Supplementary Materials
of
Analysis of Single-Cell RNA-Sequencing Data: A Step-By-Step Guide

Aanchal Malhotra[1-2,¶], Samarendra Das[1-3,¶] and Shesh N. Rai[1,2,4,5,6,7,8*]

[1]School of Interdisciplinary and Graduate Studies, University of Louisville, Louisville, KY 40292, USA

[2]Biostatistics and Bioinformatics Facility, JG Brown Cancer Center, Louisville, KY 40202, USA

[3]Division of Statistical Genetics, ICAR-Indian Agricultural Statistics Research Institute, New Delhi 110012, India

[4]Hepatobiology and Toxicology Center, University of Louisville, Louisville, KY 40202, USA

[5]Biostatistics and Informatics Facility, Center for Integrative Environmental Health Sciences, University of Louisville, Louisville, KY 40202, USA

[6]Christina Lee Brown Envirome Institute, University of Louisville, Louisville, KY 40202, USA

[7]Department of Bioinformatics and Biostatistics, School of Public Health and Information Science, University of Louisville, Louisville, KY 40202, USA

[8]Data Analysis and Sample Management Facility, University of Louisville Super Fund Center, University of Louisville, Louisville, KY 40202, USA

Authors' email addresses:

SD: samarendra.das@louisville.edu

SNR: shesh.rai@louisville.edu

AM: aanchal.malhotra@louisville.edu

***To whom correspondence should be addressed-** email: shesh.rai@louisville.edu; Telephone: +1 502-426-0016

Raw Data Download

The following command was used to download through the SRA toolkit on a local Linux OS based machine.

wget http://ftp-trace.ncbi.nlm.nih.gov/sra/sdk/2.10.2/sratoolkit.2.10.2-centos_linux64.tar.gz https://trace.ncbi.nlm.nih.gov/Traces/sra/?run=Run_ID or the data can be downloaded using the command: */fastq-dump --split-files Run_ID (e.g., Run_ID* = SRR6782109*).*

```
@ML-P2-14:9:000H003HG:1:11102:17290:1073 1:N:0:TCCTGAGC+GCGATCTA
TTTGGTAACAGCATGAATTATTCTAGCCACTAAAACTCTATGAACATCTTGTGAAGGTTTCAGATAGAGCCTGAAGTACACAGAGAACAATTCTTAAAAAA
+
AAAAAEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE<AEEEEEEE
```

Figure S1. Glimpse of the Fastq file.

Quality Control

The "FASTQ Quality Check" tool is an easy way to perform quality control. The quality control analysis is done by nine modules, which provide a quick overview of whether the data looks good, and there are no issues that may hinder downstream analysis. The result is an HTML file with evaluations in the form of charts. The file begins with the quality statistics table for data. It contains information about the input FASTQ file, type of quality score encoding, the total number of reads, read length, and GC content (Figure S2).

Basic Statistics

Measure	Value
Filename	SRR6782109_2.fastq
File type	Conventional base calls
Encoding	Sanger / Illumina 1.9
Total Sequences	109178700
Sequences flagged as poor quality	0
Sequence length	98
%GC	48

Figure S2: Basic summary statistics of raw sequence data (fastq file).

The first plot, 'Per base Sequence Quality' (Figure S3), is a box-and-whisker plot showing quality score distribution over all the sequences on the x-axis and the observed mean quality on the y-axis. The red line in the center of the box and whisker plot gives the median value, the yellow box of the plot represents the inter-quartile range (25-75%), the upper and lower whiskers of the plot represent the 10% and 90% percentile scores, and the blue line represents the mean quality of the read. The background of the graph divides the y-axis into excellent quality calls (green), reasonable quality (orange), and calls of low quality (red). A warning is flagged if the median quality for any base is less than 27, while the failure occurs if the median quality falls below 20. The screenshot of this plot for our dataset shows good quality reads with median quality above 30 for most of the reads and all the reads in the green area indicating good base calls.

