

Manuscript table1

Deisiany

2023-09-19

Table 1

```
# Load TIR1/AFB1, Aux/IAs, and ARFs so we can bind all together.
TIR <- read.csv("../final_trees/AFB_input/2023_0919_AFB_Gm_ortholog.csv")

IAA <- read.csv("../final_trees/IAA_input/IAA_Gm_ortholog.csv")
IAA2 <- read.csv("../final_trees/IAA_input/LABELS2_20230917.csv") %>% .[30:90, ]
IAA2 <- IAA2 %>% select(tair_locus, ensembl_gene_id, Class)

IAA_ortho <- merge(IAA, IAA2, by = "tair_locus", all = TRUE)
IAA_ortho <- IAA_ortho %>% dplyr::rename(`Gene ID` = `ensembl_gene_id.x`,
  `Transcript ID` = tair_locus,
  Orthology = ortholog_name,
  `Class/Clade` = Class) %>%
  select(`Transcript ID`, Orthology, `Class/Clade`)
IAA_ortho$Family <- "IAA"

expression_data <- read_csv("FINAL_manuscript_gene_INFO.csv") %>% select(-c(Average_Expression, `...22`)

## New names:
## Rows: 221 Columns: 22
## -- Column specification
## ----- Delimiter: "," chr
## (6): ensembl_gene_id, ortholog_number, tair_locus, class, name, Family dbl
## (16): AM, OF, IAM, IBM, RootTip, Cotyledon, Hypocotyl, SAM6D, SAM17D, SA...
## 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.
## * `` -> `...22`

comb_expr_df <- merge(IAA_ortho, expression_data, by.x = "Transcript ID", by.y = "tair_locus", all = TRUE)

comb_expr_df <- comb_expr_df %>% mutate(Class = coalesce(`Class/Clade`, class))

comb_expr_df <- comb_expr_df %>%
  select(-c(class, `Class/Clade`, Orthology, Family.x))

# populate NA's in class in accordance to their ensembl gene ID. if the same ID they belong to the same
comb_expr_df2 <- comb_expr_df %>%
  group_by(ensembl_gene_id) %>%
  dplyr::mutate(Class = if (all(is.na(Class))) NA else na.omit(Class)) %>%
```

```

ungroup()

# separate Class into Class and Clade
comb_expr_df2 <- comb_expr_df2 %>% separate(Class, c('Class', 'Clade'), sep = " - ")

## Warning: Expected 2 pieces. Missing pieces filled with `NA` in 120 rows [1, 2, 9, 10,
## 13, 17, 18, 19, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 38, 39, ...].

# Now combine TIR df to get their new clades.
df <- merge(comb_expr_df2, TIR, by.x = "ensembl_gene_id", by.y = "ensembl_gene_id", all = TRUE)

df <- df %>% dplyr::mutate(Clade = coalesce(Clade.x, Clade.y))

df <- df %>%
  dplyr::select(-c(Clade.x, Clade.y, class, tair_locus, ortholog_name, name.x, name.y)) %>%
  dplyr::rename(Orthology = ortholog_number)

df <- df %>%
  group_by(ensembl_gene_id) %>%
  dplyr::mutate(Clade = if(all(is.na(Clade))) NA else na.omit(Clade)) %>%
  ungroup()

# I can add clades to ARFs as I have not saved that as csv file when I built trees.
df <- df %>% dplyr::mutate(Clade = ifelse(grepl("ARF", .$Family.y) & grepl("A", .$Class), "II",
  ifelse(grepl("ARF", .$Family.y) & grepl("B", .$Class), "I",
  ifelse(grepl("ARF", .$Family.y) & grepl("C", .$Class), "III", Clade)))

df2 <-
  df %>% dplyr::mutate(Class = ifelse(grepl("Glyma.02G218100", .$ensembl_gene_id), "A", Class),
    Clade =
      ifelse(grepl("Glyma.02G218100", .$ensembl_gene_id), "I", Clade))

df2 <- df2 %>% dplyr::rename(Family = Family.y, `Gene ID` = ensembl_gene_id) %>% drop_na()

```

Table 1

```

# save Table 1 for manuscript
df2 %>% dplyr::select(`Gene ID`, Orthology, `Transcript ID`, Class, Clade, Family) %>%
  arrange(Family, Clade, Class, Orthology) #>% write_csv("Table1.csv")

```

```

## # A tibble: 221 x 6
##   `Gene ID`      Orthology      `Transcript ID`    Class      Clade Family
##   <chr>         <chr>         <chr>         <chr>    <chr> <chr>
## 1 Glyma.19G206800 GmTIR1/AFB1_A.1 Glyma.19G206800.1 TIR1/AFB1 I      AFB/TIR1
## 2 Glyma.03G209400 GmTIR1/AFB1_B.1 Glyma.03G209400.1 TIR1/AFB1 I      AFB/TIR1
## 3 Glyma.10G021500 GmTIR1/AFB1_C.1 Glyma.10G021500.1 TIR1/AFB1 I      AFB/TIR1
## 4 Glyma.10G021500 GmTIR1/AFB1_C.2 Glyma.10G021500.2 TIR1/AFB1 I      AFB/TIR1
## 5 Glyma.10G021500 GmTIR1/AFB1_C.3 Glyma.10G021500.3 TIR1/AFB1 I      AFB/TIR1
## 6 Glyma.02G152800 GmTIR1/AFB1_D.1 Glyma.02G152800.1 TIR1/AFB1 I      AFB/TIR1

```

```
## 7 Glyma.02G152800 GmTIR1/AFB1_D.2 Glyma.02G152800.2 TIR1/AFB1 I AFB/TIR1
## 8 Glyma.19G100200 GmAFB2/3_A.1 Glyma.19G100200.1 AFB2/3 II AFB/TIR1
## 9 Glyma.16G050500 GmAFB2/3_B.1 Glyma.16G050500.1 AFB2/3 II AFB/TIR1
## 10 Glyma.02G065300 GmAFB2/3_C.1 Glyma.02G065300.1 AFB2/3 II AFB/TIR1
## # i 211 more rows
```

```
# write_csv(df2, "20230919_expression_heatmap.csv")
```

Heatmap

```
# Source:
# https://stackoverflow.com/questions/43051525/how-to-draw-heatmap-plot-to-screen-and-also-save-to-file
save_pheatmap_pdf <- function(x, filename, width=7, height=4) {
  stopifnot(!missing(x))
  stopifnot(!missing(filename))
  pdf(filename, width=width, height=height)
  grid::grid.newpage()
  grid::grid.draw(x$gtable)
  dev.off()
}

# https://davetang.org/muse/2018/05/15/making-a-heatmap-in-r-with-the-pheatmap-package/

# Make a heatmap label by uniting orthology column and gene ID column
heatmap_df <- df2 %>%
  relocate(`Gene ID`, .after = `Transcript ID`) %>%
  unite("heatmap_label", `Gene ID`:Orthology, sep = "|", remove = TRUE)

# Genes with an median expression across tissues that are less than 2 TPM, will be excluded from downstream
heatmap_df2 <- heatmap_df %>%
  mutate(expr_median = round(apply(heatmap_df[,c(4:17)], 1, median), digits = 4)) %>%
  subset(., expr_median >= 2) %>% dplyr::filter(., Class != "COI1")

# we kept 133 genes for downstream normalization
# I will save this as a csv file so we can perform a tau (Tissue specificity index analysis).
#write_csv(heatmap_df2, "Tau_133_genes_with_expression_across_tissues_are_greater_than_2_SoyARCs.csv")
```

table 2 with tau values for the 133 genes.

```
heatmap_df2
```

```
## # A tibble: 133 x 20
##   `Transcript ID` heatmap_label Family AM OF IAM IBM RootTip
##   <chr>           <chr>         <chr> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Glyma.01G002100.1 Glyma.01G002100|Gm~ ARF 10.2 11.1 9.94 11.8 6.06
## 2 Glyma.01G019400.1 Glyma.01G019400|Gm~ IAA 4.02 7.57 0 1.48 3.68
## 3 Glyma.01G019400.2 Glyma.01G019400|Gm~ IAA 3.95 0 6.01 7.65 2.22
## 4 Glyma.01G019400.3 Glyma.01G019400|Gm~ IAA 22.5 35.9 20.7 26.7 16.1
## 5 Glyma.01G098000.3 Glyma.01G098000|Gm~ IAA 68.2 51.4 53.6 96.3 32.6
## 6 Glyma.01G103500.1 Glyma.01G103500|Gm~ ARF 23.4 7.20 14.1 28.2 4.92
## 7 Glyma.02G000500.1 Glyma.02G000500|Gm~ IAA 26.5 33.1 14.3 12.0 28.4
## 8 Glyma.02G065300.1 Glyma.02G065300|Gm~ AFB/T~ 8.00 6.50 7.76 11.8 3.85
```

```
## 9 Glyma.02G142500.3 Glyma.02G142500|Gm~ IAA 14.7 530. 73.1 28.5 43.2
## 10 Glyma.02G142600.1 Glyma.02G142600|Gm~ IAA 4.37 80.7 19.6 21.4 19.2
## # i 123 more rows
## # i 12 more variables: Cotyledon <dbl>, Hypocotyl <dbl>, SAM6D <dbl>,
## # SAM17D <dbl>, SAM38D <dbl>, Callus <dbl>, Leaf <dbl>, Root <dbl>,
## # Nodule <dbl>, Class <chr>, Clade <chr>, expr_median <dbl>

#read tau analysis in so we can incorporate later into our PCA analysis
tau_df <- read_csv("tau_df_contains_maximal_component_values.csv")

## Rows: 133 Columns: 18
## -- Column specification -----
## Delimiter: ","
## chr (3): Transcript ID, heatmap_label, Family
## dbl (15): AM, OF, IAM, IBM, RootTip, Cotyledon, Hypocotyl, SAM6D, SAM17D, SA...
##
## 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.

# Write tau as table 2 for sharing in supplemental material
tau_df %>% select(-c(Family, `Transcript ID`)) %>%
  rename("GeneID & Orthology" = heatmap_label) # %>%

## # A tibble: 133 x 16
## `GeneID & Orthology` AM OF IAM IBM RootTip Cotyledon Hypocotyl
## <chr> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 Glyma.01G002100|GmARF7~ 0.816 0.890 0.797 0.945 0.486 0.505 1
## 2 Glyma.01G019400|GmIAA8~ 0.382 0.718 0 0.140 0.349 0.0952 0.841
## 3 Glyma.01G019400|GmIAA8~ 0.175 0 0.266 0.339 0.0983 0.0930 0.499
## 4 Glyma.01G019400|GmIAA8~ 0.254 0.404 0.234 0.301 0.182 0.0454 1
## 5 Glyma.01G098000|GmIAA8~ 0.396 0.298 0.311 0.559 0.190 0.455 1
## 6 Glyma.01G103500|GmARF9~ 0.694 0.214 0.420 0.839 0.146 0.0621 0.0699
## 7 Glyma.02G000500|GmIAA1~ 0.252 0.315 0.137 0.114 0.271 0.0604 1
## 8 Glyma.02G065300|GmAFB2~ 0.522 0.424 0.506 0.772 0.251 0.0816 0.303
## 9 Glyma.02G142500|GmIAA1~ 0.0277 1 0.138 0.0538 0.0816 0.000382 0.216
## 10 Glyma.02G142600|GmIAA1~ 0.0541 1 0.243 0.266 0.238 0.0734 0.168
## # i 123 more rows
## # i 8 more variables: SAM6D <dbl>, SAM17D <dbl>, SAM38D <dbl>, Callus <dbl>,
## # Leaf <dbl>, Root <dbl>, Nodule <dbl>, tau <dbl>

# write_csv(., "Table2_tau_and_maximal_component_value.csv")
```

Normalization by transcript

```
Expr_data_Norm <- heatmap_df2 %>% relocate(c(Class, Clade), .after = Family) %>% select(-expr_median)

# Normalize data by performing a z-score transformation on selected columns.

# Iterate through each row of the 'Expr_data_Norm' data frame.
for (j in 1:nrow(Expr_data_Norm)) {
  # Iterate through columns starting from the 6th column to the last column.
  for (i in 6:ncol(Expr_data_Norm)) {

    # Calculate the z-score normalization for each cell in the data frame.
    # 1. Subtract the mean of the selected row (columns 6 to the last column).
```

```

# 2. Divide by the standard deviation of the selected row.
Expr_data_Norm[j,i]<-
  (Expr_data_Norm[j,i]-rowMeans(Expr_data_Norm[j,6:ncol(Expr_data_Norm)]))/
  sd(Expr_data_Norm[j,6:ncol(Expr_data_Norm)])
}
}

Expr_data_Norm <-Expr_data_Norm %>% drop_na() %>% column_to_rownames(., var = "heatmap_label")

