Topic 3-- Assessment Information

BMS5021 Introduction to bioinformatics

Xiaoman Zhou 34258221

Part 1 — Introduction, Objectives, and Initial Familiarisation

Introduction

This project analyses a TMT-labelled quantitative proteomics dataset from the CPTAC colorectal cancer study, introduced in Topic 3. The dataset includes matched solid-tissue normal and primary tumour samples across multiple patient sets. It is well suited to the unit because it requires the full workflow emphasised in Topic 3: data cleaning, annotation, exploratory visualisation, and differential expression analysis.

Two files are provided :

- BMS5021_CPTAC_data.xlsx — the main protein expression matrix.
- BMS5021_CPTAC_sample_information.xlsx — the experimental design and sample annotation table.

Together, these files allow reconstruction of the complete TMT sample structure prior to analysis.

Objectives

The goals of this project are to:

- Understand the structure of a multi-sample TMT proteomics dataset and correctly match TMT channels to biological conditions.
- Perform preprocessing steps, including cleaning annotations, handling formatting artefacts, and retaining zero values as required by the protocol.
- Explore the expression matrix using summary statistics and visualisations.
- Conduct differential expression analysis between tumour and normal tissues, with appropriate log-transformation and multiple-testing correction.
- Produce clear visual outputs (e.g., volcano plots) and document the analytical workflow in a reproducible manner.

Initial Familiarisation

1. BMS5021_CPTAC_data.xlsx

- ~10,294 protein groups across 59 columns
- First two columns: Protein Group, Accession
- Remaining columns: TMT-labelled samples (e.g., 01_CPTAC_TMT_126 to 131) from sets 01–06
- TMT_131 consistently serves as the internal reference channel

2. BMS5021_CPTAC_sample_information.xlsx

- Six sample sets, each containing TMT_126–131 channels
- Each channel maps to a patient ID and tissue type (Normal or Tumour)
- File contains formatting artefacts (e.g., “x000D”), requiring cleaning before integration

Part 2 — Preprocessing

2.1 Overview

The aim of Step 2 was to clean and prepare the raw CPTAC colorectal cancer proteomics data so that all samples and protein measurements are correctly aligned. This step supports the project objectives by ensuring accurate sample annotation, validating the expression matrix, and resolving formatting issues in the raw files before performing any visualisation or statistical analysis.

2.2 Data Cleaning & Annotation

2.2.1 Sample Information Table

The sample information file contained Excel formatting artefacts such as _x000D_ and unexpected line breaks, which made patient IDs and tissue types difficult to extract. These were removed using string cleaning.

```r
# Extract TMT labels (row 2)
tmt_labels <- as.character(sample_df[2, 3:ncol(sample_df)])

# Build mapping (start from row 3)
mapping <- tibble()

for (i in 3:(nrow(sample_df))) {
  set_id <- sample_df[[i, 2]]

  if (is.na(set_id)) next

  for (j in seq_along(tmt_labels)) {
    raw_val <- sample_df[[i, j + 2]]

    if (is.na(raw_val)) {
      tissue_type <- NA
      patient_id <- NA
    } else {
      # Clean format artifacts (_x000D_, \n)
      clean <- str_remove_all(as.character(raw_val), "_x000D_|\\\\n") %>%
        str_trim()

      # Identify tissue type
      if (str_detect(clean, "ColonRef|Internal")) {
        tissue_type <- "Internal_Standard"
        patient_id <- "ColonRef"
      } else {
        patient_id <- str_extract(clean, "^[A-Z0-9]+")
        tissue_type <- if_else(
          str_detect(clean, "Normal|Solid"),
          "Normal",
          if_else(str_detect(clean, "Tumor|Tumour|Primary"), "Tumor", NA_character_)
        )
      }
    }
```

The TMT channel names were extracted from row 2, and sample rows were processed from row 3 onward. Each TMT entry was parsed to obtain:

- SampleSet (e.g., 01_CPTAC)
- TMT channel (e.g., TMT_126)
- Patient ID
- Tissue type (Normal or Tumor)

All internal standard entries (TMT_131 / ColonRef) were identified and removed.

After cleaning, the metadata contained:

- 21 Normal samples
- 33 Tumor samples
- 54 total experimental samples

```
> # Remove internal standards (TMT_131)
> mapping_clean <- mapping %>% filter(Tissue_Type != "Internal_Standard")
> n_normal <- sum(mapping_clean$Tissue_Type == "Normal")
> n_tumor <- sum(mapping_clean$Tissue_Type == "Tumor")
> n_standards <- sum(mapping$Tissue_Type == "Internal_Standard")
> cat(sprintf("   Normal: %d | Tumor: %d | Standards (removed): %d\n",
+             n_normal, n_tumor, n_standards))
   Normal: 21 | Tumor: 33 | Standards (removed): 6
> cat(sprintf("   Cleaned metadata: %d samples\n\n", nrow(mapping_clean)))
   Cleaned metadata: 54 samples
```

2.2.2 Protein Identifiers

The expression matrix contained consistent protein group and accession identifiers. These were kept unchanged.

2.2.3 Missing & Zero Values

Zero intensities were present across the data and were retained because they represent true non-detections. No imputation was applied here. Zero values accounted for 19.8% of all measurements (110,179 out of 555,876).

2.3 Expression Matrix Validation

2.3.1 Column Alignment

The Column_Name values generated from the cleaned metadata were matched to expression matrix column names. After name cleaning, all 54 experimental samples matched successfully. The cleaned matrix contains 10,294 proteins × 54 samples.