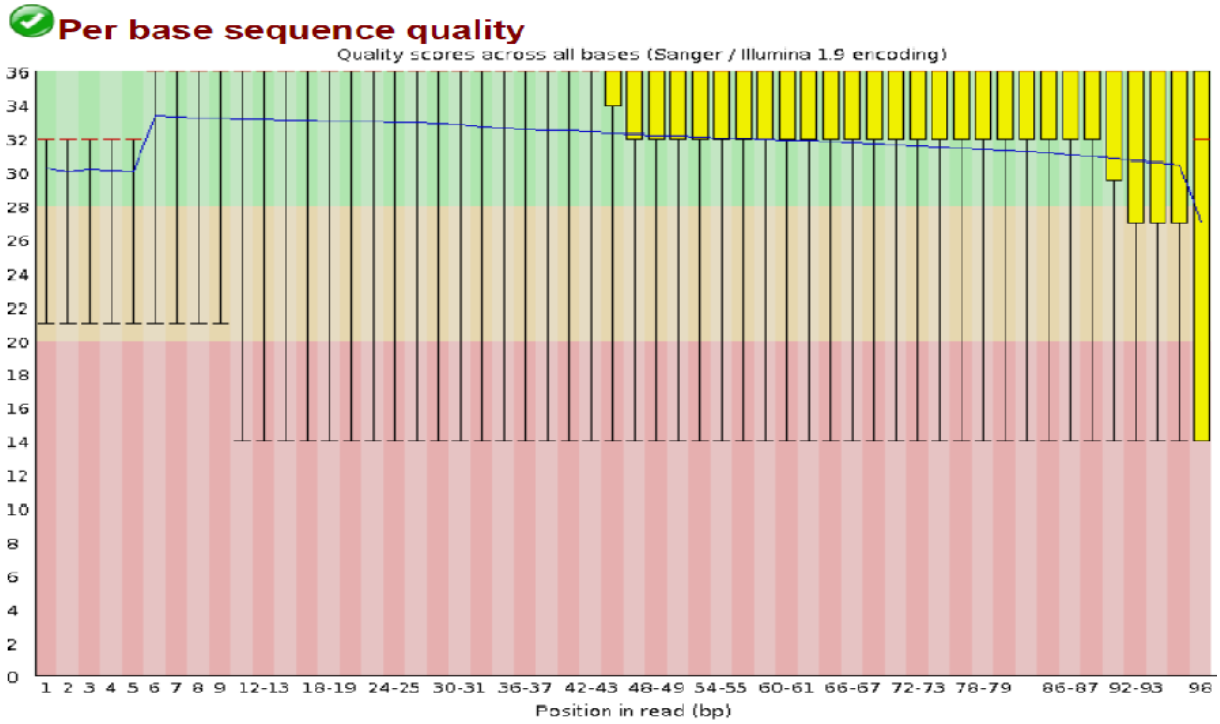

Figure S3: Per base sequence quality plot (depicting the quality of the reads in the fastq file).

The second plot, 'Per sequence quality score' (Figure S4), gives the distribution of the average quality score. This quality report allows a check if a subset of sequences has low-quality values universally. There is a problem with the run if a significant proportion of the sequences in a run have overall low quality. If the observed mean quality is below 27, which equals to a 0.2% error rate, a warning is issued. A failure is flagged when the mean quality falls below 20, which equates to a 1% error rate. What we look for in the plot is a tight distribution towards one end of the plot, which would indicate good quality. We observe that the frequently observed mean quality is around 31 for our dataset, indicating good quality read data. This module generates the following plot for our dataset:

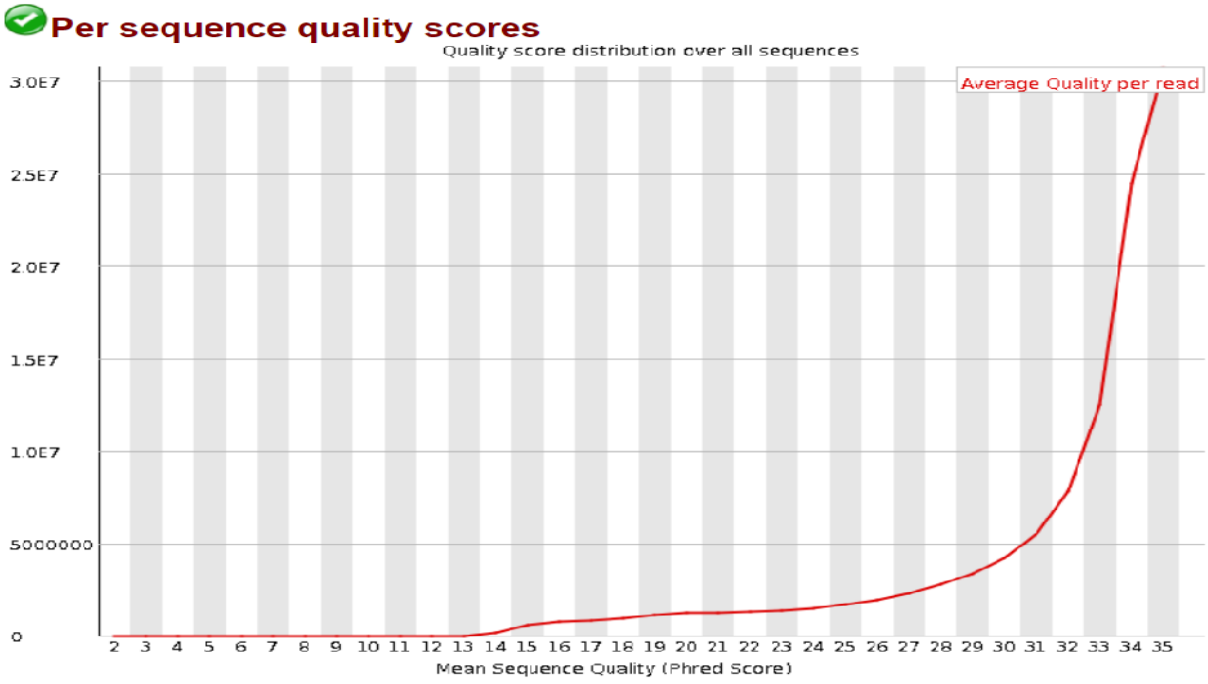

Figure S4: Distribution of per sequence quality scores.

