stein TI et al (2011) In-silico human genomics with GeneCards. Hum Genomics
455 5(6):709–717

456 29. Uhlen M, Oksvold P, Fagerberg L (2010) Towards a knowledge-based human protein atlas. Nat
457 Biotechnol 28(12):1248–1250

458 30. Subramanian A, Tamayo P, Mootha VK et al (2005) Gene set enrichment analysis: a knowledge-
459 based approach for interpreting genome-wide expression profiles. Proc Natl Acad Sci U S A
460 102(43):15545–15550

461 31. Chen J, Bardes EE, Aronow BJ et al (2009) ToppGene suite for gene list enrichment analysis and
462 candidate gene prioritization. Nucleic Acids Res 37(Web Server): W305–W311

463 32. Mootha VK, Lindgren CM, Eriksson KF et al (2003) PGC-1alpha-responsive genes involved in
464 oxidative phosphorylation are coordinately downregulated in human diabetes. Nat Genet 34(3):267–
465 273

466 33. Gene Set Enrichment Analysis (GSEA) User Guide (2010), http://software.broadinstitute.
467 org/gsea/doc/GSEAUserGuideFrame.html

468 34. Wold B, Myers RM (2008) Sequence census methods for functional genomics. Nat Methods
469 5(1):19–21

470 35. Alon U (2006) An introduction to systems biology: design principles of biological circuits. CRC
471 Press, Florida

472

473

474

475

476

477

478

479

480

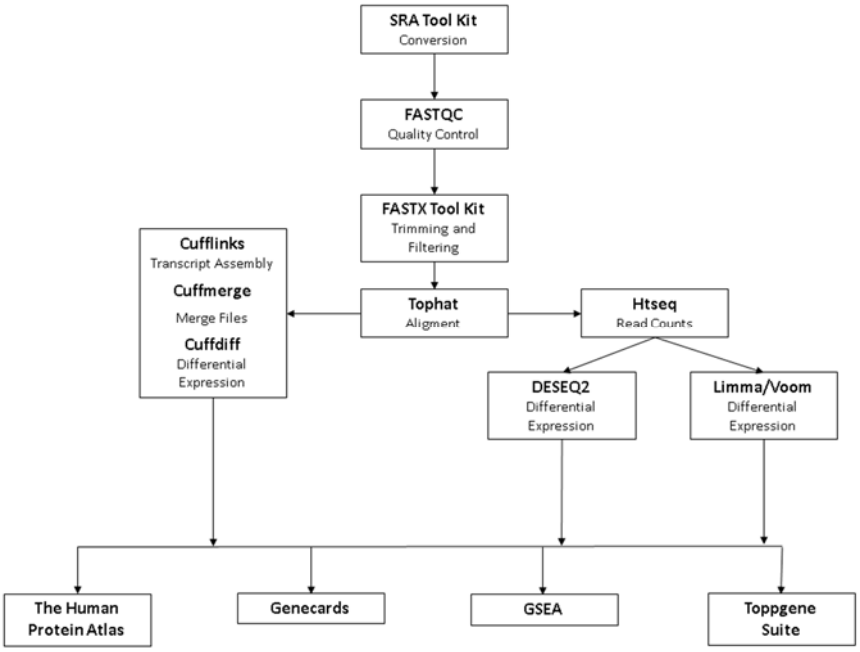


Fig. 1 Flowchart summarizing analyses from RNA seq data. The conversion step with SRA tool kit is generally only used with data from repositories

A

```
@SRR3478269.1 1 length=102
GGAAATTTAAAAAAAATACACATGGCCAGGCCCCAGCCCAAATCACTAATAACCCGGAGCATATGCTGTTTGGTCTCAGTAGGCTCCTAAATATGGGATTCCT
+SRR3478269.1 1 length=102
CCCFFFFFHFFHHJIIJJJIIIJJIJJJJIIJJJJJJJJIJIJIJJGJJJJCCCFFDFFHHDHFGIGIIJJJIIJIFHGHIJIJGGIEHJJFIJIJIIIJJJ
```

B

```
!"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNOPQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz{|}~
```

Fig. 2 FASTQ Example. Sample SRR3478269 from Dataset GSM2144086. (a) First 4 lines of the FASTQ file of the Sample SRR3478269 from Dataset GSM2144086. (b) FASTQ quality score in ascending order from the left to right

513

514 Fig. 3 FastQC report. Example FastQC report for a good quality sample (Upper Panel) and a poor
515 quality sample (Lower Pane

