# Project 2: RNA-Seq analysis

**Name: Mahroo Hajimehdi** **SID: 6104399198**

The samples under investigation in this report were sourced from a study entitled "***Differentially Expressed LncRNAs and mRNA Identified by Sequencing Analysis in Colorectal Cancer Patients***," which is accessible via the Gene Expression Omnibus (GEO) under the accession number GSE104836. This study comprises a total of 20 samples, consisting of an equal distribution of 10 normal and 10 cancerous tissues. Specifically, I selected paired samples bearing the ID 48, corresponding to the GEO accessions GSM2808515 and GSM2808516, with SRA run accessions SRR6191645 and SRR6191646, respectively.

Given that the dataset is arranged in a "paired" format, each SRA entry contains both forward and reverse reads. I processed each SRA file, converting it into paired forward and reverse Fastq.gz files using the *fastq-dump* command from the SRA Toolkit. Subsequently, these fastq files were transferred to a designated directory to undergo Quality Control (QC) analysis.

## Part a - Quality control and trimming

Quality Control (QC) is a pivotal step in the RNA sequencing workflow, ensuring that the data is of high quality and free from technical artefacts that could confound downstream analysis. In this project, FastQC, a widely-used tool for high throughput sequence data, was employed for the QC process.

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~$ fastqc -o ./project2/FASTQC/QCb ./project2/SRR/SRR6191645_1.fastq ./
project2/SRR/SRR6191645_2.fastq ./project2/SRR/SRR6191646_1.fastq ./project2/SRR/SRR6191646_2.fastq
null
null
Started analysis of SRR6191645_1.fastq
null
null
Approx 5% complete for SRR6191645_1.fastq
Approx 10% complete for SRR6191645_1.fastq
Approx 15% complete for SRR6191645_1.fastq
Approx 20% complete for SRR6191645_1.fastq
Approx 25% complete for SRR6191645_1.fastq
Approx 30% complete for SRR6191645_1.fastq
Approx 35% complete for SRR6191645_1.fastq
Approx 40% complete for SRR6191645_1.fastq
Approx 45% complete for SRR6191645_1.fastq
Approx 50% complete for SRR6191645_1.fastq
Approx 55% complete for SRR6191645_1.fastq
```

The command-line invocation used for FastQC within this project began with fastqc, which initiates the program. The -o option specifies the output directory where the FastQC reports will be saved, in this case, `./project2/FASTQC/QCb`. The command then lists the input files, which are raw sequencing files in the Fastq format:

```
./project2/SRR/SRR6191645_1.fastq
./project2/SRR/SRR6191645_2.fastq
./project2/SRR/SRR6191646_1.fastq
./project2/SRR/SRR6191646_2.fastq
```

These files represent the forward and reverse reads from two paired-end sequenced samples. By executing FastQC on the specified files, a suite of analyses was undertaken. The outcomes of these analyses are detailed in reports stored in the `Fastqc/QCb` directory. We first go through what each module tries to assess:

1. **Basic Statistics:** This provides general information about the sequencing run, including the total number of reads, read length, and whether the file is filtered. It's essentially the overview of the dataset.

2. **Per base sequence quality:** This evaluates the quality score for each base position across all reads. Quality scores are plotted as a boxplot over the length of the reads, showing the range of scores at each position.

3. **Per sequence quality scores:** This assesses the overall quality of a read. It plots the frequency of the average quality score over all bases in each read, showing if a particular set of reads is of lower overall quality.

4. **Per base sequence content:** This checks the proportion of each nucleotide (A, T, C, G) at each position across all reads. It's used to assess bias in base composition, particularly important for detecting issues like sequencing bias or contamination.

5. **Per sequence GC content:** This examines the GC distribution across the entire length of reads and compares it to the expected normal distribution. Large deviations can indicate contamination or biased library preparation.

6. **Per base N content:** This calculates the percentage of base calls at each position across all reads for which a base could not be determined (denoted as 'N'). Ideally, this should be very low or 0%.

7. **Sequence Length Distribution:** This analyzes the range and distribution of fragment sizes in the run. It checks for any discrepancies in the expected read length distribution, which could be indicative of sample-processing issues.

8. **Sequence Duplication Levels:** This module counts the number of times an identical sequence appears in the reads and calculates the degree of duplication for each sequence. High levels of duplication may indicate PCR amplification biases.

9. **Overrepresented sequences:** This identifies any single sequence that makes up a large proportion of the total. It can help detect contamination by a specific organism or overrepresented library elements like adapters.

10. **Adapter Content:** This measures the cumulative percentage of sequences that match adapter sequences as the read length progresses. Finding adapter sequences towards the ends of reads can signal that the insert was shorter than the read length, necessitating trimming.

*FastQC results before trimming:*

| | SRR6191645_1 | SRR6191645_2 | SRR6191646_1 | SRR6191646_2 |
|---|---|---|---|---|
| **Per base sequence quality** | PASS | PASS | PASS | PASS |
| **Per sequence quality scores** | PASS | PASS | PASS | PASS |
| **Per base sequence content** | FAIL | FAIL | FAIL | FAIL |
| **Per sequence GC content** | FAIL | FAIL | FAIL | FAIL |
| **Per base N content** | PASS | PASS | PASS | PASS |
| **Sequence Length Distribution** | PASS | PASS | PASS | PASS |
| **Sequence Duplication Levels** | WARN | WARN | FAIL | WARN |
| **Overrepresented sequences** | WARN | WARN | WARN | WARN |
| **Adapter Content** | WARN | WARN | WARN | WARN |

The FastQC analysis of the four FASTQ files from the samples identified both quality strengths and concerns. Basic Statistics, Per base sequence quality, and Per sequence quality scores are consistently passing, indicating reliable overall read quality and sequencing consistency. However, the repeated failure in Per base sequence content and Per sequence GC content across all files suggests a potential bias in nucleotide composition. Sequence Length Distribution consistently passed, suggesting the reads are uniformly sized, which is expected in a standardized sequencing run.

Per base N content passed for all, indicating the absence of ambiguous bases which showcases good sequencing efficacy. Yet, we observe warnings for Sequence Duplication Levels and Overrepresented sequences in all files, except for a failure in Sequence Duplication Levels for `SRR6191646_1.fastq`. These could indicate potential issues with library diversity or the presence of highly abundant sequences, which might be biological or due to PCR duplication. Adapters should not be present in high-quality sequence data, and warnings across all files in Adapter Content hint at contamination that should be trimmed before further analysis.

Particularly for flagged parameter:

- **Per base sequence content:** Showing failure, means there's a deviation from the expected distribution of nucleotide frequencies over the length of the reads which could be a sign of contamination or library preparation issues.
- **Per sequence GC content:** Failures here suggest an abnormal GC distribution that may affect the interpretation of expression levels, possibly pointing to contamination or biased amplification.
- **Sequence Duplication Levels:** High levels of duplication may reflect PCR artifacts or low complexity libraries. The single failure, as opposed to warnings in other samples, indicates a higher level of concern for `SRR6191646_1.fastq`.
- **Overrepresented sequences:** This is commonly due to biological factors such as rRNA contamination, but may also result from library preparation steps.

- **Adapter Content:** Warnings suggest residual sequencing adapters are present, which, if pervasive, can complicate the analysis and should be trimmed.

Based on the outcomes from our prior quality control assessment, it appears that trimming is essential to eliminate bases of low quality and to remove the adaptor content, which was predominantly identified as ***the Illumina Universal Adaptor***. Additionally, reads shorter than 80 base pairs should also be discarded:

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ trimmomatic PE -threads 8 -phred33 \
./SRR/SRR6191645_1.fastq ./SRR/SRR6191645_2.fastq \
./SRR_trimmed/SRR6191645_1.trimmed.fastq ./SRR_trimmed/SRR6191645_1un.trimmed.fastq \
./SRR_trimmed/SRR6191645_2.trimmed.fastq ./SRR_trimmed/SRR6191645_2un.trimmed.fastq \
ILLUMINACLIP:/home/hajimehdi/miniconda3/envs/RNAseq/share/trimmomatic/adapters/TruSeq3-PE.fa:2:30:10 \
SLIDINGWINDOW:4:30 \
MINLEN:80
TrimmomaticPE: Started with arguments:
 -threads 8 -phred33 ./SRR/SRR6191645_1.fastq ./SRR/SRR6191645_2.fastq ./SRR_trimmed/
SRR6191645_1.trimmed.fastq ./SRR_trimmed/SRR6191645_1un.trimmed.fastq ./SRR_trimmed/
SRR6191645_2.trimmed.fastq ./SRR_trimmed/SRR6191645_2un.trimmed.fastq ILLUMINACLIP:/home/hajimehdi/miniconda3/
envs/RNAseq/share/trimmomatic/adapters/TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:4:30 MINLEN:80
Using PrefixPair: 'TACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT'
ILLUMINACLIP: Using 1 prefix pairs, 0 forward/reverse sequences, 0 forward only sequences, 0 reverse only
sequences
Input Read Pairs: 64135799 Both Surviving: 29748101 (46.38%) Forward Only Surviving: 19873659 (30.99%) Reverse
Only Surviving: 2139896 (3.34%) Dropped: 12374143 (19.29%)
TrimmomaticPE: Completed successfully
```