```

Build heatmap with normalized data

```

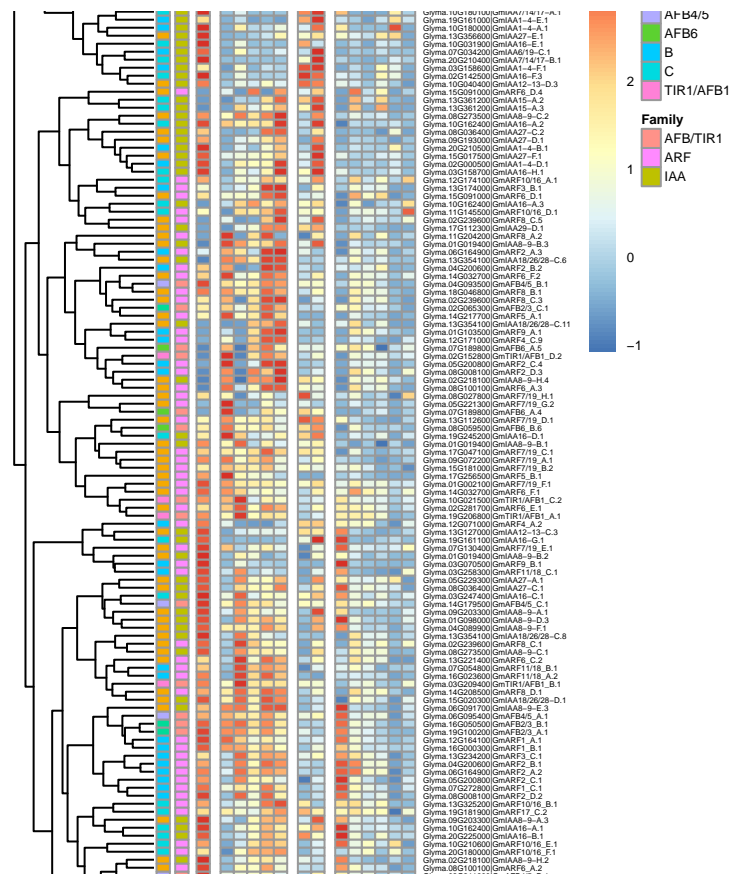
# Specify colors
Class_df1_colors = list(Family = c(ARF = "Purple", IAA = "yellow", `AFB/TIR1`="lightblue"), Class = c(A

```

```

pheatmap::pheatmap(mat = Expr_data_Norm[,5:18], annotation_row = Expr_data_Norm[,c(2,3)],
  fontsize = 5,
  fontsize_row = 3,
  cellwidth = 5,
  cutree_cols = 4,
  cellheight = 3, na_col = "black",
  clustering_distance_rows = "euclidean",
  main = "Expression analysis")

```



These genes are relatively lower in expression compared to the mean expression of those genes across the samples or conditions under analysis.

Principal component analysis

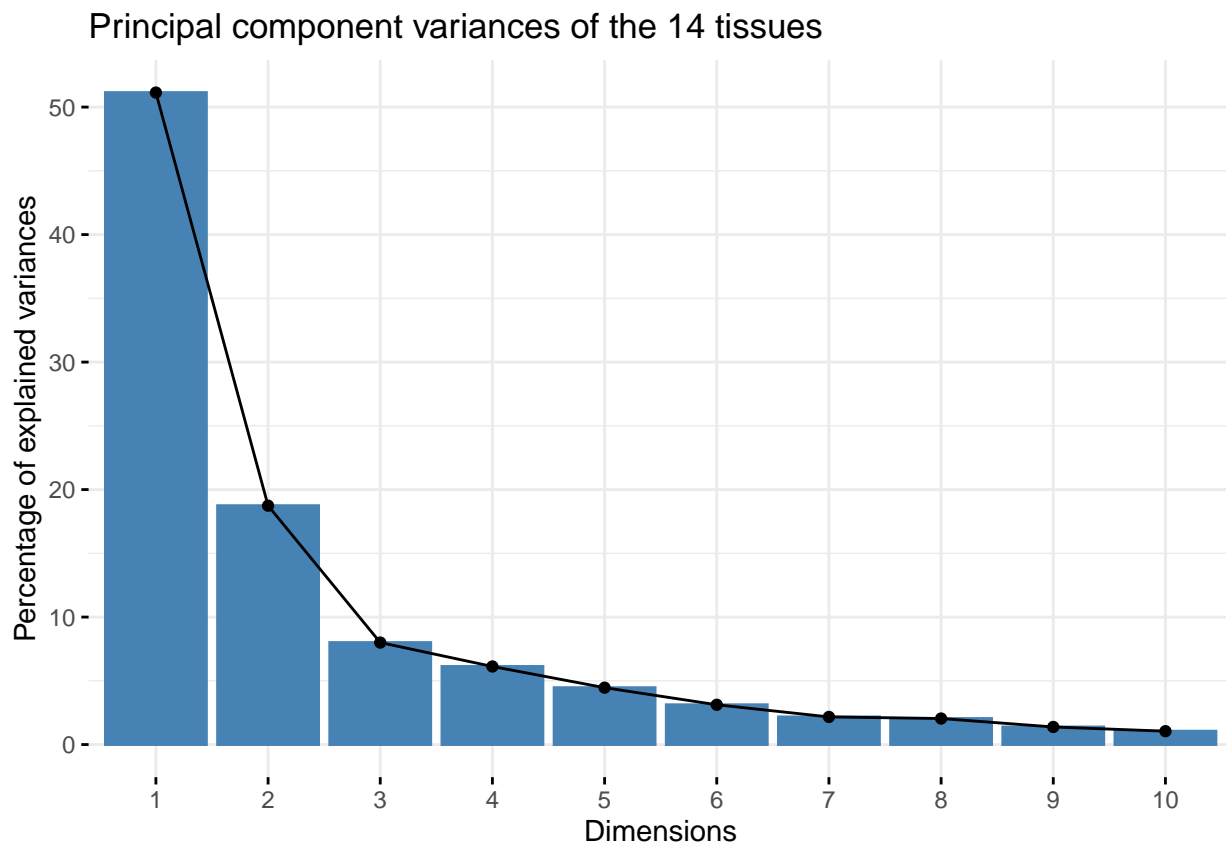
Using the full data set in which data median expression was greater than 2. In our PCA we are not using the normalized data. That is because we are already transforming our data once by using `scale = true`, which is known to change the range of the data whereas normalized changes its shape distribution.

```
pca_data <- heatmap_df2 %>% relocate(c(Class, Clade), .after = Family) %>% select(-expr_median)

# for analysis with all tissues
pca <- prcomp(pca_data[, -(1:5)],
              scale. = T, center = T)
summary(pca)

## Importance of components:
##              PC1    PC2    PC3    PC4    PC5    PC6    PC7
## Standard deviation  2.6757 1.6196 1.05841 0.92601 0.79038 0.66102 0.55083
## Proportion of Variance 0.5114 0.1874 0.08002 0.06125 0.04462 0.03121 0.02167
## Cumulative Proportion 0.5114 0.6988 0.77878 0.84003 0.88465 0.91586 0.93753
##              PC8    PC9    PC10    PC11    PC12    PC13    PC14
## Standard deviation  0.53473 0.43946 0.38317 0.32336 0.26775 0.2275 0.14373
## Proportion of Variance 0.02042 0.01379 0.01049 0.00747 0.00512 0.0037 0.00148
## Cumulative Proportion 0.95796 0.97175 0.98224 0.98971 0.99483 0.9985 1.00000

factoextra::fviz_eig(pca, main = "Principal component variances of the 14 tissues")
```



```
# ggsave("20230927_PCScreePlot_allTissues.png", dpi = 1000, width = 10, height = 8)
```

PC1 and PC2 for all 14 tissues

```
Family <- pca_data$Family

# extract the loading so we can change the arrows more easily
PCA_loadings <- data.frame(Variables=rownames(pca$rotation), pca$rotation)

comb_pca_df <- cbind(pca_data, pca$x[, 1:4])
```

PC1 and PC2 outside ellipse labels for the 14 tissues

```
PC1_2all <- ggbiplot::ggbiplot(pca, obs.scale = 1, var.scale = .5,
                               varname.adjust = 3, alpha = .1,
                               groups = Family, var.axes = F, varname.size = 2,
                               ellipse = TRUE, circle = FALSE, ellipse.prob = .70)

## -----
## You have loaded plyr after dplyr - this is likely to cause problems.
## If you need functions from both plyr and dplyr, please load plyr first, then dplyr:
## library(plyr); library(dplyr)
## -----
##
## Attaching package: 'plyr'
##
## The following objects are masked from 'package:plotly':
##
##   arrange, mutate, rename, summarise
##
## The following objects are masked from 'package:dplyr':
##
##   arrange, count, desc, failwith, id, mutate, rename, summarise,
##   summarize
##
## The following object is masked from 'package:purrr':
##
##   compact
##
## Attaching package: 'scales'
##
## The following object is masked from 'package:viridis':
##
##   viridis_pal
##
## The following object is masked from 'package:purrr':
##
##   discard
##
## The following object is masked from 'package:readr':
##
##   col_factor
```



```
# Extract components so we can select out
PC1_2all$data
```

##	xvar	yvar	groups
## 1	-1.304753081	0.253967120	ARF
## 2	-1.934745295	0.185928231	IAA
## 3	-1.616343442	-0.113511655	IAA
## 4	0.071187241	0.274717011	IAA
## 5	7.459612561	-0.600002016	IAA
## 6	-0.749858791	0.859463281	ARF
## 7	-0.005956317	-0.427960267	IAA
## 8	-1.524808613	0.431662229	AFB/TIR1
## 9	1.574336185	-5.041777156	IAA
## 10	-1.125575851	-0.558512920	IAA
## 11	-0.578750074	-0.811787304	IAA
## 12	-0.102457801	0.788703488	AFB/TIR1
## 13	-1.734416069	0.129024379	AFB/TIR1
## 14	0.542400680	-0.670223439	IAA
## 15	2.276729634	0.750445550	IAA
## 16	-0.307178253	0.330600703	ARF
## 17	0.923944787	1.334152162	ARF
## 18	-1.417357564	0.157370925	ARF
## 19	-0.428112834	0.500704876	ARF
## 20	1.418306200	-2.036229235	ARF
## 21	1.623282025	0.380380191	ARF
## 22	-2.129452835	-0.114441199	IAA
## 23	3.425337748	-3.402760865	IAA
## 24	0.306505583	0.466557952	AFB/TIR1
## 25	13.392512855	-1.806412389	IAA
## 26	-1.395820127	0.019944454	ARF
## 27	-0.071996269	0.256945271	IAA
## 28	-0.636996614	0.476207228	AFB/TIR1
## 29	3.120519585	1.142973848	ARF
## 30	-0.431602510	0.803815826	ARF
## 31	-1.866388546	0.375426589	ARF
## 32	1.337994430	0.015559355	ARF
## 33	6.702648045	3.579346703	ARF
## 34	-2.050727705	0.494196522	ARF
## 35	-0.617206008	-0.218288936	IAA
## 36	15.037892920	4.746237323	IAA
## 37	-0.488627623	0.398718486	AFB/TIR1
## 38	3.271439778	1.250223968	ARF
## 39	5.275584046	2.305407189	ARF
## 40	-1.369724965	-0.788772964	IAA
## 41	-0.667462730	0.526307555	ARF
## 42	-0.350550067	0.913360463	ARF
## 43	0.191018481	0.098979610	ARF
## 44	-0.450954600	0.394826074	ARF
## 45	-1.955375052	0.413220346	AFB/TIR1
## 46	-1.668697519	0.421500106	AFB/TIR1
## 47	-1.422813511	0.323815463	ARF
## 48	0.144182276	0.393931829	ARF
## 49	2.741786531	0.753157423	ARF
## 50	1.148171748	1.245116561	ARF