516

```
@CO	PI: Yang_Yanan_m107100
@CO	RLIMS ID: 1415404747 RLIMS Project ID: NGSLBP-1412503 Project IRB: N/A
@CO	Read Type: PAIRED_END Application Type: MRNAS Capture Kit: N/A
@CO	RUN: 140903_SN7001164_0280_BC5FDPACXX Lane: 1 Index: CTTGTA Flowcell: HiSeq Flow Cell v3 Sbs: TruSeq SBS Kit v3 Pe: TruSeq PE Cluster Kit v3 - cBot - HS
@CO	HCS: 2.0.12.0 RTA: 1.17.21.3 FPGA 3.10.3 CPLD 3.0.14 CASAVA v1.8.2
1	163	chr1	11478	0	51M	=	11690	263	CCCGGAGCATATGCTGTTTGGTCTCAGTAGGCTCCTAAATATGGGATTCCT	CCCFFDFFHHDHFGIGIJJJIIJIFHGHIJIJGGGIEHJJFIJIJIIIJJJ	RG:Z:HCC827_3-1_a
1	83	chr1	11690	0	51M	=	11478	-263	TTATTAGTGATTTGGGCTGGGGCCTGGCCATGTGTATTTTTTTAAATTTCC	JJJJGJJIJIJIJJJJJJJJJIIJJJJIJJIIIJJJIIJHHFFHFFFFCCC	RG:Z:HCC827_3-1_a
2	99	chr1	11887	0	51M	=	11966	130	TTTTCTTTGACCTCTTCTTTCTGTTCATGTGTATTTGCTGTCTCTTAGCCC	@@@FFFFFDFDHAFHIGBHIEBHHDHBHD>CCCEHGHFHHEDFHHHGGEG?D	RG:Z:HCC827_3-1_a
2	147	chr1	11966	0	51M	=	11887	-130	TTTGAGAGGTCACAGGGTCTTGATGCTGTGGTCTTCATCTGCAGGTGTCTG	BCEHAGFGEFD?B?G@HGBGHFCBGIIHFBBC4DGBEFBFHFFDDBBD8??	RG:Z:HCC827_3-1_a
3	163	chr1	12138	0	51M	=	12222	135	CTGCATGTAACTTAATACCACAACCAGGCATAGGGGAAAGATTGGAGGAAA	?BEFFFFEHHHHHHJJJJJIJJJJIJJJIJGHHIJJJGIIIIIHJJJGIIJIH	RG:Z:HCC827_3-1_a
3	83	chr1	12222	0	51M	=	12138	-135	AGGCCAGTAAGTAGTGCTTGTGCTCATCTCCTTGGCTGTGATACGTGGCCG	IJIIJJJJJIHEIJIIJGIHCIIHGIHIIJJJIGHF?BHHGFFFFFCCC	RG:Z:HCC827_3-1_a
4	163	chr1	12858	0	51M	=	13012	205	TGCTGCTTGCGGTGGCCTATAAAGCCTCCTAGTCTGGCTCCAAGGCCTGGC	?:@?=DD=C=F?OB::CG@ACH@H<:FF?CGD??FFGIDBD<?BFBFB:BC	RG:Z:HCC827_3-1_a
4	83	chr1	13012	0	51M	=	12858	-205	GGTGGAGAACCTGTGCATGAAGGCTGTCAACCAGTCCATAGGCAAGCCTGG	FB<FFBDD:?3>BC@D93EF@C<FEB;CA<;@>FEC::4FF;<DDDDD:<@	RG:Z:HCC827_3-1_a
5	163	chr1	13032	15	51M	=	13151	170	AGGCTGTCAACCAGTCCATAGGCAAGCCTGGCTGCCTCCAGCTGGGTCGAC	?=?DDDDAB?DHADBHBGH?CCEC>BCCBDFHGKEBHDFQDDDFGH@?A@@	RG:Z:HCC827_3-1_a
5	83	chr1	13151	0	51M	=	13032	-170	GGAAGGAGAAGGGGATGCACTGTTGGGGAGGCAGCTGTAACTCAAAGCCTT	HGJJIGHEIIIIGGEDGBCGHCIGHFIIJJIIJIJIHFDAFDHFFFDD@B?	RG:Z:HCC827_3-1_a
6	99	chr1	13545	0	51M	=	13655	161	GTCATGGGCCTGGTCTGCAGGGATCCTGCTACAAAGGTGAAACCCAGGAGA	B@BDFFFFHHHHHIIIJIJIJJHGJJHIJGHIJJJJJ:BBFHHIIGJIJJG	RG:Z:HCC827_3-1_a
6	147	chr1	13655	0	51M	=	13545	-161	ACAGGGGAATCCCGAAGAAATGGTGGGTCTTGGCCATCCGTGAGATCTTCC	JJIJJJJIHDGJIGIHGIHGGGGHIIIGCGGJJIIGJJHHGHHFFDFFBC@	RG:Z:HCC827_3-1_a
12	99	chr1	14412	0	51M	=	14542	181	CAGTTCTTTATTGATTGGTGTGCCGTTTTCTCTGGAAGCCTCTTAAGAACA	???=A?DDF4DF?<CGFEDFFEEFIFGII@GCGFIIIEFEFFGGBDFFFCF	RG:Z:HCC827_3-1_a
11	163	chr1	14415	0	51M	=	14539	175	TTCTTTATTGATTGGTGTGCCGTTTTCTCTGGAAGCCTCTTAAGAACACTG	CCCFFFFFHHHHHJIHIIJIJJJJJJJJJJJJJJJIJJJJIJJIIJJJJJJ	RG:Z:HCC827_3-1_a
8	99	chr1	14417	0	51M	=	14497	131	CTTTATTGATTGGTGTGCCGTTTTCTCTGGAAGCCTCTTAAGAACACTGTG	BCCFDDFDFHHGHHGHHHHHHIJJGIHIJJJJJJJIJIIJJJJIIJJJJGI	RG:Z:HCC827_3-1_a
10	163	chr1	14433	0	51M	=	14615	233	GCCGTTTTCTCTGGAAGCCTCTTAAGAACACAGTGGCGCAGGCTGGGTGGA	CCCFFFFFFHHHGHJJJJIIIJJJJJJJJIIIJGHIIJJJJIJGJJJ=FHI	RG:Z:HCC827_3-1_a
9	163	chr1	14433	0	51M	=	14497	115	GCCGTTTTCTCTGGAAGCCTCTTAAGAACACAGTGGCGCAGGCTGGGTGGA	CCCFFDFFFHHHHIJJIJGIIJIIGGIJGIJGHIIIIGIJJIJIIFHII	RG:Z:HCC827_3-1_a
26	163	chr1	14443	0	51M	=	14646	254	CTGGAAGCCTCTTAAGAACACAGTGGCGCAGGCTGGGTGGAGCCGTCCCCC	CC@FFFFDHHHHHIGIJJJIJIIHIHIJIJIJJJ9@F;CCHGGIJHGH	RG:Z:HCC827_3-1_a
10	99	chr1	14449	0	51M	=	14510	120	GCCTCTTAAGAACACAGTGGCGCAGGCTGGGTGGAGCCGTCCCCCCATGGA	CCCFFFFFHHHHHHJJJJJJFIGIIGDHGIJHIIEHJIIJJGHGIFEFFEE	RG:Z:HCC827_3-1_a
13	99	chr1	14451	0	51M	=	14545	145	CTCTTAAGAACACAGTGGCGCAGGCTGGGTGGAGCCGTCCCCCCATGGAGC	CCCFFDFEHHHHHHJJGEHIGIHIJJDIIIDFECDHII@FEHEGH8?CEFDC	RG:Z:HCC827_3-1_a
7	99	chr1	14457	0	51M	=	14493	87	AGAACACAGTGGCGCAGGCTGGGTGGAGCCGTCCCCCCATGGAGCACAGGC	CCCFFFFDHCFHGJJJIGIJJJJCFGGGGIJJIJGIJHHHHHFFFEEDD	RG:Z:HCC827_3-1_a
17	99	chr1	14460	0	51M	=	14604	187	GCGCAGGCTGGGTGGAGCCGTCCCCCCATGGAGCACAGGCAGACAAAAGTC	CC@FFFFFHFHH?FH?FGID@FHIIIII@FHGIGEHHCGEEHEHGFFF@>C	RG:Z:HCC827_3-1_a
```