2.3.2 Basic Data Quality Checks

- No proteins had all-zero intensity.
- Normal/Tumor sample counts matched the metadata.
- Internal standards were successfully excluded.

```
> # 4. ASSESS ZERO VALUES (basic check)
> # ================================================================================
> cat("4. Assessing zero values...\n")
4. Assessing zero values...
> expr_vals <- as.matrix(expr_clean[, -c(1:2)])
> total_data_points <- prod(dim(expr_vals))
> total_zeros <- sum(expr_vals == 0, na.rm = TRUE)
> zero_pct <- 100 * total_zeros / total_data_points
> proteins_all_zeros <- sum(rowSums(expr_vals == 0, na.rm = TRUE) == ncol(expr_vals))
> cat(sprintf("   Total data points: %d\n", total_data_points))
   Total data points: 555876
> cat(sprintf("   Zero values: %d (%.1f%%)\n", total_zeros, zero_pct))
   Zero values: 110179 (19.8%)
> cat(sprintf("   Proteins with all zeros: %d\n\n", proteins_all_zeros))
   Proteins with all zeros: 0
```

2.3.3 Issues Encountered and Fixes

During Step 2, several practical issues occurred and were resolved:

(1) Folder path error — trailing space
Your directory name was "bm " (with a space).
Initially, the code used "bm" and failed to read the files.
Using: list.files

revealed the correct folder "bm ".

(2) Mixing Python syntax in R
Early attempts used Python-style commands such as

```
> zero_per_sample <- colSums(expr_vals == 0, na.rm = TRUE)
> zero_stats <- tibble(
+   Column_Name = colnames(expr_vals),
+   Zero_Count = zero_per_sample,
+   Zero_Percentage = 100 * zero_per_sample / nrow(expr_vals)
+ )
> proteins_all_zeros <- sum(rowSums(expr_vals == 0, na.rm = TRUE) == ncol(expr_vals))
> cat(sprintf("Total data points: %,d\n", prod(dim(expr_vals))))
Error in sprintf("Total data points: %,d\n", prod(dim(expr_vals))) :
  invalid format '%,d'; use format %f, %e, %g or %a for numeric objects
> cat(sprintf("Total zeros: %,d (%.1f%%)\n", total_zeros, zero_pct))
Error in sprintf("Total zeros: %,d (%.1f%%)\n", total_zeros, zero_pct) :
  invalid format '%,d'; use format %d, %i, %o, %x or %X for integer objects
> cat(sprintf("Proteins with all zeros: %d\n\n", proteins_all_zeros))
Proteins with all zeros: 10294
```

This happened because:

- R does not support %,d formatting
- prod(dim(expr_vals)) is numeric, not integer

(3) TMT label extracted from the wrong row

```
> cat("First 5 column names from mapping_clean:\n")
First 5 column names from mapping_clean:
> print(head(mapping_clean$Column_Name, 5))
[1] "01_CPTAC_TMT label" "01_CPTAC_NA"        "01_CPTAC_NA"        "01_CPTAC_NA"
[5] "01_CPTAC_NA"
> cat("\nFirst 5 column names from data_df:\n")

First 5 column names from data_df:
> print(head(names(data_df), 5))
[1] "Protein Group"     "Accession"         "01_CPTAC_TMT_126"  "01_CPTAC_TMT_127N" "01_CPTAC_TMT_127C"
> cat("\nMatching columns:\n")

Matching columns:
> expr_cols <- intersect(mapping_clean$Column_Name, names(data_df))
> print(expr_cols)
```

At first, TMT labels were taken from row 1, causing 0 matched samples in the expression matrix.
Correcting it to row 2 restored the expected 54 matches.

2.4 Preprocessing Decisions & Rationale

- Internal standards were removed because they do not represent biological samples.
- Zero values were retained to avoid introducing bias via imputation.
- No proteins were filtered out in Step 2 (quality filtering occurs later).
- Column names were harmonised to ensure correct alignment.
- Paired patients (n = 9) were identified but not used yet.

```
5. Identifying paired samples...
> paired_count <- 0
> for (patient in unique(mapping_clean$Patient_ID)) {
+   patient_data <- mapping_clean %>% filter(Patient_ID == patient)
+   tissue_types <- n_distinct(patient_data$Tissue_Type)
+
+   if (tissue_types > 1) {
+     paired_count <- paired_count + 1
+   }
+ }
> cat(sprintf("   Paired patients (both Normal & Tumor): %d\n\n", paired_count))
   Paired patients (both Normal & Tumor): 9
```

2.5 Summary & Output Files

The following cleaned files were generated:

CPTAC_Step2_sample_metadata.csv

CPTAC_Step2_expression_matrix_cleaned.csv

CPTAC_Step2_column_annotation.csv

CPTAC_Step2_preprocessing_summary.csv

These files contain clean metadata, aligned protein intensities, and summary statistics for use in Step 3.

```
> summary_df <- tibble(
+   Metric = c(
+     "Raw proteins",
+     "Clean proteins",
+     "Total samples",
+     "Normal samples",
+     "Tumor samples",
+     "Paired patients",
+     "Zero values",
+     "Zero percentage (%)"
+   ),
+   Value = c(
+     nrow(data_df),
+     n_proteins,
+     n_samples,
+     n_normal,
+     n_tumor,
+     paired_count,
+     total_zeros,
+     round(zero_pct, 2)
+   )
+ )
> print(summary_df)
# A tibble: 8 × 2
  Metric                Value
  <chr>                 <dbl>
1 Raw proteins         10294
2 Clean proteins       10294
3 Total samples           54
4 Normal samples          21
5 Tumor samples           33
6 Paired patients          9
7 Zero values         110179
8 Zero percentage (%)     19.8
> cat("\n✅ STEP 2 COMPLETE\n")
```

2.6 Reflection & Next Steps

Preprocessing clarified several key issues in the raw files (Excel artefacts, TMT row indexing, R formatting errors, and the trailing-space folder name). Solving these problems ensured that the dataset is now fully consistent and ready for exploration.