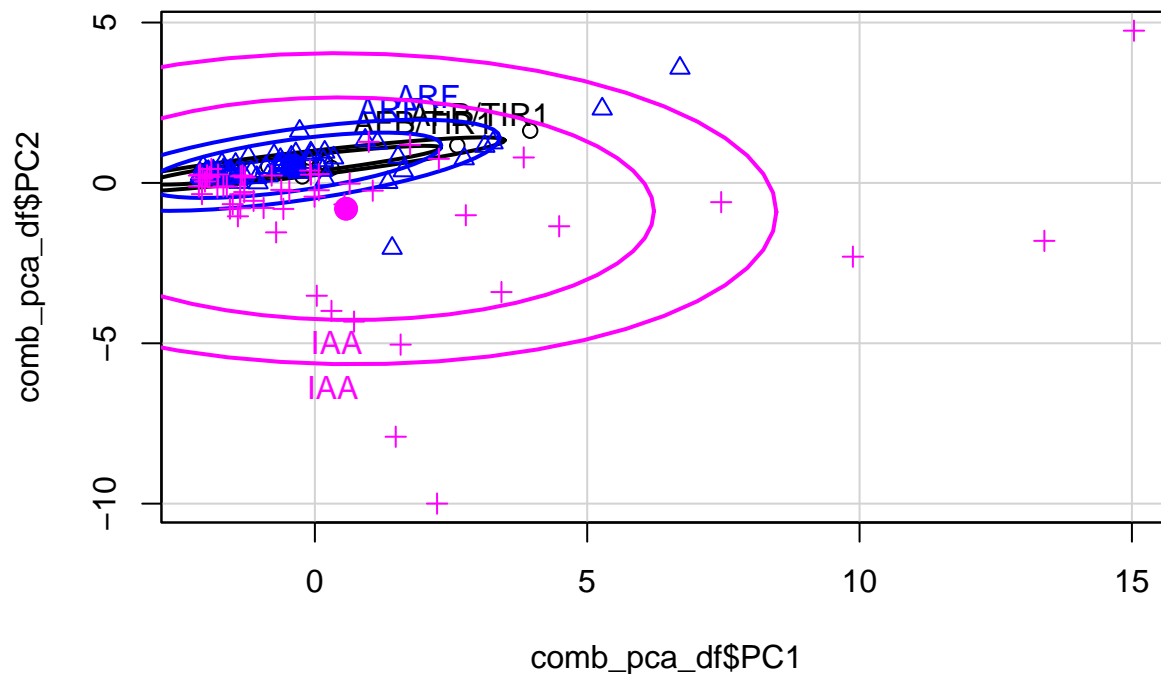
## 51	-2.105944003	0.202653740	ARF
## 52	0.639274022	-0.023380766	IAA
## 53	-2.063221418	0.141276678	IAA
## 54	-1.837152604	0.261934275	AFB/TIR1
## 55	-0.632126448	0.732872508	ARF
## 56	-1.068012274	0.168806405	ARF
## 57	-1.789096929	-0.140394034	IAA
## 58	-1.329711750	0.231182561	IAA
## 59	3.834394420	0.793449297	IAA
## 60	-1.552833562	0.395883644	ARF
## 61	-2.130228985	0.229001671	IAA
## 62	1.062351513	-0.241185818	IAA
## 63	-0.468794376	-0.249922455	IAA
## 64	0.077758331	0.524357821	AFB/TIR1
## 65	-2.084695899	0.369234056	AFB/TIR1
## 66	-0.856921604	0.507222423	AFB/TIR1
## 67	-1.508225563	-0.662268126	IAA
## 68	-1.559775037	-0.800512633	IAA
## 69	-2.018657725	0.043427973	IAA
## 70	-1.909610168	0.277764356	ARF
## 71	4.485099923	-1.351421276	IAA
## 72	2.771186411	-1.008109821	IAA
## 73	-1.778294112	0.262406297	IAA
## 74	-0.711451800	-1.541788959	IAA
## 75	0.304514187	-3.991108850	IAA
## 76	2.243284561	-9.997776235	IAA
## 77	-1.044218657	0.004571344	ARF
## 78	-0.860081508	0.223425301	ARF
## 79	-1.540300976	0.174794131	ARF
## 80	-0.075389131	0.937109194	ARF
## 81	-1.455953728	0.582823128	ARF
## 82	-2.063880601	0.149412964	ARF
## 83	-0.902465346	0.479261947	ARF
## 84	1.526821368	0.804159673	ARF
## 85	-1.705462800	0.589012162	ARF
## 86	-1.600893255	0.192216165	ARF
## 87	-1.791100183	0.339552665	ARF
## 88	-1.737520983	0.145743103	ARF
## 89	-1.679057092	-0.108332600	IAA
## 90	-1.994650072	0.350064536	ARF
## 91	-0.056763350	0.882006175	ARF
## 92	0.344926556	0.780242803	ARF
## 93	-0.446664212	0.458750897	ARF
## 94	-0.564238437	0.443802208	ARF
## 95	0.988840347	1.281462617	IAA
## 96	-0.793439779	0.229054413	IAA
## 97	-1.900821570	0.435235820	IAA
## 98	-1.372216366	-0.353257233	IAA
## 99	-1.413117840	-1.040984678	IAA
## 100	-1.362338626	0.207864794	IAA
## 101	-1.285762155	0.171175760	IAA
## 102	-1.166597927	0.338183452	ARF
## 103	-0.098539817	0.642577969	ARF
## 104	-0.230981637	0.201050662	AFB/TIR1

```
## 105  0.071596867 -0.228022083    IAA
## 106  0.156497195  0.696034794    ARF
## 107 -1.219875713  0.814553891    ARF
## 108 -1.297431165 -0.285475152    IAA
## 109 -1.979546144  0.237586347    IAA
## 110  1.748018509  1.187868663    IAA
## 111 -0.603884550  0.638749151    ARF
## 112 -1.676009054  0.418512190    ARF
## 113 -1.955130698  0.355458289    ARF
## 114 -1.994415698  0.311126791    ARF
## 115 -0.353407486  0.428375254    ARF
## 116  0.180816068  0.960752443    ARF
## 117  3.952871915  1.625931704 AFB/TIR1
## 118 -1.649072111  0.285122668    ARF
## 119 -2.107793036  0.262895090    IAA
## 120 -0.279830679  1.613426920    ARF
## 121 -0.598906053  0.577784192    ARF
## 122 -2.099471731  0.273681105    ARF
## 123  2.615232113  1.164031145 AFB/TIR1
## 124 -2.073025165 -0.349419050    IAA
## 125  0.719480304 -4.324426296    IAA
## 126  0.034697877 -3.518281284    IAA
## 127 -1.545395901  0.285127917    ARF
## 128  0.193178086  0.633239300 AFB/TIR1
## 129 -0.088423804  0.385551512    IAA
## 130 -1.883285723  0.267885408    ARF
## 131  1.484478940 -7.913429916    IAA
## 132 -0.941227933 -0.781215430    IAA
## 133  9.877340758 -2.300225618    IAA
```

```
build <- ggplot_build(PC1_2all)$data
points <- build[[1]]
```

```
# co-ordinates of the ellipses
```

```
ell_points <- car::dataEllipse(comb_pca_df$PC1,
                               comb_pca_df$PC2,
                               as.factor(comb_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point

ell_ARF <- as.data.frame(ell_points$ARF$`0.7`)
ell_IAA <- as.data.frame(ell_points$IAA$`0.7`)
ell_TIR <- as.data.frame(ell_points$`AFB/TIR1`$`0.7`)

# Find which points are outside (!) the ellipse, and add this to the data
library(sp)
dat_TIR <- data.frame(
  points[2:3],
  in_ell_TIR = as.logical(point.in.polygon(points$x, points$y, ell_TIR$x, ell_TIR$y))
)

dat_IAA <- data.frame(
  points[2:3],
  in_ell_IAA = as.logical(point.in.polygon(points$x, points$y, ell_IAA$x, ell_IAA$y))
)

dat_ARF <- data.frame(
  points[2:3],
  in_ell_ARF = as.logical(point.in.polygon(points$x, points$y, ell_ARF$x, ell_ARF$y))
)

# as.logical(point..) equals to TRUE indicated points are inside ellipses

# Combining data points for labeling
transcript_expr_logic <- cbind(comb_pca_df, dat_TIR, dat_IAA, dat_ARF) %>% select(., - c(x, y))

expr_logic <- transcript_expr_logic %>% mutate(., in_ell = case_when(Family=="ARF" & `in_ell_ARF` == TRUE ~ "TRUE",
  Family=="IAA" & `in_ell_IAA` == TRUE ~ "TRUE",
```

```

Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
mutate(in_ell = coalesce(in_ell, "FALSE"))
expr_logic[which(expr_logic$in_ell == FALSE),]

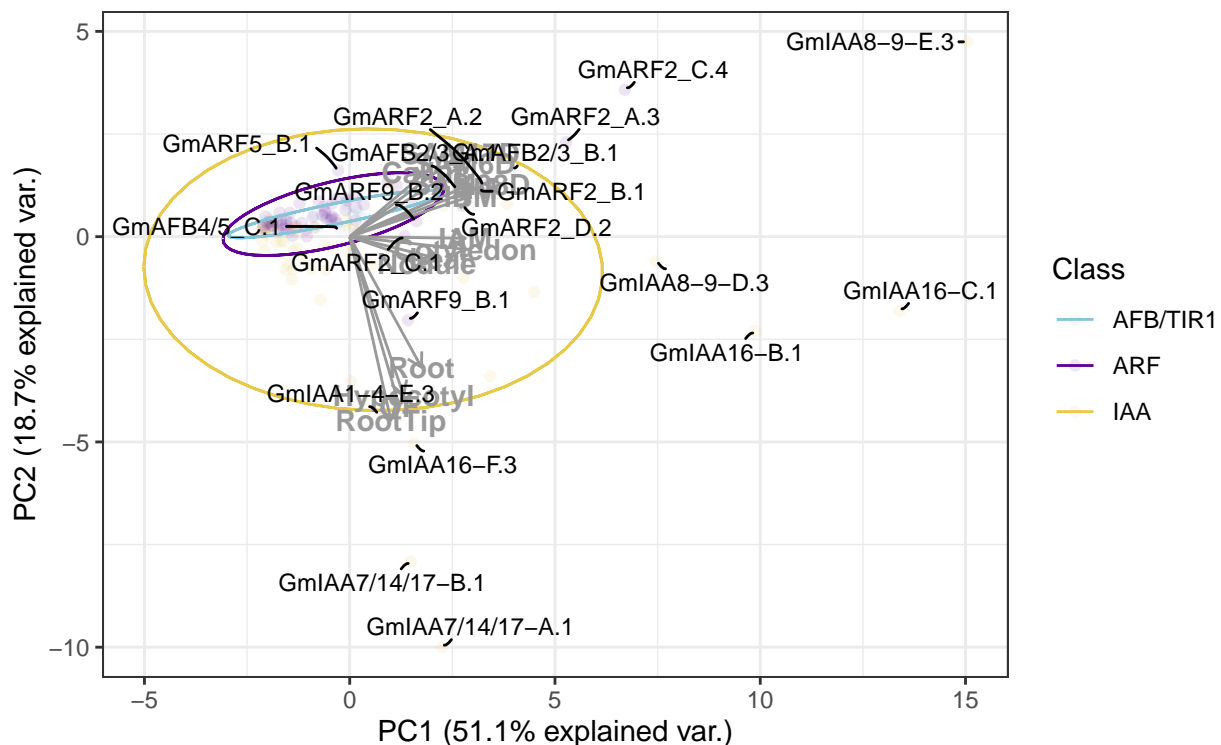
```

