HW4_FinalProject_Dougherty

Mark Dougherty

2023-05-10

Unsupervised Clustering and Heatmap Analysis of Human Schwannoma Metabolites

Mark Dougherty
Project Homework #4 - BIOL 4386: Intro to Scientific Computing
Due May 1, 2023
Github repo: https://github.com/Intro-Sci-Comp-UIowa/biol-4386-course-project-doughertymc

Reference Masalha, W., Daka, K., Woerner, J. *et al.* Metabolic alterations in meningioma reflect the clinical course. *BMC Cancer* 21, 211 (2021). https://rdcu.be/c5yzG

Introduction Schwannomas are benign (WHO grade 1) tumors that grow on peripheral nerves, originating from the Schwann cells that myelinate the nerve axons. Vestibular schwannomas are those that occur on the vestibular nerve intracranially, and account for about 8% of all primary brain tumors. Surgery and radiation are first-line treatments for these tumors, but if they fail there is no second-line therapy. Thus, novel medical treatments are needed. What's more, although all schwannomas are considered Grade 1 tumors, some are more aggressive than others; we do not currently have a biological explanation for this inter-tumor heterogeneity or a good way to predict this behavior in order to modify our clinical care. Recent literature has suggested a possible role of the tumor-immune microenvironment, as there are differences in macrophage infiltration between different tumors. Another recent development in the literature is that DNA methylation profiling can identify tumors with more aggressive phenotypes better than genomic or transcriptomic analysis, suggesting that schwannomas may be driven largely by *epigenetic* changes rather than the classical genetic mutation paradigm in cancer.

My research focuses on trying to improve our understanding of the underlying biology of schwannomas in hope of finding vulnerabilities that could serve as drug targets, as well as further explain differences in tumor recurrence/aggressiveness. Specifically, I have been using metabolomics, which is a method of analyzing levels of many (~100-150) metabolites in a tissue at a given time. Metabolomic analysis has not previously been used to study schwannomas, but has shown promise in finding novel treatment targets in other tumors/cancers. Thus, our aims are twofold: identify novel drug targets in metabolic pathways, and identify biological differences that might explain differences in tumor behavior. In the future, I also hope to integrate this analysis with other '-omics' data and clinical outcomes, but that is likely beyond the scope of the current project.

In this project, my aim is to use R to process data from metabolomic analysis of primary human schwannoma samples, and then use unsupervised clustering analysis and a heatmap to evaluate whether there are meaningful clusters of tumors that seem to be metabolically similar. I will use the same unsupervised PAM cluster analysis method as the authors of this paper use, although at this time I do not know the specifics of this cluster method or why they chose it over other unsupervised clustering methods. This is an exploratory analysis, so it is possible that I will not find clean clustering as is the case with the reference figure. On the other hand, if I do identify strong data clusters, further steps would then be needed to determine what the groups/clusters mean. In order to evaluate the clustering visually, I will combine the clustering analysis to arrange the metabolites, and then visualize with a heatmap as in Figure 2A of Masalha *et al.* Notably, I do not intend to perform the analyses in parts B & C of the same figure, nor do I anticipate including a 'Silhouette width' graph as they do at the top part of their figure.

As a secondary aim of this project, I hope to apply a similar process to describe the effect of radiation on patient-derived schwannoma xenografts. As with the primary schwannomas, we already have data from these specimens, but unlike the primary tumors we also have treatment groups (radiation/control) that can be compared.

Figure to reproduce: Figure 2A

Materials and Methods

Specimen collection

- Schwannoma specimens are collected directly from surgical patients at UIHC. A *primary tumor* specimen is flash-frozen in liquid nitrogen in the operating room. When available, additional tissue is implanted in 8-9 nude mice per human tumor (*patient-derived xenografts*); after the mice recover (~2-4 weeks), these xenografts are treated with radiation (0, 10, 20 Gy) and harvested 72 hours post-treatment. The *primary tumors* and *xenografts* are then metabolically profiled with GC-MS and/or LC-MS (AKA metabolomics).

Data Preprocessing & Cleaning

- The UI Metabolomics Core performs the mass spectrometry analysis and provides the data to our lab as relative concentrations of each metabolite in a **labeled Excel spreadsheet**. Each sample has ~100-150 metabolite levels measured. *Critically, these are relative levels rather than absolute concentrations. This means that we can compare one metabolite between different samples (e.g. glutamine 2x higher in Sample X than in Sample Y), but we cannot directly compare levels of different metabolites (e.g. cannot state "glutamine is 2x higher than glutamate").*
- Non-metabolic information must then be manually associated with samples. For example, in the above figure 2A this would include Edema, Proliferation, Gender, and WHO Grade at the bottom of the heatmap. Our samples are labeled with: NF2 status (categorical), prior radiation (categorical), prior surgery (categorical), and proliferation (continuous; from EdU assay, [xenografts only]). Some samples also have freeze time data (continuous)
- Data will then be imported from Excel into R

Materials & Methods - Data Analysis Part One: Primary Tumor Samples

- Double check that undesired samples are excluded from further analyses (eg 2022.3.11 (S35) known ischemic sample) **DONE**
- **Clustering & Heatmap**
  - As described in the Methods of the journal article cited above, I set out to perform cluster analysis on my data using their R package **AutoPipe**, which they make available on GitHub.