The next plot, 'Per base sequence content' (Figure S5), gives the percentage of bases called for each of the four nucleotides at each position across all reads in the read file. For any sequencing run, the proportion of all nucleotides, A, T, G, C, is expected to be in equal proportion indicating that the lines in this plot should run parallel with each other. With most RNA-Seq library preparation methods, there is a non-uniform distribution of bases for the first 10-15 nucleotides. This non-uniform distribution is normal and expected, depending on the type of library kit which is used in the library preparation. This module marks this distribution as Failed by FastQC even though the sequence is perfectly acceptable. If the difference between A and T, or G and C is greater than 10% in any position, a warning is given, and if more than 20%, it raises a failure.

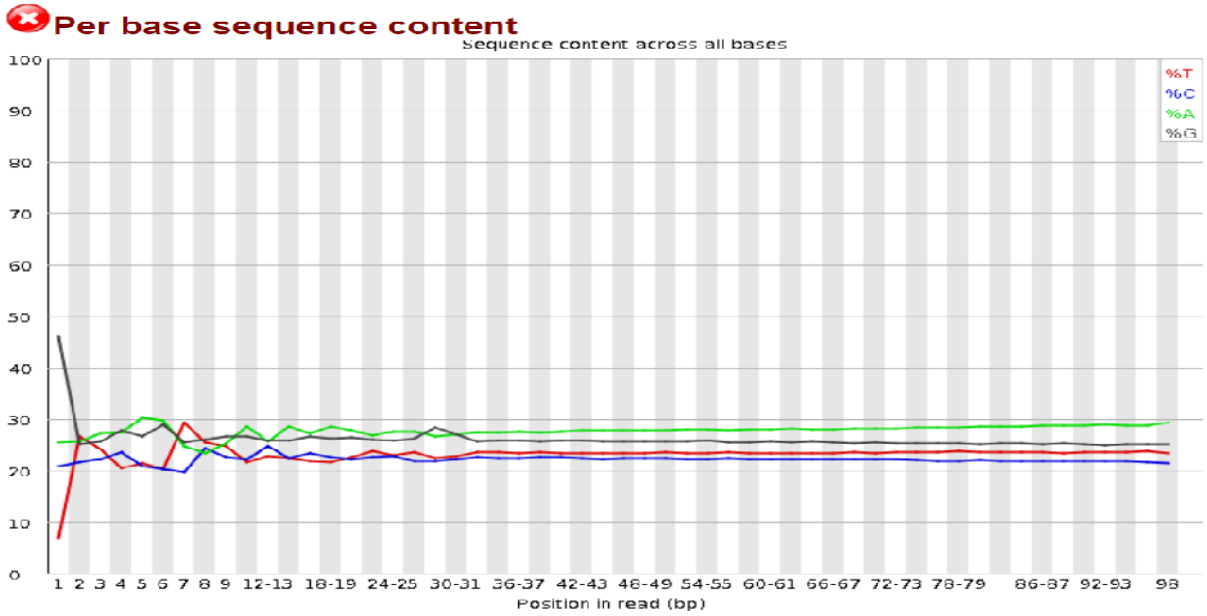

Figure S5: Distribution of per base sequence content.

The following Figure S6 is of the sequence GC content and plots the number of reads vs. GC percentage per read. There may be a greater or lesser distribution of mean GC content among transcripts. A roughly normal distribution of GC content is expected where the peak corresponds to the overall GC content. An anomaly can lead to the observed plot to be broader or narrower than the ideal plot of the normal distribution, indicating contamination. If the sum of the deviations from the normal distribution represents more than 15% of the reads, a warning is issued while a fail is raised for more than 30% reads. For our dataset, the GC content is 48%.