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ trimmomatic PE -threads 8 -phred33 \
./SRR/SRR6191646_1.fastq ./SRR/SRR6191646_2.fastq \
./SRR_trimmed/SRR6191646_1.trimmed.fastq ./SRR_trimmed/SRR6191646_1un.trimmed.fastq \
./SRR_trimmed/SRR6191646_2.trimmed.fastq ./SRR_trimmed/SRR6191646_2un.trimmed.fastq \
ILLUMINACLIP:/home/hajimehdi/miniconda3/envs/RNAseq/share/trimmomatic/adapters/TruSeq3-PE.fa:2:30:10 \
SLIDINGWINDOW:4:30 \
MINLEN:80
TrimmomaticPE: Started with arguments:
 -threads 8 -phred33 ./SRR/SRR6191646_1.fastq ./SRR/SRR6191646_2.fastq ./SRR_trimmed/
SRR6191646_1.trimmed.fastq ./SRR_trimmed/SRR6191646_1un.trimmed.fastq ./SRR_trimmed/
SRR6191646_2.trimmed.fastq ./SRR_trimmed/SRR6191646_2un.trimmed.fastq ILLUMINACLIP:/home/hajimehdi/miniconda3/
envs/RNAseq/share/trimmomatic/adapters/TruSeq3-PE.fa:2:30:10 SLIDINGWINDOW:4:30 MINLEN:80
Using PrefixPair: 'TACACTCTTTCCCTACACGACGCTCTTCCGATCT' and 'GTGACTGGAGTTCAGACGTGTGCTCTTCCGATCT'
ILLUMINACLIP: Using 1 prefix pairs, 0 forward/reverse sequences, 0 forward only sequences, 0 reverse only
sequences
Input Read Pairs: 57973670 Both Surviving: 31328814 (54.04%) Forward Only Surviving: 14801383 (25.53%) Reverse
Only Surviving: 1957765 (3.38%) Dropped: 9885708 (17.05%)
TrimmomaticPE: Completed successfully
```

- `-phred33` or **`-phred64`** specifies the base quality score encoding.
- `SRR6191646_1.fastq` and `SRR6191646_2.fastq` are the input files for forward and reverse reads.
- `SRR6191646_1.trimmed.fastq` and `SRR6191646_2.trimmed.fastq` are the output files for forward and reverse reads where the adaptor sequence has been trimmed and reads are still in a pair.
- `SRR6191646_1un.trimmed.fastq` and `SRR6191646_2un.trimmed.fastq` are the output files where corresponding paired reads were not found after trimming.
- `ILLUMINACLIP:adapters.fasta:2:30:10` specifies the fasta file containing adaptor sequences followed by clipping options: maximum mismatch count, palindrome clip threshold, and simple clip threshold.
- `SLIDINGWINDOW:4:30` scans in a 4-base wide sliding window, cutting when the average quality per base drops below 30.
- `MINLEN:80` drops the read if it is below a specified length (80 in this case).

Following the trimming process, a subsequent QC analysis was conducted using FastQC. The results of these analyses are detailed in reports located within the `Fastqc/QCa` directory.

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~$ fastqc -t 10 -o project2/FASTQC/QCa project2/SRR_trimmed/
SRR6191645_1.trimmed.fastq project2/SRR_trimmed/SRR6191645_2.trimmed.fastq project2/SRR_trimmed/
SRR6191645_1un.trimmed.fastq  project2/SRR_trimmed/SRR6191645_2un.trimmed.fastq
null
null
Started analysis of SRR6191645_1.trimmed.fastq
null
null
Started analysis of SRR6191645_2.trimmed.fastq
Started analysis of SRR6191645_1un.trimmed.fastq
Started analysis of SRR6191645_2un.trimmed.fastq
Approx 5% complete for SRR6191645_2un.trimmed.fastq
Approx 10% complete for SRR6191645_2un.trimmed.fastq
Approx 15% complete for SRR6191645_2un.trimmed.fastq
Approx 20% complete for SRR6191645_2un.trimmed.fastq
```