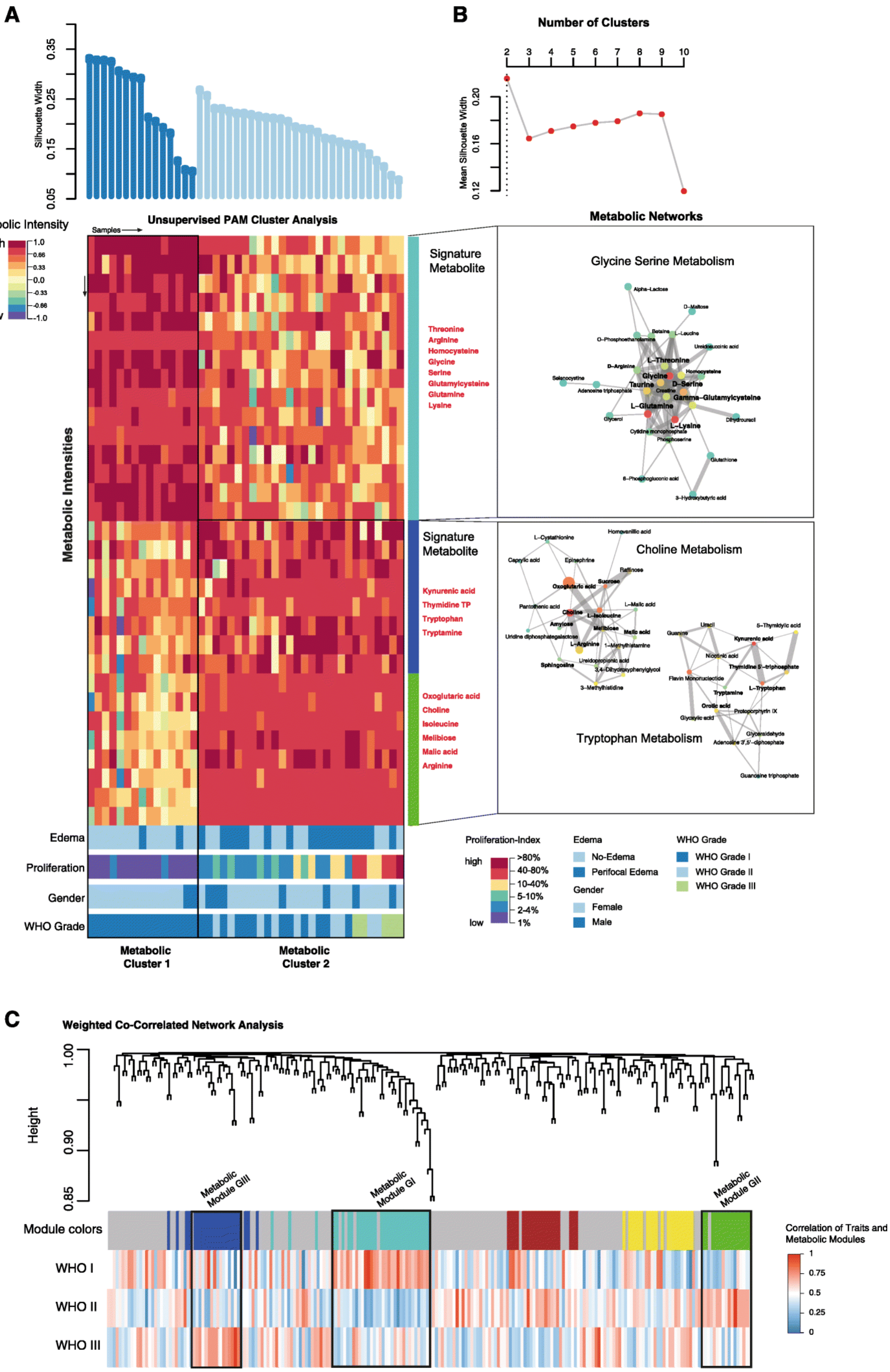


Figure 1: Figure 2A taken from Masalha *et al*

Materials & Methods - Data Analysis Part Two: Radiation effect in Schwannoma Xenografts

- **Normalization**: Fold change calculations (xenografts only)
  - *Xenograft fold change calculations are complicated because want to normalize to mean of control group for each specific tumor*
  - Example: Tumor from patient 123 was implanted into 9 mice, and those 9 mice were randomized to 0, 10, or 20 Gy radiation treatment (3 per group). Tumor from patient 456 was also implanted into 9 mice. However, to evaluate the fold change of a given metabolite after radiation, we want to compare the 10 & 20 Gy treatment groups from Patient 123 to the control tumors of Patient 123, and Radiated Tumors from Patient 456 to Control Tumors from Patient 456.
- **Outlier detection**: Grubbs' test, alpha = 0.01.
- **Test for normality** - Shapiro-Wilk test
- **Transformation** (when needed) - for non-normally distributed metabolites, LogTransform the values
- **Statistical analysis (xenografts)**
  - Correlation with radiation dose (per metabolite)

- Two-way ANOVA with Holm-Sidak test for xenografts to compare radiation treatment doses
- **Graphs**
- Graph of average fold change by radiation dose (0-10-20 Gy) per metabolite, only selecting the metabolites with correlation with radiation dose > 0.25

Results

Results - Data Analysis Part 1:

- Broadly speaking, although it took significant effort to implement and the R Package from the Masalha publication was NOT user-friendly, I was able to apply the clustering functions from their package 'AutoPipe' to test for the optimal cluster number (n=2 was best), and then apply PAM (partitions around medioids) clustering to my data. Thus, the baseline goal was achieved
- However, the figure that was produced with this was not easily modified to include metabolite names or improved formatting. Thus, in its current state it is not suitable for publication, but hopefully I can find a way to improve upon this visually. Ironically, although the PAM clustering was successful, due to the poor visualization it is difficult to see exactly how the tumors clustered and which metaboiltes were the basis for said clustering.

```
# Part 1: Vestibular Schwannoma Primary Tumor Clustering Data Analysis
# Source of AutoPipe: https://github.com/falafel19/AutoPipe
# Reference: Masalha et al (2021). https://rdcu.be/c5yzG

# Import raw data file from CSV to tibble using read_csv
vs_primary_metabolomics_raw <- read_csv("C:/Users/mark1/Dropbox/BIOL_4386/Project_Folder/Formatted_Data/
```

```
## New names:
## Rows: 43 Columns: 166
## -- Column specification
## -------------------------------------------------------- Delimiter: "," chr
## (2): Sample_Label, Location dbl (152): Freeze_Time_Seconds, Prior_surgery,
## Prior_Radiation, NF2, 2-Hydro... lgl (12): 3-Hydroxyanthranilic acid,
## Aminoadipate, Gluconic acid, Histamine...
## i Use ‘spec()‘ to retrieve the full column specification for this data. i
## Specify the column types or set ‘show_col_types = FALSE‘ to quiet this message.
## * ‘‘ -> ‘...113‘
```

```
# Remove missing metabolite columns and save as curated tibble. Note that columns 1-6 are metadata.
vs_primary_curated <- vs_primary_metabolomics_raw %>% select(!where(is_logical))
metabolite_names_primary <- colnames(vs_primary_curated)[-(1:6)]
# Note that must remove sample S15 to run clustering because S15 is missing all LCMS data (~half of met
vs_primary_curated <- vs_primary_curated %>% filter(Sample_Label != "S15") %>% filter(Sample_Label != "
view(vs_primary_curated)

vs_clinical_data <- vs_primary_curated[c(1,3,4,5,6)]
vs_clinical_data_df <- vs_clinical_data %>% column_to_rownames(var = "Sample_Label") %>% as.data.frame(

#Remove metabolite columns with missing data, and remove metadata columns 2:6
vs_primary_no_missing <- vs_primary_curated[-(2:6)] %>% select(where(~all(!is.na(.))))
view(vs_primary_no_missing)
```

```
### NOTE: CANNOT RUN THIS STUFF PRIOR TO USING TIDYVERSE IN ABOVE CHUNK BECAUSE LOADING THESE PACKAGES
#Must convert from tibble to dataframe for AutoPipe::TopPAM to work; this also converts the column Samp
vs_primary_df <- vs_primary_no_missing %>% column_to_rownames(var = "Sample_Label") %>% as.data.frame(.
class(vs_primary_df)
```

```
## [1] "data.frame"
```

```
vs_transposed <- t(vs_primary_df)
# Run AutoPipe's TopPAM feature to calcluate optimal number of clusters using PAM clustering. NOTE this
res <- AutoPipe::TopPAM(vs_transposed, max_clusters = 15, TOP=139, B=100, clusterboot=FALSE)
```