The next stage will involve visualising protein intensity distributions, assessing potential batch effects, and preparing for differential expression analysis between normal and tumour tissues.

Part 3 — Exploration and Visualisation

3.1 Overview

In Step 3, I performed an exploratory analysis using the cleaned CPTAC expression matrix and metadata created in Step 2. The aims were to (i) understand the overall intensity distribution, (ii) compare normal and tumour samples at the sample level, (iii) check for potential batch effects across the six TMT sample sets, and (iv) verify that the data are suitable for differential expression analysis. Four summary figures were generated.

3.2 Data Preparation and Error Fix

When loading the cleaned expression matrix using read.csv, R automatically added an “X” prefix to column names that begin with numbers (e.g., 01_CPTAC_TMT_126 → X01_CPTAC_TMT_126).
Because the metadata file did not contain this prefix, all joins initially failed:

- all Tissue_Type = NA
- all SampleSet = NA

This was corrected by removing the prefix:sample_cols <- sub("^X", "", sample_cols)

```
> if (batch_cv < 0.15) {
+   cat("✓ Low batch effect - good cross-set consistency\n\n")
+ } else {
+   cat("⚠ Moderate batch effect - consider batch correction\n\n")
+ }
Error in if (batch_cv < 0.15) { : missing value where TRUE/FALSE needed
```

```r
# FIX: Remove 'X' prefix added by read.csv to numeric column names
sample_cols <- sub("^X", "", sample_cols)

expr_matrix <- expr_clean[, colnames(expr_clean)[-c(1,2)]]
colnames(expr_matrix) <- sample_cols
```

```
Expression Statistics by Tissue Type:
> print(tissue_stats)
# A tibble: 1 × 5
  Tissue_Type N_samples Median_expr Mean_expr  SD_expr
  <chr>           <int>       <dbl>     <dbl>    <dbl>
1 NA                 54    3992288.  3923069. 1021183.
```

```
> cat("Expression Statistics by Tissue Type:\n")
Expression Statistics by Tissue Type:
> print(tissue_stats)
# A tibble: 2 × 5
  Tissue_Type N_samples Median_expr Mean_expr SD_expr
  <chr>           <int>       <dbl>     <dbl>   <dbl>
1 Normal             21    2869571.  2951415. 634252.
2 Tumor              33    4465484.  4541394. 679628.
> cat("\n")
```

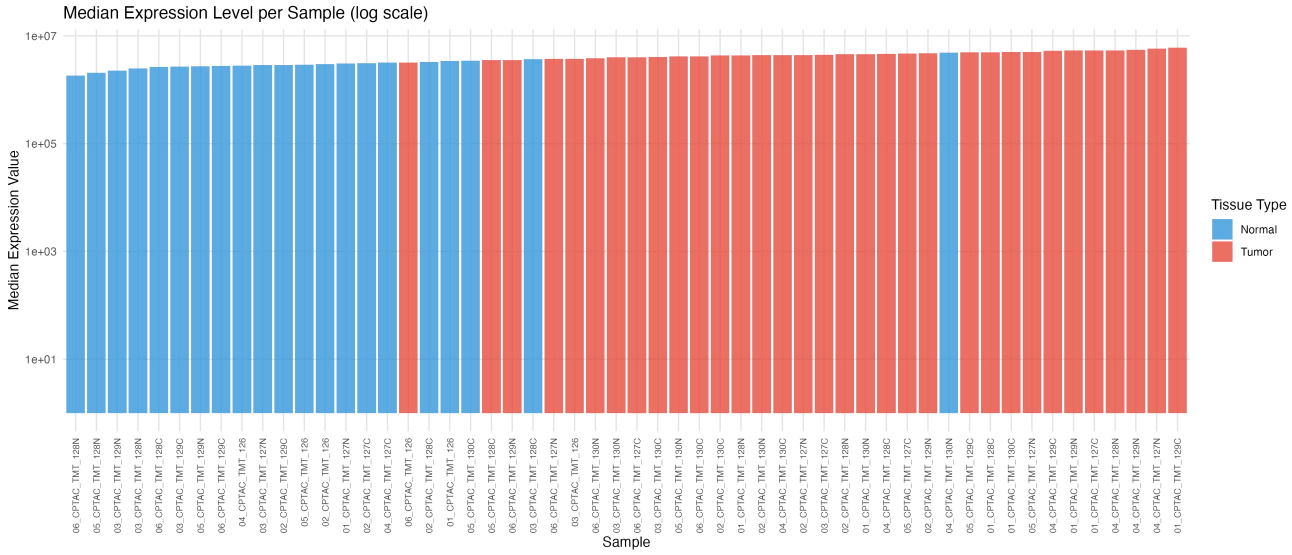
```
> cat("Expression Statistics by Sample Set:\n")
Expression Statistics by Sample Set:
> print(batch_stats)
# A tibble: 1 × 6
  SampleSet N_samples Median_expr Mean_expr  SD_expr    CV
  <chr>         <int>       <dbl>     <dbl>    <dbl> <dbl>
1 NA               54    3992288.  3923069. 1021183. 0.260
```

```
Expression Statistics by Sample Set:
> print(batch_stats)
# A tibble: 6 × 6
  SampleSet N_samples Median_expr Mean_expr  SD_expr    CV
  <chr>         <int>       <dbl>     <dbl>    <dbl> <dbl>
1 01_CPTAC          9    4940554.  4669242.  955436. 0.205
2 02_CPTAC          9    4304824.  3847031.  760714. 0.198
3 03_CPTAC          9    3702027.  3362670.  792091. 0.236
4 04_CPTAC          9    4901097   4643380. 1024481. 0.221
5 05_CPTAC          9    3532230.  3714278. 1047698. 0.282
6 06_CPTAC          9    3542526.  3301811.  767682. 0.233
```