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~$ fastqc -t 10 -o project2/FASTQC/QCa project2/SRR_trimmed/
SRR6191646_1.trimmed.fastq project2/SRR_trimmed/SRR6191646_2.trimmed.fastq project2/SRR_trimmed/
SRR6191646_1un.trimmed.fastq  project2/SRR_trimmed/SRR6191646_2un.trimmed.fastq
null
null
Started analysis of SRR6191646_1.trimmed.fastq
null
null
Started analysis of SRR6191646_2.trimmed.fastq
Started analysis of SRR6191646_1un.trimmed.fastq
Started analysis of SRR6191646_2un.trimmed.fastq
Approx 5% complete for SRR6191646_2un.trimmed.fastq
Approx 10% complete for SRR6191646_2un.trimmed.fastq
Approx 15% complete for SRR6191646_2un.trimmed.fastq
Approx 20% complete for SRR6191646_2un.trimmed.fastq
Approx 25% complete for SRR6191646_2un.trimmed.fastq
Approx 30% complete for SRR6191646_2un.trimmed.fastq
Approx 5% complete for SRR6191646_1un.trimmed.fastq
Approx 35% complete for SRR6191646_2un.trimmed.fastq
Approx 40% complete for SRR6191646_2un.trimmed.fastq
Approx 45% complete for SRR6191646_2un.trimmed.fastq
Approx 50% complete for SRR6191646_2un.trimmed.fastq
```

***FastQC results after trimming:***

| | SRR6191645_1 | SRR6191645_2 | SRR6191646_1 | SRR6191646_2 |
|---|---|---|---|---|
| **Per base sequence quality** | PASS | PASS | PASS | PASS |
| **Per sequence quality scores** | PASS | PASS | PASS | PASS |
| **Per base sequence content** | FAIL | FAIL | FAIL | FAIL |
| **Per sequence GC content** | FAIL | FAIL | FAIL | FAIL |
| **Per base N content** | PASS | PASS | PASS | PASS |
| **Sequence Length Distribution** | WARN | WARN | WARN | WARN |
| **Sequence Duplication Levels** | WARN | WARN | WARN | WARN |
| **Overrepresented sequences** | WARN | WARN | WARN | WARN |
| **Adapter Content** | PASS | PASS | PASS | PASS |

## Part a - Questions

1. **What is the average number of reads across samples before and after the read trimming?**

| | Before | After |
|---|---|---|
| **SRR6191645_1** | 64135799 | 29748101 |
| **SRR6191646_1** | 57973670 | 31328814 |

Average number of reads before trimming:

$$\frac{(64135799 + 57973670)}{2} = 360554845$$

Average number of reads after trimming:

$$\frac{(29748101 + 31328814)}{2} = 305384575$$

2. **Compare the read length averages in different samples before and after the read trimming?**

*Before trimming:*
The read length was consistently 150 nucleotides.

*After trimming:*
To find the average read length after trimming, you need to take into account the frequencies of each length interval. For intervals (like 80-81), you will assume the midpoint to represent the average length for all reads in that interval (for example, 80 + 81 / 2 = 80.5 for the 80-81 interval), and then multiply it by the frequency of reads for that interval. Once you sum up these products, you divide by the total number of reads to get the average read length after trimming:

$$(805 \times 48190) + (825 \times 50668) + (845 \times 55061) + \cdots + (1495 \times 1951903) = 4230898131$$
$$4230898131 / 29748101 = 142224142$$

| | Before | After |
|---|---|---|
| **SRR6191645_1** | 150 | 142.224142 |
| **SRR6191646_1** | 150 | 146.4611697 |

For SRR6191645_1 trimming reduced the average read length by

$$150 - 142224142 \approx 7775858\, nucleotides$$

For SRR6191646_1 trimming reduced the average read length by

$$150 - 1464611697 \approx 35388303\, nucleotides$$

From these calculations, we can conclude that trimming has indeed shortened the average read length in both samples. However, the extent of the reduction differs between the two samples. Sample SRR6191645_1 experienced a larger decrease in average read length due to trimming

than sample SRR6191646_1. This can be due to various reasons such as differences in quality score thresholds, the presence of adapters, or other sequence-specific considerations that trimming aims to address.

3. **Compare the read quality distributions over all sequences before and after the read trimming.**

When comparing the read quality distributions before and after trimming, one can observe the following:

SRR6191645_1 (Forward Reads):

- Before Trimming: Quality scores ranged between 12 and 40, with a marked exponential increase from a Phred score of 30-33, indicating a substantial number of lower-quality reads.
- After Trimming: The range tightens to between 35 and 40, with an exponential rise more pronounced from 39-40. This tightening and shift towards higher quality scores after trimming suggest that lower-quality reads (or regions within reads) have been removed.

SRR6191645_2 (Reverse Reads):

- Before Trimming: Quality scores varied more widely from 5 to 40, with an exponential increase from 16-20, peaking at 38-39 and then declining sharply. This indicates a substantial quantity of low-quality reads and a sharp drop-off in quality as the read progresses.
- After Trimming: The range narrows to between 34 and 40, with the most substantial increase from 37-38, peaking sharply at 39. The post-trimming distribution indicates that much of the low-quality data has been removed, akin to the forward reads but with a previously observed peak that is now more pronounced, indicating a sudden drop-off in quality at the very end of the reads.

*Why the Average Quality per Read in Reverse (SRR6191645_2) and Forward (SRR6191645_1) are Different?*

- The sequencing technology, while recording the forward and reverse reads, might intrinsically produce different quality scores. Typically, reverse reads (also known as R2 in paired-end sequencing) might exhibit a decline in quality towards the end of sequencing. This can be due to the nature of the sequencing chemistry as the polymerase is more likely to incorporate errors as the sequencing reaction progresses.
- The specifics of library preparation can also cause discrepancies. For example, if the sample has been through PCR amplification, the quality of the starting template, among other factors, can affect the quality of reverse reads more than forward reads.
- In paired-end sequencing, degradation of the sequencing reagents or instruments' performance deterioration over time can lead to a decrease in quality in reverse reads because they are sequenced after the forward reads.

4. **What does the Adaptor Content warning indicate?**

An Adaptor Content warning indicates that a significant portion of the sequence reads contain adaptor sequences. Adaptors are short, artificially synthesized DNA sequences that are ligated to the ends of DNA fragments to facilitate their introduction into sequencing platforms. They are

crucial for the initial steps in the sequencing process but are not part of the target DNA and hence should not be present in the final sequencing reads. Some implications of such a warning are:

- Sequence Contamination: The presence of adaptor sequences in the read data suggests that the library preparation or the sequencing process may have had some issues, potentially leading to contamination of the sequence data with these unwanted adaptor sequences.

- Data Quality Impact: Adaptor sequences can interfere with downstream analysis, such as alignment/mapping to reference genomes, variant calling, and de novo assembly, by producing false alignments or confusing the assembly algorithms.

5. **Why do we first remove the Adapter sequences for the reads and then the low-quality bases?**

Adapters can be accurately identified and trimmed when the full sequence is present. If the low-quality bases were removed first, it could potentially make it harder to identify adapter sequences, as the lower-quality regions of the read may actually contain part of the adapter sequence that has been read with less accuracy.

Most adapter removal algorithms assume that the reads are intact. They work best on the full-length reads rather than fragmented or quality-trimmed sequences, which might lack the adapter sequences in their entirety or present them in a truncated form that is more difficult to recognize.

Risk of False Positives in Quality Trimming: Low-quality bases towards the end of reads might be mistakenly identified as adapters, leading to incorrect trimming. If adapters are removed first, it ensures that the subsequent quality trimming is acting on actual sequence data rather than on sequence artifacts.

Keeping adapter sequences in reads might influence the downstream analyses adversely, such as alignments, where reads with adapters could misalign to the reference genome. Additionally, if the quality trimming were performed first, short stretches of high-quality adapter sequence might be retained in the dataset, leading to issues in subsequent analysis steps.

Post-adapter trimming, the remaining sequence is what will be mapped to the reference genome or used in other analyses. Determining the quality of these sequences (and trimming as necessary) accurately represents the real quality of the data that will be used in downstream processes.

6. **What does the quality of bases mean, and how is it obtained?**

The quality of bases refers to the prediction of the error probability for each base call during sequencing. This quality score represents the likelihood that a particular base has been incorrectly identified. It's essential for determining the accuracy of the sequencing reads and is a critical component of data quality assessment and downstream analyses.

These quality scores are obtained through a process known as base calling, which is performed by the sequencing instrument's software during the sequencing run. During base calling, the software analyzes the intensity of the fluorescent signal emitted as each base is incorporated into a growing DNA strand. Different bases emit different wavelengths of light, and the intensity of the signal correlates with the confidence of the base call made by the sequencer.

Each base is assigned a quality score, which is logarithmically linked to the error probability. The most common scale used to represent these scores is the Phred quality score (Q), defined as:

$$Q = -10 \log_{10} \mathrm{P}$$

where P is the probability of an incorrect base call. Therefore, a Phred quality score of 20 represents a 1 in 100 chance that the base call is wrong, while a score of 30 represents a 1 in 1,000 chance of being incorrect.

In the FASTQ file format, these quality scores are encoded as ASCII characters, following a specific offset (usually 33 or 64, depending on the sequencing platform). Each character corresponds to a Phred quality score, which can be converted back into the error probability to analyze the quality of the sequencing run.

For example, with an offset of 33, the ASCII character '!' represents the lowest possible score (Phred score 0). High-quality bases (e.g., Phred scores > 30) imply a very accurate base call, while lower quality scores indicate less confidence in the base call, and these may be considered for trimming or filtering during the data cleanup stage before analysis.

## Part b - Read mapping

in this step I map the reads to the reference genome using the HISAT2 software. First the reference genome `Homo_sapiens.GRCh38.dna.toplevel.fa.gz` was unzipped using the command:

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ gunzip Homo_sapiens.GRCh38.dna.toplevel.fa.gz
```