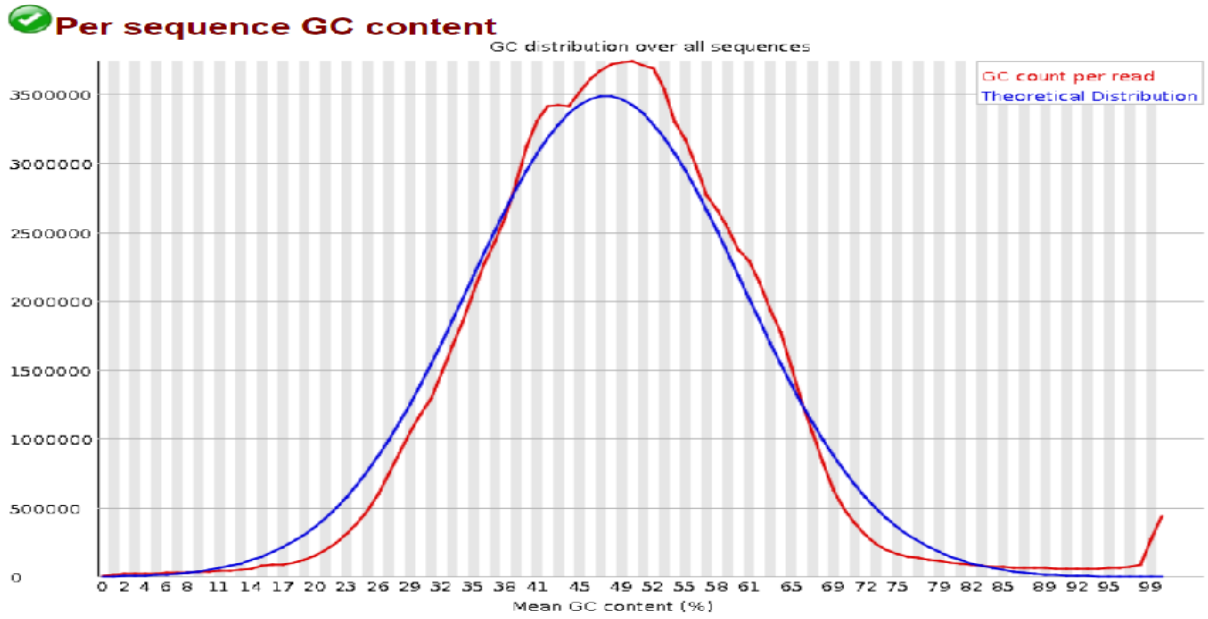

Figure S6: Distribution of per sequence GC content.

The next plot,' per base N content,'(Figure S7) gives the percentage of bases at each position for which N is the base call. If a sequencer cannot call a base with confidence, it substitutes an N rather than a conventional base. If this curve rises noticeably above zero, it indicates a problem that occurred during the sequencing run. If a position in the plot shows an N content of >5%, it's a warning, and for >20%, it's a failure. The following is the screenshot of our dataset.

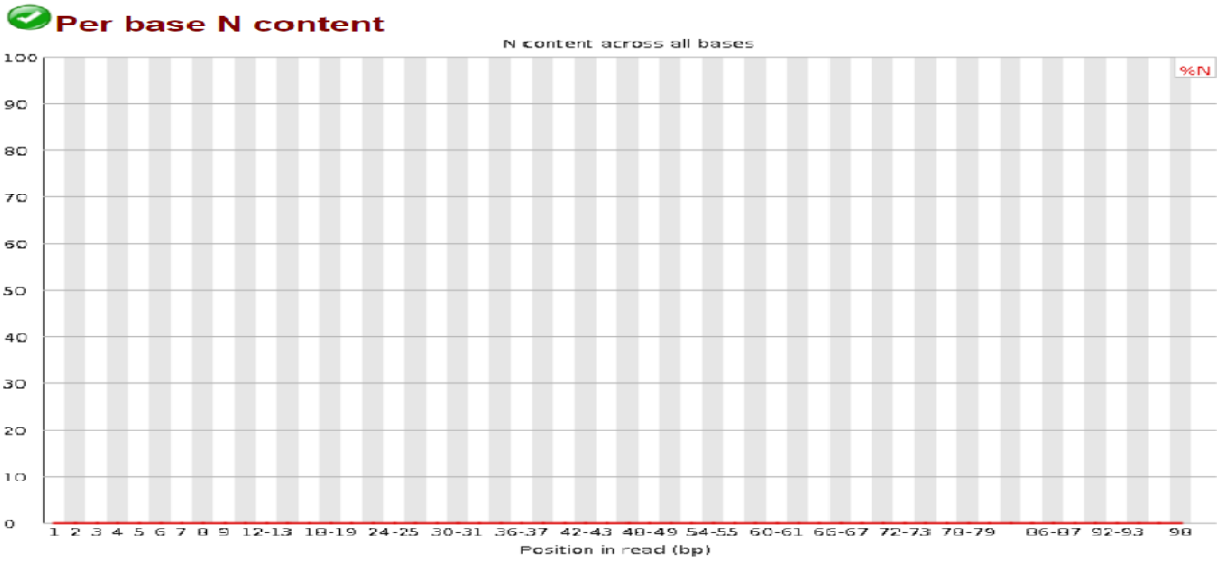

Figure S7: Distribution of per base N content.