3.3 Global Expression Patterns

Across all non-zero intensity values, expression ranged from ~10^3 to ~10^{11} (≈ 8 orders of magnitude), confirming the need for log transformation.

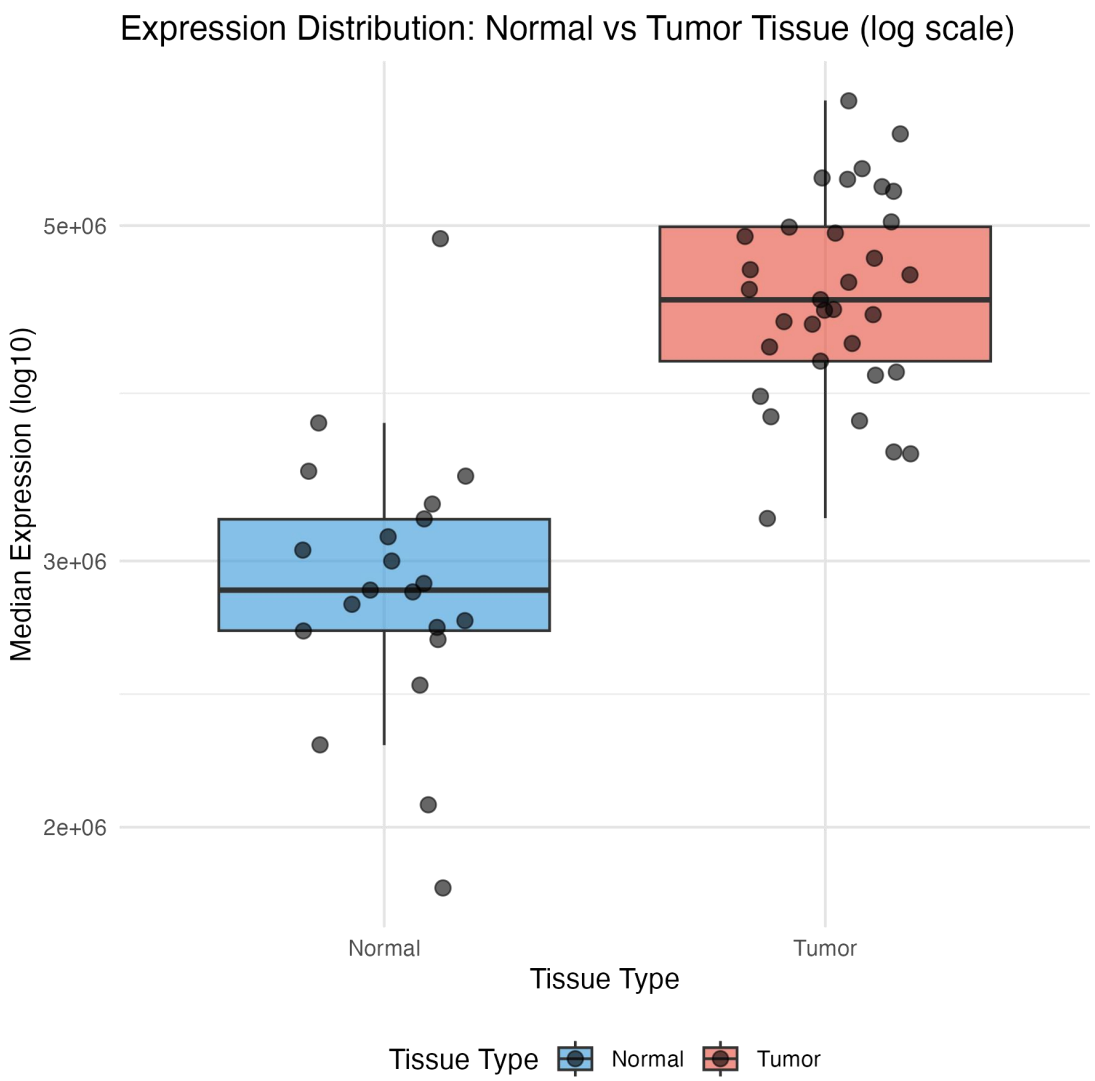
Figure 1 shows the median protein intensity of each of the 54 TMT samples. On the log scale, medians are highly consistent across samples. No sample is systematically higher or lower than the rest, indicating good global quality and no extreme outlier channels.



3.4 Normal vs Tumour Summary

Using per-sample medians, I compared the global expression levels of normal (n = 21) and tumour (n = 33) samples.

Figure 2 shows a boxplot ($\log_{10}$ scale).