Then we need to index the reference genome so I used this command to build the index files

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ hisat2-build -p 8 Homo_sapiens.GRCh38.dna.toplevel.fa
indexed_gene
Settings:
  Output files: "indexed_gene.*.ht2l"
  Line rate: 7 (line is 128 bytes)
  Lines per side: 1 (side is 128 bytes)
  Offset rate: 4 (one in 16)
  FTable chars: 10
  Strings: unpacked
  Local offset rate: 3 (one in 8)
  Local fTable chars: 6
  Local sequence length: 57344
  Local sequence overlap between two consecutive indexes: 1024
  Endianness: little
  Actual local endianness: little
  Sanity checking: disabled
  Assertions: disabled
  Random seed: 0
```

8 files were generated:

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ ls
FASTQC                          indexed_gene.4.ht2l  output.sam
Homo_sapiens.GRCh38.dna.toplevel.fa  indexed_gene.5.ht2l  SRR
indexed_gene.1.ht2l             indexed_gene.6.ht2l  SRR_trimmed
indexed_gene.2.ht2l             indexed_gene.7.ht2l
indexed_gene.3.ht2l             indexed_gene.8.ht2l
```

I moved all these 8 files to a folder named `indexed_genome` using this command:

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ mv indexed_gene.*.ht2l ./indexed_genome/
```

By using the following command, I performed read mapping first for SRR6191645 and subsequently for SRR6191646.

```
(RNAseq) hajimehdi@ibbadmin-X10DAi:~/project2$ hisat2 -p 10 -x ./indexed_genome/indexed_gene -1 ./SRR_trimmed/SRR6191645_1.trimmed.fastq -2 ./SRR_trimmed/SRR6191645_2.trimmed.fastq -S ./SRR6191645.sam
```

the result where 2 SAM files that needed to be converted to BAM for further analysis. Using this command the conversion can take place:

```
(samtool) hajimehdi@ibbadmin-X10DAi:~/project2$ samtools view -Sb SRR6191646.sam > SRR6191646.bam
(samtool) hajimehdi@ibbadmin-X10DAi:~/project2$ samtools view -Sb SRR6191645.sam > SRR6191645.bam
```

## Part b - Questions

1. **What is the difference between SAM and BAM files?**

SAM (Sequence Alignment/Map) and BAM (Binary Alignment/Map) files are two formats used for storing sequence data, particularly the alignments of sequence reads to a reference genome. Both formats are designed to contain the same information, but they are structured differently with distinct use cases.
SAM Files:

- Text-based format, which makes them human-readable.
- They tend to be significantly larger in size compared to BAM files, which makes them less efficient for storage and slower to process.
- Easy to manipulate using standard text-processing tools.
- Header section is optional, but if present, it provides detailed information about the alignment, including the reference sequence names and lengths, and may contain other metadata.

BAM Files:

- Binary version of SAM, which makes them not human-readable without specialized software like SAMtools to view or manipulate them.
- More storage-efficient because they are compressed, usually reducing the file size by about 4 to 10 times compared to SAM files.
- Faster to process due to their compressed nature, which makes them more suitable for large-scale analyses and pipelines.

- Require indexing for efficient access to individual alignments or regions, which is particularly useful for visualizing the data in genome browsers or for analyses that need to access specific genomic coordinates rapidly.
- Header section contains the same information as a SAM file and is also optional, but is stored in binary form.

Both formats store information about individual reads, their alignment to the reference, mapping quality, and optional tag-value pairs that provide additional information about each read. Data in SAM and BAM files include fields like QNAME (query template NAME), FLAG (bitwise FLAG), RNAME (Reference sequence NAME), POS (1-based leftmost mapping POSition), MAPQ (MAPping Quality), CIGAR (CIGAR string), RNEXT, PNEXT, TLEN, SEQ (segment SEQuence), and QUAL (quality scores).

**2. What is the purpose of indexing the genome?**

a. Speed Up Searches: Indexing a genome is analogous to indexing a book. Just like an index helps you quickly find specific information in a book without reading every page, a genome index lets computational tools rapidly locate sequences without scanning the entire genome.

b. Efficient Storage: Genome indexes can efficiently store information about the structure and content of the genome, thereby facilitating quick access to various genomic regions.

c. Mapping Reads: The primary utility of a genome index is to aid in aligning sequencing reads to the reference genome. During alignment, the process looks for matching regions or similar sequences within the indexed reference, speeding up what would otherwise be a very time- and resource-intensive process.

d. Random Access: Particularly in BAM files, indexing allows for random access to the data, which means quick retrieval of read information from any specific genomic location without having to go through the entire file.

e. Visualization: Genome browsers and visualization tools rely on indexed genomes to display a particular genomic region without delay. This capability is essential when working with large genomic datasets.

f. Reduce Computational Load: By avoiding a full scan of the reference genome during analyses such as read alignment or variant calling, indexing conserves computational resources, reducing the time and CPU/memory needed to accomplish these tasks.

A genome index is usually built using specialized algorithms that create data structures like B-trees, hash tables, or suffix arrays/Tries (as in the Burrows-Wheeler Transform). These structures are optimized to reduce lookup times drastically and form the backbone of many fast sequence alignment tools such as BWA and Bowtie.

**3. Report mapping percentages of all samples in a table. Please explain why a low percentage of reads cannot be mapped.**

To do this we need to follow the following steps:

*1- Sort and Index the BAM files:*

Before proceeding with the viewing or processing of BAM files, they should be sorted, and an index should be created.

```
samtools sort -@ 4 -o SRR6191645_sorted.bam SRR6191645.bam
samtools index -@ 4 SRR6191645_sorted.bam

samtools sort -@ 4 -o SRR6191646_sorted.bam SRR6191646.bam
samtools index -@ 4 SRR6191646_sorted.bam
```

*2- Generate Alignment Statistics:*

SAMtools provides the flagstat command, which gives us a summary of various statistics about the alignments, including the number of reads that are mapped, unmapped, properly paired, etc.


| samtools flagstat SRR6191645_sorted.bam | samtools flagstat SRR6191646_sorted.bam |
|---|---|
| a83687004 + 0 in total (QC-passed reads + QC-failed reads) | 88910136 + 0 in total (QC-passed reads + QC-failed reads) |
| 24190802 + 0 secondary | 26252508 + 0 secondary |
| 0 + 0 supplementary | 0 + 0 supplementary |
| 0 + 0 duplicates | 0 + 0 duplicates |
| **81482119 + 0 mapped (97.37% : N/A)** | **87184381 + 0 mapped (98.06% : N/A)** |
| 59496202 + 0 paired in sequencing | 62657628 + 0 paired in sequencing |
| 29748101 + 0 read1 | 31328814 + 0 read1 |
| 29748101 + 0 read2 | 31328814 + 0 read2 |
| 55544646 + 0 properly paired (93.36% : N/A) | 59145316 + 0 properly paired (94.39% : N/A) |
| 56531706 + 0 with itself and mate mapped | 60180688 + 0 with itself and mate mapped |
| 759611 + 0 singletons (1.28% : N/A) | 751185 + 0 singletons (1.20% : N/A) |
| 471720 + 0 with mate mapped to a different chr | 438728 + 0 with mate mapped to a different chr |
| 329727 + 0 with mate mapped to a different chr (mapQ>=5) | 282925 + 0 with mate mapped to a different chr (mapQ>=5) |

From the information above we have what we want:

**SRR6191646 ⟼ 98.06%**

**SRR6191645 ⟼ 97.37%**

A low percentage of reads mapping to the reference sequence can be due to several reasons, including:

1. Poor quality reads, which include a high number of ambiguous bases or low-quality scores, might not map reliably to the reference genome.

2. Issues during the library preparation phase can also lead to poor mapping. This includes problems like DNA shearing, adapter contamination, or inefficient PCR amplification.

3. Genomes often contain repetitive sequences. Reads that come from these regions may map to multiple locations or not map uniquely, causing them to be filtered out or marked as low mapping quality.

4. Significant genetic variants (such as SNPs, indels, or structural variants) between the sample and the reference genome could cause alignment algorithms to fail to map the reads.

5. RNA-Seq Specific Challenges: If working with RNA-Seq data, issues such as high levels of rRNA contamination, RNA editing, or bias in RNA selection could result in low mapping rates.

6. Errors introduced by the sequencing platform, such as sequence-specific biases or adapter remnants, might impede proper alignment.

7. If the experiment targets non-coding DNA or RNA, but the reference is coding DNA, large portions of the sequence might go unmapped.

8. Multiplexing errors or barcoding issues can mix up reads from different samples, leading to poor mapping results.

## Part c - Building gene expression matrix

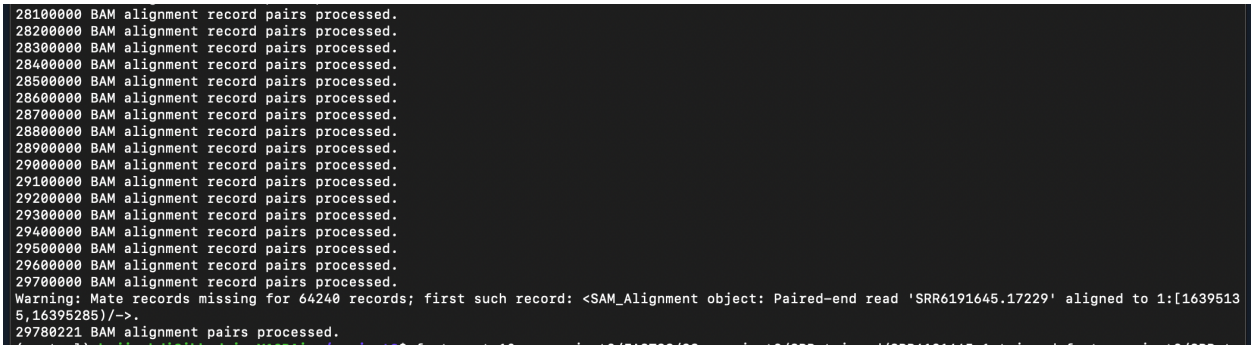
In this step we'll run *htseq-count* on count aligned reads for differential expression analysis. We first move `Homo_sapiens.GRCh38.106.chr.gtf.gz` to the directory and unzip it.