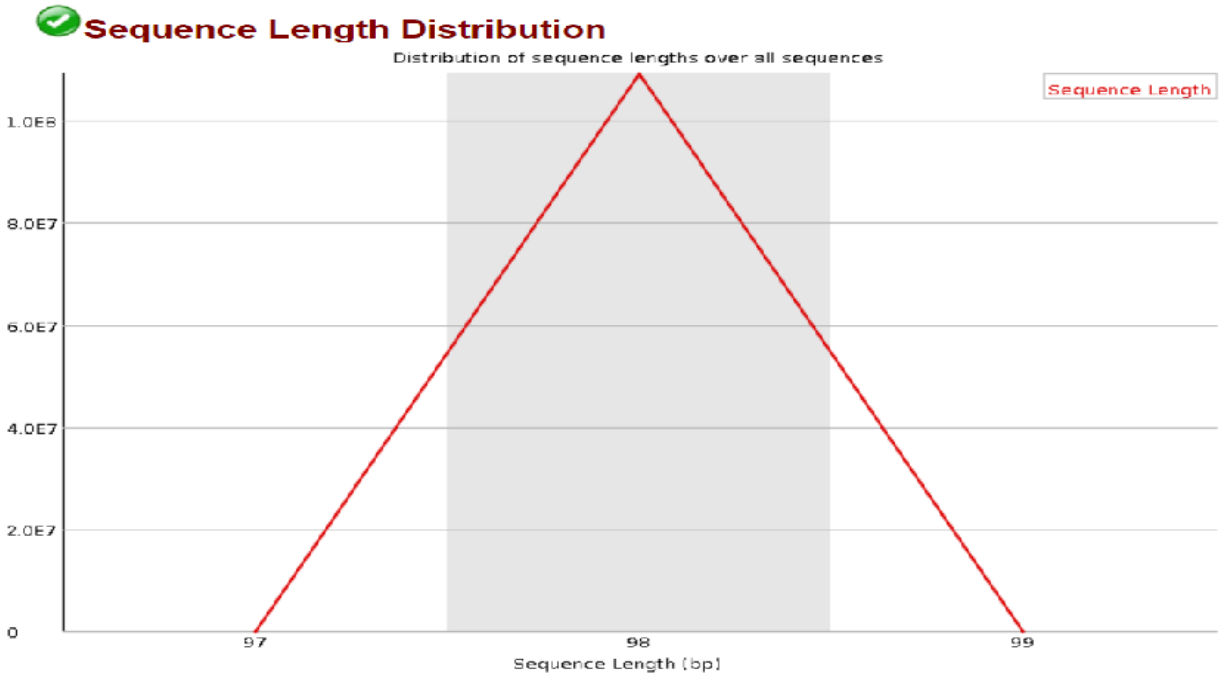

Figure S8: Sequence length distribution plot.

The plot is for the sequence length distribution (Figure S8). It gives the distribution of fragment sizes in the file. This module will produce a simple graph showing a peak only at one height. A warning occurs if all sequences are not the same length and failure is due to an error leading to the sequences with zero length. For our dataset, we get a peak of the curve at 98, depicting that all sequences are 98 bases long.

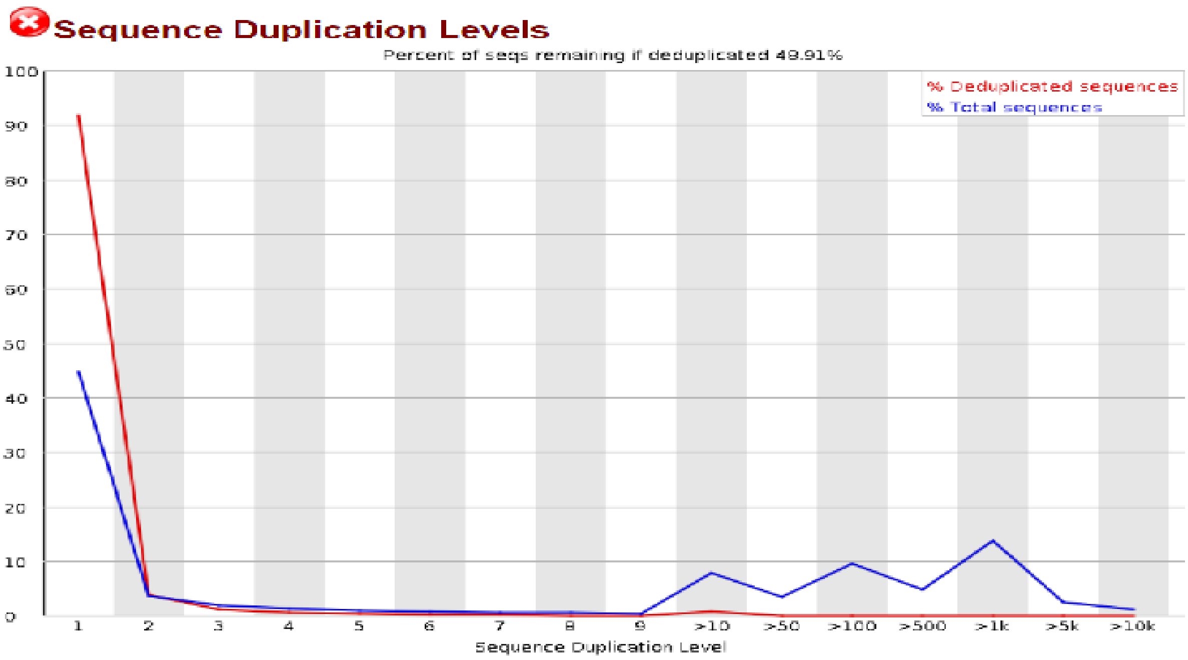

Figure S9: Sequence duplication levels plot.