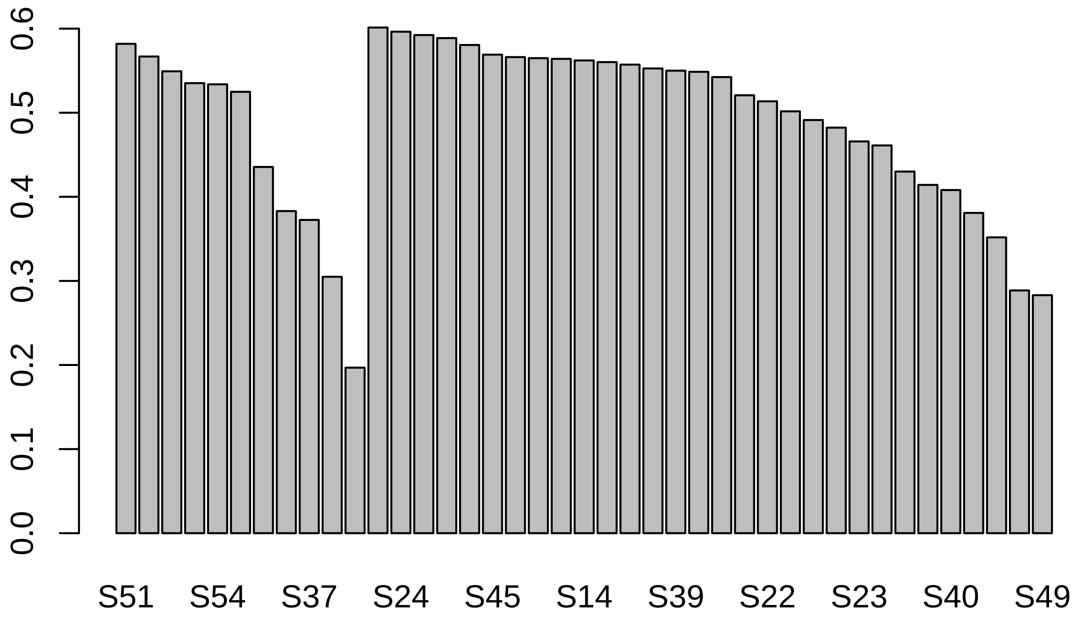
```
## [1] "Cluster with k=2"
## [1] "Cluster with k=3"
## [1] "Cluster with k=4"
## [1] "Cluster with k=5"
## [1] "Cluster with k=6"
## [1] "Cluster with k=7"
## [1] "Cluster with k=8"
## [1] "Cluster with k=9"
## [1] "Cluster with k=10"
## [1] "Cluster with k=11"
## [1] "Cluster with k=12"
## [1] "Cluster with k=13"
## [1] "Cluster with k=14"
## [1] "Cluster with k=15"
```

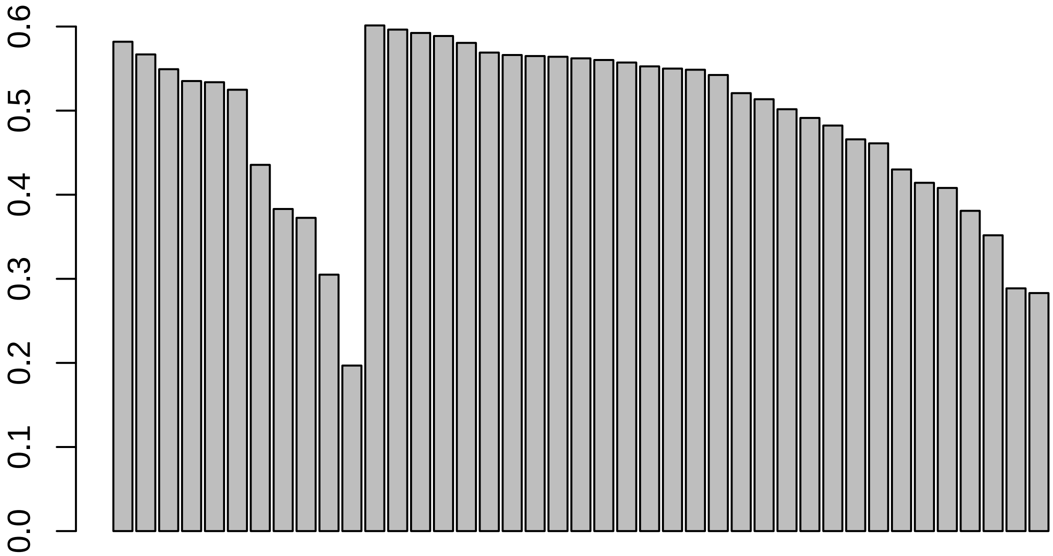
```
# TopPAM result -> 2 groups are best for PAM clustering, but one of them is just sample S46

me_TOP <- res[[1]]
dim(me_TOP)
```

```
## [1] 139  41
```

```
number_of_k <- res[[3]]
File_genes <- AutoPipe::Groups_Sup(me_TOP, me = vs_transposed, number_of_k,TRw=-1)
```