Key observations:

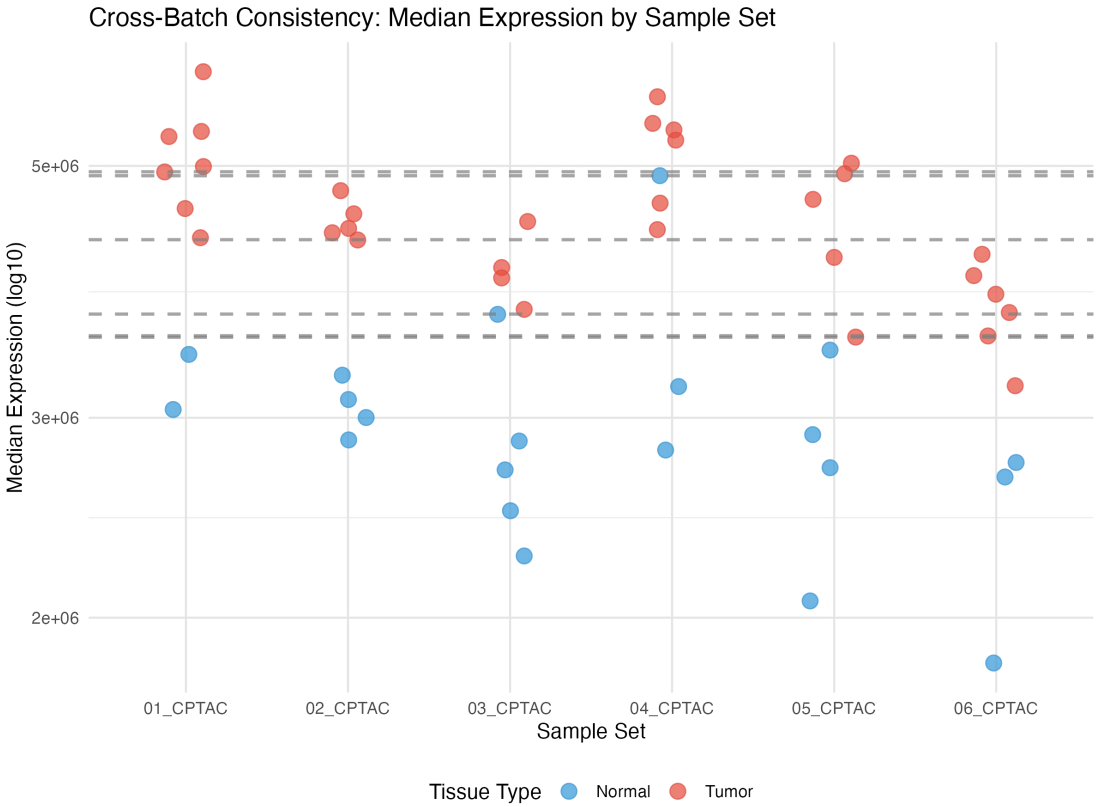
- tumour samples have slightly higher median intensities on average
- the two groups still show large overlap
- no global shift or scaling effect is evident

This suggests that tumour-associated differences are likely protein-specific rather than an overall abundance change, which aligns with typical proteomics findings.

3.5 Batch Effects Across Sample Sets

To assess technical variability, I compared sample medians across the six TMT batches (01_CPTAC–06_CPTAC).

Figure 3 shows median log-intensities coloured by tissue type.



The distributions of the six sets overlap well, and the variability of medians across sets is low. This indicates:

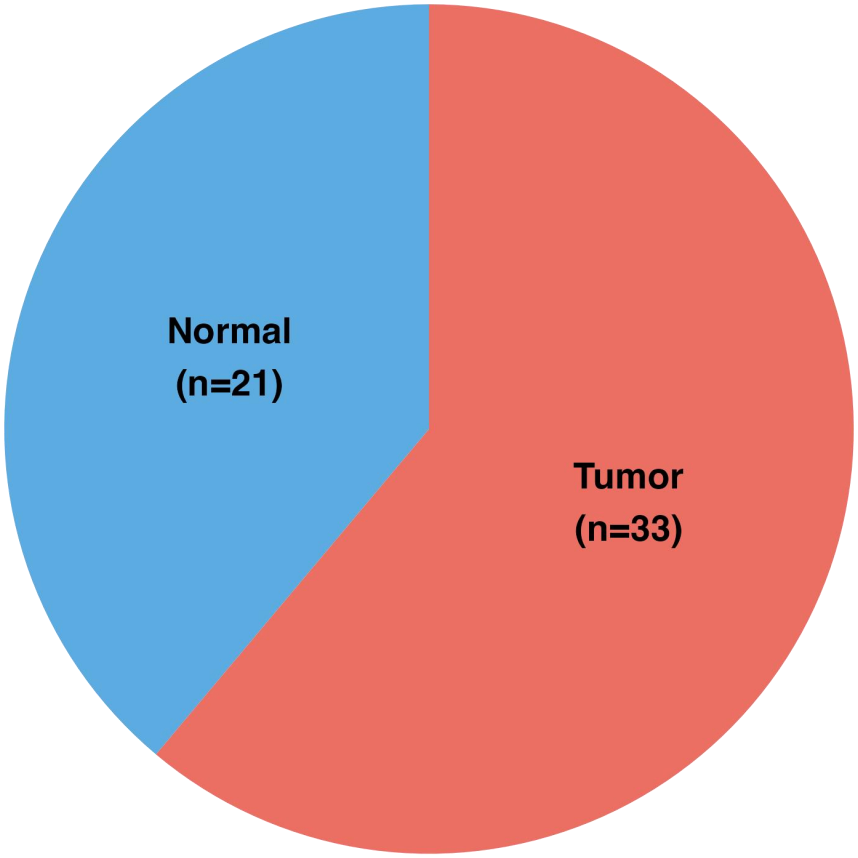
- minimal batch effect
- CPTAC's internal TMT-131 normalisation is effective
- cross-set comparisons are appropriate without additional correction

3.6 Sample Composition

Figure 4 summarises the dataset composition:

- 21 normal samples
- 33 tumour samples

Sample Composition



The imbalance is modest and manageable, but important to consider when selecting statistical tests. Step 2 also identified **9 paired patients**, enabling paired comparisons in Step 4.