The sequence duplication level plot (Figure S9) shows the relative number of sequences with different degrees of duplication. A low duplication level is indicative of a higher level of coverage of the file, while a higher recurrence indicates a bias usually from the PCR amplification. There are two lines on the plot. The blue line takes the full sequence set and shows how its duplication levels are distributed in the original data. The red line shows the duplication proportions of the deduplicated group. For RNA-seq, there will be some very highly abundant transcripts for which duplicate reads are expected and is always flagged. If the duplicated sequences are more than 20% of the total, a warning is issued, while for more than 50%, it is a failure.

Overrepresented sequences

Sequence	Count	Percentage	Possible Source
GGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGGG	674042	0.6173750008014384	No Hit

Figure S10: Chart of the overrepresented sequences.

The overrepresented sequence lists the sequence, which appears more than expected in the file (Figure S10). An overrepresented sequence in the file either means that it has some biological significance or indicates a contaminated library or not a diverse library as expected. In the case of a typical high-throughput library, it will contain different sequences, and no one sequence that makes up a small fraction of the whole set of sequences. A sequence is considered overrepresented if it is accounted for ≥ 0.1% of the total reads for which a warning is raised and a failure if it is >1%. For our dataset, the over-represented sequence is of poly Gs constituting approximately 0.6% of the total with no possible link to contamination or primers.

The last plot for our data quality control is the' Adapter content'(Figure S11). The graph shows the percentage of adapter content in each position. It is always useful to know if the library contains an adapter in order to be trimmed for downstream analysis. Our dataset is free of the primers as seen in the plot below:

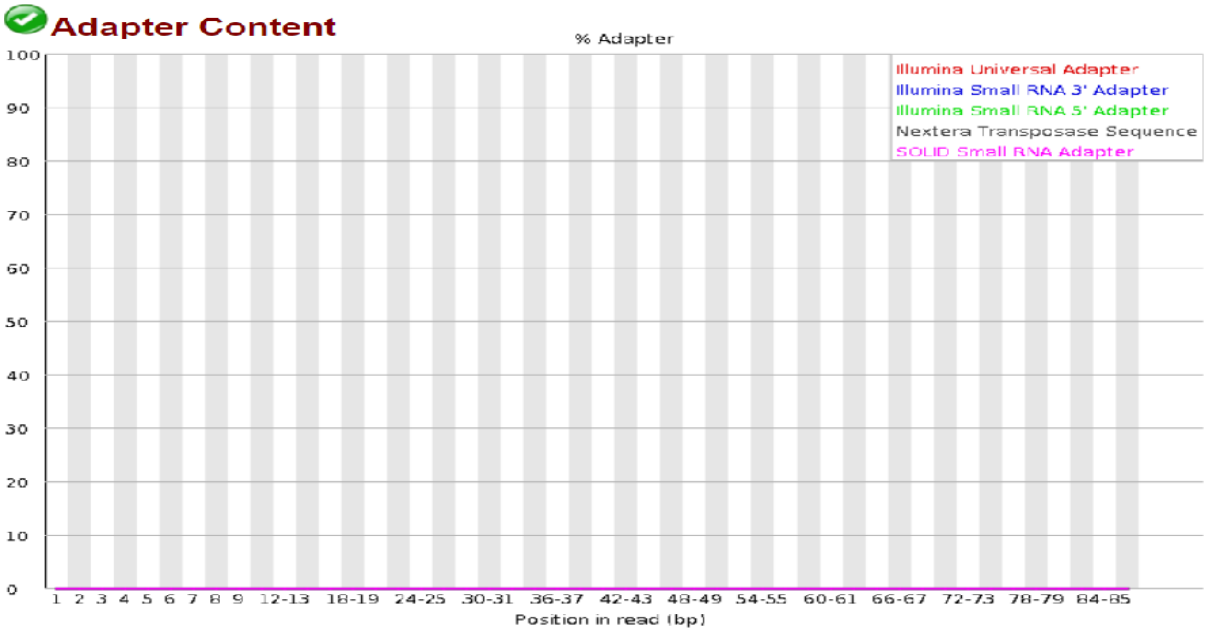

Figure S11: Distribution of percentage of Adapter content.

Document S1: Linux commands for various analyses.