Fig. 4 Example SAM file. First several lines of the SAM file from Sample SRR3478269 from Dataset GSM2144086.

gene_id	gene	locus	Group_1	Group_2	status	value_1	value_2	log2(fold_change)	test_stat	p_value	q_value
XLOC_000126	EFHD2	chr1:15736390-15756839	Control	ZEB1	OK	202.415	95.5723	-1.08265	-2.95758	5.00E-05	0.000518
XLOC_000221	HMGN2	chr1:26798901-26803133	Control	ZEB1	OK	161.569	235.038	0.540743	1.63467	0.0131	0.0473428
XLOC_000351	CDC20	chr1:43824625-43828873	Control	ZEB1	OK	225.636	318.984	0.499487	1.62614	0.0133	0.0478431
XLOC_000370	RPS8	chr1:45241245-45244412	Control	ZEB1	OK	2517.32	1969.36	-0.354161	-1.67685	0.01245	0.045781
XLOC_000377	UROD	chr1:45477804-45481341	Control	ZEB1	OK	71.9628	180.767	1.32881	1.89676	0.0012	0.0079017
XLOC_000456	PGM1	chr1:64058946-64125916	Control	ZEB1	OK	142.789	88.7964	-0.685316	-1.38816	0.00725	0.0320071
XLOC_000512	CYR61	chr1:86046443-86049648	Control	ZEB1	OK	701.992	521.97	-0.427489	-1.96439	0.0023	0.0135386
XLOC_000730	PSMB4	chr1:151372040-151374412	Control	ZEB1	OK	254.892	435.399	0.772454	2.2223	0.00425	0.0207689
XLOC_000956	QSOX1	chr1:180123967-180169859	Control	ZEB1	OK	264.486	123.342	-1.10053	-2.63871	5.00E-05	0.000518
XLOC_000970	LAMC2	chr1:183155173-183214262	Control	ZEB1	OK	177.744	13.4283	-3.72645	-4.96459	5.00E-05	0.000518
XLOC_001020	ELF3	chr1:201979689-201986315	Control	ZEB1	OK	123.944	28.7934	-2.10589	-3.44096	5.00E-05	0.000518
XLOC_001074	G0S2	chr1:209848669-209849735	Control	ZEB1	OK	291.39	16.4758	-4.14453	-2.88957	0.00595	0.0278503
XLOC_001153	GALNT2	chr1:230193535-230417876	Control	ZEB1	OK	99.1635	167.39	0.755334	1.55569	0.00275	0.0147362
XLOC_001311	ENO1	chr1:8921058-8939943	Control	ZEB1	OK	2012.97	2502.59	0.314096	1.46414	0.01085	0.0417349
XLOC_001572	SLC2A1	chr1:43391045-43449029	Control	ZEB1	OK	281.788	139.861	-1.01062	-3.46746	5.00E-05	0.000518
XLOC_001573	EBNA1BP2	chr1:43629844-43720029	Control	ZEB1	OK	124.382	212.616	0.773473	1.53116	0.0117	0.0437062
XLOC_001594	PRDX1	chr1:45965855-45988562	Control	ZEB1	OK	528.434	802.64	0.60303	1.88551	0.0008	0.006035
XLOC_001609	PDZK1IP1	chr1:47649260-47655771	Control	ZEB1	OK	358.718	37.0944	-3.27358	-3.42087	0.0076	0.0328067
XLOC_001672	JAK1	chr1:65210777-65432187	Control	ZEB1	OK	116.975	62.8851	-0.895414	-2.55947	5.00E-05	0.000518
XLOC_001756	F3	chr1:94994731-95007413	Control	ZEB1	OK	295.327	72.2413	-2.03142	-4.48758	5.00E-05	0.000518
XLOC_001933	S100A10	chr1:151955385-151966714	Control	ZEB1	OK	337.267	168.961	-0.997202	-2.53059	0.0017	0.0104833