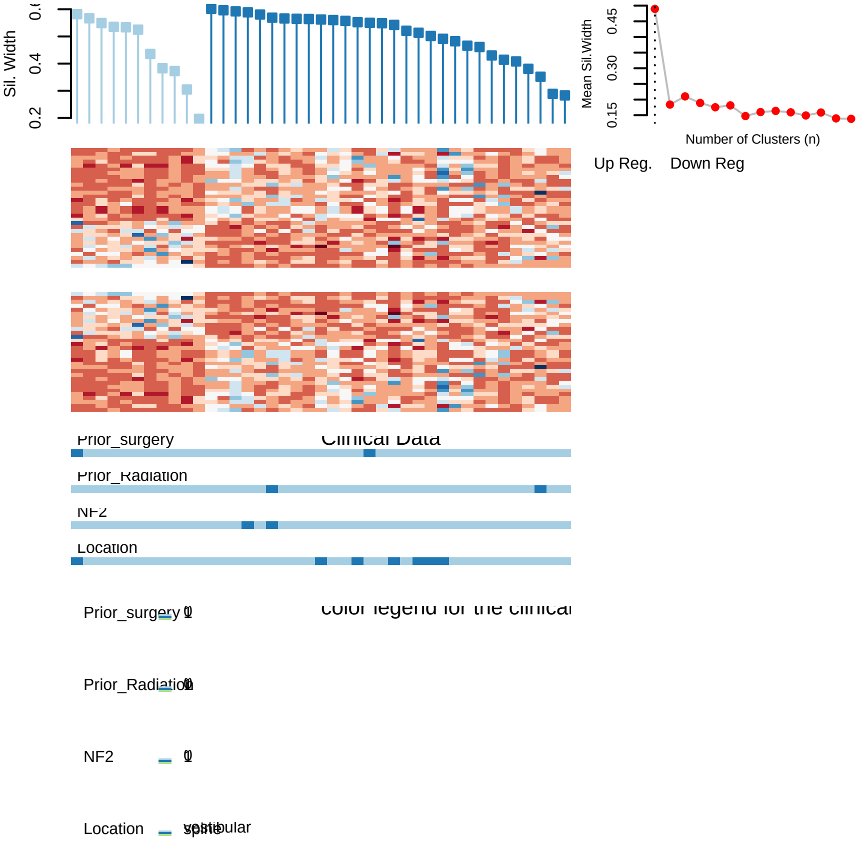


```
groups_men=File_genes[[2]]

AutoPipe::Supervised_Cluster_Heatmap(groups_men = groups_men, gene_matrix=File_genes[[1]], TOP_Cluster=
```

```
## 12345678910111213141516171819202122232425262728293012Fold 1 :1234567891011121314151617181920212223242
## Fold 2 :123456789101112131415161718192021222324252627282930
## Fold 3 :123456789101112131415161718192021222324252627282930
## Fold 4 :123456789101112131415161718192021222324252627282930
## Fold 5 :123456789101112131415161718192021222324252627282930
## Fold 6 :123456789101112131415161718192021222324252627282930
## Fold 7 :123456789101112131415161718192021222324252627282930
## Fold 8 :123456789101112131415161718192021222324252627282930
## Fold 9 :123456789101112131415161718192021222324252627282930
## Fold 10 :123456789101112131415161718192021222324252627282930
##       id                                    1-score 2-score
##  [1,] GSH                                   3.2572  -1.1943
##  [2,] Mannose                               0.2066  -0.0757
##  [3,] XMP                                   -0.2044 0.075
##  [4,] UDP                                   0.1333  -0.0489
##  [5,] NADH                                  -0.107  0.0392
##  [6,] alpha-Keto-beta-Methylvalerate (KMV) -0.1035 0.038
##  [7,] alpha-Ketoisovalerate (KIV)          -0.1005 0.0369
##  [8,] CMP                                   -0.0982 0.036
##  [9,] N-Acetylaspartate                     0.0969  -0.0355
## [10,] UMP                                   -0.088  0.0323
## [11,] N-Acetyltyrosine                    0.0865  -0.0317
## [12,] Alanine                             0.0839  -0.0308
## [13,] dGDP                                -0.0834 0.0306
## [14,] dAMP                                -0.0693 0.0254
## [15,] Glycerate                           -0.0679 0.0249
## [16,] Malonate                            0.06    -0.022
## [17,] Pyridoxal (PL)                      -0.0592 0.0217
## [18,] AMP                                 -0.0539 0.0198
## [19,] Xylose                              -0.0536 0.0196
## [20,] Fructose                            -0.0475 0.0174
## [21,] O-Phosphoethanolamine               0.0379  -0.0139
## [22,] Lauric acid                         0.0366  -0.0134
## [23,] Tryptophan                          -0.0318 0.0117
## [24,] Adonitol                            -0.0261 0.0096
## [25,] Gamma-aminobutyric acid (GABA)      0.0248  -0.0091
## [26,] Tridecanoic acid                    0.0174  -0.0064
## [27,] GSSG                                -0.0137 0.005
## [28,] Indolelactic acid                   -0.0056 0.002
## [29,] Glycerol                            0.0038  -0.0014
## [30,] Uridine                             -0.002  7e-04
## [1] "########################## Finish with PAMR ##################################################"
## [1] "Use Layout Format 6"
##       [,1] [,2] [,3]
##  [1,]    0    1   14
##  [2,]    0    2   12
##  [3,]    0    3   13
##  [4,]    0    4    0
##  [5,]    0    5    0
##  [6,]    0    6    0
##  [7,]    0    7    0
##  [8,]    0    8    0
##  [9,]    0    9    0
## [10,]    0   10    0
## [11,]    0   11    0

## [[1]]
## [1] "GSH"                "Mannose"           "UDP"                "N-Acetylaspartate"
##
## [[2]]
## [1] "XMP"                                "NADH"                               "alpha-Keto-beta-Me
```



```
## [[1]]
## [[1]][[1]]
##                                              S51        S53        S56        S47        S54        S3
## GSH                                   27033.6576 24049.4603 25257.6690 35347.4814 36076.9449 22745.628
## Mannose                                 229.0508  1746.7314   519.9360   262.7155  3122.9268  2618.66
## UDP                                    1666.5110  3781.9862  1363.9533  2280.9557  1794.5275  2485.16
## N-Acetylaspartate                      5465.4364  1327.4143  4900.7770  4777.5202  2600.3269  1472.00
## N-Acetyltyrosine                        867.2784  2111.8300  1104.6687  1796.5931  1573.7357  1216.78
## Alanine                                1143.3077  1962.0972  1583.5345  1426.5174  1809.3231  1301.81
## Malonate                                713.7648   907.4581   846.5766  1001.1333   900.6584  1333.43
## O-Phosphoethanolamine                  1125.9608  2139.0583  1037.1305  1531.1157  1135.4877   825.09
## Lauric acid                             465.2389   606.0785   632.1433   464.0117   553.0871  1267.01
## Gamma-aminobutyric acid (GABA)         1316.6368   840.8793   653.6077   580.9792  1345.9058  1308.51
## Tridecanoic acid                        544.2898  1192.0673   825.4181   914.5099   970.8812   774.50
## Glycerol                               1566.0056   263.1227   217.8842   398.3235   488.8416   353.81
## Uridine                                 588.4780   378.4008   409.0787   119.4623   147.6769   233.57
## Indolelactic acid                       408.4165   614.2979   861.7769   558.7625   696.2295   574.17
## GSSG                                    597.5166   750.0290   542.1755   733.2523   580.0724   502.33
## GSSG                                    597.5166   750.0290   542.1755   733.2523   580.0724   502.33
## Adonitol                                511.1742   799.8732  1615.5049  1225.7662  2365.6002   679.52
## Tryptophan                              371.7378   521.5220   778.6022   516.9112   711.8080   532.90
## Fructose                               1413.8101   208.9646   383.2496   405.0455   333.0364  1943.16
## Xylose                                  246.7350   266.7551   401.1817   176.9352   179.7314   258.11
## AMP                                     279.8044   381.3476   263.8068   535.8180   562.6455   453.16
## Pyridoxal (PL)                         1434.1181   207.9487   371.8950   405.7698   334.4769  1925.16
## Glycerate                               519.5884   640.0269   770.3814   614.1804   675.1256   444.62
## dAMP                                   203.1264   202.6435   122.7627   128.3604   373.6099   314.51
## dGDP                                   328.7280   345.6700   253.6828   557.1888   473.1294   407.79
## UMP                                    281.2936   250.2468   229.5017   322.2028   390.1482   545.64
## CMP                                    405.0425   147.3781   217.2898   387.0245   182.4002   583.04
## alpha-Ketoisovalerate (KIV)            733.9569   624.6872  1340.6122   642.5226   542.6647   410.44
## alpha-Keto-beta-Methylvalerate (KMV)  1047.0644   804.5426  1092.0138   618.7436   815.2827   272.20
## NADH                                   526.2167   435.0981   726.5945   356.2425   367.6365  1256.11
## XMP                                    283.3155   247.8460   273.5755   453.5294   135.3956   200.84
##
## [[1]][[2]]
##                                             S51        S53        S56        S47        S54        S
## XMP                                    283.3155   247.8460   273.5755   453.5294   135.3956   200.84
## NADH                                   526.2167   435.0981   726.5945   356.2425   367.6365  1256.11
## alpha-Keto-beta-Methylvalerate (KMV)  1047.0644   804.5426  1092.0138   618.7436   815.2827   272.20
## alpha-Ketoisovalerate (KIV)            733.9569   624.6872  1340.6122   642.5226   542.6647   410.44
## CMP                                    405.0425   147.3781   217.2898   387.0245   182.4002   583.04
## UMP                                    281.2936   250.2468   229.5017   322.2028   390.1482   545.64
## dGDP                                   328.7280   345.6700   253.6828   557.1888   473.1294   407.79
## dAMP                                   203.1264   202.6435   122.7627   128.3604   373.6099   314.51
## Glycerate                              519.5884   640.0269   770.3814   614.1804   675.1256   444.62
## Pyridoxal (PL)                        1434.1181   207.9487   371.8950   405.7698   334.4769  1925.16
## AMP                                    279.8044   381.3476   263.8068   535.8180   562.6455   453.16
## Xylose                                 246.7350   266.7551   401.1817   176.9352   179.7314   258.11
## Fructose                              1413.8101   208.9646   383.2496   405.0455   333.0364  1943.16
## Tryptophan                             371.7378   521.5220   778.6022   516.9112   711.8080   532.90
## Adonitol                               511.1742   799.8732  1615.5049  1225.7662  2365.6002   679.52
## Adonitol                               511.1742   799.8732  1615.5049  1225.7662  2365.6002   679.52
## GSSG                                   597.5166   750.0290   542.1755   733.2523   580.0724   502.33
## Indolelactic acid                      408.4165   614.2979   861.7769   558.7625   696.2295   574.17
## Uridine                                588.4780   378.4008   409.0787   119.4623   147.6769   233.57
## Glycerol                              1566.0056   263.1227   217.8842   398.3235   488.8416   353.81
## Tridecanoic acid                       544.2898  1192.0673   825.4181   914.5099   970.8812   774.50
## Gamma-aminobutyric acid (GABA)        1316.6368   840.8793   653.6077   580.9792  1345.9058  1308.51
## Lauric acid                            465.2389   606.0785   632.1433   464.0117   553.0871  1267.01
## O-Phosphoethanolamine                 1125.9608  2139.0583  1037.1305  1531.1157  1135.4877   825.09
## Malonate                               713.7648   907.4581   846.5766  1001.1333   900.6584  1333.43
## Alanine                               1143.3077  1962.0972  1583.5345  1426.5174  1809.3231  1301.81
## N-Acetyltyrosine                       867.2784  2111.8300  1104.6687  1796.5931  1573.7357  1216.78
## N-Acetylaspartate                     5465.4364  1327.4143  4900.7770  4777.5202  2600.3269  1472.00
## UDP                                   1666.5110  3781.9862  1363.9533  2280.9557  1794.5275  2485.16
## Mannose                                229.0508  1746.7314   519.9360   262.7155  3122.9268  2618.66
## GSH                                  27033.6576 24049.4603 25257.6690 35347.4814 36076.9449 22745.62
##
##
## [[2]]
## [[2]][[1]]
##                                          Sig Test
## GSH                                   3.2572    1
## Mannose                               0.2066    1
## UDP                                   0.1333    1
## N-Acetylaspartate                     0.0969    1
## N-Acetyltyrosine                      0.0865    1
## Alanine                               0.0839    1
## Malonate                              0.0600    1
## O-Phosphoethanolamine                 0.0379    1
## Lauric acid                           0.0366    1
## Gamma-aminobutyric acid (GABA)        0.0248    1
## Tridecanoic acid                      0.0174    1
## Glycerol                              0.0038    1
## Uridine                              -0.0020    1
## Indolelactic acid                    -0.0056    1
## GSSG                                 -0.0137    1
## Adonitol                             -0.0261    1
## Tryptophan                           -0.0318    1
## Fructose                             -0.0475    1
## Xylose                               -0.0536    1
## AMP                                  -0.0539    1
## Pyridoxal (PL)                       -0.0592    1
## Glycerate                            -0.0679    1
## dAMP                                 -0.0693    1
## dGDP                                 -0.0834    1
## UMP                                  -0.0880    1
## CMP                                  -0.0982    1
## alpha-Ketoisovalerate (KIV)          -0.1005    1
## alpha-Keto-beta-Methylvalerate (KMV) -0.1035    1
## NADH                                 -0.1070    1
## XMP                                  -0.2044    1
## 
## [[2]][[2]]
##                                          Sig Test
## XMP                                   0.0750    1
## NADH                                  0.0392    1
## alpha-Keto-beta-Methylvalerate (KMV)  0.0380    1
## alpha-Ketoisovalerate (KIV)           0.0369    1
## CMP                                   0.0360    1
## UMP                                   0.0323    1
## dGDP                                  0.0306    1
## dAMP                                  0.0254    1
## Glycerate                             0.0249    1
## Pyridoxal (PL)                        0.0217    1
## AMP                                   0.0198    1
## Xylose                                0.0196    1
## Fructose                              0.0174    1
## Tryptophan                            0.0117    1
## Adonitol                              0.0096    1
## GSSG                                  0.0050    1
## Indolelactic acid                     0.0020    1
## Uridine                               0.0007    1
## Glycerol                             -0.0014    1
## Tridecanoic acid                     -0.0064    1
## Gamma-aminobutyric acid (GABA)       -0.0091    1
## Lauric acid                          -0.0134    1
## O-Phosphoethanolamine                -0.0139    1
## Malonate                             -0.0220    1
## Alanine                              -0.0308    1
## N-Acetyltyrosine                     -0.0317    1
## N-Acetylaspartate                    -0.0355    1
## UDP                                  -0.0489    1
## Mannose                                  -0.0757    1
## GSH                                      -1.1943    1
```

```
# Un-load the following packages because they otherwise interfere with dplyr in part 2
detach("package:org.Hs.eg.db")
detach("package:AutoPipe")
detach("package:BiocManager")
detach("package:AnnotationDbi")
detach("package:Biobase")
detach("package:IRanges")
detach("package:S4Vectors")
detach("package:BiocGenerics")
```