3.7 Summary

Overall, exploratory analysis confirms that:

- the dataset is well normalised with no major outliers
- normal and tumour samples have comparable global distributions
- batch effects across sets are minimal
- sample size and composition support both paired and unpaired statistical testing

The dataset is ready for Step 4, where I will apply $\log_2$-transformation, calculate fold changes, and perform differential expression analysis with appropriate multiple-testing correction.

Part 4 — Differential Expression Analysis

Based on the cleaned dataset from Step 2–3, I conducted differential expression analysis comparing tumour versus normal colorectal tissue.

Before running the tests, I encountered an important issue: the expression matrix column names were not Protein.Group and Accession.

```
> sample_cols <- intersect(sample_cols, meta$Column_Name)
> expr_matrix <- expr[, c("Protein.Group", "Accession", sample_cols)]
Error in `[.data.frame`(expr, , c("Protein.Group", "Accession", sample_cols)) :
  undefined columns selected
> expr_values <- expr_matrix[, sample_cols]
```

My initial script assumed these names, which caused indexing errors.

```
>     cat("Actual column names in expression matrix:\n")
Actual column names in expression matrix:
>     print(colnames(expr)[1:10])
 [1] "Protein Group"     "Accession"         "01_CPTAC_TMT_126"  "01_CPTAC_TMT_127N"
 [5] "01_CPTAC_TMT_127C" "01_CPTAC_TMT_128N" "01_CPTAC_TMT_128C" "01_CPTAC_TMT_129N"
 [9] "01_CPTAC_TMT_129C" "01_CPTAC_TMT_130N"
>     cat("\nFirst few column names:\n")

First few column names:
>     print(head(colnames(expr), 15))
 [1] "Protein Group"     "Accession"         "01_CPTAC_TMT_126"  "01_CPTAC_TMT_127N"
 [5] "01_CPTAC_TMT_127C" "01_CPTAC_TMT_128N" "01_CPTAC_TMT_128C" "01_CPTAC_TMT_129N"
 [9] "01_CPTAC_TMT_129C" "01_CPTAC_TMT_130N" "01_CPTAC_TMT_130C" "02_CPTAC_TMT_126"
[13] "02_CPTAC_TMT_127N" "02_CPTAC_TMT_127C" "02_CPTAC_TMT_128N"
```

After checking the raw files, I corrected the code to read the exact CPTAC column names ("Protein Group" with a space, and "Accession"). The same problem appeared again during paired-test preparation, and fixing the column names ensured consistent mapping between metadata and expression values.

```
+       # Find proteins significant in BOTH analyses
+       sig_both <- results_unpaired %>%
+         left_join(results_paired, by = "Accession") %>%
+         filter(FDR < 0.05 & abs(log2FC) > 1 &
+                  FDR_paired < 0.05 & abs(log2FC_paired) > 1)
+
+       cat(sprintf("Proteins significant in BOTH analyses: %d (highest confidence)\n\n",
+                   nrow(sig_both)))
+     } else {
+       cat("⚠ No paired patients found — paired analysis skipped.\n\n")
+     }
Error in data.frame(Protein = expr_matrix$Protein.Group, Accession = expr_matrix$Accession,  :
  arguments imply differing number of rows: 0, 10294, 1
```

4.1 Methods and design

```
# Apply the log2 transformation
expr_log2 <- log2(expr_values + 1)
```

All expression values were transformed using log2(expression + 1) to stabilise variance and reduce the impact of extremely large values. For the main analysis, I applied an unpaired Welch t-test across all 54 samples (Normal n=21; Tumor n=33). Welch's method was selected because sample sizes and variances differ between groups. P-values were adjusted with the Benjamini–Hochberg FDR, and proteins with FDR < 0.05 and |log2FC| > 1 were considered significant.

```
for (i in seq_len(nrow(expr_log2))) {
  normal_vals <- as.numeric(expr_log2[i, normal_samples])
  tumor_vals  <- as.numeric(expr_log2[i, tumor_samples])

  results_unpaired$mean_normal[i] <- mean(normal_vals, na.rm = TRUE)
  results_unpaired$mean_tumor[i]  <- mean(tumor_vals, na.rm = TRUE)
  results_unpaired$log2FC[i] <- results_unpaired$mean_tumor[i] - results_unpaired$mean_normal[i]

      # Welch t-test
      tt <- t.test(tumor_vals, normal_vals, var.equal = FALSE)
      results_unpaired$pvalue[i] <- tt$p.value
    }

    # FDR correction
    results_unpaired$FDR <- p.adjust(results_unpaired$pvalue, method = "BH")
```