Fig. 5 Cuffdiff output. Top 20 differentially expressed genes from Cuffdiff analysis

Symbol	Counts
A1BG-AS1	1
AAAS	3
AACS	2
AAGAB	5
AAK1	1
AAMP	4
AAR2	1
AARS	5
AASS	1
AATF	3
ABCA1	10
ABL2	11
ACTG1	205
CEP41	1
EEF1A1	419
GAPDH	453
TTC3	12
TTYH3	12
TUBA1B	200
TUBA1C	202

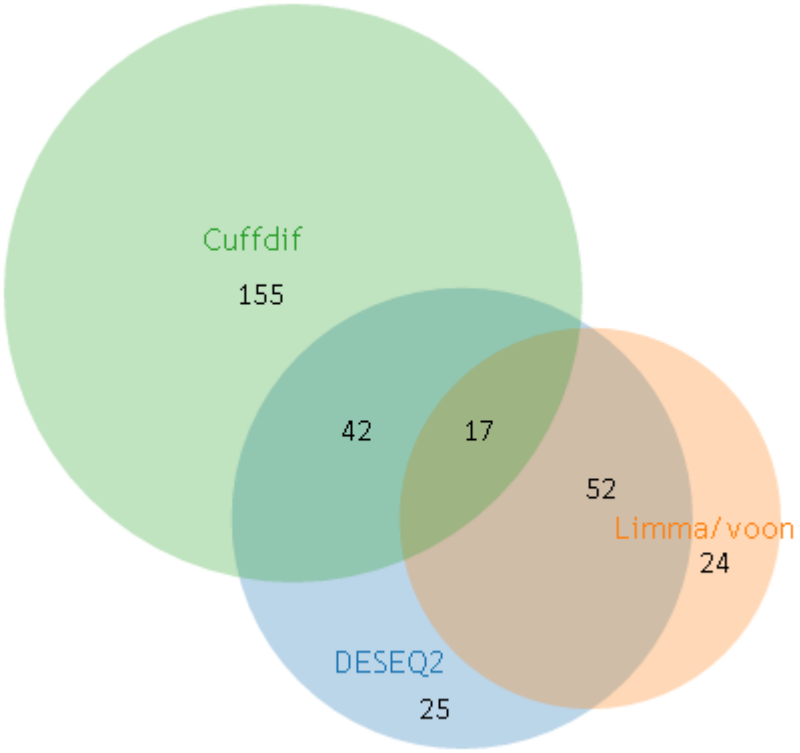
Fig. 6 HTSeq output. Example of count results from sample GSM2144086