Extracting Barcodes and UMI

The following command from the UMI-tool outputs a text file containing the list.

umi_tools whitelist --stdin read1(R1)file.*fastq –bc-pattern="*Pattern sequence*"* *--log2stderr > whitelist.txt*

where, read1(R1).fastq is the read file e.g., SRR6782109_1.fastq; Pattern sequence is the given pattern sequence, e.g., *CCCCCCCCCCCCCCCCNNNNNNNNN.*

The whitelist file that is generated as follows:

```
AAACCTGCAGACACTT        AAAACTGCAGACACTT,AAACATGCAGACACTT,AAACCAGCAGACACTT,AAACCCGCAGACACTT,AAACCGGCAGACACTT,AA
CTGCAGAAACTT,AAACCTGCAGACAATT,AAACCTGCAGACACAT,AAACCTGCAGACACCT,AAACCTGCAGACACGT,AAACCTGCAGACACTA,AAACCTGCAGACA
T,AAACCTGCAGACGCTT,AAACCTGCAGACTCTT,AAACCTGCAGAGACTT,AAACCTGCAGATACTT,AAACCTGCAGCCACTT,AAACCTGCAGGCACTT,AAACCTG
GACACTT,AAACCTNCAGACACTT,AAACCTTCAGACACTT,AAACGTGCAGACACTT,AAACTTGCAGACACTT,AAATCTGCAGACACTT,AACCCTGCAGACACTT,A
CCTGCAGACACTT,GAACCTGCAGACACTT,NAACCTGCAGACACTT,TAACCTGCAGACACTT        77199   29,51,63,30,5,24,75,18,18,4,20,
2
AAAGATGCACATTTCT        AAACATGCACATTTCT,AAAGAAGCACATTTCT,AAAGACGCACATTTCT,AAAGAGGCACATTTCT,AAAGANGCACATTTCT,AA
ATGCACAGTTCT,AAAGATGCACATATCT,AAAGATGCACATCTCT,AAAGATGCACATGTCT,AAAGATGCACATTACT,AAAGATGCACATTCCT,AAAGATGCACATT
T,AAAGATGCACATTTTT,AAAGATGCACCTTTCT,AAAGATGCACGTTTCT,AAAGATGCACTTTTCT,AAAGATGCAGATTTCT,AAAGATGCATATTTCT,AAAGATG
CATTTCT,AAAGCTGCACATTTCT,AAAGGTGCACATTTCT,AAAGNTGCACATTTCT,AAAGTTGCACATTTCT,AAATATGCACATTTCT,AACGATGCACATTTCT,A
GATGCACATTTCT,GAAGATGCACATTTCT,NAAGATGCACATTTCT,TAAGATGCACATTTCT        75590   15,42,15,9,15,58,16,22,29,31,17
4,19
AAAGATGTCCTTTACA        AAAAATGTCCTTTACA,AAACATGTCCTTTACA,AAAGAAGTCCTTTACA,AAAGACGTCCTTTACA,AAAGAGGTCCTTTACA,AA
ATGTACTTTACA,AAAGATGTCATTTACA,AAAGATGTCCATTACA,AAAGATGTCCCTTACA,AAAGATGTCCGTTACA,AAAGATGTCCTATACA,AAAGATGTCCTCT
A,AAAGATGTCCTTTACC,AAAGATGTCCTTTACG,AAAGATGTCCTTTACT,AAAGATGTCCTTTAGA,AAAGATGTCCTTTATA,AAAGATGTCCTTTCCA,AAAGATG
CTTTACA,AAAGATTTCCTTTACA,AAAGCTGTCCTTTACA,AAAGGTGTCCTTTACA,AAAGNTGTCCTTTACA,AAAGTTGTCCTTTACA,AAATATGTCCTTTACA,A
GATGTCCTTTACA,ATAGATGTCCTTTACA,CAAGATGTCCTTTACA,GAAGATGTCCTTTACA,NAAGATGTCCTTTACA,TAAGATGTCCTTTACA        88646
8,8,2,17,23,5,12,1,11,8,13,1,1,7,28,17,20
```

Figure S12: Part of the whitelist file.

To extract UMIs the command is in this form:

*umi_tools extract --bc-pattern="*Pattern sequence*" --stdin SRR6782109_1.fastq --stdout SRR6782109_1_extracted.fastq.gz –read2-in SRR6782109_2.fastq.gz --read2-out= SRR6782109_2_extracted.fastq.gz --filter-cell-barcode --whitelist=whitelist.txt*

Mapping

The following command was used to generate the BAM files:

STAR --runThreadN 12 --genomeDir ./Human-genome-indices --readFilesIn /SRX3742050/SRR6782109_2_extracted.fastq.gz --outFilterMultimapNmax 1 --outSAMtype BAM SortedByCoordinate --outFileNamePrefix SRX3742050/SRR6782019_star_output

Generating count matrix

The featureCounts package was run using the following command:

featureCounts -a hg38_ucsc.gtf -o gene_assigned_file -R BAM out.bam -T 4

To sort the file, we used the following commands:

samtools sort subread-2.0.0-Linux-x86_64/bin/SRR6782019_star_outputAligned.sortedByCoord.out.bam.featureCounts.bam -o assigned_sorted.bam samtools index assigned_sorted.bam

The following command was used to generate a count data file (i.e., gene expression data matrix).

umi_tools count --per-gene --gene-tag=XT --assigned-status-tag=XS --per-cell -I assigned_sorted.bam -S counts.tsv.gz