Results - Data Analysis Part 2

Part 2.1: Normalization

```
# Import raw data file from CSV to tibble using read_csv
vs_xeno_metabolomics_raw <- read_csv("C:/Users/mark1/Dropbox/BIOL_4386/Project_Folder/Formatted_Data/23
```

```
## New names:
## Rows: 53 Columns: 165
## -- Column specification
## ----------------------------------------------------------------------------------------------------
## (2): Sample Label, Corresponding Primary/Xenograft dbl (148): RT Dose (Gy) - Xenografts, Prior RT, N
## Citrate, Citrulline, Creatinine, Cysteine, Dihydroxyace... lgl (15): 3-Hydroxyanthranilic acid, Amin
## i Use ‘spec()‘ to retrieve the full column specification for this data. i Specify the column types o
## * ‘‘ -> ‘...112‘
```

```
## NEW CONSOLIDATED CODE 4.1.23 for data wrangling - calculates fold changes as desired
vs_xeno_curated <- vs_xeno_metabolomics_raw %>% select(!where(is_logical)) %>% rename(primary_tumor = "
metabolite_names_xeno <- colnames(vs_xeno_curated)[-(1:5)]
vs_xeno_fc <- vs_xeno_curated %>% select(primary_tumor, dose, all_of(metabolite_names_xeno)) %>% group_
```

Part 2.2: Outlier detection

Grubbs’ test, alpha = 0.01 - Overall this was successful and I was able to create a list of the metabolites with outliers as identified by Grubbs’ test - However, I did not successfully incorportate this outlier information into the subsequent analysis, such that all the stats and graphs include any outliers that Grubbs’ test may have found. This was due to the difficulty of working with single NA values in multiple columns (57 of 144 columns had significant outliers), with those NA values not occurring in the same row for each column and thus not straightforward to remove those values from statistical calculations, normality testing, etc. In the future, I hope to re-run all subsequent analyses without these outlier values.

```
################# Outlier Detection - NOW WORKING AS OF 5.9.23 PM
##If not already installed, run: install.packages("outliers")
library(outliers)

# Run Grubbs' test for outliers on each column name in vs_xeno_fc that appears in the vector list metab
grubbs_results <- map_dfr(metabolite_names_xeno, ~{
  metabolite_col <- .x
  test_result_high <- grubbs.test(vs_xeno_fc[[metabolite_col]], opposite=FALSE, type=10)
  test_result_low <- grubbs.test(vs_xeno_fc[[metabolite_col]], opposite=TRUE, type=10)
  list(metabolite_name = metabolite_col,
       high_value = max(vs_xeno_fc[[metabolite_col]], na.rm=TRUE),
       p_value_high = test_result_high$p.value,
       low_value = min(vs_xeno_fc[[metabolite_col]], na.rm=TRUE),
       p_value_low = test_result_low$p.value)
})
# Filter for only the metabolites with p<0.01 on either high or low Grubbs test ('high' tests largest v
outlier_df <- grubbs_results %>% filter(., p_value_high<=0.01 | p_value_low<=0.01)
outlier_list <- outlier_df$metabolite_name
## NOTE THAT NONE OF THE LOW VALUES WERE SIGNIFICANT; 57 HIGH VALUES WERE SIGNIFICANT OUTLIERS PER GRUB
# Loop over each metabolite in outlier_df and replace the high value with NA in vs_xeno_fc
vs_xeno_fc_outliers_removed <- vs_xeno_fc
for (metabolite_name in outlier_df$metabolite_name) {
  vs_xeno_fc_outliers_removed <- vs_xeno_fc_outliers_removed %>%
    mutate(!!sym(metabolite_name) := if_else(!!sym(metabolite_name) == outlier_df$high_value[outlier_df$
}
```