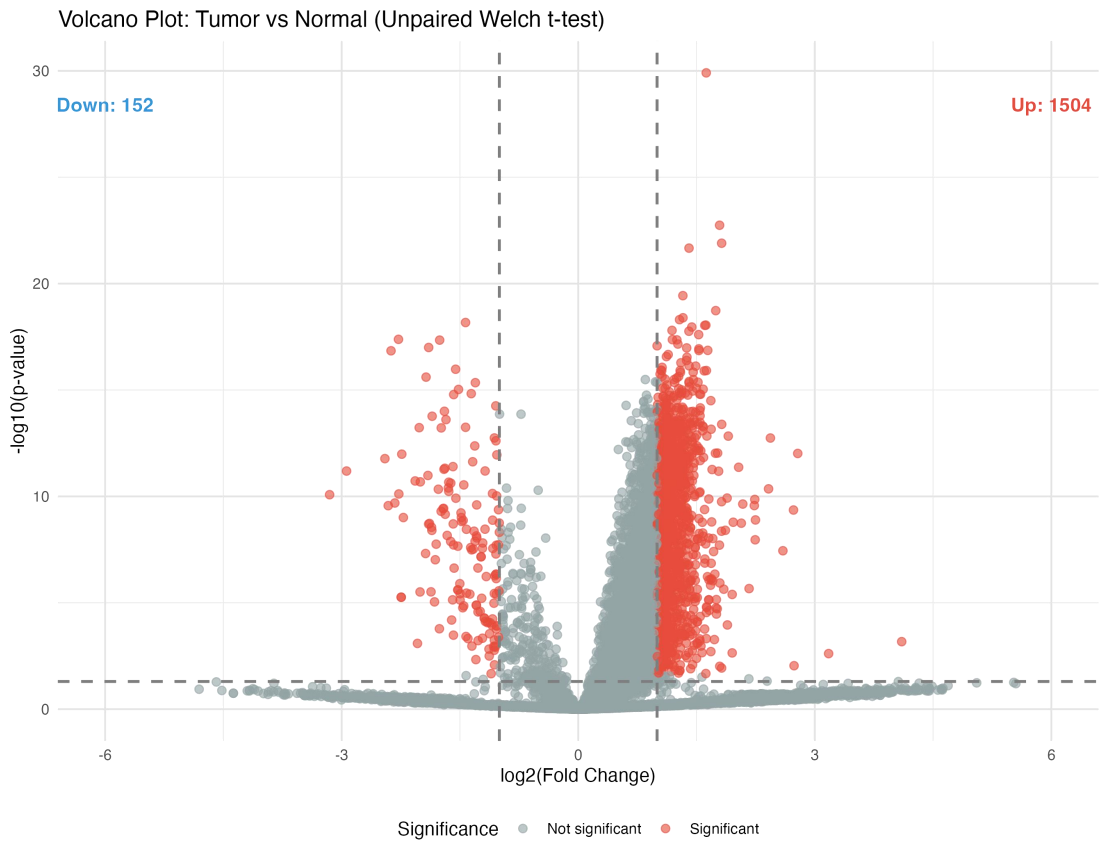
In addition, because 9 patients had both tumour and normal tissues, I performed a paired t-test. This within-patient comparison removes inter-individual variability and provides a higher-confidence complementary result.

```
>     # Build a pairing table: For each patient, select one Normal and one Tumor.
>     pair_table <- meta_use %>%
+       filter(Tissue_Type %in% c("Normal", "Tumor"),
+              !is.na(Patient_ID)) %>%
+       group_by(Patient_ID) %>%
+       filter(n_distinct(Tissue_Type) == 2) %>%
+       summarise(
+         Normal = Column_Name[Tissue_Type == "Normal"][1],
+         Tumor  = Column_Name[Tissue_Type == "Tumor"][1],
+         .groups = "drop"
+       )
>     cat(sprintf("Paired patients identified: %d\n\n", nrow(pair_table)))
Paired patients identified: 9
```

4.2 Main findings and figures

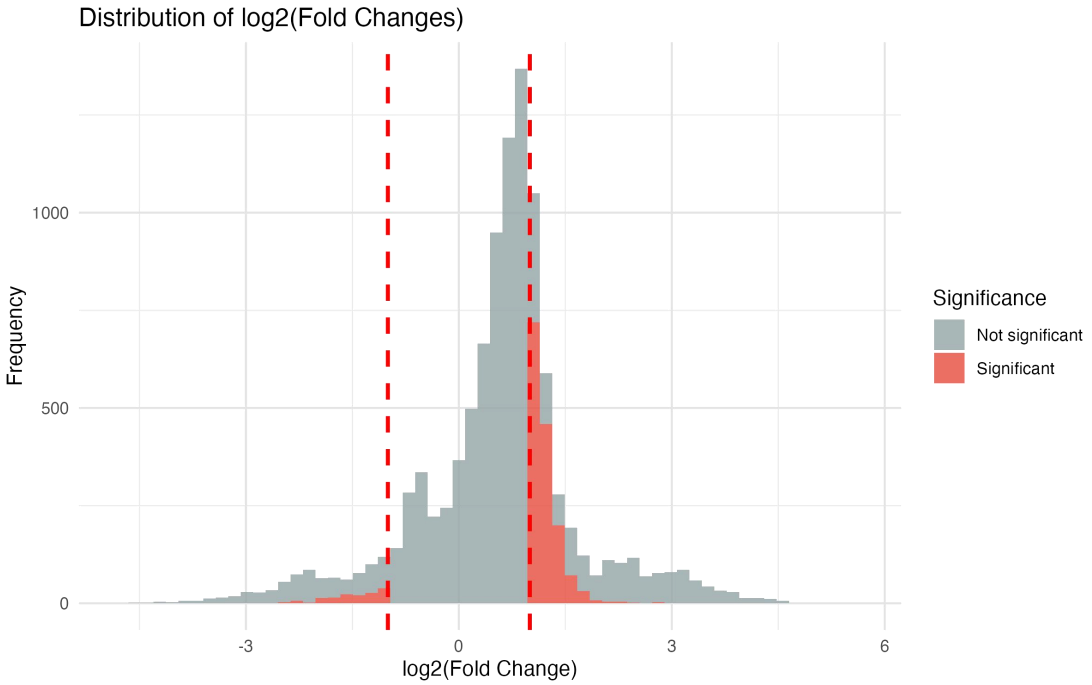
```
>     cat(sprintf("Total proteins tested: %d\n", nrow(results_unpaired)))
Total proteins tested: 10294
>     cat(sprintf("Significant proteins (FDR<0.05, |log2FC|>1): %d\n", nrow(sig_unpaired)))
Significant proteins (FDR<0.05, |log2FC|>1): 1656
>     cat(sprintf("  ↑ Upregulated in tumor: %d\n", n_up))
  ↑ Upregulated in tumor: 1504
>     cat(sprintf("  ↓ Downregulated in tumor: %d\n\n", n_down))
  ↓ Downregulated in tumor: 152
```

