Constructing a gene co-expression network for Head-Neck Squamous Cell Carcinoma using WCGNA

The database used: The Cancer Genome Atlas (TCGA)

I used R to perform Weighted Gene Co-expression Network Analysis (WGCNA) using gene expression data of Head-Neck Squamous Cell Carcinoma from the TCGA database using the code (TCGA-HNSC).

Following is the breakdown of how the WGSNA analysis was done to create a heatmap of gene expression data.

Cancer type: Head-neck squamous Cell Carcinoma (HNSC)

Head and neck squamous cell carcinoma (HNSC) is a significant type of cancer that arises from the epithelial cells lining the head and neck region, including areas such as the oral cavity, pharynx, and larynx. HNSC accounts for approximately 3% of new cancer cases globally and is associated with high mortality rates, particularly in advanced stages. The incidence of HNSC has been influenced by factors such as tobacco use, alcohol consumption, and human papillomavirus (HPV) infection, with HPV-positive cases showing improved survival outcomes compared to HPV-negative cases (Therapeutic approaches for the treatment of head and neck squamous cell carcinoma–An update on clinical trials).

WCGNA Analysis performed on R

1. Loading Required Libraries

```
library(WGCNA)
library(DESeq2)
library(GEOquery)
library(tidyverse)
library(CorLevelPlot)
library(gridExtra)
```

2. Fetching Data from TCGA

Used the TCGAbiolinks package to query gene expression data from TCGA (The Cancer Genome Atlas) specifically for HNSC (Head and Neck Squamous Cell Carcinoma) using RNA-Seq data.

It filters out only tumor primary (TP) and normal tissue (NT) samples and selects a small subset (10 samples each) for demonstration purposes.

```
allowWGCNAThreads()          # allow multi-threading (optional)

# 1. Fetch Data from TCGA
------------------------------------------------
if (!requireNamespace("TCGAbiolinks", quietly = TRUE)) {
  BiocManager::install("TCGAbiolinks")
}
library(TCGAbiolinks)

query_TCGA_HNSC <- GDCquery(
  project = "TCGA-HNSC",
  data.category = "Transcriptome Profiling",
  experimental.strategy = "RNA-Seq",
  workflow.type = "STAR - Counts",
  data.type = "Gene Expression Quantification")
#get a full list of barcodes
samplesDown <- getResults(query_TCGA_HNSC, cols=c("cases"))

#identify barcodes with Tumor Primary samples
dataSmTP <- TCGAquery_SampleTypes(
  barcode = samplesDown,
  typesample = "TP"
)

#identify barcodes with Normal Tissue
dataSmNT <- TCGAquery_SampleTypes(
  barcode = samplesDown,
  typesample = "NT"
)

#Choose a small subset of 10 from each.
#Omit this step for full analysis.
dataSmTP_short <- dataSmTP[1:10]
dataSmNT_short <- dataSmNT[1:10]
```

It downloads the selected data and prepares it for further processing by saving it as a .RData file and performing basic preprocessing steps like filtering out low-quality data.

```
#Create a query for only the selected samples
query.selected.samples <- GDCquery(
  project = "TCGA-HNSC",
  data.category = "Transcriptome Profiling",
  experimental.strategy = "RNA-Seq",
  data.type = "Gene Expression Quantification",
  workflow.type = "STAR - Counts",
  barcode = c(dataSmTP_short, dataSmNT_short) #only for a subset of
barcodes
)

#Download the data
GDCdownload(query = query.selected.samples)
##Confirm in the console if the download starts
```

3. Preparing a summary of Data

```
#prepare a summary of all the data
# Prepare a summary of all the data using the temp directory
dataPrep <- GDCprepare(
  query = query.selected.samples,
  save = TRUE,
  save.filename =
"TCGA-HNSCTranscriptome_ProfilingSun_Nov_24_22_01_44_2024.RData", #
specified the filename to solve the error
  summarizedExperiment = TRUE  # this ensures proper data structure
)

data_counts <- TCGAanalyze_Preprocessing(
  object = dataPrep,
  cor.cut = 0.6,  # filter out if correlation between samples is below this
threshold
  datatype = "unstranded",
  filename = file.path( "preprocessing_plot.pdf"),  # save plot to temp
directory
  width = 1000,
  height = 1000
)
```

4. Quality Control (QC) - Outlier Detection

Used the goodSamplesGenes() function from WGCNA to detect genes that may be outliers based on the gene expression data.

It applies hierarchical clustering (hclust) and Principal Component Analysis (PCA) to detect outlier samples. These outliers may skew the results, so they are identified and excluded from the data.

```
# detect outlier genes

gsg <- goodSamplesGenes(t(data_counts))
summary(gsg)
gsg$allOK

table(gsg$goodGenes)
table(gsg$goodSamples)

# remove genes that are detected as outliers
data_counts <- data_counts[gsg$goodGenes == TRUE,]

# detect outlier samples - hierarchical clustering
htree <- hclust(dist(t(data_counts)), method = "average")
plot(htree)

# PCA for detecting sample outliers
pca <- prcomp(t(data_counts))
pca.dat <- pca$x

pca.var <- pca$sdev^2
pca.var.percent <- round(pca.var/sum(pca.var)*100, digits = 2)

pca.dat <- as.data.frame(pca.dat)

ggplot(pca.dat, aes(PC1, PC2)) +
  geom_point() +
  geom_text(label = rownames(pca.dat)) +
  labs(x = paste0('PC1: ', pca.var.percent[1], ' %'),
       y = paste0('PC2: ', pca.var.percent[2], ' %'))
```

```
### NOTE: If there are batch effects observed, correct for them
before moving ahead

##If no outlier samples to be excluded
data.subset <- data_counts
```

5. Normalization (Using DESeq2)

The script uses DESeq2 to normalize the data by stabilizing the variance across genes. This step ensures that gene expression values are comparable across different samples.

Genes with low expression (less than 15 counts in more than 75% of the samples) are excluded to focus on the most reliable data.

```
# 3. Normalization (using variance stabilization)
# create a deseq2 dataset
# get the required colData table for dds

reqBarcodes <- colnames(data.subset)
req_clin_info <- c("shortLetterCode", "tumor_grade", "gender",
"alcohol_history")
req_colData <- as.data.frame(colData(dataPrep)[reqBarcodes, req_clin_info])

# create dds
dds <- DESeqDataSetFromMatrix(countData = data.subset,
                              colData = req_colData,
                              design = ~ 1) # not specifying model

## remove all genes with counts < 15 in more than 75% of samples
(31*0.75=23.25)
## suggested by WGCNA on RNAseq FAQ

dds75 <- dds[rowSums(counts(dds) >= 15) >= 15,]
nrow(dds75) # 13284 genes

# perform variance stabilization
dds_norm <- vst(dds75)

# get normalized counts
norm.counts <- assay(dds_norm) %>%
```

```
  t()
```

6. Network Construction

The script uses WGCNA's pickSoftThreshold() function to choose a "soft-thresholding" power that determines the network's topology (how genes are interconnected) → creates a scale-free network. This step involves determining the optimal power for constructing a network of gene co-expression.