Part 2.3: Test for normality wtih Shapiro-Wilk test

- This was largely successful, and was incorporated into subsequent steps.
- However, as noted above, it does NOT exclude the outliers that were identified with Grubbs' test, so it is possible that the results would change without those values.

```
################# Test for Normality with Shapiro-Wilk test (NOTE: THIS DOES NOT YET ACCOUNT FOR VALUES
# Initialize an empty tibble to store the p-values from the Shapiro-Wilk test
shapiro_pvalues <- tibble(metabolite = character(),
                          p_value = double())

# Loop over the outcome variables and perform the Shapiro-Wilk test
for (i in 1:length(metabolite_names_xeno)) {
  # Extract the outcome variable
  outcome_var <- metabolite_names_xeno[i]
  # Perform the Shapiro-Wilk test
  shapiro_test <- shapiro.test(vs_xeno_fc[[outcome_var]])
  # Store the variable name and p-value in the tibble
  shapiro_pvalues <- shapiro_pvalues %>%
    add_row(metabolite = outcome_var, p_value = shapiro_test$p.value)
}

# Sort the tibble by p-values and filter for p-values > 0.05 (non-normally distributed) and < 0.05 (nor
# non-normal list:
shapiro_pvalues_nonnormal <- shapiro_pvalues %>% arrange(p_value) %>% filter(p_value > 0.05)
# normally distributed:
shapiro_pvalues_normal <- shapiro_pvalues %>% arrange(p_value) %>% filter(p_value <= 0.05)
```

Part 2.4: Log transformation of non-normally distributed metabolites

- This was largely successsful.

```
################ Transformation of LogNormal metabolites
# Create tibble to identify the columns that are normal
vs_xeno_fc_normal <- vs_xeno_fc %>% group_by(primary_tumor, dose) %>% select(all_of(shapiro_pvalues_norm
```

```
## Adding missing grouping variables: ‘primary_tumor‘, ‘dose‘
```

```
# Create new tibble with only the columns to log transform
vs_xeno_fc_nonnormal <- vs_xeno_fc %>% group_by(primary_tumor, dose) %>% select(all_of(shapiro_pvalues_
```

```
## Adding missing grouping variables: ‘primary_tumor‘, ‘dose‘
```

```
# Log-transform the columns
vs_xeno_fc_log_transformed <- vs_xeno_fc_nonnormal %>% mutate_if(is.numeric, ~ ifelse(. > 0, log(.), NA
```

```
## ‘mutate_if()‘ ignored the following grouping variables:
## * Columns ‘primary_tumor‘, ‘dose‘
```

```
# IF DESIRED CAN RE-BIND THESE VALUES TO THE NORMALLY DISTRIBUTED VALUES USING CODE SIMILAR TO THE FOLL
## Combine the log-transformed columns with the rest of the original tibble
# my_tibble_transformed <- bind_cols(my_tibble %>% select(-all_of(var_names_to_log)), my_tibble_log_tra
## View the resulting tibble
# my_tibble_transformed
```

Part 2.5: Statistical Analysis

- Correlation with radiation dose (per metabolite): successful.
- Two-way ANOVA with Holm-Sidak test for xenografts to compare radiation treatment doses
  - The two-way ANOVA was done, and I believe it was done correctly.
  - However, I was unable to figure out how to do the Holm-Sidak test in an efficient manner across the long list of metabolites without manually copy-pasting a new line of code for each metabolite column. Further work will need to be done to apply the same code across all columns automatically.

```
################STATISTICAL TESTS (note: does not yet account for outliers)
######### Correlation with radiation
# Create an empty list to store the correlations
my_correlations <- list()
# Loop over the output variables and compute the correlations
for (i in 3:ncol(vs_xeno_fc)) {
  output_var <- names(vs_xeno_fc)[i]
  cor_test <- cor.test(vs_xeno_fc$dose, vs_xeno_fc[[output_var]], method = "pearson")
  my_correlations[[output_var]] <- cor_test$estimate
}
# Combine the correlations into a data frame
cor_df <- data.frame(output_var = names(my_correlations),
                     correlation = unlist(my_correlations))
cor_df_filtered_sorted <- cor_df %>% arrange(desc(correlation)) %>% filter(correlation > 0.2)

#### REPEAT THE ABOVE FOR ONLY THE NORMALLY DISTRIBUTED METABOLITES
```

```
my_correlations <- list()
for (i in 3:ncol(vs_xeno_fc_normal)) {
  output_var <- names(vs_xeno_fc_normal)[i]
  cor_test <- cor.test(vs_xeno_fc_normal$dose, vs_xeno_fc_normal[[output_var]], method = "pearson")
  my_correlations[[output_var]] <- cor_test$estimate
}
# Combine the correlations into a data frame
cor_df_normal <- data.frame(output_var = names(my_correlations),
                     correlation = unlist(my_correlations))
cor_df_normal_filtered_sorted <- cor_df_normal %>% arrange(desc(correlation)) %>% filter(correlation >

#### REPEAT THE ABOVE FOR ONLY LOG-TRANSFORMED METABOLITES
my_correlations <- list()
for (i in 3:ncol(vs_xeno_fc_log_transformed)) {
  output_var <- names(vs_xeno_fc_log_transformed)[i]
  cor_test <- cor.test(vs_xeno_fc_log_transformed$dose, vs_xeno_fc_log_transformed[[output_var]], metho
  my_correlations[[output_var]] <- cor_test$estimate
}
# Combine the correlations into a data frame
cor_df_log_normal <- data.frame(output_var = names(my_correlations),
                         correlation = unlist(my_correlations))
cor_df_log_normal_filtered_sorted <- cor_df_log_normal %>% arrange(desc(correlation)) %>% filter(correla

#Create data frame with both normal and log transformed correlations that are >0.2 by pearson test and
cor_df_all_filtered_sorted <- rbind(cor_df_normal_filtered_sorted, cor_df_log_normal_filtered_sorted) %


################ Two-way ANOVA with Holm-Sidak test ****************INCOMPLETE - AS OF 5.9.23 PM, STIL
######ALSO NOTE: DOES NOT ACCOUNT FOR OUTLIERS YET (as of 5.9.23 PM)
library(broom)
# Normally distributed metabolites:
## First pivot longer to reformat the output columns for anova function
normal_data_long <- vs_xeno_fc_normal %>%
  pivot_longer(cols = 3:ncol(.), names_to = "metabolite", values_to = "value")
# Then run the two-way ANOVA by the first two columns  across all metabolites
####NOTE: primary_tumor + dose does not include the interaction term between primary_tumor and dose (my
normal_anova <- normal_data_long %>%
  group_by(metabolite) %>%
  do(tidy(aov(value ~ primary_tumor + dose, data = .)))
## Then filter and sort for just the significant metabolites by radiation dose:
normal_anova_dose_significant <- normal_anova %>%
  filter(term == "dose" & p.value <= 0.05) %>%
  arrange(p.value) %>%
  mutate(normality = 'normal')

# Log-transformed metabolites:
## First pivot longer to reformat the output columns for anova function
logtransform_data_long <- vs_xeno_fc_log_transformed %>%
  pivot_longer(cols = 3:ncol(.), names_to = "metabolite", values_to = "value")
# Then run the two-way ANOVA by the first two columns  across all metabolites
####NOTE: primary_tumor + dose does not include the interaction term between primary_tumor and dose (my
logtransform_anova <- logtransform_data_long %>%
  group_by(metabolite) %>%
  do(tidy(aov(value ~ primary_tumor + dose, data = .)))
## Then filter and sort for just the significant metabolites by radiation dose:
logtransform_anova_dose_significant <- logtransform_anova %>%
  filter(term == "dose" & p.value <= 0.05) %>%
  arrange(p.value) %>%
  mutate(normality = 'lognormal')

all_anova_dose_significant <- rbind(normal_anova_dose_significant, logtransform_anova_dose_significant)
  arrange(p.value)
significant_metabolites <- all_anova_dose_significant$metabolite
significant_metabolites_normal <- normal_anova_dose_significant$metabolite
significant_metabolites_logtransform <- logtransform_anova_dose_significant$metabolite
```