##	Transcript ID	heatmap_label	Family	Class	Clade		
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I		
## 9	Glyma.02G142500.3	Glyma.02G142500 GmIAA16-F.3	IAA	C	III		
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I		
## 21	Glyma.03G070500.2	Glyma.03G070500 GmARF9_B.2	ARF	B	I		
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III		
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I		
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I		
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I		
## 36	Glyma.06G091700.3	Glyma.06G091700 GmIAA8-9-E.3	IAA	A	I		
## 38	Glyma.06G164900.2	Glyma.06G164900 GmARF2_A.2	ARF	B	I		
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I		
## 49	Glyma.08G008100.2	Glyma.08G008100 GmARF2_D.2	ARF	B	I		
## 76	Glyma.10G180100.1	Glyma.10G180100 GmIAA7/14/17-A.1	IAA	C	III		
## 104	Glyma.14G179500.1	Glyma.14G179500 GmAFB4/5_C.1	AFB/TIR1	AFB4/5	IV		
## 117	Glyma.16G050500.1	Glyma.16G050500 GmAFB2/3_B.1	AFB/TIR1	AFB2/3	II		
## 120	Glyma.17G256500.1	Glyma.17G256500 GmARF5_B.1	ARF	A	II		
## 123	Glyma.19G100200.1	Glyma.19G100200 GmAFB2/3_A.1	AFB/TIR1	AFB2/3	II		
## 125	Glyma.19G161000.3	Glyma.19G161000 GmIAA1-4-E.3	IAA	B	II		
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III		
## 133	Glyma.20G225000.1	Glyma.20G225000 GmIAA16-B.1	IAA	C	III		
##	AM	OF	IAM	IBM	RootTip	Cotyledon	
## 5	68.216252	51.350688	53.553189	96.275020	32.628226	78.301786	
## 9	14.688070	529.887081	73.110411	28.516509	43.238091	0.202364	
## 20	14.963834	9.149436	10.014337	18.303006	6.361899	6.335987	
## 21	45.008611	32.497606	40.924358	42.192675	8.439588	8.282666	
## 25	123.752976	182.226043	143.672394	136.882838	68.294552	100.483878	
## 29	59.132821	34.776950	54.023677	104.014185	5.893899	15.036259	
## 32	23.736932	7.300337	13.387625	33.516112	4.197497	21.693739	
## 33	106.794815	41.848269	90.206750	89.149798	3.924654	25.580965	
## 36	120.096829	73.349658	80.928913	112.799969	14.116324	123.198735	
## 38	62.736767	31.362845	56.712867	104.850125	1.621706	14.695636	
## 39	78.161251	91.333242	102.461124	95.846621	1.569549	29.653419	
## 49	41.952212	22.684099	48.464588	54.600720	2.166574	14.976262	
## 76	7.941917	536.895072	6.035997	7.033665	214.246604	30.729826	
## 104	21.750232	21.417969	15.854160	17.435871	11.207461	8.506090	
## 117	55.698420	35.061948	32.831068	58.586254	13.107811	27.899216	
## 120	7.845772	2.581028	3.311743	17.662479	0.302999	1.222648	
## 123	42.087683	43.031580	27.129734	42.689001	13.615904	12.545145	
## 125	4.478037	404.803343	44.077715	17.846549	72.730750	41.028328	
## 131	4.160975	265.136463	15.528614	13.540402	130.804062	41.664671	
## 133	83.900583	259.425512	113.461714	100.588102	53.025576	53.477803	
##	Hypocotyl	SAM6D	SAM17D	SAM38D	Callus	Leaf	Root
## 5	172.112449	101.538840	108.035003	57.568952	3.602672	72.181052	53.728970
## 9	114.709942	7.356839	6.637677	7.455989	1.051705	20.562404	67.383610
## 20	4.159165	21.230577	13.709403	9.639602	2.603449	42.803542	126.598709
## 21	7.071054	47.672295	55.806546	37.757871	0.685890	8.647167	43.277837
## 25	215.724837	75.229555	108.926253	87.081914	65.420159	178.440543	98.055423
## 29	13.614824	40.036100	67.500984	33.565553	5.088175	22.636836	12.527492

##	32	20.950298	31.857638	31.063118	16.461606	4.368482	44.030052	25.338768		
##	33	38.494199	75.216825	290.336670	100.526085	25.977418	0.000000	0.000000		
##	36	118.328990	229.139072	393.546142	181.391747	71.756006	45.788590	20.983248		
##	38	13.618860	48.102339	51.456817	31.907033	8.665198	27.396630	14.501406		
##	39	18.983798	38.298793	237.526834	76.153152	6.048278	3.674760	1.606524		
##	49	26.934453	45.899996	104.119336	30.814124	3.509392	48.206033	26.688307		
##	76	350.297556	2.600212	4.426392	5.209396	2.744771	20.850299	94.160633		
##	104	17.456177	22.806197	19.605122	9.950014	12.500697	19.094595	14.355242		
##	117	27.176499	82.133333	87.939197	38.297183	47.396469	32.138566	17.503017		
##	120	1.046225	28.072725	24.136925	7.085973	69.182090	2.954026	4.491654		
##	123	19.247151	60.919431	75.317808	31.290701	44.433814	27.410816	21.526502		
##	125	177.214792	3.197441	2.314000	2.924909	0.843634	6.919914	6.338750		
##	131	633.844120	1.661192	0.821472	1.444420	0.244966	0.972353	62.263900		
##	133	102.248657	92.557209	144.677356	86.120903	10.711991	21.058177	104.233361		
##		Nodule	PC1	PC2	PC3	PC4	in.ell_TIR			
##	5	22.341412	7.4596126	-0.60000202	2.0307657	0.4458205	FALSE			
##	9	1.533396	1.5743362	-5.04177716	-0.6172365	0.6633938	FALSE			
##	20	27.491901	1.4183062	-2.03622924	5.7800703	1.3629861	FALSE			
##	21	3.279099	1.6232820	0.38038019	0.1987720	1.1554167	FALSE			
##	25	5.415049	13.3925129	-1.80641239	2.1698302	-5.4199946	FALSE			
##	29	10.758737	3.1205196	1.14297385	0.1671277	1.3202221	FALSE			
##	32	21.300657	1.3379944	0.01555935	2.7843694	0.4682530	FALSE			
##	33	0.000000	6.7026480	3.57934670	-3.3730284	1.3243790	FALSE			
##	36	22.794609	15.0378929	4.74623732	-1.3006126	-0.9309800	FALSE			
##	38	9.673077	3.2714398	1.25022397	0.2925031	0.9717274	FALSE			
##	39	0.000000	5.2755840	2.30540719	-2.8806692	1.9555757	FALSE			
##	49	8.836310	2.7417865	0.75315742	0.9842749	0.4962433	FALSE			
##	76	5.269519	2.2432846	-9.99777623	-1.0514431	-0.9241382	FALSE			
##	104	3.394860	-0.2309816	0.20105066	0.5268626	-0.5560466	FALSE			
##	117	8.945171	3.9528719	1.62593170	0.5054186	-1.6929049	FALSE			
##	120	0.392173	-0.2798307	1.61342692	0.2055221	-3.4292754	FALSE			
##	123	7.313215	2.6152321	1.16403114	0.5868563	-1.5858479	FALSE			
##	125	0.000000	0.7194803	-4.32442630	-2.0960569	-0.4665481	FALSE			
##	131	0.495348	1.4844789	-7.91342992	-2.4573024	-0.8906494	FALSE			
##	133	53.539749	9.8773408	-2.30022562	3.5022307	4.6373748	FALSE			
##		in.ell_IAA	in.ell_ARF	in.ell						
##	5	FALSE	FALSE	FALSE						
##	9	FALSE	FALSE	FALSE						
##	20	TRUE	FALSE	FALSE						
##	21	TRUE	FALSE	FALSE						
##	25	FALSE	FALSE	FALSE						
##	29	TRUE	FALSE	FALSE						
##	32	TRUE	FALSE	FALSE						
##	33	FALSE	FALSE	FALSE						
##	36	FALSE	FALSE	FALSE						
##	38	TRUE	FALSE	FALSE						
##	39	FALSE	FALSE	FALSE						
##	49	TRUE	FALSE	FALSE						
##	76	FALSE	FALSE	FALSE						
##	104	TRUE	TRUE	FALSE						
##	117	TRUE	FALSE	FALSE						
##	120	TRUE	FALSE	FALSE						
##	123	TRUE	FALSE	FALSE						
##	125	FALSE	FALSE	FALSE						

```
## 131      FALSE      FALSE  FALSE
## 133      FALSE      FALSE  FALSE
```

```
PC1_2all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
                                         xend=(PC1*8.75), yend=(PC2*8.75)),
              arrow = arrow(length = unit(1/2, "picas"), color="gray60") +
              annotate("text", x=(PCA_loadings$PC1*8.75), #add the tissue names to it manually
                      y=(PCA_loadings$PC2*8.75),
                      label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
              theme(panel.background = element_rect(fill = "white", linewidth = 1)) +
              theme_bw() +
              scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  ggrepel::geom_text_repel(data = expr_logic %>%
                          as_tibble(rownames = "name") %>%
                          filter(as.logical(in_ell == FALSE)),
                          aes(PC1, PC2, label=sub(".*\\|", "", heatmap_label)),
                          size=3, max.overlaps = 100, min.segment.length = 0,
                          segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()
```



```
# ggsave("20230927_PC1_2_allTissues.png", dpi = 1000, width = 10, height = 8)
# ggsave("20230927_PC1_2_allTissues.pdf", dpi = 1000, width = 10, height = 8)
```

Lets now repeat with the tau values

```
# combine data frames expr_logic with tau values
expr_logic2 <- cbind(expr_logic, tau_df$tau)

# Define the intervals for expr_logic2$tau
expr_logic2$tau_interval <- cut(expr_logic2$tau,
```

```

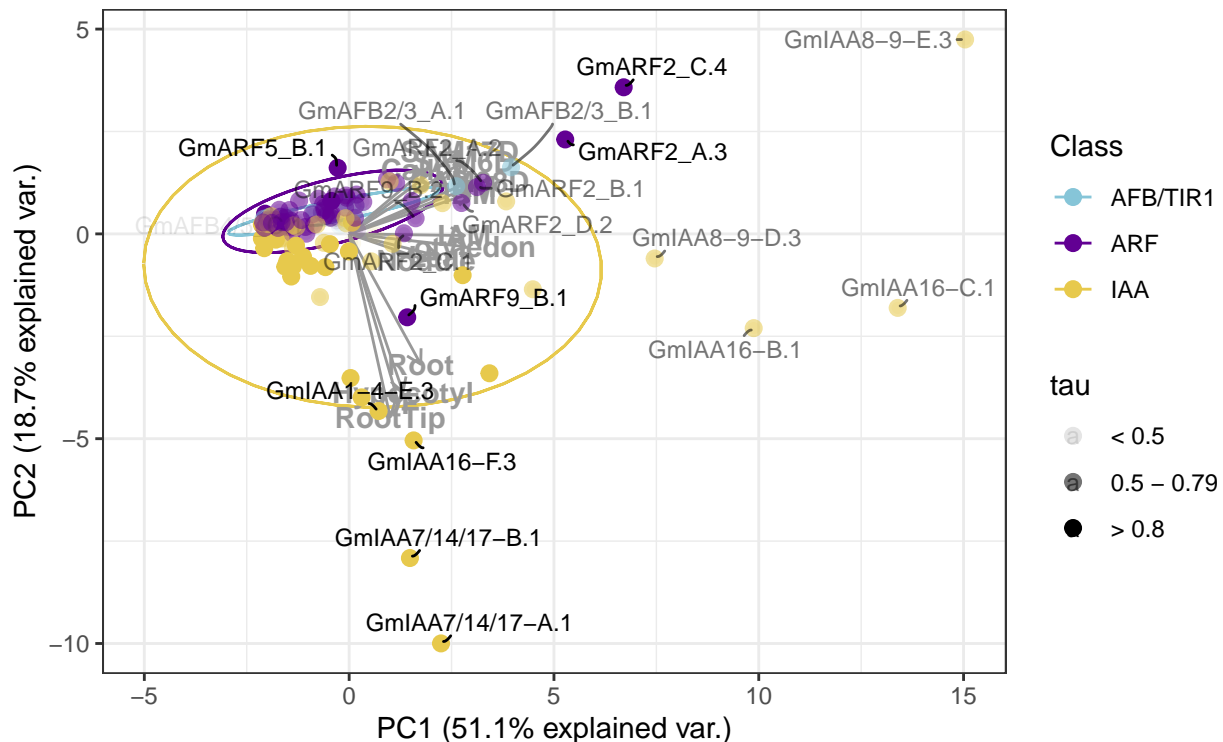
breaks = c(-Inf, 0.5, 0.8, Inf),
labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC1_2all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
                                         xend=(PC1*8.75), yend=(PC2*8.75)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings$PC1*8.75), #add the tissue names to it manually
           y=(PCA_loadings$PC2*8.75),
           label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw() +
  geom_point(data = expr_logic2 %>% as_tibble(rownames = "name"),
            aes(PC1, PC2, alpha = tau_interval, color = Family), size=2.5) +
  ggrepel::geom_text_repel(data = expr_logic2 %>%
                           as_tibble(rownames = "name") %>%
                           filter(as.logical(in_ell == FALSE)),
                           aes(PC1, PC2, label=sub(".*\\|", "", heatmap_label),
                               alpha = tau_interval),
                           size=3, max.overlaps = 100, min.segment.length = 0,
                           segment.curvature = -0.1) +

  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C"))+
  labs(color = "Class", shape = "tau", alpha = "tau") +
  theme_bw()