Displays how well the network satisfies the scale-free topology criterion for each power. A threshold (red line) of R^2 = 0.8 is often used. Based on the visualization, the first best soft power (16) was chosen.

WGCNA's blockwiseModules() function is applied to identify gene modules (groups of genes with similar expression patterns) based on the selected soft threshold.

```
# 4. Network Construction
-----------------------------------------------------
# Choose a set of soft-thresholding powers
power <- c(c(1:10), seq(from = 12, to = 50, by = 2))

# Call the network topology analysis function
sft <- pickSoftThreshold(norm.counts,
                         powerVector = power,
                         networkType = "signed",
                         verbose = 5)

sft.data <- sft$fitIndices

# visualization to pick power
a1 <- ggplot(sft.data, aes(Power, SFT.R.sq, label = Power)) +
  geom_point() +
  geom_text(nudge_y = 0.1) +
  geom_hline(yintercept = 0.8, color = 'red') +
  labs(x = 'Power', y = 'Scale free topology model fit, signed R^2')
+
  theme_classic()

a2 <- ggplot(sft.data, aes(Power, mean.k., label = Power)) +
```

```
  geom_point() +
  geom_text(nudge_y = 0.1) +
  labs(x = 'Power', y = 'Mean Connectivity') +
  theme_classic()

grid.arrange(a1, a2, nrow = 2)

# convert matrix to numeric
norm.counts[] <- sapply(norm.counts, as.numeric)

soft_power <- 16
temp_cor <- cor
cor <- WGCNA::cor

# memory estimate w.r.t blocksize
bwnet <- blockwiseModules(norm.counts,
                          maxBlockSize = 14000,
                          TOMType = "signed",
                          power = soft_power,
                          mergeCutHeight = 0.25,
                          numericLabels = FALSE,
                          randomSeed = 1234,
                          verbose = 3)

cor <- temp_cor
```

7. Module Eigengenes

The script computes the **module eigengenes**, which represent the first principal component (PC) of each gene module. These are used to summarize the gene expression profiles of the modules.

Dendrogram and Color Plot: It visualizes the hierarchical clustering of modules in a dendrogram and shows how modules were merged or split based on their expression similarity.

Ensembl IDs for genes in each module are mapped to gene symbols and names using the org.Hs.eg.db package. This provides more interpretable information about the genes in each module.

```
# 5. Module Eigengenes
```

```
-----------------------------------------------------------
module_eigengenes <- bwnet$MEs

# Print out a preview
head(module_eigengenes)

# get number of genes for each module
table(bwnet$colors)

# Plot the dendrogram and the module colors before and after merging
underneath
plotDendroAndColors(bwnet$dendrograms[[1]],
cbind(bwnet$unmergedColors, bwnet$colors),
                    c("unmerged", "merged"),
                    dendroLabels = FALSE,
                    addGuide = TRUE,
                    hang= 0.03,
                    guideHang = 0.05)

# grey module = all genes that doesn't fall into other modules were
assigned to the grey module

##Get gene symbols and names from IDs
if (!requireNamespace("org.Hs.eg.db", quietly = TRUE)) {
  BiocManager::install("org.Hs.eg.db")
}
library(org.Hs.eg.db)

# List of Ensembl IDs
col_oi <- "cyan"
  ensembl_ids <- names(which(bwnet$colors==col_oi))
  ensembl_ids <- sapply(ensembl_ids, function(x) strsplit(x, split =
"\\.")[[1]][1])

  # Map Ensembl IDs to gene symbols
  gene_symbols <- mapIds(
    org.Hs.eg.db,
    keys = ensembl_ids,
    column = "SYMBOL",   # Retrieve HGNC symbols
```

```
    keytype = "ENSEMBL", # Input type
    multiVals = "first"  # If multiple matches, return the first
  )

  # Map Ensembl IDs to gene names
  gene_names <- mapIds(
    org.Hs.eg.db,
    keys = ensembl_ids,
    column = "GENENAME", # Retrieve gene names/descriptions
    keytype = "ENSEMBL", # Input type
    multiVals = "first"  # If multiple matches, return the first
  )

  # Combine results into a data frame
  gene_info <- data.frame(
    Ensembl_ID = ensembl_ids,
    Gene_Symbol = gene_symbols,
    Gene_Name = gene_names,
    stringsAsFactors = FALSE
  )

  # View the results
  View(gene_info)

  ## expression patterns of module genes
  expression_oi <- norm.counts[,names(which(bwnet$colors==col_oi))]
  colLabels_oi <- apply(req_colData[rownames(req_colData) %in%
rownames(norm.counts), ], 1, paste, collapse = "_")
  heatmap(margins = c(15,15), x = t(expression_oi), labCol =
colLabels_oi, labRow = gene_info$Gene_Name)
```

8. Heatmap of Gene Expression

The script generates a heatmap to visualize the expression patterns of genes in a selected module (e.g., cyan module). It plots the expression levels of module genes across the samples, which can help identify patterns related to tumor vs. normal tissue or other clinical factors.

Parameters

1. Downloading the TCGA-HNSC dataset

Data category: Transcriptome profiling
Experimental strategy: RNA sequencing data.

2. Tumor type

TP: First 10 primary tumor
NT: First 10 Normal tumor

3. req_colData columns for plotting heatmap (x-axis)

Type of tumor, tumor grade that is reported, gender of the patient, and if there's any alcohol history (as alcohol intake is one of the major factors contributing to the development of HNSC).