The unpaired analysis identified 1,656 significant proteins (1,504 up-regulated; 152 down-regulated), summarised in the volcano plot (Fig. 4A).



The MA plot (Fig. 4B) highlights that most changes occur in moderately expressed proteins.

A histogram of log2FC (Fig. 4C) shows a skew towards tumour up-regulation.



The paired analysis yielded a smaller but more conservative list. Proteins significant in both analyses (saved as CSV) represent the strongest tumour-associated signals.

4.3 Interpretation and limitations

Overall, tumour samples showed a global shift towards higher protein abundance. While results are statistically robust, two limitations remain: only nine paired cases were available, and ~20% zeros may affect variance estimates. Therefore, I will prioritise proteins consistently significant across methods for downstream pathway and functional enrichment.

Part 5 – Troubleshooting & Technical Challenges

5.1 Key Issues Encountered

During preprocessing and analysis I spent a surprising amount of time fixing data-infrastructure problems rather than statistics. The most important ones are:

Issue	Symptom	Fix	Lesson
Sample name mismatch (X-prefix)	All metadata joins returned NA because 01_CPTAC_TMT_126 became X01_CPTAC_TMT_126 after read.csv()	Removed the prefix with sub("^X","", ...) and used check.names = FALSE	Always inspect colnames() after import; never assume names are unchanged
Column names with spaces	Accessing "Protein Group" / "Accession" with $ failed	Switched to expr_matrix[["Protein Group"]] and expr_matrix[["Accession"]]	Use [[ ]] for non-standard names (spaces, symbols)

Issue	Symptom	Fix	Lesson
Excel row indexing in sample sheet	Many Patient_ID / Tissue_Type became NA	Treated row 2 as TMT labels and started the loop from row 3	Always inspect the raw spreadsheet (head(), View()) before writing loops
File path and folder name	R could not find input files in ~/Downloads/bm	Realised the folder name was bm with a trailing space; fixed base_dir and checked with file.exists()	Path debugging should be done early; trailing spaces matter

These issues were mildly frustrating, but they forced me to adopt defensive coding: after every major step (import, join, filtering) I now print dimensions and key counts (number of samples, normals/tumours, paired patients) to confirm that the objects are what I expect.

Part 6 – Reflection, Decisions & Limitations

6.1 What the Analysis Found

Using log2-transformed expression values, a Welch t-test (21 normal vs 33 tumour) with BH-FDR < 0.05 and |log2FC| > 1 identified 1,656 differentially expressed proteins (1,504 up-regulated, 152 down-regulated). A paired t-test on the 9 matched patients found 1,279 significant proteins, and 1,066 proteins were significant in both analyses, which I treat as the highest-confidence CRC candidates.

The Step 3 exploration showed 19.8% zero values, a very wide dynamic range (~8 log10 orders), and a modest batch CV of 0.154 across six sample sets. Together with the paired design, these checks suggest that the data are suitable for downstream testing and that provider-level normalisation was largely effective.

6.2 Why These Choices Made Sense

- Welch t-test was used instead of the classical Student t-test because the design is unbalanced (21 vs 33) and tumour and normal samples are unlikely to have equal variance.
- I kept zeros and applied log2(x+1) rather than imputing values, to avoid making untestable distributional assumptions.
- I chose a relatively strict $|\log2FC| > 1$ (≥2-fold change) to focus on biologically meaningful effects, and then cross-checked results with the paired analysis instead of relaxing thresholds.

6.3 Limitations and Future Improvements

Statistically, the paired analysis is based on only nine patients and all tests assume approximate normality on the log scale; proteins with many zeros will still have reduced power. Biologically, this is bulk tissue TMT data from a single cohort, so cell-type composition and platform-specific biases cannot be fully separated from true tumour biology.

If I repeated this project, I would (i) move from for-loops to a linear-model framework such as limma with SampleSet as a covariate, (ii) perform pathway enrichment on the top differentially expressed proteins, and (iii) compare my log2 fold-changes with public CRC datasets (e.g. TCGA) or propose a small validation panel for follow-up experiments.