```

Warning: Using alpha for a discrete variable is not advised.



```

# ggsave("20240117_PC1_2_allTissues.png", dpi = 1000, width = 10, height = 8)
# ggsave("20240117_PC1_2_allTissues.pdf", dpi = 1000, width = 10, height = 8)

```


PC2 and PC3 outside ellipse labels for the 14 tissues

```
PC2_3all <- ggbiplot::ggbiplot(pca, obs.scale = 1, var.scale = .5, choices = c(2, 3),
                              varname.adjust = 3, alpha = .1,
                              groups = Family, var.axes = F, varname.size = 2,
                              ellipse = TRUE, circle = FALSE, ellipse.prob = .70)
```

```
# Extract components so we can select out
PC2_3all$data
```

##	xvar	yvar	groups
## 1	0.253967120	0.192765997	ARF
## 2	0.185928231	0.323910990	IAA
## 3	-0.113511655	0.829138289	IAA
## 4	0.274717011	-0.929827851	IAA
## 5	-0.600002016	2.030765709	IAA
## 6	0.859463281	-0.543671987	ARF
## 7	-0.427960267	-0.820157717	IAA
## 8	0.431662229	0.097838721	AFB/TIR1
## 9	-5.041777156	-0.617236529	IAA
## 10	-0.558512920	0.212488063	IAA
## 11	-0.811787304	-1.025141950	IAA
## 12	0.788703488	-0.633363301	AFB/TIR1
## 13	0.129024379	0.388439486	AFB/TIR1
## 14	-0.670223439	1.484374427	IAA
## 15	0.750445550	-1.880148999	IAA
## 16	0.330600703	0.213426698	ARF
## 17	1.334152162	-1.082153419	ARF
## 18	0.157370925	-0.395001161	ARF
## 19	0.500704876	0.080303779	ARF
## 20	-2.036229235	5.780070252	ARF
## 21	0.380380191	0.198771991	ARF
## 22	-0.114441199	-0.085693178	IAA
## 23	-3.402760865	-3.388444330	IAA
## 24	0.466557952	0.166835894	AFB/TIR1
## 25	-1.806412389	2.169830163	IAA
## 26	0.019944454	1.074396091	ARF
## 27	0.256945271	0.489833024	IAA
## 28	0.476207228	-0.051671359	AFB/TIR1
## 29	1.142973848	0.167127745	ARF
## 30	0.803815826	-0.548952095	ARF
## 31	0.375426589	0.157488574	ARF
## 32	0.015559355	2.784369433	ARF
## 33	3.579346703	-3.373028396	ARF
## 34	0.494196522	0.018029747	ARF
## 35	-0.218288936	0.162561278	IAA
## 36	4.746237323	-1.300612613	IAA
## 37	0.398718486	0.569404529	AFB/TIR1
## 38	1.250223968	0.292503137	ARF
## 39	2.305407189	-2.880669155	ARF
## 40	-0.788772964	-0.633847587	IAA
## 41	0.526307555	0.141808591	ARF
## 42	0.913360463	-0.656456898	ARF

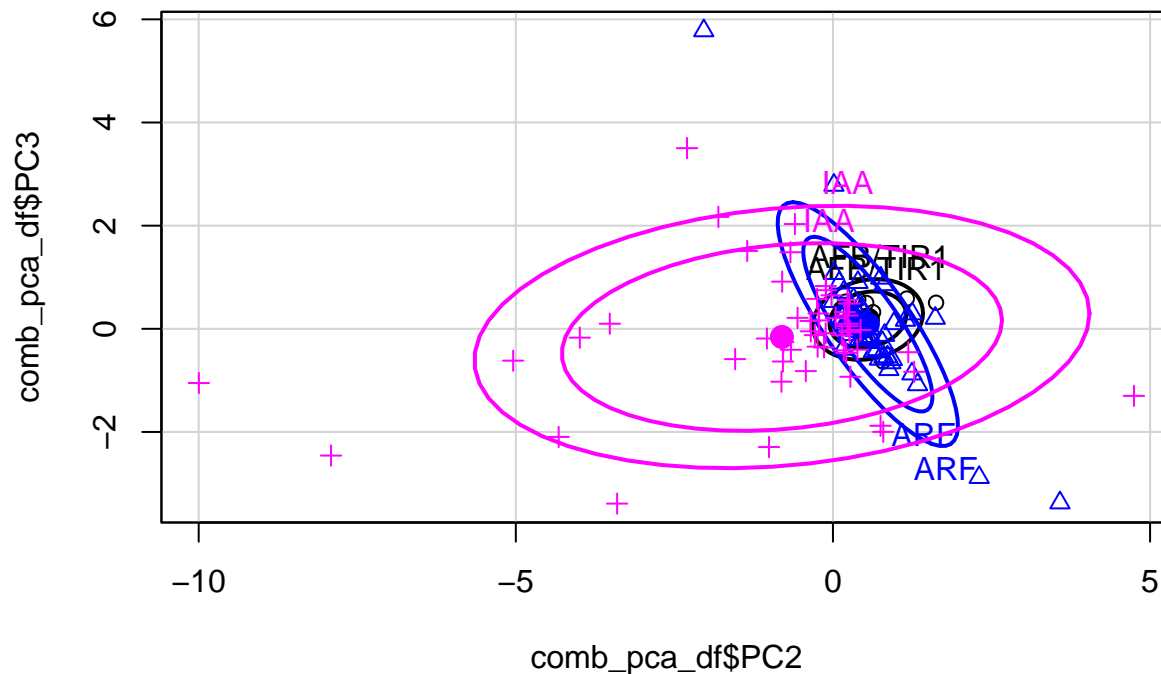
## 43	0.098979610	0.889037817	ARF
## 44	0.394826074	0.286525824	ARF
## 45	0.413220346	-0.086074243	AFB/TIR1
## 46	0.421500106	0.078045033	AFB/TIR1
## 47	0.323815463	0.576287661	ARF
## 48	0.393931829	0.899632729	ARF
## 49	0.753157423	0.984274929	ARF
## 50	1.245116561	-0.873135543	ARF
## 51	0.202653740	0.140445647	ARF
## 52	-0.023380766	0.649133026	IAA
## 53	0.141276678	-0.125741994	IAA
## 54	0.261934275	0.016357767	AFB/TIR1
## 55	0.732872508	-0.587549057	ARF
## 56	0.168806405	0.710504593	ARF
## 57	-0.140394034	-0.360521593	IAA
## 58	0.231182561	0.257918902	IAA
## 59	0.793449297	-1.998468838	IAA
## 60	0.395883644	0.150013982	ARF
## 61	0.229001671	0.049754903	IAA
## 62	-0.241185818	-0.345426570	IAA
## 63	-0.249922455	0.580533544	IAA
## 64	0.524357821	0.502244955	AFB/TIR1
## 65	0.369234056	0.114997043	AFB/TIR1
## 66	0.507222423	-0.214148722	AFB/TIR1
## 67	-0.662268126	-0.406399893	IAA
## 68	-0.800512633	0.914386871	IAA
## 69	0.043427973	0.206015394	IAA
## 70	0.277764356	0.074614796	ARF
## 71	-1.351421276	1.509715283	IAA
## 72	-1.008109821	-2.292586475	IAA
## 73	0.262406297	-0.029598759	IAA
## 74	-1.541788959	-0.587753821	IAA
## 75	-3.991108850	-0.171191334	IAA
## 76	-9.997776235	-1.051443078	IAA
## 77	0.004571344	0.531597276	ARF
## 78	0.223425301	-0.109188343	ARF
## 79	0.174794131	0.317995304	ARF
## 80	0.937109194	-0.594875791	ARF
## 81	0.582823128	0.004808883	ARF
## 82	0.149412964	0.324427333	ARF
## 83	0.479261947	-0.173718424	ARF
## 84	0.804159673	0.857615531	ARF
## 85	0.589012162	-0.146999936	ARF
## 86	0.192216165	0.092837656	ARF
## 87	0.339552665	0.275671087	ARF
## 88	0.145743103	0.204583749	ARF
## 89	-0.108332600	0.749279943	IAA
## 90	0.350064536	-0.025807891	ARF
## 91	0.882006175	-0.788482703	ARF
## 92	0.780242803	-0.264669941	ARF
## 93	0.458750897	0.352511287	ARF
## 94	0.443802208	-0.049101215	ARF
## 95	1.281462617	-0.833524403	IAA
## 96	0.229054413	0.559131560	IAA

```
## 97 0.435235820 -0.022773811 IAA
## 98 -0.353257233 0.152202391 IAA
## 99 -1.040984678 -0.188391551 IAA
## 100 0.207864794 -0.465096502 IAA
## 101 0.171175760 -0.417437561 IAA
## 102 0.338183452 0.239714138 ARF
## 103 0.642577969 -0.433940124 ARF
## 104 0.201050662 0.526862594 AFB/TIR1
## 105 -0.228022083 0.297046621 IAA
## 106 0.696034794 -0.401907770 ARF
## 107 0.814553891 -0.125386206 ARF
## 108 -0.285475152 -0.122534533 IAA
## 109 0.237586347 0.097778388 IAA
## 110 1.187868663 -0.453351398 IAA
## 111 0.638749151 -0.442597992 ARF
## 112 0.418512190 -0.204069007 ARF
## 113 0.355458289 0.229584716 ARF
## 114 0.311126791 0.219278723 ARF
## 115 0.428375254 0.457407776 ARF
## 116 0.960752443 0.101790408 ARF
## 117 1.625931704 0.505418632 AFB/TIR1
## 118 0.285122668 -0.017733146 ARF
## 119 0.262895090 -0.064799773 IAA
## 120 1.613426920 0.205522094 ARF
## 121 0.577784192 -0.221521206 ARF
## 122 0.273681105 0.113040052 ARF
## 123 1.164031145 0.586856325 AFB/TIR1
## 124 -0.349419050 -0.048212877 IAA
## 125 -4.324426296 -2.096056941 IAA
## 126 -3.518281284 0.098701172 IAA
## 127 0.285127917 0.193201295 ARF
## 128 0.633239300 0.321861245 AFB/TIR1
## 129 0.385551512 -0.405549799 IAA
## 130 0.267885408 0.074216762 ARF
## 131 -7.913429916 -2.457302433 IAA
## 132 -0.781215430 -0.258947266 IAA
## 133 -2.300225618 3.502230742 IAA
```

```
build_PC2_3all <- ggplot_build(PC2_3all)$data
points_PC2_3all <- build_PC2_3all[[1]]
```

```
# co-ordinates of the ellipses
```

```
ell_points_PC2_3all <- car::dataEllipse(comb_pca_df$PC2,
                                          comb_pca_df$PC3,
                                          as.factor(comb_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point

ell_ARF_PC2_3all <- as.data.frame(ell_points_PC2_3all$ARF$`0.7`)

ell_IAA_PC2_3all <- as.data.frame(ell_points_PC2_3all$IAA$`0.7`)

ell_TIR_PC2_3all <- as.data.frame(ell_points_PC2_3all$`AFB/TIR1`$`0.7`)

# Find which points are outside (!) the ellipse, and add this to the data

dat_TIR_PC2_3all <- data.frame(
  points_PC2_3all[2:3],
  in_ell_TIR = as.logical(point.in.polygon(points_PC2_3all$x, points_PC2_3all$y, ell_TIR_PC2_3all$x, ell_TIR_PC2_3all$y))
)

dat_IAA_PC2_3all <- data.frame(
  points_PC2_3all[2:3],
  in_ell_IAA = as.logical(point.in.polygon(points_PC2_3all$x, points_PC2_3all$y, ell_IAA_PC2_3all$x, ell_IAA_PC2_3all$y))
)

dat_ARF_PC2_3all <- data.frame(
  points_PC2_3all[2:3],
  in_ell_ARF = as.logical(point.in.polygon(points_PC2_3all$x, points_PC2_3all$y, ell_ARF_PC2_3all$x, ell_ARF_PC2_3all$y))
)

# as.logical(point..) equals to TRUE indicated points are inside ellipses

# Combining data points for labeling
transcript_expr_logic_PC2_3all <- cbind(comb_pca_df, dat_TIR_PC2_3all, dat_IAA_PC2_3all, dat_ARF_PC2_3all)

expr_logic_PC2_3all <- transcript_expr_logic_PC2_3all %>% mutate(., in_ell = case_when(Family=="ARF" &
  Family=="IAA" & `in_ell_IAA` == TRUE ~ "TRUE",
```

```

Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
mutate(in_ell = coalesce(in_ell, "FALSE"))
expr_logic_PC2_3all[which(expr_logic_PC2_3all$in_ell == FALSE),]

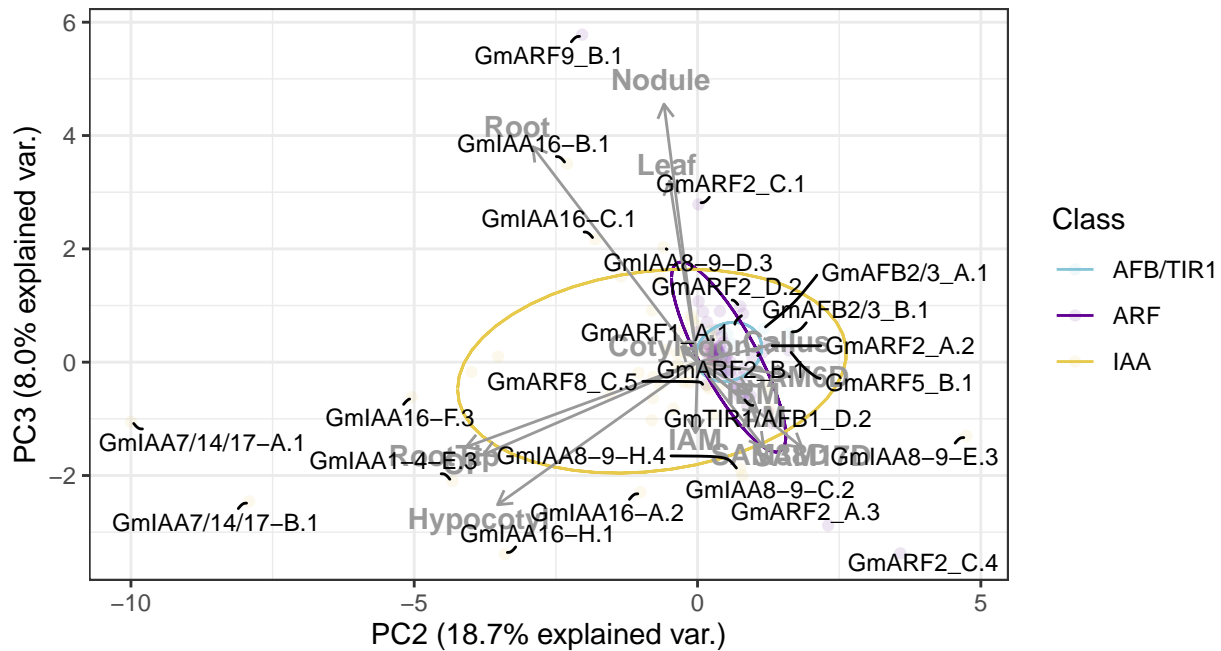
```