4. Soft thresholding power selection plot based on R^2

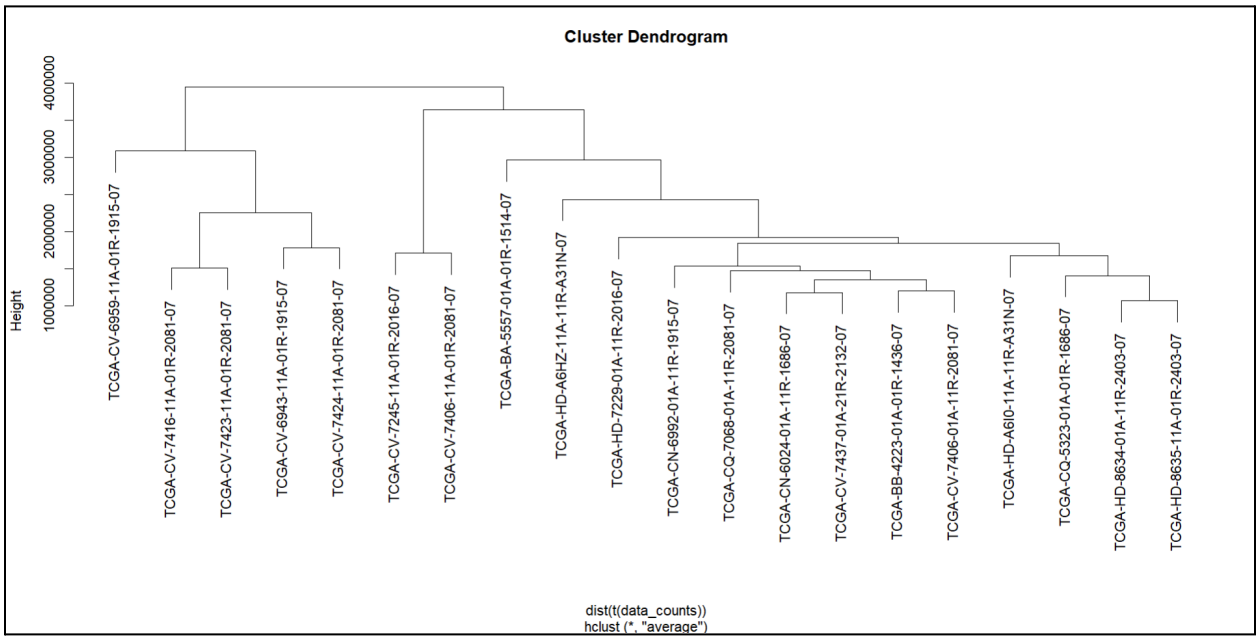
The soft scale thresholding value I chose was **16**, as it is the first best value above the threshold. I could have chosen a value higher than this (which would have given a better scale-free topology), but this could have resulted in high noise and we could lose on important networks.

5. Color selection for the dataset which needs to be plotted on heatmap

I tried with different color datasets (turquoise, cyan, and pink). This was randomly chosen based on the high/low dataset (gene) available in that color.

Plots and data

Cluster Dendrogram




```
plot(htree)
```

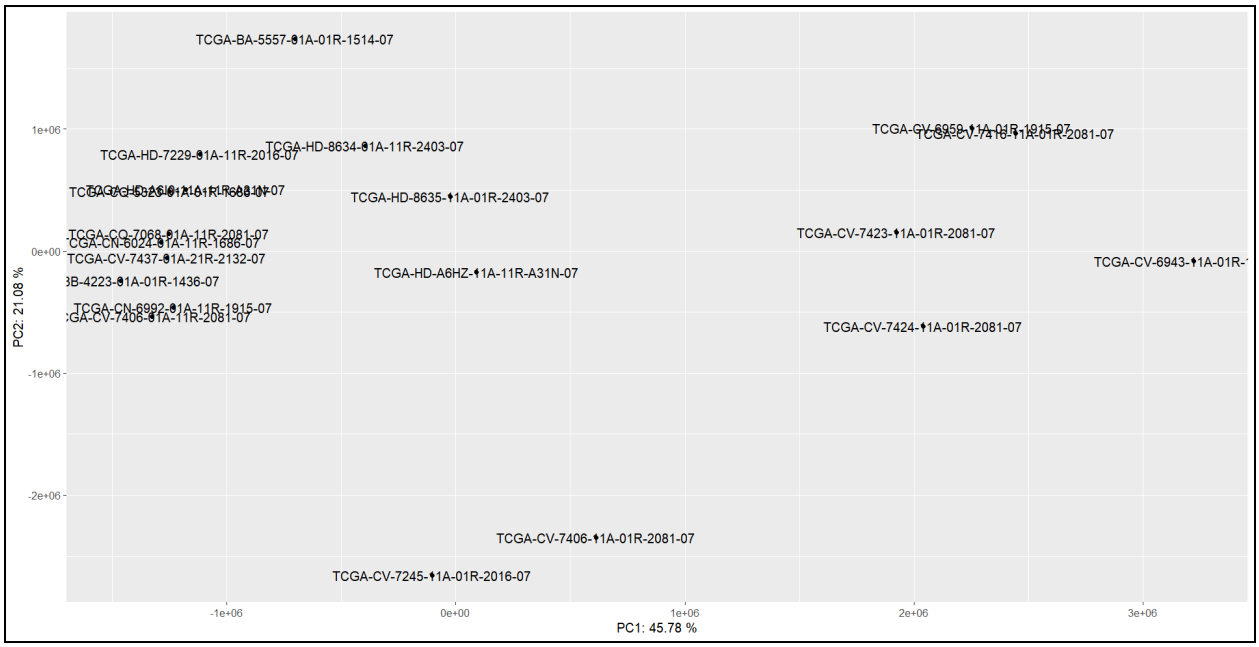
This is a **cluster dendrogram**, typically used in hierarchical clustering to display the relationships between samples or features based on a distance metric.

X-Axis (Labels): Represents the individual samples or data points (e.g., TCGA identifiers). Each label corresponds to a sample being clustered.

Y-Axis (Height): Represents the distance or dissimilarity between clusters. The greater the height at which two branches merge, the more dissimilar the clusters are.