```
(samtool) hajimehdi@ibbadmin-X10DAi:~$ gunzip Homo_sapiens.GRCh38.106.chr.gtf.gz
```

Then we run these two commands, each for one sample:

```
htseq-count -f bam -r pos -s no -t exon -i gene_id SRR6191646.bam
Homo_sapiens.GRCh38.106.chr.gtf > ./count2/counts.txt
```


```
htseq-count -f bam -r pos -s no -t exon -i gene_id SRR6191645.bam
Homo_sapiens.GRCh38.106.chr.gtf > ./count/counts.txt
```




The output consists of two `counts.txt` files, each containing two columns: the first column lists gene names in ***Ensembl*** format, and the second column reports the corresponding read counts. The next step involves merging these result files into a single matrix to facilitate further analysis. The code for doing so can be found in `merge_counts.R` located in `Part-c` directory and the merged file `merged_data.txt` is in the same directory.

## Part c - Questions

1. **How many genes are not expressed in control and tumor samples? Explain the results.**

we typically need to establish a threshold for what constitutes 'not expressed' in our data. We set this value to 0, but some analyses may set a higher threshold to account for background noise or low-level contamination in sequencing data.
The code to count number of 0 in the files can be found in `merge_counts.R`. The results indicate that there are `18,543` genes not expressed in sample SRR6191645, `20,183` genes not expressed in sample SRR6191646, and `14,297` genes that are not expressed in either of the two samples. *But why are some genes not expressed?*

Gene expression can vary considerably across different cell types, tissue states (such as normal or cancerous), and conditions due to a complex interplay of biological and technical factors. Here are some reasons why certain genes may not be expressed in a dataset:

1. Cell Type Specificity: Some genes are specifically expressed in certain cell types and are silent in others as a result of cellular differentiation and specialization.
2. Tissue Specificity: Similar to cell types, entire tissues can have unique gene expression profiles. Genes may be expressed in one type of tissue and not in another.
3. Developmental Stage: The expression of some genes is tightly regulated during development. They may be transcriptionally active at some stages and inactive at others.
4. Disease States: The progression of diseases such as cancer can activate or repress the expression of genes that are not commonly active or inactive in a normal state.
5. Environmental Factors: Factors such as stress, diet, chemical exposure, and other environmental stimuli can influence gene expression patterns.
6. Epigenetic Regulation: Chemical modifications of DNA or histones (proteins around which DNA winds) can cause gene silencing without changing the genetic code itself.
7. Transcription Factor Binding: Some genes may not be expressed because the transcription factors necessary to initiate their transcription are not present or active.
8. RNA Decay: If mRNA is not stable or is rapidly degraded, it may not be captured in a gene expression profile, even if it is initially transcribed.
9. Experimental Design: The sensitivity of the experimental method used to measure gene expression (like RNA-Seq or microarrays) may not be high enough to detect lowly expressed genes.
10. Technical Variability: Factors such as RNA degradation during sample preparation, sequencing errors, or low sequencing depth can result in an absence of detectable expression for some genes.
11. Statistical Thresholding: In data analysis, thresholds are often set to determine what constitutes 'expression' versus 'no expression,' potentially leading to genes being categorized as not expressed due to stringent cutoffs.

There are 4 particular rows that are not present in the main study's count matrix:

1. ***__alignment_not_unique***: Reads that didn't align uniquely to one location in the genome
2. ***__ambiguous***: Reads that aligned to multiple genes and hence are ambiguous
3. ***__no_feature***: Reads that didn't align to any annotated gene
4. ***__not_aligned***: Reads that didn't align to the genome at all
5. ***__too_low_aQual***: Reads that aligned but didn't meet the quality threshold

| GeneEns | X48C_COUNT | Percentage X48C | X48N_COUNT | Percentage X48N |
|---|---|---|---|---|
| **__alignment_not_unique** | 5015043 | 16.8401806 | 5681202 | 18.10515467 |
| **__ambiguous** | 3689169 | 12.38798396 | 4444355 | 14.16350531 |
| **__no_feature** | 9667769 | 32.46372483 | 8228576 | 26.22326071 |
| **__not_aligned** | 722637 | 2.426566949 | 487285 | 1.552905581 |
| **__too_low_aQual** | 697135 | 2.34093293 | 677709 | 2.159758844 |
| **total** | 29780221 | | 31378920 | |

To explain the results in the context of these specific categories listed in the matrix:

- **__alignment_not_unique**: High numbers here suggest that a lot of reads are coming from repetitive or duplicated regions. It's not a direct measure of gene expression.
- **__ambiguous**: This reflects the complexity of the transcriptome, where reads can map to multiple genes.

- **__no_feature**: If this number is high, it indicates that a significant portion of reads cannot be assigned to any gene in the annotation, which might mean many genes are not being detected or that there is a lot of off-target sequencing.
- **__not_aligned**: This could suggest poor-quality sequences, contamination, or issues with the reference genome.
- **__too_low_aQual**: High numbers here might indicate that many reads did not pass the quality threshold.

2. **Compare the matrix obtained at this stage with the corresponding gene expression submatrix of the main study. Discuss the differences.**

To compare our submatrix with the corresponding part of the gene expression matrix from the main study, we should consider several factors:

- Gene Identifiers: Ensure that the gene identifiers in our submatrix match those in the main study. There is a difference in gene names; the main study uses mix of both Ensembl IDs and gene symbols, while we only have Ensembl IDs.

- Sample Pair Comparisons: Our subset contains only one of 10 pairs of cancerous and non-cancerous sample data, namely pairId = 48.

- Normalization: The main study includes both raw counts and TPM (transcripts per million) normalized data. We only have the raw counts.

- Differences in Data Processing: There may be preprocessing steps (e.g., filtering low-expression genes, batch correction) applied to the main study data compared to our subset.

The `GSE104836_gene_exp.txt` dataset comprises `37,362` gene entries, while our matrix encompasses a larger set with `61,503` gene entries. When we focus on the overlapping set of genes, which is cataloged in the `common_genes.txt` file, and apply the gene identifier conversions provided by the `comparison.R` script, we can compare their counts. This comparison yields the divergent count values between cancerous and normal tissues for these shared genes:

```
comparison_ratio between X48C_COUNT in both tables
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.2778  0.3931     Inf  0.5103     Inf     938

comparison_ratio between X48N_COUNT in both tables
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.    NA's
 0.0000  0.3065  0.4286     Inf  0.5521     Inf    1110
```

3. **What are other software available to do this step? Name two other software and discuss their advantages and disadvantages.**

Two other software applications commonly used for counting aligned reads for differential expression analysis are FeatureCounts from the Subread package and Cufflinks.

**1. *FeatureCounts*:**

- Advantages:
  - Speed: FeatureCounts is known for its efficient processing speed and low memory consumption.
  - Compatibility: It can handle various types of aligned files (BAM/SAM).
  - Simplicity: It has straightforward usage and a simpler output that facilitates downstream analysis.
  - Multithreading: It supports multithreading, which can make the counting process much faster on systems with multiple cores.
  - Accuracy: FeatureCounts is also credited with high accuracy in read assignment.

- Disadvantages:
  - Flexibility: It has fewer options for fine-tuning the algorithm compared to tools like Cufflinks that have more parameters to adjust.

**2. *Cufflinks*:**

- Advantages:
  - Isoform resolution: It can handle the quantification at the isoform level, providing more detailed expression data.
  - Transcript discovery: Cufflinks has the ability to identify novel transcripts, which HTSeq and FeatureCounts do not do.
  - Integrated suite: It is part of a larger suite (Cufflinks/Cuffdiff) that handles differential expression analysis aside from just counting reads.

- Disadvantages:
  - Complexity: The software has a steeper learning curve due to its longer list of parameters and configuration options.
  - Computationally Intensive: It is more resource-intensive, requiring more memory and computational power than FeatureCounts.
  - Time-consuming: It generally takes longer to run compared to HTSeq and FeatureCounts.

Each software tool for quantifying gene expression from RNA-seq data has its specific use-case scenarios. The choice between them depends on the specific needs of the analysis, the available computational resources, and the level of detail required. While HTSeq-count is commonly used and integrates well with edgeR for differential expression analysis, FeatureCounts offers a fast alternative and Cufflinks provides more granular control over transcript detection and quantification.

## Part d - Differential gene expression analysis

For a detailed explanation of the code, please refer to the accompanying R script in the `Part-d/` folder, which includes comprehensive comments that gives information about the code and steps taken.

## Part d - QUESTIONS

1. **How many genes are given to edgeR? How many of them are differentially expressed in tumor versus normal samples? How do you define statistical significance in this context?**

There are total of 37362 genes. We then filter for genes that the mean expression value of them is greater than or equal to 1. The purpose here is to filter out genes with low average expression across samples, which are often considered to be at the noise level and not informative for further analysis. We end up with 17166 genes which we give it to edgeR.

The measure of significancy is calculated with the following criteria:

- estimateCommonDisp():
  This function estimates the common dispersion, which is a measure of the variability among the biological replicates that is not explained by the mean of the counts. This is an important step as it helps to model the variability and apply it to the statistical tests.
- estimateTagwiseDisp():
  The function estimateTagwiseDisp refines the dispersion estimate for each gene (tag). This allows for gene-specific dispersion estimates, which usually provides a better model for the count data, leading to more accurate DEG calls.
- exactTest(dge, pair=c(1, 2)):
  The exactTest function performs pairwise tests for differential expression between two groups of samples. The pair argument corresponds to the levels of the group factor defined earlier: typically, here 1 would refer to normal and 2 to tumor samples, or vice-versa.
- topTags()
  After performing the exact test, topTags is used to sort genes by their level of differential expression, with n=Inf indicating that all genes should be returned, not just a subset. The $table extracts the results table from the topTags output, which includes statistics for each gene like log-fold changes, log-counts per million, p-values, and false discovery rate (FDR) values.

For defining the statistical significance we would count the number of genes in the DEG table with a p-value (or, more commonly, an adjusted p-value such as FDR) below a certain threshold, which we set to determine statistical significance.
Statistical significance in this context is defined by:

- The p-value, indicating the probability of observing at least as extreme a difference in expression between the two conditions by chance alone.
- The false discovery rate (FDR), which adjusts for multiple testing. A common threshold for significance is an FDR of 0.05 or less.

For example, if we want to apply an FDR cutoff we can have 7 differentially expressed genes that meet our criterion for statistical significance, which in this case is an FDR of 0.05 or less.

2. **Determine the percentage of differentially expressed genes with |log2FoldChange| > 1.5.**

By setting the threshold, 1.258301% of genes are DEGs.

3. **Explain the difference between P-value and FDR?**

***P-value:***

- The p-value is the probability of obtaining an effect at least as extreme as the one in your sample data, assuming the null hypothesis is true.
- It does not take into account the number of tests you are performing.
- A low p-value (typically ≤ 0.05) is often interpreted as meaning that the observed effect is statistically significant, i.e., unlikely to have occurred by chance alone.

***False Discovery Rate (FDR):***

- The FDR is a proportion that measures the expected proportion of false positives (incorrectly rejected null hypotheses) among all rejected hypotheses.
- FDR is particularly important in situations where multiple comparisons or tests are conducted, and the chance of making Type I errors (false positives) increases with the number of tests.
- FDR-controlling procedures adjust p-values or significance levels to account for the multiplicity of tests, providing a way to control the expected proportion of "discoveries" or rejected null hypotheses that are false.

So the difference can be explained as:

- Scope of Control: A p-value measures the evidence against a specific null hypothesis for a single test, whereas FDR controls the expected rate of false positives across multiple tests.
- P-values do not inherently adjust for multiple comparisons; this requires additional correction methods like Bonferroni or Holm. FDR methods, such as the Benjamini-Hochberg procedure, directly adjust for multiple comparisons.