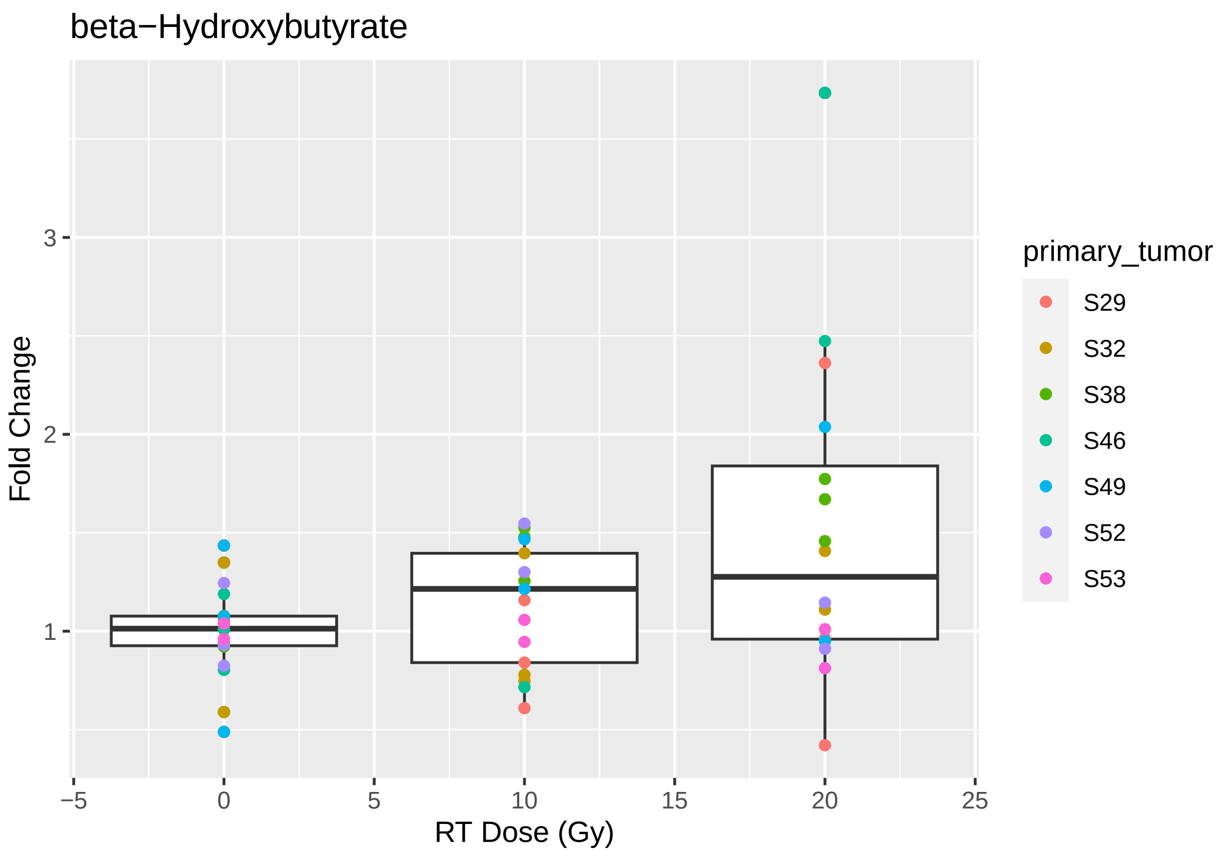
Part 2.6: Graphs of metabolites that are significantly correlated with radiation dose (limit to ~top 20 candidates)

- I was able to use ggplot2 to graph three metabolties that were highly correlated with radiation.
- As noted above, this did not exclude any outliers that were identified in part 2.2
- Next I would like to learn to automate this graphing process, such that it would do the same graph for all metabolites that meet a given significance threshold. Unfortunately I have not yet been able to figure this out.

```
################ Graphs of metabolites that are significantly correlated with radiation dose (limit to
  ############################INCOMPLETE AS OF 5.9.23 - ONLY HAVE A FEW GRAPHS DONE
# beta-Hydroxybutyrate fold change boxplot with color labels
ggplot(vs_xeno_fc, aes(x = `dose`, y = `beta-Hydroxybutyrate (3-Hydroxybutyrate)`)) +
  geom_boxplot(aes(group=`dose`)) +
  geom_point(aes(color = `primary_tumor`)) +
  labs(title = "beta-Hydroxybutyrate", x = "RT Dose (Gy)", y = "Fold Change")
```