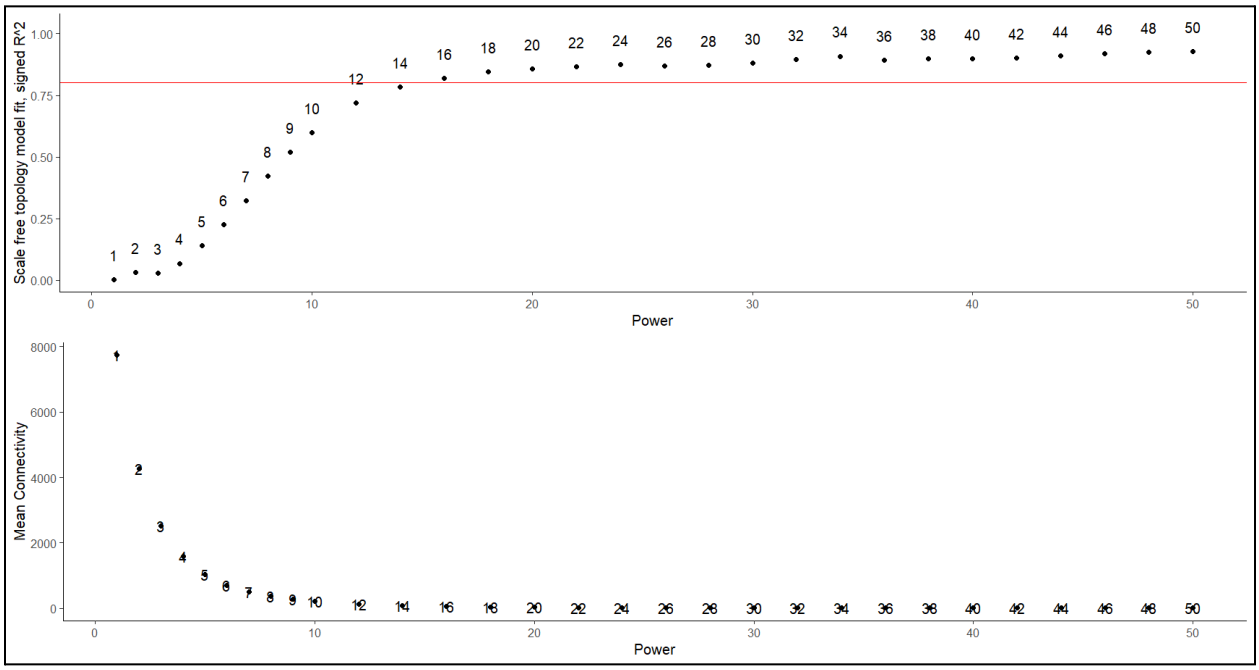
Tree Structure (Dendrogram): The tree-like structure shows how samples are hierarchically grouped based on similarity. Closely related samples are merged at lower heights, while less related samples join at higher points.

PCA



This graph helps identify outliers. The distant samples are outliers.

Soft thresholding power selection plot based on R^2



In WGCNA, one crucial step is determining the **soft-thresholding power** to achieve a scale-free topology in the network (followed by most biological networks). The goal is to identify a power where the network approximates a scale-free topology (fit index ≈ 0.85) while maintaining sufficient connectivity among genes.

Biological networks tend to exhibit a scale-free property, meaning most genes have few connections, while a few hub genes have many connections.

The **soft scale thresholding value I chose was 16**, as it is the first best value above the threshold. I could have chosen a value higher than this (which would have given a better scale-free topology), but this could have resulted in high noise and we could lose on important networks.

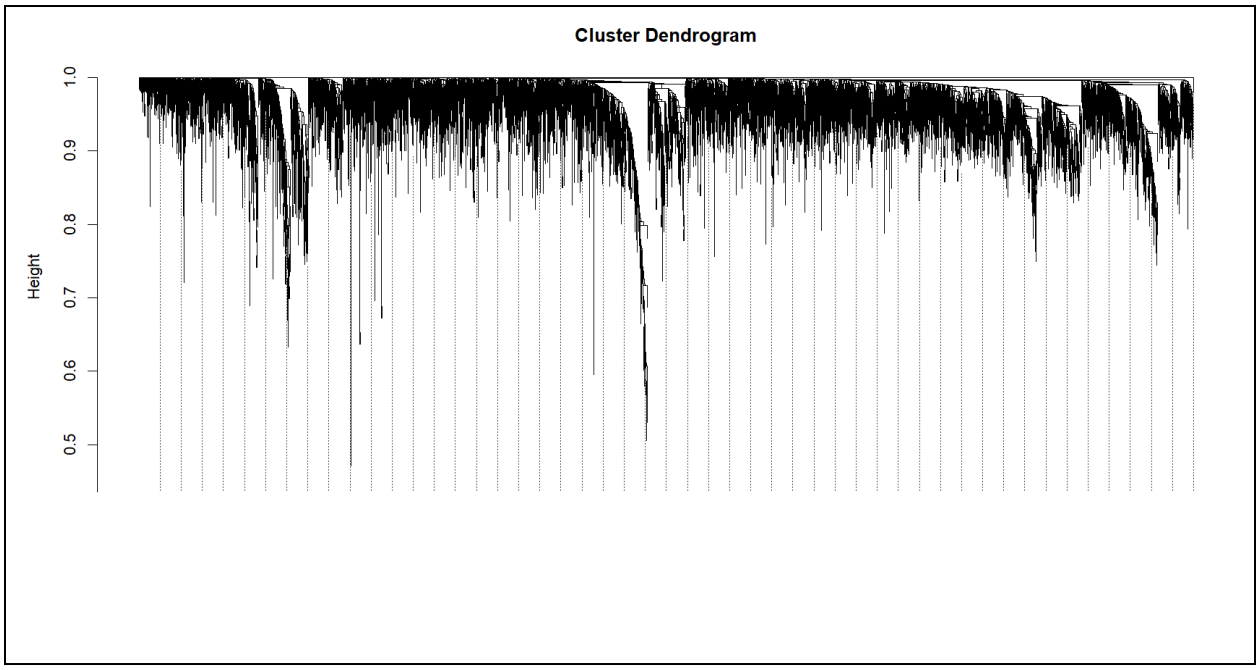
List of colors for different modules

```
> table(bwnet$colors)

      black        blue       brown        cyan       green greenyellow        grey       grey60   lightcyan 
        356        1027         993          88         391         151        8796          47          48 
lightgreen lightyellow     magenta midnightblue        pink      purple         red   royalblue      salmon 
         34          32         183          50         206         169         372          31          90 
        tan   turquoise      yellow 
        112        1314         977 
```

For the analysis and heatmap plotting, I first chose turquoise but didn't get a heatmap with a clear pattern. Then I randomly plotted for different colors.

Cluster Dendrogram for merged and unmerged modules



I tried to plot the table for different colors and make the merged and unmerged table, but it didn't happen and was showing an **error in the code.**

Gene Info table (first few rows)