Symbol	GeneID	Count_Control_1	Count_Control_2	Count_Control_3	Count_ZEB1_1	Count_ZEB1_2	Count_ZEB1_3	baseMean	log2FoldChange	lfcSE	stat	pvalue	padj
KRT7	3855	150	164	131	25	23	27	88.32087747	-2.719804965	0.21121846	-12.8767	6.09E-38	2.97E-34
FN1	2335	85	68	75	19	10	17	46.51829	-2.440348331	0.27796973	-8.77919	1.65E-18	4.02E-15
ACTG1	71	222	205	190	85	116	118	155.1586327	-1.136943225	0.13453789	-8.45073	2.89E-17	4.71E-14
MT2A	4502	41	34	20	112	139	128	74.02538642	1.783708582	0.21425014	8.325355	8.41E-17	1.03E-13
LAMC2	3918	50	43	49	4	1	4	25.99212881	-3.852052972	0.46562369	-8.27289	1.31E-16	1.28E-13
KRT18	3875	121	126	101	46	48	43	81.07128371	-1.514812939	0.18871538	-8.02697	9.99E-16	8.13E-13
CXCL1	2919	54	42	28	3	0	0	21.87722077	-4.679926755	0.60298745	-7.76123	8.41E-15	5.87E-12
LCN2	3934	40	51	29	0	1	1	20.97514876	-4.914503581	0.63920592	-7.68845	1.49E-14	9.09E-12
JUP	3728	45	32	35	4	4	2	20.96486972	-3.393701917	0.46634748	-7.2772	3.41E-13	1.66E-10
TNFAIP2	7127	34	42	41	4	3	6	22.22526049	-3.150144598	0.43211452	-7.29007	3.10E-13	1.66E-10
LAMB3	3914	42	28	30	1	2	1	17.94658771	-4.17558093	0.58055798	-7.19236	6.37E-13	2.83E-10
VIM	7431	11	14	10	59	70	64	35.42369692	2.206368812	0.30934126	7.132475	9.86E-13	4.01E-10
CDH1	999	22	38	34	1	1	0	16.59529808	-4.596894063	0.65133486	-7.05765	1.69E-12	6.36E-10
KRT8	3856	115	131	129	69	58	64	94.06276097	-1.152540615	0.1702325	-6.77039	1.28E-11	4.48E-09
LCP1	3936	21	34	24	0	0	0	13.64243193	-5.136608256	0.76190433	-6.7418	1.56E-11	5.09E-09
SAA1	6288	25	29	23	1	0	0	13.4714644	-4.685365375	0.70570744	-6.63925	3.15E-11	9.62E-09
PIP4K2C	79837	94	85	81	37	43	40	63.29305063	-1.287364914	0.20737209	-6.208	5.37E-10	1.54E-07
PLAU	5328	55	55	34	11	15	15	31.04926848	-1.936850937	0.31515463	-6.14572	7.96E-10	2.16E-07
CXCL2	2920	10	20	27	0	0	0	9.925612632	-4.723188071	0.78409378	-6.02375	1.70E-09	4.38E-07
PLEC	5339	70	87	64	36	30	30	52.83260945	-1.365602851	0.23073187	-5.91857	3.25E-09	7.93E-07

Fig. 7 DESeq2 output. Top 20 differentially expressed genes from DESeq2 analyzes

Symbol	logFC	AveExpr	t	P.Value	adj.P.Val	B
LCP1	-5.85418	6.573509	-16.9375	2.28E-07	0.001111	7.125779
FAM83A	-4.33212	5.812479	-13.5217	1.22E-06	0.002822	5.874455
ESRP1	-4.31032	5.801578	-12.7464	1.90E-06	0.002822	5.520751
CXCL2	-5.31476	6.303797	-12.0713	2.83E-06	0.002822	5.186808
COL4A2	-4.89651	6.094674	-11.781	3.38E-06	0.002822	5.035188
IGFN1	3.975878	5.77204	11.56598	3.87E-06	0.002822	4.919587
COL4A1	-3.67745	5.485143	-11.419	4.25E-06	0.002822	4.838877
LAMB3	-4.36474	7.659404	-10.9862	5.64E-06	0.002822	4.593108
CYLD	3.482421	5.525311	10.80534	6.36E-06	0.002822	4.486671
LCN2	-5.38404	7.395079	-10.7332	6.68E-06	0.002822	4.443573
LRRC6	-3.58095	5.436892	-10.6768	6.94E-06	0.002822	4.409643
NR5A2	3.939291	5.753746	10.40548	8.37E-06	0.002822	4.243081
CGN	-3.58095	5.436892	-10.3526	8.68E-06	0.002822	4.210012
ECH1	3.402085	5.485143	10.35119	8.69E-06	0.002822	4.209112
SAA1	-5.31258	6.831029	-10.2951	9.04E-06	0.002822	4.173766
GALNT3	-3.40411	5.348473	-10.263	9.24E-06	0.002822	4.153441
SCN9A	3.225246	5.396724	9.984322	1.13E-05	0.003079	3.973631
CDA	-3.70996	5.501396	-9.97544	1.13E-05	0.003079	3.967801
TNFRSF10A	-3.28325	5.288044	-9.67495	1.41E-05	0.003625	3.766785
SPTLC3	3.583743	5.575972	9.609286	1.48E-05	0.003625	3.72187
CDH3	-3.45332	7.44935	-9.45097	1.67E-05	0.00386	3.612084

Fig. 8 Limma/Voom output. Top 20 differentially expressed genes from Limma/Voom analyses