- A single p-value does not give information about the rate of type I errors in multiple tests, while FDR provides a framework for understanding and limiting the proportion of type I errors in the context of multiple hypotheses testing.

## Part e - Gene Ontology enrichment analysis

For a detailed explanation of the code, please refer to the accompanying R script in the directory `Part-e/`, which includes comprehensive comments that gives information about the code and steps taken.

1. **Display results related to Biological Process, Molecular Function, Cellular Component, and KEGG as separate plots using an R package of your choice.**

The visual outputs have been organized within the `Part-e/` directory. Functional enrichment analysis was conducted using the `enrichKEGG` function for KEGG pathway annotation, and `enrichGO` for Gene Ontology (GO) terms, covering *Biological Processes (BP), Molecular Functions (MF), and Cellular Components (CC)*. Under the threshold of `qvalueCutoff` set to 0.05, no significant enrichment was detected for Cellular Components (CC) annotations. Consequently, the `goseq` package was utilized as an alternative for the analysis of GO:CC. Additionally, these results were aggregated with findings from the ***Enrichr*** website to provide a comprehensive view of the functional annotations.

2. **Do a brief study of each of the significant terms and discuss which terms you think may play an important role.**

***a. KEGG Pathways:***

- *Rheumatoid arthritis*: Although primarily associated with joint inflammation, the pathway involves inflammatory cytokines that can play a role in tumorigenesis through chronic inflammation.
- *IL-17 signaling pathway*: IL-17 can contribute to cancer-related inflammation and has dual roles in tumor promotion and immune surveillance, which can be critical in colorectal cancer.
- *Lipid and atherosclerosis*: This suggests altered lipid metabolism, which can be crucial for supplying energy and building blocks for rapidly proliferating cancer cells.
- *Cytokine-cytokine receptor interaction*: Key for cell signaling, its alteration can impact immune evasion and inflammation in cancer.
- *Transcriptional misregulation in cancer*: Highlights the role of transcription factors and regulatory RNAs in the abnormal proliferation of cells during cancer.

***b. GO Terms:***

- *Receptor ligand activity (BP):* Critical for cell signaling and communication, its deregulation can alter tumor microenvironment interactions.
- *Metallopeptidase activity (MF):* Involves enzymes like MMPs that degrade the extracellular matrix, which is a key process in cancer invasion and metastasis.
- *Catenin Complex (CC):* It's an essential part of cell adhesion. Deregulation could lead to decreased cell-cell adhesion, encouraging metastatic spread.
- *Humoral immune response and antimicrobial humoral response (BP):* This suggests an atypical engagement of the body's defenses and possibly aberrant inflammation in colorectal cancer.
- *Collagen metabolic process and associated terms (BP):* Point to extracellular matrix alteration, which could facilitate tumor expansion and metastasis.
- *adherens junction (CC):* Indicate disruptions in cell-cell adhesion structures; such dysfunction is a typical step toward increased invasiveness of cancer cells.

**3. Write a general biological conclusion about the final results of the project.**

The final results of the study on "***Differentially Expressed LncRNAs and mRNA Identified by Sequencing Analysis in Colorectal Cancer Patients"*** indicate a multifaceted impact on critical cellular pathways and functions. There is a notable enrichment of genes involved in inflammatory processes, ECM remodeling, and immune responses, which are all pivotal in the cancer progression landscape. The identified DEGs, namely, ***CST1, MMP3, MMP1, REG1B, MMP7, REG1A, CXCL5, INHBA, IL8, CDH3, and OTOP2*** disrupt various biological processes and cellular components that are usually responsible for *maintaining cellular architecture and tissue homeostasis, such as the extracellular matrix and cell adhesion junctions.*

Prominently, pathways related to inflammation such as IL-17 signaling and those connected to matrix remodeling via metalloproteinases hint at a tumor microenvironment conducive to cancer growth and metastasis. Additionally, disturbances in cytokine and chemokine signaling underscored by the KEGG pathways may contribute to an immune landscape that permits tumor escape and progression.

The KEGG pathways such as "Rheumatoid arthritis," "IL-17 signaling pathway," "Cytokine-cytokine receptor interaction," "Coronavirus disease - COVID-19," "TNF signaling pathway," and "Transcriptional misregulation in cancer" involve several key processes important in inflammation and immune response. Chronic inflammation is known to contribute to cancer progression, including colorectal cancer. For example, cytokine signaling pathways typically mediate cell communication in the immune system, and their deregulation might lead to inflammatory responses that promote tumor development or enhance tumor survival and growth.

DEGs involved in "Lipid and atherosclerosis" could suggest a role for lipid metabolism in the cancer, as dysregulated lipid metabolism has been implicated in cancer cell proliferation and survival. MMP1, MMP3, and MMP7 are matrix metalloproteinases, which contribute to extracellular matrix remodeling, playing a role in "extracellular matrix organization" as found in the GO Biological Process (BP) category. This is crucial in tumor invasion and metastasis.

The DEGs noted in "Transcriptional misregulation in cancer" might affect gene expression that can lead to uncontrolled cell proliferation and avoid apoptosis. Pathways like "IL-17 signaling" and "TNF signaling" are deeply involved in modulating the tumor microenvironment, which can affect cancer progression and response to therapies.

The enriched GO BP terms indicate a significant impact on immune response activities, such as "humoral immune response" and "antimicrobial humoral response." This suggests that there may be an alteration in how the body's defenses are organized against pathogens, which is consistent with the idea of an inflammation-cancer link.

Moreover, terms like "collagen metabolic process" and "extracellular matrix organization" are associated with the structure and integrity of tissues. Since MMPs are among your DEGs, and they are known to degrade the extracellular matrix, this might facilitate tumor invasion and metastasis.

The GO CC categories like "Catenin Complex" and "Adherens Junction" imply that cell adhesion mechanisms are affected. Deregulation of cell adhesion is a hallmark of cancer, as it promotes cancer cells' detachment and dissemination to distant organs.

The "Vesicle" and "Cell-Cell Junction" components are involved in processes such as cell communication, signaling, and transport of molecules, all of which are crucial for maintaining normal cellular functions and whose disruption may contribute to cancer progression.

In summary, the deregulation of these pathways and cellular components suggests a complex interplay between inflammatory responses, immune function, extracellular matrix remodeling, cell adhesion, and cell communication in the progression of colorectal cancer.