##	Transcript ID	heatmap_label	Family	Class	Clade	
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I	
## 9	Glyma.02G142500.3	Glyma.02G142500 GmIAA16-F.3	IAA	C	III	
## 12	Glyma.02G152800.2	Glyma.02G152800 GmTIR1/AFB1_D.2	AFB/TIR1	TIR1/AFB1	I	
## 15	Glyma.02G218100.4	Glyma.02G218100 GmIAA8-9-H.4	IAA	A	I	
## 18	Glyma.02G239600.5	Glyma.02G239600 GmARF8_C.5	ARF	A	II	
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I	
## 23	Glyma.03G158700.1	Glyma.03G158700 GmIAA16-H.1	IAA	C	III	
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III	
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I	
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I	
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I	
## 36	Glyma.06G091700.3	Glyma.06G091700 GmIAA8-9-E.3	IAA	A	I	
## 38	Glyma.06G164900.2	Glyma.06G164900 GmARF2_A.2	ARF	B	I	
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I	
## 49	Glyma.08G008100.2	Glyma.08G008100 GmARF2_D.2	ARF	B	I	
## 59	Glyma.08G273500.2	Glyma.08G273500 GmIAA8-9-C.2	IAA	A	I	
## 72	Glyma.10G162400.2	Glyma.10G162400 GmIAA16-A.2	IAA	C	III	
## 76	Glyma.10G180100.1	Glyma.10G180100 GmIAA7/14/17-A.1	IAA	C	III	
## 84	Glyma.12G164100.1	Glyma.12G164100 GmARF1_A.1	ARF	B	I	
## 117	Glyma.16G050500.1	Glyma.16G050500 GmAFB2/3_B.1	AFB/TIR1	AFB2/3	II	
## 120	Glyma.17G256500.1	Glyma.17G256500 GmARF5_B.1	ARF	A	II	
## 123	Glyma.19G100200.1	Glyma.19G100200 GmAFB2/3_A.1	AFB/TIR1	AFB2/3	II	
## 125	Glyma.19G161000.3	Glyma.19G161000 GmIAA1-4-E.3	IAA	B	II	
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III	
## 133	Glyma.20G225000.1	Glyma.20G225000 GmIAA16-B.1	IAA	C	III	
##	AM	OF	IAM	IBM	RootTip	Cotyledon
## 5	68.216252	51.350688	53.553189	96.275020	32.628226	78.301786
## 9	14.688070	529.887081	73.110411	28.516509	43.238091	0.202364
## 12	22.069192	16.181094	21.004047	22.849022	12.164769	15.200032
## 15	58.683503	91.015780	54.923525	50.038109	25.887821	8.588162
## 18	13.757865	13.859804	8.771196	7.354321	6.099923	14.166313
## 20	14.963834	9.149436	10.014337	18.303006	6.361899	6.335987
## 23	62.978354	188.282417	63.415971	50.273803	93.913232	11.797675
## 25	123.752976	182.226043	143.672394	136.882838	68.294552	100.483878
## 29	59.132821	34.776950	54.023677	104.014185	5.893899	15.036259
## 32	23.736932	7.300337	13.387625	33.516112	4.197497	21.693739
## 33	106.794815	41.848269	90.206750	89.149798	3.924654	25.580965
## 36	120.096829	73.349658	80.928913	112.799969	14.116324	123.198735
## 38	62.736767	31.362845	56.712867	104.850125	1.621706	14.695636
## 39	78.161251	91.333242	102.461124	95.846621	1.569549	29.653419
## 49	41.952212	22.684099	48.464588	54.600720	2.166574	14.976262
## 59	60.235552	42.618679	62.597901	70.694115	15.151090	25.020319
## 72	50.429036	62.648515	44.799041	49.047824	54.886130	14.549869
## 76	7.941917	536.895072	6.035997	7.033665	214.246604	30.729826
## 84	29.907851	25.658848	21.848792	33.978092	12.158648	15.573356
## 117	55.698420	35.061948	32.831068	58.586254	13.107811	27.899216
## 120	7.845772	2.581028	3.311743	17.662479	0.302999	1.222648
## 123	42.087683	43.031580	27.129734	42.689001	13.615904	12.545145

## 125	4.478037	404.803343	44.077715	17.846549	72.730750	41.028328	
## 131	4.160975	265.136463	15.528614	13.540402	130.804062	41.664671	
## 133	83.900583	259.425512	113.461714	100.588102	53.025576	53.477803	
##	Hypocotyl	SAM6D	SAM17D	SAM38D	Callus	Leaf	Root
## 5	172.112449	101.538840	108.035003	57.568952	3.602672	72.181052	53.728970
## 9	114.709942	7.356839	6.637677	7.455989	1.051705	20.562404	67.383610
## 12	20.202937	29.352679	33.917530	17.176745	17.838179	0.000000	0.000000
## 15	51.413131	79.581114	54.407495	42.568039	6.383128	0.000000	0.000000
## 18	33.826894	0.000000	11.829332	13.561473	1.551810	0.000000	0.000000
## 20	4.159165	21.230577	13.709403	9.639602	2.603449	42.803542	126.598709
## 23	382.933284	20.163323	39.919056	73.535631	0.583159	6.250057	14.892682
## 25	215.724837	75.229555	108.926253	87.081914	65.420159	178.440543	98.055423
## 29	13.614824	40.036100	67.500984	33.565553	5.088175	22.636836	12.527492
## 32	20.950298	31.857638	31.063118	16.461606	4.368482	44.030052	25.338768
## 33	38.494199	75.216825	290.336670	100.526085	25.977418	0.000000	0.000000
## 36	118.328990	229.139072	393.546142	181.391747	71.756006	45.788590	20.983248
## 38	13.618860	48.102339	51.456817	31.907033	8.665198	27.396630	14.501406
## 39	18.983798	38.298793	237.526834	76.153152	6.048278	3.674760	1.606524
## 49	26.934453	45.899996	104.119336	30.814124	3.509392	48.206033	26.688307
## 59	166.602217	77.910628	90.255018	63.645599	6.462795	11.030950	6.618096
## 72	258.563372	52.919869	43.482386	62.182188	12.401001	7.425099	12.077243
## 76	350.297556	2.600212	4.426392	5.209396	2.744771	20.850299	94.160633
## 84	22.344234	27.195518	37.963940	21.692550	51.460680	19.141297	21.871309
## 117	27.176499	82.133333	87.939197	38.297183	47.396469	32.138566	17.503017
## 120	1.046225	28.072725	24.136925	7.085973	69.182090	2.954026	4.491654
## 123	19.247151	60.919431	75.317808	31.290701	44.433814	27.410816	21.526502
## 125	177.214792	3.197441	2.314000	2.924909	0.843634	6.919914	6.338750
## 131	633.844120	1.661192	0.821472	1.444420	0.244966	0.972353	62.263900
## 133	102.248657	92.557209	144.677356	86.120903	10.711991	21.058177	104.233361
##	Nodule	PC1	PC2	PC3	PC4	in.ell_TIR	
## 5	22.341412	7.4596126	-0.60000202	2.0307657	0.445820518	FALSE	
## 9	1.533396	1.5743362	-5.04177716	-0.6172365	0.663393838	FALSE	
## 12	0.000000	-0.1024578	0.78870349	-0.6333633	-0.606570751	FALSE	
## 15	0.000000	2.2767296	0.75044555	-1.8801490	0.882776840	FALSE	
## 18	0.000000	-1.4173576	0.15737092	-0.3950012	-0.059740809	FALSE	
## 20	27.491901	1.4183062	-2.03622924	5.7800703	1.362986140	FALSE	
## 23	2.016265	3.4253377	-3.40276087	-3.3884443	0.846071762	FALSE	
## 25	5.415049	13.3925129	-1.80641239	2.1698302	-5.419994579	FALSE	
## 29	10.758737	3.1205196	1.14297385	0.1671277	1.320222051	TRUE	
## 32	21.300657	1.3379944	0.01555935	2.7843694	0.468252980	FALSE	
## 33	0.000000	6.7026480	3.57934670	-3.3730284	1.324379018	FALSE	
## 36	22.794609	15.0378929	4.74623732	-1.3006126	-0.930979975	FALSE	
## 38	9.673077	3.2714398	1.25022397	0.2925031	0.971727378	FALSE	
## 39	0.000000	5.2755840	2.30540719	-2.8806692	1.955575677	FALSE	
## 49	8.836310	2.7417865	0.75315742	0.9842749	0.496243282	FALSE	
## 59	1.380951	3.8343944	0.79344930	-1.9984688	0.825223890	FALSE	
## 72	0.000000	2.7711864	-1.00810982	-2.2925865	-0.002501458	FALSE	
## 76	5.269519	2.2432846	-9.99777623	-1.0514431	-0.924138217	FALSE	
## 84	7.347936	1.5268214	0.80415967	0.8576155	-2.169356529	FALSE	
## 117	8.945171	3.9528719	1.62593170	0.5054186	-1.692904946	FALSE	
## 120	0.392173	-0.2798307	1.61342692	0.2055221	-3.429275402	FALSE	
## 123	7.313215	2.6152321	1.16403114	0.5868563	-1.585847922	FALSE	
## 125	0.000000	0.7194803	-4.32442630	-2.0960569	-0.466548138	FALSE	
## 131	0.495348	1.4844789	-7.91342992	-2.4573024	-0.890649441	FALSE	

```
## 133 53.539749 9.8773408 -2.30022562 3.5022307 4.637374786 FALSE
##      in_ell_1AA in_ell_ARF in_ell
## 5      FALSE      FALSE FALSE
## 9      FALSE      FALSE FALSE
## 12     TRUE       TRUE  FALSE
## 15     FALSE      FALSE FALSE
## 18     TRUE      FALSE FALSE
## 20     FALSE      FALSE FALSE
## 23     FALSE      FALSE FALSE
## 25     FALSE      FALSE FALSE
## 29     TRUE      FALSE FALSE
## 32     FALSE      FALSE FALSE
## 33     FALSE      FALSE FALSE
## 36     FALSE      FALSE FALSE
## 38     TRUE      FALSE FALSE
## 39     FALSE      FALSE FALSE
## 49     TRUE      FALSE FALSE
## 59     FALSE      FALSE FALSE
## 72     FALSE      FALSE FALSE
## 76     FALSE      FALSE FALSE
## 84     TRUE      FALSE FALSE
## 117    TRUE      FALSE FALSE
## 120    TRUE      FALSE FALSE
## 123    TRUE      FALSE FALSE
## 125    FALSE      FALSE FALSE
## 131    FALSE      FALSE FALSE
## 133    FALSE      FALSE FALSE
```

```
PC2_3all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
                                         xend=(PC2*8), yend=(PC3*8)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings$PC2*8.75), #add the tissue names to it manually
           y=(PCA_loadings$PC3*8.75),
           label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw()+
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C"))+
  ggrepel::geom_text_repel(data = expr_logic_PC2_3all %>%
                           as_tibble(rownames = "name") %>%
                           filter(as.logical(in_ell == FALSE)),
                           aes(PC2, PC3, label=sub(".*\\|", "", heatmap_label)),
                           size=3, max.overlaps = 100, min.segment.length = 0,
                           segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()
```



```
# ggsave("20230927_PC2_3_allTissues.png", dpi = 1000, width = 10, height = 8)
# ggsave("20230927_PC2_3_allTissues.pdf", dpi = 1000, width = 10, height = 8)
```