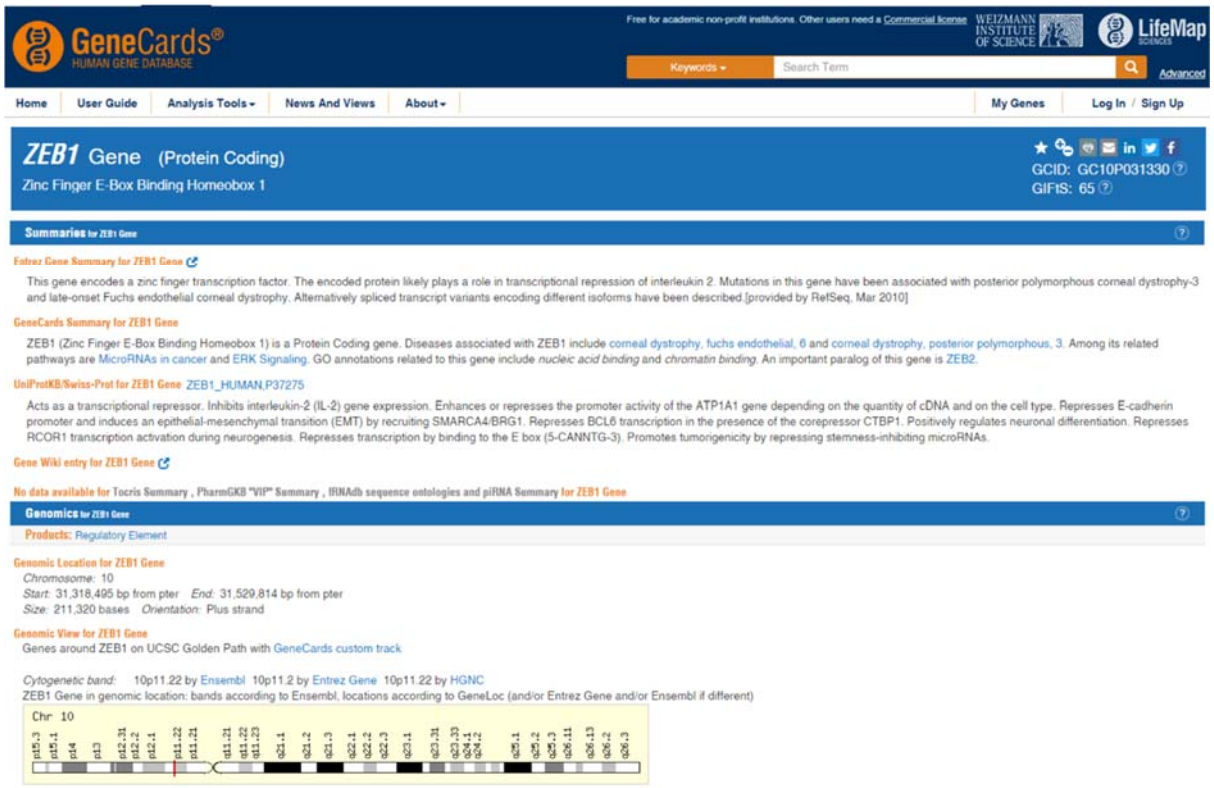


569

570 Fig. 9 Venny output. Proportional Venn diagram comparing the list of differentially expressed genes
571 uncovered by Cuffdiff, DESeq2 and Limma/Voom, where differential expression was determined at
572 adjusted p-value of < 0.05 for all the three methods