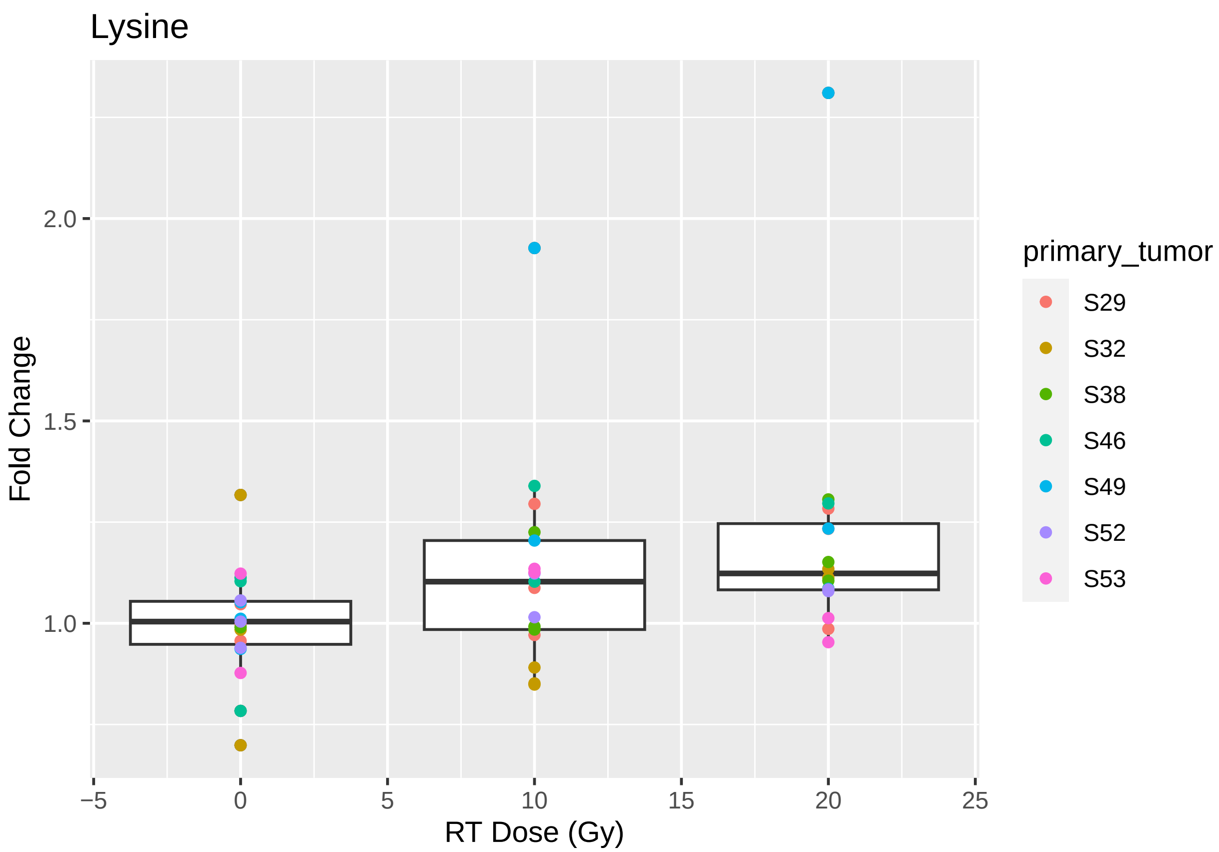
```
## Warning: Removed 1 rows containing non-finite values (`stat_boxplot()`).
```

```
## Warning: Removed 1 rows containing missing values (`geom_point()`).
```



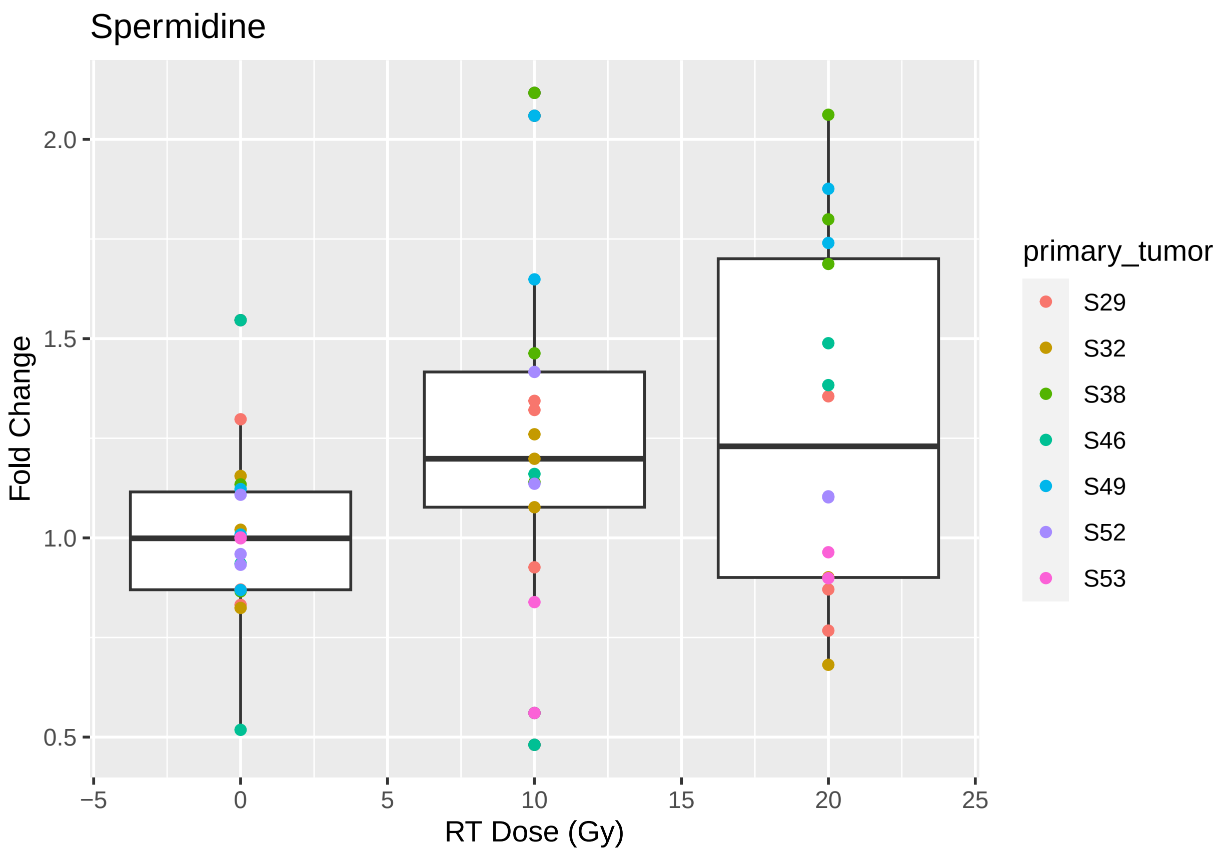
```
# Lysine fold change boxplot with color labels
ggplot(vs_xeno_fc, aes(x = `dose`, y = `Lysine`)) +
  geom_boxplot(aes(group=`dose`)) +
  geom_point(aes(color = `primary_tumor`)) +
  labs(title = "Lysine", x = "RT Dose (Gy)", y = "Fold Change")
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_boxplot()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```



```r
# Fold change boxplot for Spermidine - with color labels
ggplot(vs_xeno_fc, aes(x = `dose`, y = `Spermidine`)) +
  geom_boxplot(aes(group=`dose`)) +
  geom_point(aes(color = `primary_tumor`)) +
  labs(title = "Spermidine", x = "RT Dose (Gy)", y = "Fold Change")
```

```
## Warning: Removed 1 rows containing non-finite values (‘stat_boxplot()‘).
## Removed 1 rows containing missing values (‘geom_point()‘).
```



Appendix

```
## APPENDIX (part 2.1): view results of fold change calculations
view(vs_xeno_fc)

# APPENDIX (part 2.2): Print Grubbs' test results (grubbs_results) and the VS xenograft tibble with tho
view(grubbs_results)
view(vs_xeno_fc_outliers_removed)

## APPENDIX (part 2.4): View results of normal and lognormal data (Appendix)
view(vs_xeno_fc_normal)
view(vs_xeno_fc_log_transformed)

# APPENDIX (part 2.5): view all the metabolites with correlation coeff > 0.2, sorted greatest to least:
view(cor_df_all_filtered_sorted)
# Print top 10 metabolites by correlation values:
cor_df_all_filtered_sorted[1:10,1]
```

```
##  [1] "beta-Hydroxybutyrate (3-Hydroxybutyrate)" "Lysine"                                   "Spermidi
```

```
### APPENDIX (part 2.5): View list of significant metabolites based on Two-Way ANOVA (no post-hoc test
view(significant_metabolites)
view(all_anova_dose_significant)
```