Lets now repeat with the tau values

```
# combine data frames expr_logic with tau values
expr_logic_PC2_3 <- cbind(expr_logic_PC2_3all, tau_df$tau)

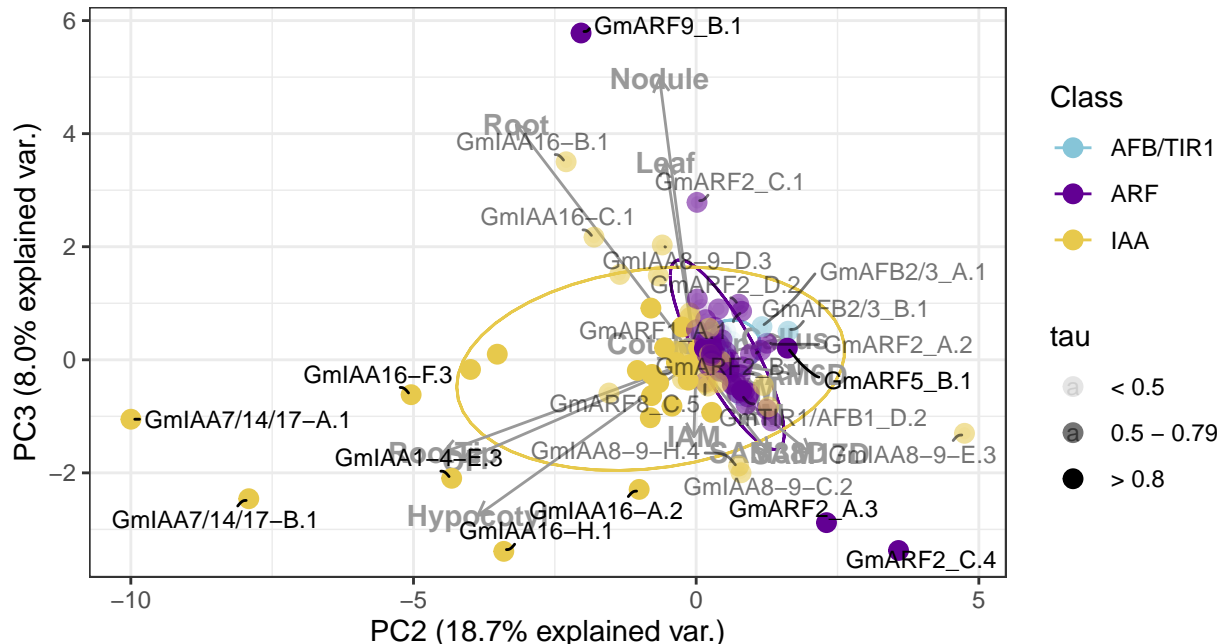
# Define the intervals for expr_logic2$tau
expr_logic_PC2_3$tau_interval <- cut(expr_logic_PC2_3$tau,
  breaks = c(-Inf, 0.5, 0.8, Inf),
  labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC2_3all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
    xend=(PC2*8.75), yend=(PC3*8.75)),
    arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings$PC2*8.75), #add the tissue names to it manually
    y=(PCA_loadings$PC3*8.75),
    label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw() +
  geom_point(data = expr_logic_PC2_3 %>% as_tibble(rownames = "name"),
    aes(PC2, PC3, alpha = tau_interval, color = Family),
    size=3) +
  ggrepel::geom_text_repel(data = expr_logic_PC2_3 %>%
    as_tibble(rownames = "name") %>%
    filter(as.logical(in_ell == FALSE)),
    aes(PC2, PC3, label=sub(".*\\|", "", heatmap_label),
      alpha = tau_interval),
    size=3, max.overlaps = 100, min.segment.length = 0,
    segment.curvature = -0.1) +
```



```
scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C"))+
labs(color = "Class", shape = "tau", alpha = "tau") +
theme_bw()
```

Warning: Using alpha for a discrete variable is not advised.



```
# ggsave("20240117_PC2_3_allTissues.png", dpi = 1000, width = 10, height = 8)
# ggsave("20240117_PC2_3_allTissues.pdf", dpi = 1000, width = 10, height = 8)
```

PC3 and PC4 for all 14 tissues

```
PC3_4all <- ggbiplot::ggbiplot(pca, obs.scale = 1, var.scale = .5, choices = c(3,4),
  varname.adjust = 3, alpha = .1,
  groups = Family, var.axes = F, varname.size = 2,
  ellipse = TRUE, circle = FALSE, ellipse.prob = .70)
```

```
# Extract components so we can select out
PC3_4all$data
```

##	xvar	yvar	groups
## 1	0.192765997	-0.329157357	ARF
## 2	0.323910990	-0.475687016	IAA
## 3	0.829138289	-0.036476522	IAA
## 4	-0.929827851	-0.596814655	IAA
## 5	2.030765709	0.445820518	IAA
## 6	-0.543671987	0.467432769	ARF
## 7	-0.820157717	0.205971511	IAA
## 8	0.097838721	0.071591356	AFB/TIR1
## 9	-0.617236529	0.663393838	IAA
## 10	0.212488063	0.372205118	IAA
## 11	-1.025141950	0.497830190	IAA

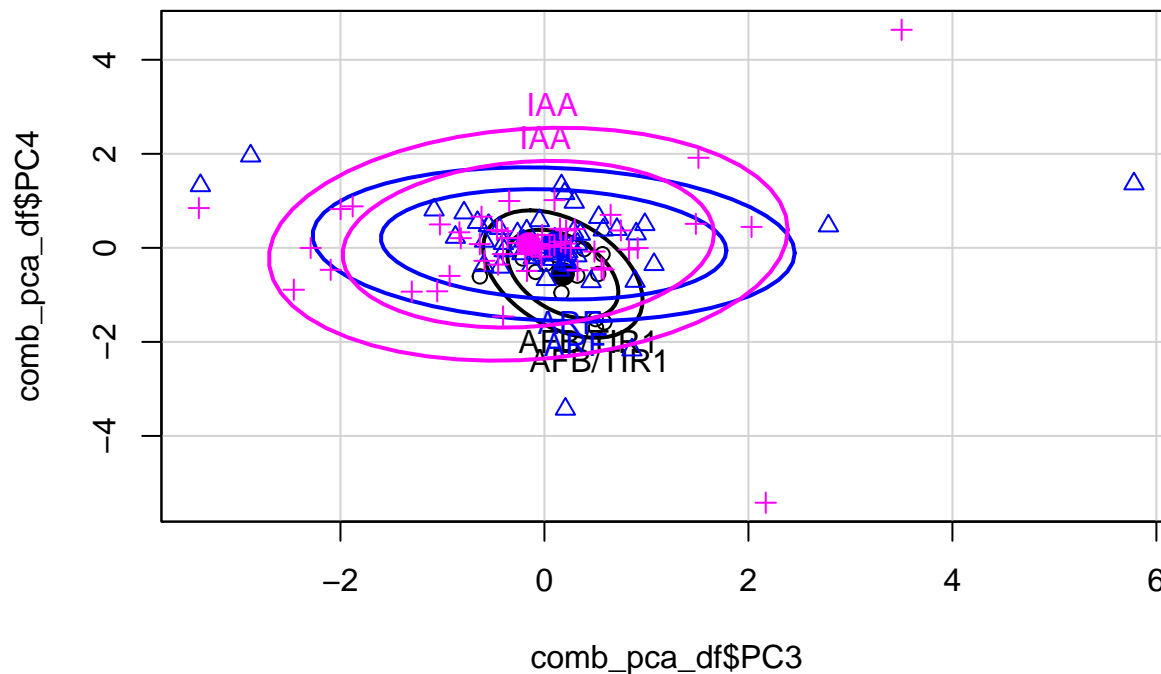
## 12	-0.633363301	-0.606570751	AFB/TIR1
## 13	0.388439486	-0.027544084	AFB/TIR1
## 14	1.484374427	0.510884280	IAA
## 15	-1.880148999	0.882776840	IAA
## 16	0.213426698	-0.342539454	ARF
## 17	-1.082153419	0.807398759	ARF
## 18	-0.395001161	-0.059740809	ARF
## 19	0.080303779	-0.243991322	ARF
## 20	5.780070252	1.362986140	ARF
## 21	0.198771991	1.155416668	ARF
## 22	-0.085693178	0.034427672	IAA
## 23	-3.388444330	0.846071762	IAA
## 24	0.166835894	-0.952600938	AFB/TIR1
## 25	2.169830163	-5.419994579	IAA
## 26	1.074396091	-0.353111432	ARF
## 27	0.489833024	-0.078333073	IAA
## 28	-0.051671359	-0.075470037	AFB/TIR1
## 29	0.167127745	1.320222051	ARF
## 30	-0.548952095	0.475445406	ARF
## 31	0.157488574	0.046630592	ARF
## 32	2.784369433	0.468252980	ARF
## 33	-3.373028396	1.324379018	ARF
## 34	0.018029747	-0.672254946	ARF
## 35	0.162561278	0.130306707	IAA
## 36	-1.300612613	-0.930979975	IAA
## 37	0.569404529	-0.135589979	AFB/TIR1
## 38	0.292503137	0.971727378	ARF
## 39	-2.880669155	1.955575677	ARF
## 40	-0.633847587	0.080138588	IAA
## 41	0.141808591	0.005907838	ARF
## 42	-0.656456898	0.540261125	ARF
## 43	0.889037817	-0.709455025	ARF
## 44	0.286525824	0.333019125	ARF
## 45	-0.086074243	-0.512682037	AFB/TIR1
## 46	0.078045033	-0.027875326	AFB/TIR1
## 47	0.576287661	0.380679666	ARF
## 48	0.899632729	0.298409644	ARF
## 49	0.984274929	0.496243282	ARF
## 50	-0.873135543	0.226144958	ARF
## 51	0.140445647	-0.084054512	ARF
## 52	0.649133026	0.701903529	IAA
## 53	-0.125741994	0.095581276	IAA
## 54	0.016357767	-0.335554133	AFB/TIR1
## 55	-0.587549057	0.144105430	ARF
## 56	0.710504593	0.391861533	ARF
## 57	-0.360521593	0.209868199	IAA
## 58	0.257918902	-0.003108430	IAA
## 59	-1.998468838	0.825223890	IAA
## 60	0.150013982	-0.471664728	ARF
## 61	0.049754903	0.022310438	IAA
## 62	-0.345426570	0.996380369	IAA
## 63	0.580533544	-0.463754306	IAA
## 64	0.502244955	-1.514606539	AFB/TIR1
## 65	0.114997043	-0.155428110	AFB/TIR1