573

574



575

576 Fig. 10 ZEB1 edited description in Genecards

577

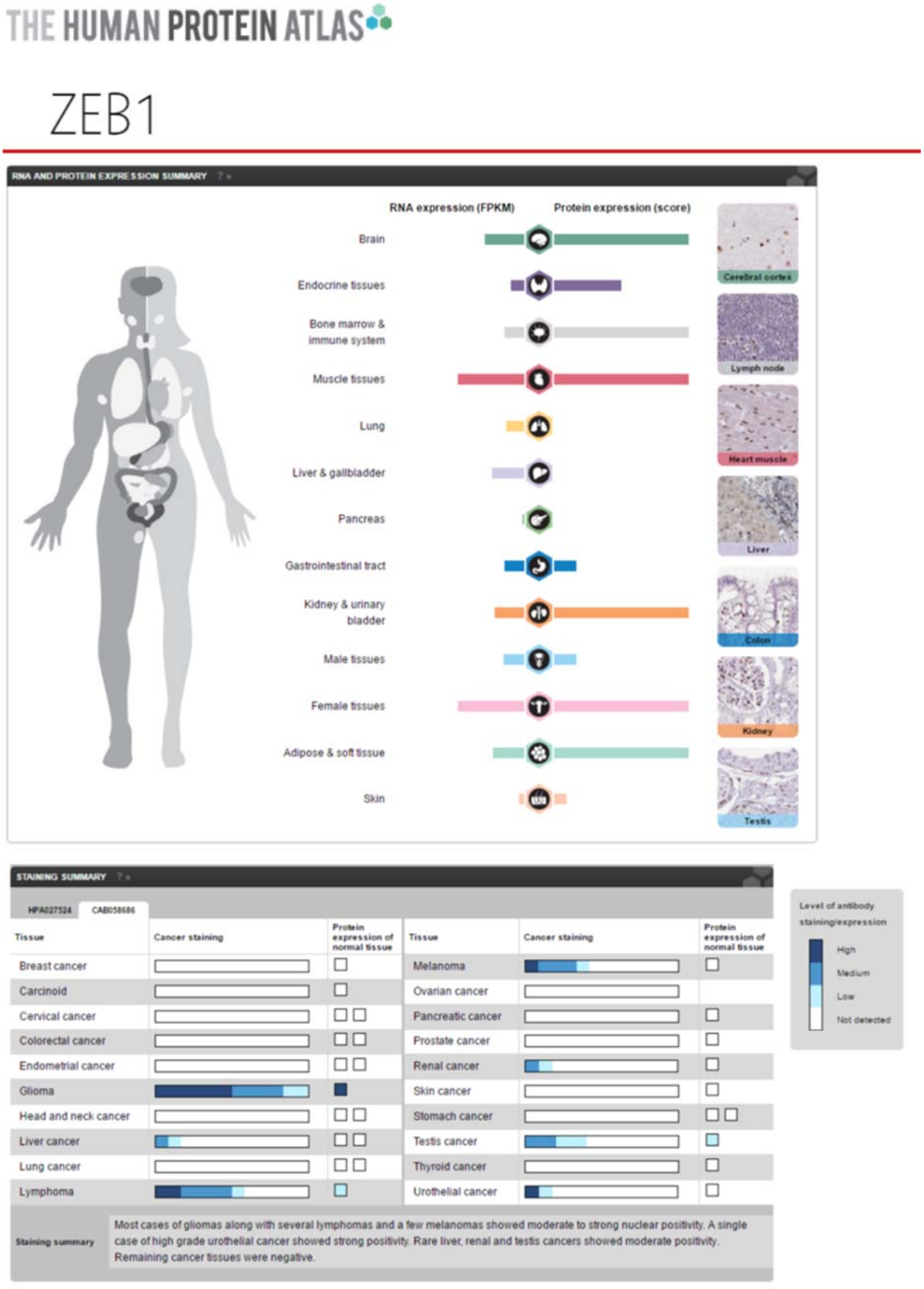


Fig. 11 ZEB1 edited description in The Human Protein Atlas website

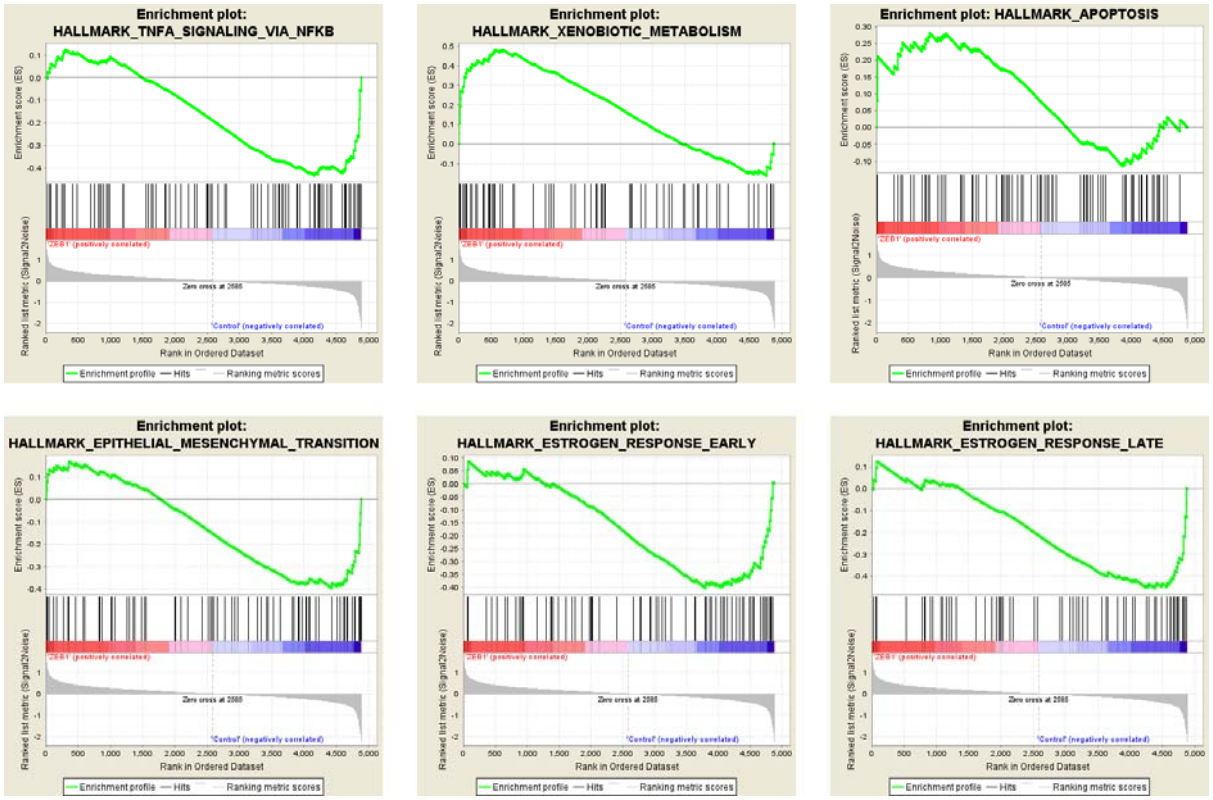


Fig. 12 Results from GSEA desktop version (using the transformed table from Limma/Voom as input)