	Ensembl_ID	Gene_Symbol	Gene_Name
ENSG00000011478.13	ENSG00000011478	QPCTL	glutaminyl-peptide cyclotransferase like
ENSG00000018610.15	ENSG00000018610	STEEP1	STING1 ER exit protein 1
ENSG00000019995.6	ENSG00000019995	ZRANB1	zinc finger RANBP2-type containing 1
ENSG00000035403.18	ENSG00000035403	VCL	vinculin
ENSG00000070778.13	ENSG00000070778	PTPN21	protein tyrosine phosphatase non-receptor type 21
ENSG00000077312.9	ENSG00000077312	SNRPA	small nuclear ribonucleoprotein polypeptide A
ENSG00000080371.6	ENSG00000080371	RAB21	RAB21, member RAS oncogene family
ENSG00000087088.20	ENSG00000087088	BAX	BCL2 associated X, apoptosis regulator
ENSG00000089248.7	ENSG00000089248	ERP29	endoplasmic reticulum protein 29
ENSG00000100028.12	ENSG00000100028	SNRPD3	small nuclear ribonucleoprotein D3 polypeptide
ENSG00000100296.14	ENSG00000100296	THOC5	THO complex subunit 5
ENSG00000102078.16	ENSG00000102078	SLC25A14	solute carrier family 25 member 14
ENSG00000103174.13	ENSG00000103174	NAGPA	N-acetylglucosamine-1-phosphodiester alpha-N-ace...
ENSG00000105185.12	ENSG00000105185	PDCD5	programmed cell death 5
ENSG00000106299.8	ENSG00000106299	WASL	WASP like actin nucleation promoting factor
ENSG00000107771.17	ENSG00000107771	CCSER2	coiled-coil serine rich protein 2
ENSG00000108239.9	ENSG00000108239	TBC1D12	TBC1 domain family member 12
ENSG00000109381.20	ENSG00000109381	ELF2	E74 like ETS transcription factor 2
ENSG00000109686.19	ENSG00000109686	SH3D19	SH3 domain containing 19
ENSG00000109756.9	ENSG00000109756	RAPGEF2	Rap guanine nucleotide exchange factor 2
ENSG00000111775.3	ENSG00000111775	COX6A1	cytochrome c oxidase subunit 6A1
ENSG00000113742.14	ENSG00000113742	CPEB4	cytoplasmic polyadenylation element binding protei...
ENSG00000115128.7	ENSG00000115128	SF3B6	splicing factor 3b subunit 6

The table shows the entire list of genes listed in the dataset. The ensemble ID's and the respective gene names. —> Some shown on heatmap.

Req_ColData (Data to be plotted on heatmap)

	shortLetterCode	tumor_grade	gender	alcohol_history
TCGA-BA-5557-01A-01R-1514-07	TP	Not Reported	female	Yes
TCGA-BB-4223-01A-01R-1436-07	TP	Not Reported	male	No
TCGA-CN-6024-01A-11R-1686-07	TP	Not Reported	male	Yes
TCGA-CN-6992-01A-11R-1915-07	TP	Not Reported	male	Yes
TCGA-CQ-5323-01A-01R-1686-07	TP	Not Reported	male	Yes
TCGA-CQ-7068-01A-11R-2081-07	TP	Not Reported	female	No
TCGA-CV-7406-01A-11R-2081-07	TP	Not Reported	male	Yes
TCGA-CV-7437-01A-21R-2132-07	TP	Not Reported	male	No
TCGA-HD-7229-01A-11R-2016-07	TP	Not Reported	male	No
TCGA-HD-8634-01A-11R-2403-07	TP	Not Reported	female	Yes
TCGA-CV-6943-11A-01R-1915-07	NT	Not Reported	male	Yes
TCGA-CV-6959-11A-01R-1915-07	NT	Not Reported	male	Yes
TCGA-CV-7245-11A-01R-2016-07	NT	Not Reported	male	No
TCGA-CV-7406-11A-01R-2081-07	NT	Not Reported	male	Yes
TCGA-CV-7416-11A-01R-2081-07	NT	Not Reported	female	Not Reported
TCGA-CV-7423-11A-01R-2081-07	NT	Not Reported	male	No
TCGA-CV-7424-11A-01R-2081-07	NT	Not Reported	male	No
TCGA-HD-8635-11A-01R-2403-07	NT	Not Reported	female	Yes
TCGA-HD-A6HZ-11A-11R-A31N-07	NT	Not Reported	female	No
TCGA-HD-A6I0-11A-11R-A31N-07	NT	Not Reported	male	Yes

There were more columns listed in the table, I removed some of the columns which did not have a difference and would not be helpful in the interpretation of the data.

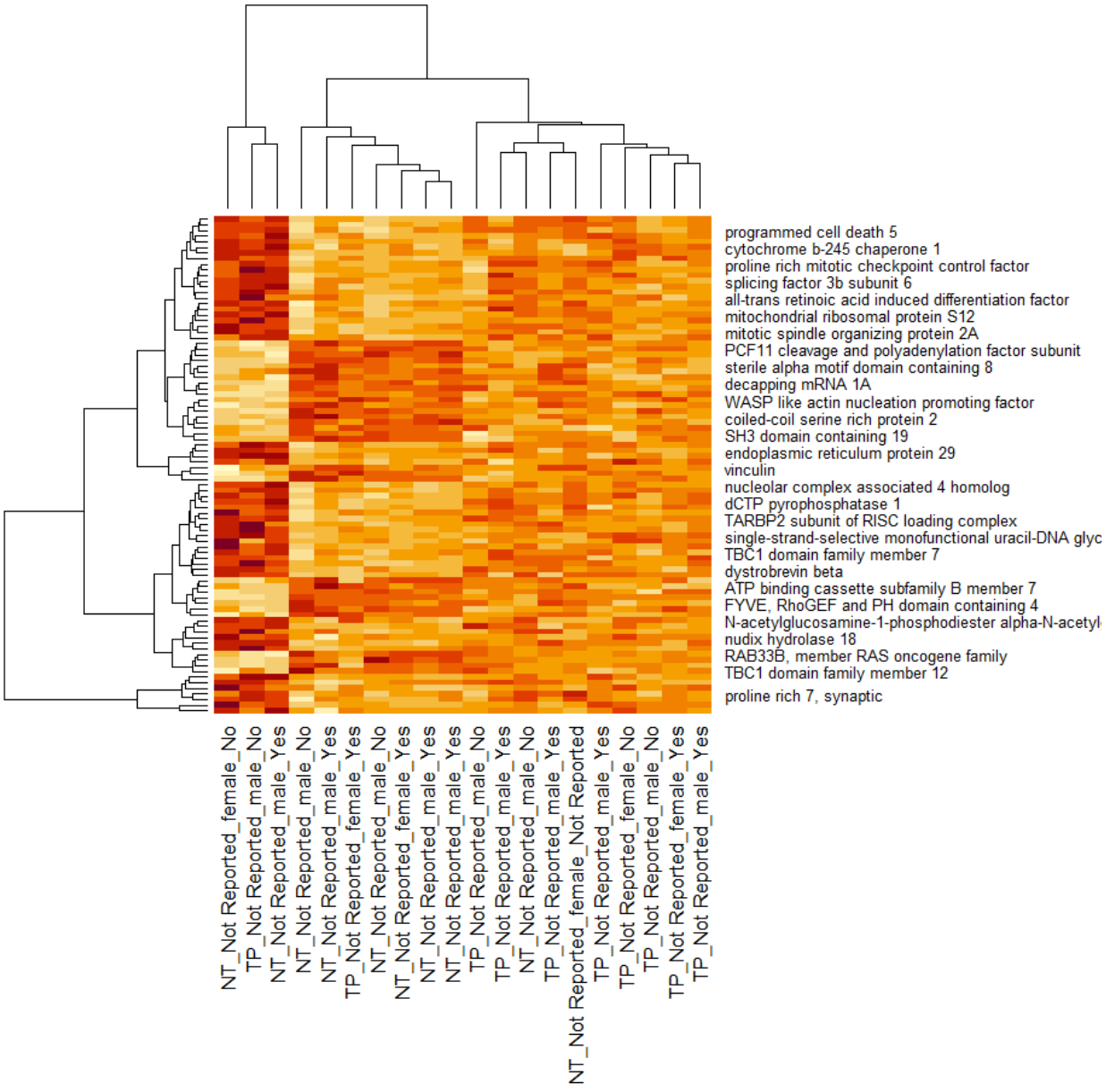
```
reqBarcodes <- colnames(data.subset)
req_clin_info <- c("shortLetterCode", "tumor_grade", "gender",
"alcohol_history")
req_colData <- as.data.frame(colData(dataPrep)[reqBarcodes,
req_clin_info])
```