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	gene	AAACCTGC	AAAGATGC	AAAGATGT	AAAGCAAC	AAAGCAAC	AAATGCCC	AAATGCCT	AACACGTC	AACACGTC	AACACGTT	AACCATGA	AACCATGA
2	ENST00000001008.6	1	1	4	1	0	2	1	1	7	1	0	2
3	ENST00000002165.11	7	7	0	3	7	4	5	6	3	3	4	1
4	ENST00000002501.10	0	0	0	0	0	0	0	0	0	0	0	0
5	ENST00000002596.6	1	0	0	0	5	0	0	0	0	2	2	0
6	ENST00000002829.8	0	0	0	0	0	0	0	0	0	0	0	0
7	ENST00000003084.10	0	0	0	0	0	0	0	0	0	0	0	0
8	ENST00000003100.13	0	0	0	0	0	0	0	0	0	0	0	0
9	ENST00000003302.8	0	0	0	0	1	0	0	0	0	0	0	0
10	ENST00000004982.6	0	0	0	0	0	0	0	0	0	0	0	0

Figure S13: Glimpse of resulting count matrix.

***Reduce the number of genes when many cells have zero counts**:*

For 60% reduction, those genes are deleted, which have '0' expression in 60% of cells. 1201 genes are used to plot the clustering index and the number of clusters in Figure S11. The number of clusters was found to be 10.

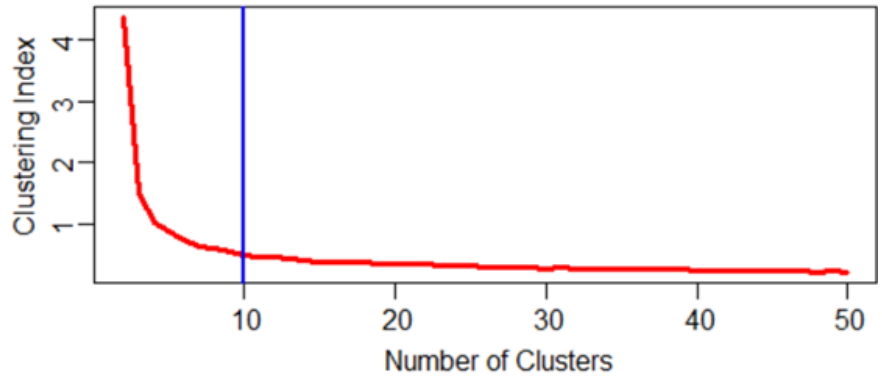

Figure S14: Number of optimal clusters with a 60% reduction in genes.

Differential Expression Analysis

Differential Expression Analysis is also performed on the expression data to reduce the size of the data. It is also called as gene selection in gene expression genomics [1-4]

Document S2: Normalization

DEsingle integrates a modified median normalization method that originated from DESeq to normalize the read counts data [5-8]. Let Y_{ij} denote the read counts of the gene i in cell j, then the size factor of cell j is estimated by:

$$s_j = median \frac{Y_{ij}}{(\prod_{v-1}^{m} Y_{iv})^{1/m}}$$

Where $Y_{ij} \neq 0$, which means only the non-zero read counts of each gene, are used. The normalized read counts of i^{th} gene in j^{th} is calculated using the formula, Y_{ij}/S_j.

Table S1. List of top 500 differentially expressed genes detected through DESeq2 and DEsingle for adenocarcinoma cell lines (.xlsx).

Reference:

1. Das S, Meher PK, Rai A, Bhar LM, Mandal BN (2017) Statistical Approaches for Gene Selection, Hub Gene Identification and Module Interaction in Gene Co-Expression Network Analysis: An Application to Aluminum Stress in Soybean (Glycine max L.). *PLoS ONE* 2017, 12(1): e0169605. https://doi.org/10.1371/journal.pone.0169605
2. Das, S.; Rai, S.N. Statistical Approach for Biologically Relevant Gene Selection from High-Throughput Gene Expression Data. *Entropy* 2020, 22(11), 1205; https://doi.org/10.3390/e22111205
3. Das, S. Rai, A., Mishra, D.C., Rai, S.N. Statistical approach for selection of biologically informative genes. *Gene* 2018, 655: 71-83. doi:10.1016/j.gene.2018.02.044
4. Das, S.; Rai, A.; Merchant, M.L.; Cave, M.C.; Rai, S.N. A Comprehensive Survey of Statistical Approaches for Differential Expression Analysis in Single-cell RNA Sequencing Studies. *Genes* 2021, 12(12), 1947. https://doi.org/10.3390/genes12121947
5. Bacher R, Kendziorski C. Design and computational analysis of single-cell RNA-sequencing experiments. *Genome Biology*. 2016, 17, 63. doi.org/10.1186/s13059-016-0927-y
6. Das, S. and Rai, S.N. SwarnSeq: An improved statistical approach for differential expression analysis of single-cell RNA-seq data. *Genomics* 2021; 113(3), 1308-1324. doi.org/10.1016/j.ygeno.2021.02.014
7. Das, S.; Rai, S.N. Statistical methods for analysis of single-cell RNA-sequencing data. *MethodsX* 2021, *8*, 101580. https://doi.org/10.1016/j.mex.2021.101580.