Gene Set Name [# Genes (K)]	Description	# Genes in Overlap (k)	k/K	p-value	FDR q-value
HALLMARK_EPITHELIAL_MESENCHYMAL_TRANSITION [200]	Genes defining epithelial-mesenchymal transition, as in wound healing, fibrosis and metastasis.	14		$1.19\ e^{-15}$	$5.96\ e^{-14}$
HALLMARK_TNFA_SIGNALING_VIA_NFKB [200]	Genes regulated by NF-kB in response to TNF [GeneID=7124].	12		$8.77\ e^{-13}$	$2.19\ e^{-11}$
HALLMARK_ESTROGEN_RESPONSE_EARLY [200]	Genes defining early response to estrogen.	10		$4.44\ e^{-10}$	$5.55\ e^{-9}$
HALLMARK_XENOBIOTIC_METABOLISM [200]	Genes encoding proteins involved in processing of drugs and other xenobiotics.	10		$4.44\ e^{-10}$	$5.55\ e^{-9}$
HALLMARK_ESTROGEN_RESPONSE_LATE [200]	Genes defining late response to estrogen.	9		$8.53\ e^{-9}$	$8.53\ e^{-8}$
HALLMARK_ANDROGEN_RESPONSE [101]	Genes defining response to androgens.	7		$2.09\ e^{-8}$	$1.75\ e^{-7}$
HALLMARK_APOPTOSIS [161]	Genes mediating programmed cell death (apoptosis) by activation of caspases.	8		$2.73\ e^{-8}$	$1.95\ e^{-7}$
HALLMARK_CHOLESTEROL_HOMEOSTASIS [74]	Genes involved in cholesterol homeostasis.	6		$8.64\ e^{-8}$	$5.4\ e^{-7}$
HALLMARK_APICAL_JUNCTION [200]	Genes encoding components of apical junction complex.	8		$1.46\ e^{-7}$	$6.63\ e^{-7}$
HALLMARK_HYPOXIA [200]	Genes up-regulated in response to low oxygen levels (hypoxia).	8		$1.46\ e^{-7}$	$6.63\ e^{-7}$

Fig. 13 GSEA web tool Analysis. Top 10 Hallmarks gene sets enriched for the DE gene list obtained from DESeq2

1: GO: Molecular Function [Display Chart] 557 annotations before applied cutoff / 18661 genes in category

	ID	Name	Source	pValue	FDR B&H	FDR B&Y	Bonferroni	Genes from Input	Genes in Annotation
1	GO:0005198	structural molecule activity		8.892E-8	4.953E-5	3.418E-4	4.953E-5	21	762
2	GO:0008201	heparin binding		2.937E-6	8.181E-4	5.645E-3	1.636E-3	9	167
3	GO:0045236	CXCR chemokine receptor binding		5.301E-6	9.841E-4	6.791E-3	2.952E-3	4	17
4	GO:0005200	structural constituent of cytoskeleton		1.312E-5	1.827E-3	1.261E-2	7.310E-3	7	110
5	GO:0005539	glycosaminoglycan binding		2.609E-5	2.906E-3	2.006E-2	1.453E-2	9	219

Show 22 more annotations

2: GO: Biological Process [Display Chart] 3331 annotations before applied cutoff / 18623 genes in category

	ID	Name	Source	pValue	FDR B&H	FDR B&Y	Bonferroni	Genes from Input	Genes in Annotation
1	GO:0060429	epithelium development		2.597E-10	8.652E-7	7.517E-6	8.652E-7	32	1296
2	GO:0042127	regulation of cell proliferation		5.978E-10	9.956E-7	8.650E-6	1.991E-6	36	1666
3	GO:0040012	regulation of locomotion		1.500E-9	1.500E-6	1.303E-5	4.996E-6	25	866
4	GO:0040011	locomotion		1.801E-9	1.500E-6	1.303E-5	5.999E-6	36	1735
5	GO:0048870	cell motility		2.897E-9	1.609E-6	1.398E-5	9.651E-6	32	1428

Show 45 more annotations

3: GO: Cellular Component [Display Chart] 360 annotations before applied cutoff / 19061 genes in category

	ID	Name	Source	pValue	FDR B&H	FDR B&Y	Bonferroni	Genes from Input	Genes in Annotation
1	GO:0005925	focal adhesion		3.882E-12	7.361E-10	4.758E-9	1.397E-9	20	393
2	GO:0005924	cell-substrate adherens junction		4.888E-12	7.361E-10	4.758E-9	1.760E-9	20	398
3	GO:0030055	cell-substrate junction		6.134E-12	7.361E-10	4.758E-9	2.208E-9	20	403
4	GO:0005912	adherens junction		2.181E-11	1.963E-9	1.269E-8	7.853E-9	21	484
5	GO:0070161	anchoring junction		4.466E-11	3.216E-9	2.079E-8	1.608E-8	21	503

Show 45 more annotations

589

590 Fig. 14 Results from TOPPFUN. Gene Ontology terms enriched for the gene list obtained from
591 DESEQ2 are presented