I chose the parameters: Type of tumor, tumor grade that is reported, gender of the patient and if there's any alcohol history (as alcohol intake is one of the major factors contributing to the development of HNSC).

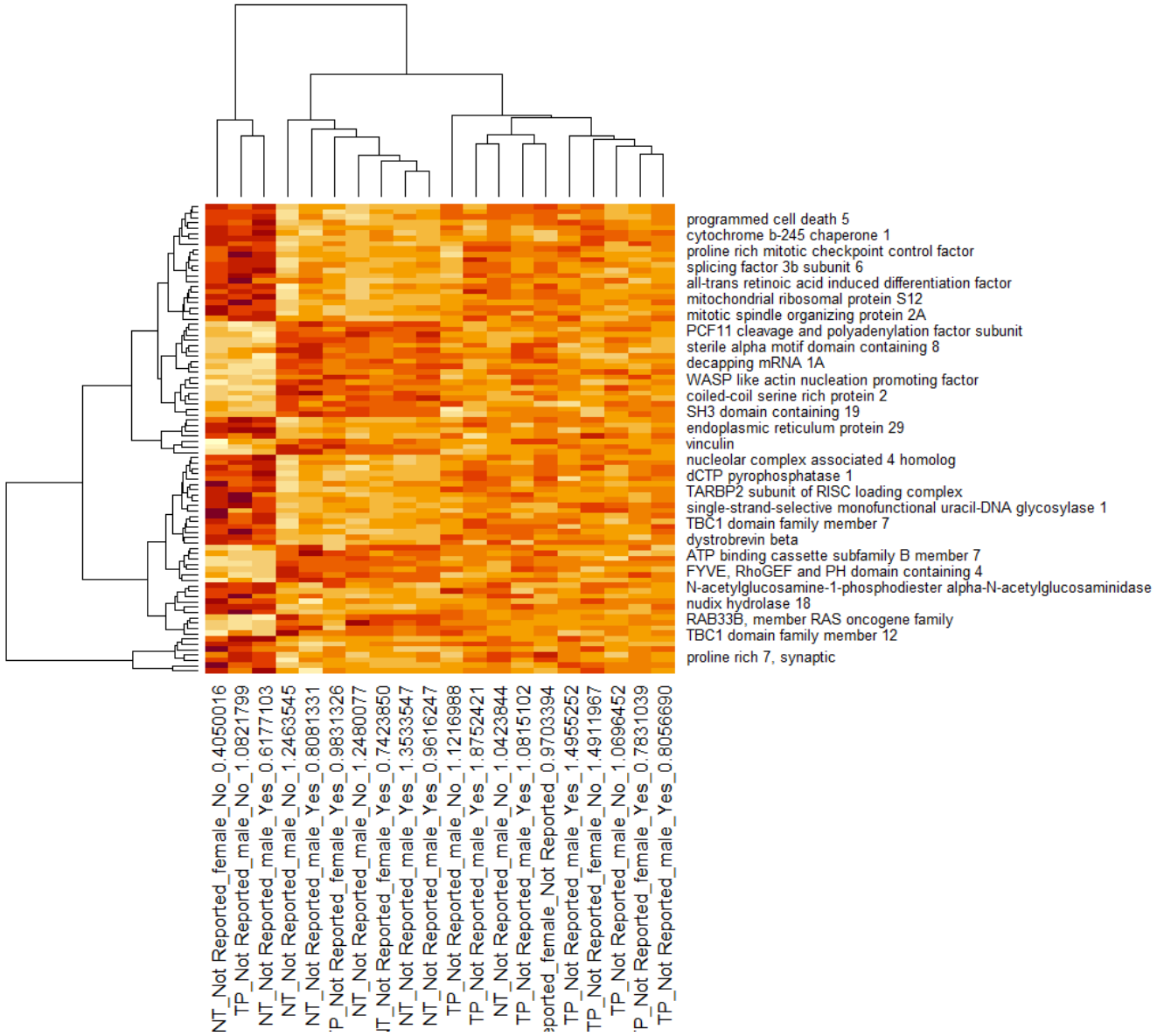
Analysis and Interpretation of the Heatmap for HSNA Cancer from TCGA Data

Heatmap plot: Turquoise

First I plotted the color turquoise for the heatmap. So all the data in this module was observed. However, I did not see any clear pattern in gene regulation based on cancer. This module has a large dataset of 1314.



Heatmap plot: Cyan



Features:

Columns: Represent samples, and patient data categorized into different conditions or attributes:

- TP: Tumor Primary
- NT: Normal Tissue
- Gender or clinical data annotated (e.g., "male", "female", "Not Reported/reported (tumor grade)").
- Alcohol history: Yes/No

Rows: Represent individual genes, which are labeled with their names. These genes might be linked to biological pathways or processes relevant to cancer.

Color Scale: Indicates the relative expression level of genes:

- **Red/Orange**: High expression.
- **Yellow/White**: Low expression.

Clustering:

- **Samples (Columns)**: Grouped based on similarity in gene expression patterns.
- **Genes (Rows)**: Grouped based on co-expression or similarity in behavior across samples.

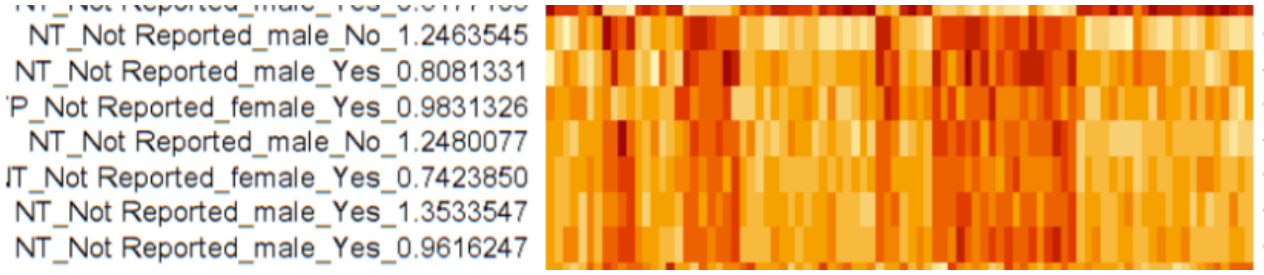
Interpretation and Insights:

The first three columns exhibit a distinct expression pattern with certain genes consistently overexpressed (red/orange) and others underexpressed (white/yellow). This block-like pattern suggests systematic differences rather than random noise, indicating that these samples are biologically or technically unique compared to the rest.
The pattern is unlikely to be cancer-specific, as it is absent in other tumor samples. It could result from comorbidities (e.g., systemic inflammation, infections), ethnic or geographic variations (e.g., population-specific genetic polymorphisms or environmental influences), or other conditions affecting gene expression. Differences in metabolic genes or immune response genes between populations from distinct geographic regions (e.g., African vs. Asian cohorts) → affected by specific diet and lifestyle.

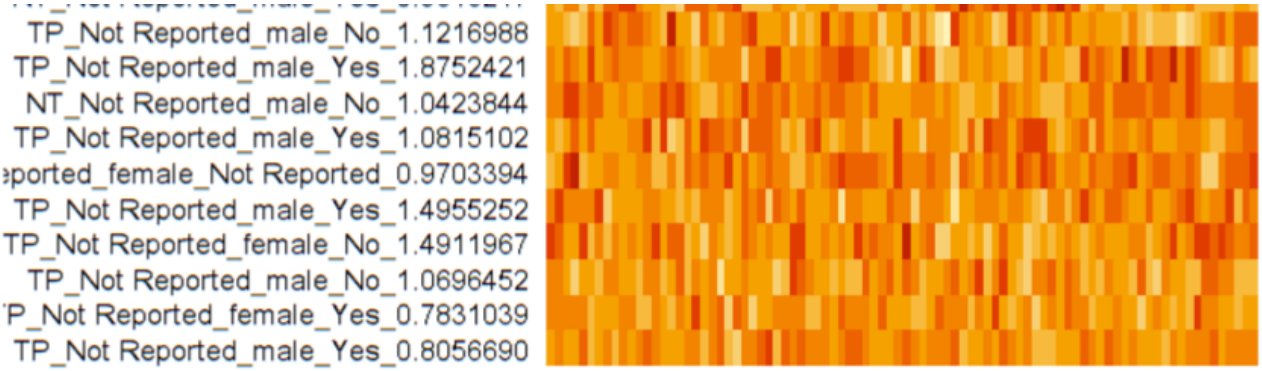
(Note: I have taken screenshots of parts of the heatmap, to explain the pattern. The screenshot is just rotated to fit)

In the next seven columns, the gene expression pattern is opposite to that observed in the first three columns. Regions, where certain genes are highly expressed (red/orange) in the first three columns, show lower expression (white/yellow) in these seven columns, and vice versa. While this trend is not extremely strong, it is still noticeable and worth investigating.
While most of the data from the 7 columns are for normal and only 1 for tumor, we can't really tell if the observation is cancer-dependent. However, the pattern in this compared to the first 3 columns

could be due to differences in other factors such as age, geographic locations, type of diet and lifestyle and any other underlying factors.



Overall, I cannot observe any pattern that can show cancer (HSCN) dependent gene expression. The pattern is quite random, with certain patterns but the reason behind it cannot be specified just on the basis of this data.



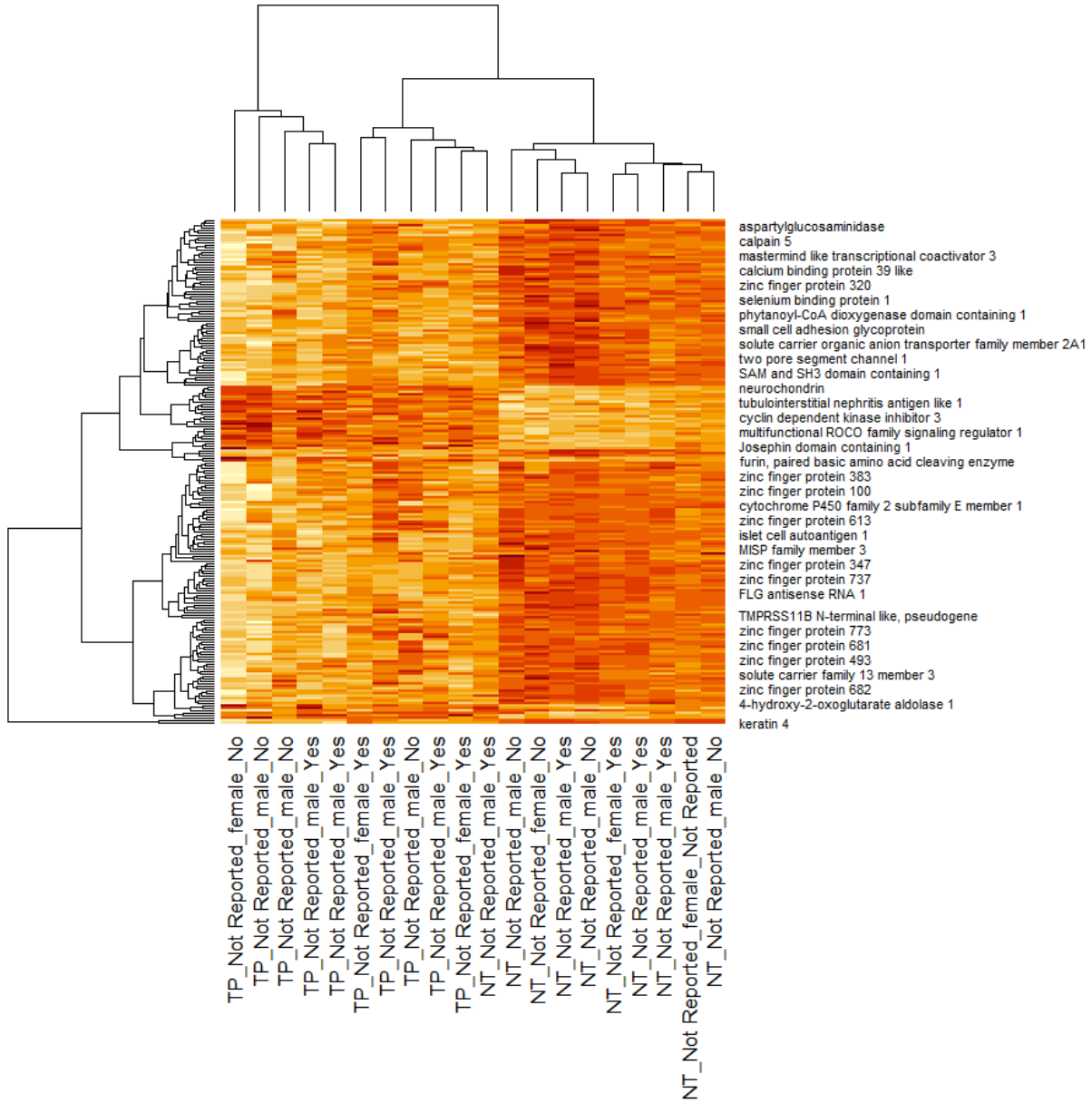
Heatmap plot: Pink

In the above two plots, I did not see a clear pattern of gene expression data based on the tumor type, so I plotted the heatmap for another color - Pink.

Sample Types:

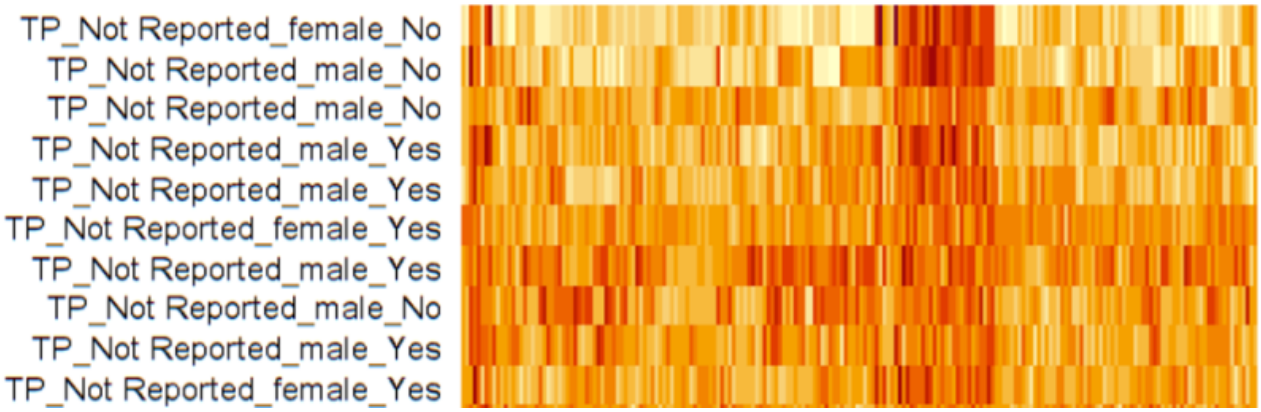
- **TP**: Primary tumors.
- **NT**: Normal adjacent tissues.
- Gender and alcohol history are also noted for each sample, providing demographic and clinical stratification.

Gene Clustering: Genes are hierarchically clustered based on expression patterns. Some genes show distinct expression in certain clusters of samples, suggesting differential expression between tumor and normal samples or based on gender/alcohol consumption history.

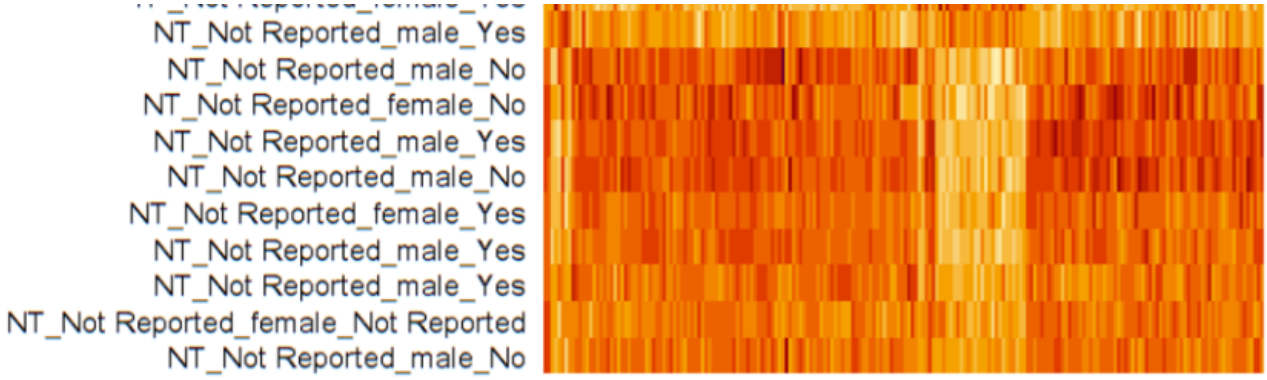


The first 10 columns are for primary tumor type and the next 10 are for normal tumor type.

Primary tumor gene expression pattern: There are three main blocks of expression pattern. There are set of genes that are highly expressed (red/orange), and another set where another set of genes are less expressed comparatively (white/yellow/orange.



Normal tumor gene expression pattern: In comparison to the TP columns, for the NT, gene expression is high overall. With a small block where gene expression is low.



Outlier: There is one NT column that matches the pattern of the TP dataset.

Overall, there are certain sets of genes that are highly expressed in normal tumors and those same genes are less expressed in cancer types. And this is a major trend. Biologically, this pattern could be because this cancer type could result in the downregulation of certain pathways that lead to the downregulation of genes involved in the particular pathway.

Conclusion

The gene expression heatmap analysis for Head-Neck Squamous Cell Carcinoma (HNSC) showed distinct patterns among different colors (modules). For certain modules, I could not draw a clear gene expression pattern. However, in the Pink module, there was a clear pattern in the primary tumor and normal tumor. There were sets of genes differentially expressed between the two.

Further exploration using Weighted Gene Co-expression Network Analysis (WGCNA) highlights co-expressed gene modules, which could uncover meaningful biological associations. Integrating clinical data with gene expression profiles is crucial to understanding these discrepancies. Identifying patient-specific factors that contribute to this variability is essential for advancing personalized medicine approaches in HNSC.