## 66	-0.214148722	-0.225376445	AFB/TIR1
## 67	-0.406399893	-0.132299390	IAA
## 68	0.914386871	-0.003733875	IAA
## 69	0.206015394	0.088858545	IAA
## 70	0.074614796	-0.126598662	ARF
## 71	1.509715283	1.915691304	IAA
## 72	-2.292586475	-0.002501458	IAA
## 73	-0.029598759	-0.187635626	IAA
## 74	-0.587753821	-0.275831866	IAA
## 75	-0.171191334	-0.490329128	IAA
## 76	-1.051443078	-0.924138217	IAA
## 77	0.531597276	0.648999766	ARF
## 78	-0.109188343	-0.208434682	ARF
## 79	0.317995304	-0.168331019	ARF
## 80	-0.594875791	-0.375545404	ARF
## 81	0.004808883	0.018108365	ARF
## 82	0.324427333	0.078558501	ARF
## 83	-0.173718424	0.362553950	ARF
## 84	0.857615531	-2.169356529	ARF
## 85	-0.146999936	0.007800111	ARF
## 86	0.092837656	-0.001694144	ARF
## 87	0.275671087	0.014778960	ARF
## 88	0.204583749	-0.247017881	ARF
## 89	0.749279943	0.370701122	IAA
## 90	-0.025807891	-0.090310742	ARF
## 91	-0.788482703	0.744876265	ARF
## 92	-0.264669941	0.300848696	ARF
## 93	0.352511287	0.291391237	ARF
## 94	-0.049101215	0.579170451	ARF
## 95	-0.833524403	0.330486790	IAA
## 96	0.559131560	-0.426860938	IAA
## 97	-0.022773811	0.273444399	IAA
## 98	0.152202391	0.406918506	IAA
## 99	-0.188391551	-0.132620836	IAA
## 100	-0.465096502	0.378210247	IAA
## 101	-0.417437561	0.354610124	IAA
## 102	0.239714138	-0.372805353	ARF
## 103	-0.433940124	-0.408984924	ARF
## 104	0.526862594	-0.556046556	AFB/TIR1
## 105	0.297046621	0.399860511	IAA
## 106	-0.401907770	0.093790074	ARF
## 107	-0.125386206	-0.371265410	ARF
## 108	-0.122534533	0.019319698	IAA
## 109	0.097778388	0.057486747	IAA
## 110	-0.453351398	-0.355495023	IAA
## 111	-0.442597992	0.413317924	ARF
## 112	-0.204069007	0.144534631	ARF
## 113	0.229584716	-0.081833467	ARF
## 114	0.219278723	-0.136465251	ARF
## 115	0.457407776	-0.717654679	ARF
## 116	0.101790408	0.189752408	ARF
## 117	0.505418632	-1.692904946	AFB/TIR1
## 118	-0.017733146	-0.152751925	ARF
## 119	-0.064799773	0.081077732	IAA

```
## 120  0.205522094 -3.429275402      ARF
## 121 -0.221521206 -0.123086248      ARF
## 122  0.113040052 -0.006051997      ARF
## 123  0.586856325 -1.585847922 AFB/TIR1
## 124 -0.048212877 -0.018020025      IAA
## 125 -2.096056941 -0.466548138      IAA
## 126  0.098701172  1.015184517      IAA
## 127  0.193201295  0.035834145      ARF
## 128  0.321861245 -0.590150626 AFB/TIR1
## 129 -0.405549799 -1.459565623      IAA
## 130  0.074216762  0.099160724      ARF
## 131 -2.457302433 -0.890649441      IAA
## 132 -0.258947266  0.101572514      IAA
## 133  3.502230742  4.637374786      IAA
```

```
build_PC3_4all <- ggplot_build(PC3_4all)$data
points_PC3_4all <- build_PC3_4all[[1]]
```

```
# co-ordinates of the ellipses
ell_points_PC3_4all <- car::dataEllipse(comb_pca_df$PC3,
                                         comb_pca_df$PC4,
                                         as.factor(comb_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point

ell_ARF_PC3_4all <- as.data.frame(ell_points_PC3_4all$ARF$`0.7`)
ell_IAA_PC3_4all <- as.data.frame(ell_points_PC3_4all$IAA$`0.7`)
ell_TIR_PC3_4all <- as.data.frame(ell_points_PC3_4all$`AFB/TIR1`$`0.7`)

# Find which points are outside (!) the ellipse, and add this to the data
```

```

dat_TIR_PC3_4all <- data.frame(
  points_PC3_4all[2:3],
  in_ell_TIR = as.logical(point.in.polygon(points_PC3_4all$x, points_PC3_4all$y, ell_TIR_PC3_4all$x, ell_TIR_PC3_4all$y))
)

dat_IAA_PC3_4all <- data.frame(
  points_PC3_4all[2:3],
  in_ell_IAA = as.logical(point.in.polygon(points_PC3_4all$x, points_PC3_4all$y, ell_IAA_PC3_4all$x, ell_IAA_PC3_4all$y))
)

dat_ARF_PC3_4all <- data.frame(
  points_PC3_4all[2:3],
  in_ell_ARF = as.logical(point.in.polygon(points_PC3_4all$x, points_PC3_4all$y, ell_ARF_PC3_4all$x, ell_ARF_PC3_4all$y))
)

# as.logical(point..) equals to TRUE indicated points are inside ellipses

# Combining data points for labeling
transcript_expr_logic_PC3_4all <- cbind(comb_pca_df, dat_TIR_PC3_4all, dat_IAA_PC3_4all, dat_ARF_PC3_4all)

expr_logic_PC3_4all <- transcript_expr_logic_PC3_4all %>% mutate(., in_ell = case_when(Family=="ARF" &
  Family=="IAA" & `in_ell_IAA` == TRUE ~ "TRUE",
  Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
  mutate(in_ell = coalesce(in_ell, "FALSE"))

expr_logic_PC3_4all[which(expr_logic_PC3_4all$in_ell == FALSE),]

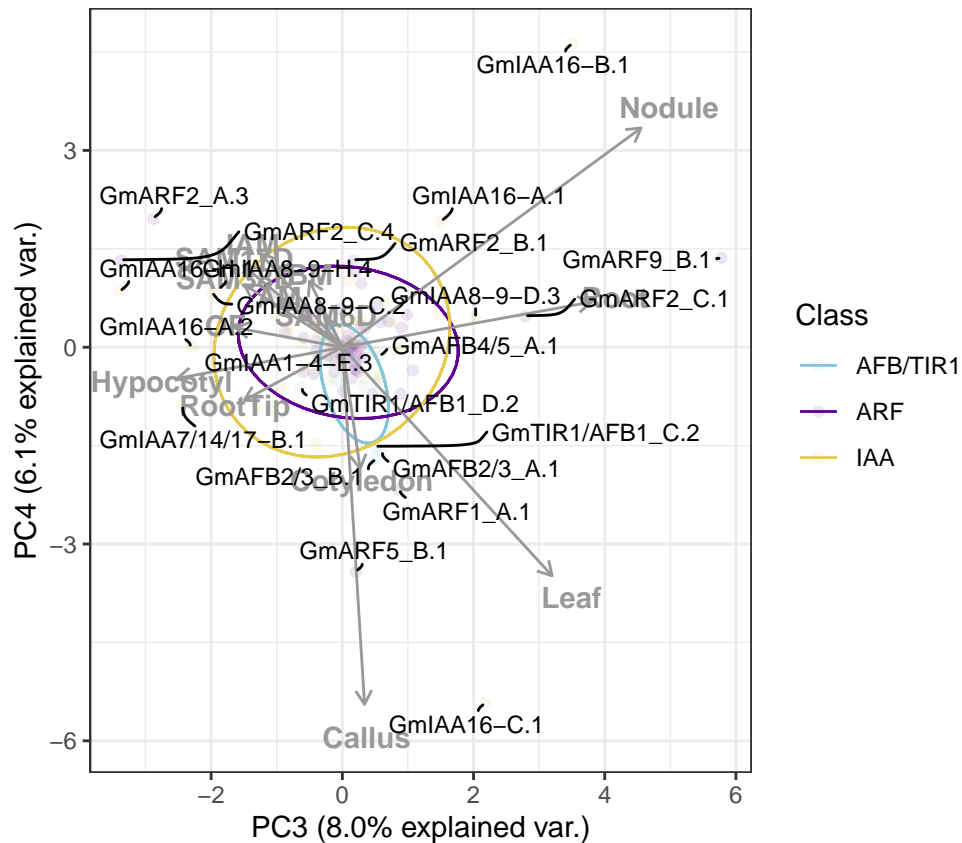
```

##	Transcript ID	heatmap_label	Family	Class	Clade		
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I		
## 12	Glyma.02G152800.2	Glyma.02G152800 GmTIR1/AFB1_D.2	AFB/TIR1	TIR1/AFB1	I		
## 15	Glyma.02G218100.4	Glyma.02G218100 GmIAA8-9-H.4	IAA	A	I		
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I		
## 23	Glyma.03G158700.1	Glyma.03G158700 GmIAA16-H.1	IAA	C	III		
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III		
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I		
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I		
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I		
## 37	Glyma.06G095400.1	Glyma.06G095400 GmAFB4/5_A.1	AFB/TIR1	AFB4/5	IV		
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I		
## 59	Glyma.08G273500.2	Glyma.08G273500 GmIAA8-9-C.2	IAA	A	I		
## 64	Glyma.10G021500.2	Glyma.10G021500 GmTIR1/AFB1_C.2	AFB/TIR1	TIR1/AFB1	I		
## 71	Glyma.10G162400.1	Glyma.10G162400 GmIAA16-A.1	IAA	C	III		
## 72	Glyma.10G162400.2	Glyma.10G162400 GmIAA16-A.2	IAA	C	III		
## 84	Glyma.12G164100.1	Glyma.12G164100 GmARF1_A.1	ARF	B	I		
## 117	Glyma.16G050500.1	Glyma.16G050500 GmAFB2/3_B.1	AFB/TIR1	AFB2/3	II		
## 120	Glyma.17G256500.1	Glyma.17G256500 GmARF5_B.1	ARF	A	II		
## 123	Glyma.19G100200.1	Glyma.19G100200 GmAFB2/3_A.1	AFB/TIR1	AFB2/3	II		
## 125	Glyma.19G161000.3	Glyma.19G161000 GmIAA1-4-E.3	IAA	B	II		
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III		
## 133	Glyma.20G225000.1	Glyma.20G225000 GmIAA16-B.1	IAA	C	III		
##	AM	OF	IAM	IBM	RootTip	Cotyledon	Hypocotyl
## 5	68.216252	51.350688	53.553189	96.27502	32.628226	78.301786	172.112449
## 12	22.069192	16.181094	21.004047	22.84902	12.164769	15.200032	20.202937
## 15	58.683503	91.015780	54.923525	50.03811	25.887821	8.588162	51.413131

## 20	14.963834	9.149436	10.014337	18.30301	6.361899	6.335987	4.159165
## 23	62.978354	188.282417	63.415971	50.27380	93.913232	11.797675	382.933284
## 25	123.752976	182.226043	143.672394	136.88284	68.294552	100.483878	215.724837
## 29	59.132821	34.776950	54.023677	104.01418	5.893899	15.036259	13.614824
## 32	23.736932	7.300337	13.387625	33.51611	4.197497	21.693739	20.950298
## 33	106.794815	41.848269	90.206750	89.14980	3.924654	25.580965	38.494199
## 37	16.624289	16.028732	15.517072	13.11757	3.675868	9.842971	17.361197
## 39	78.161251	91.333242	102.461124	95.84662	1.569549	29.653419	18.983798
## 59	60.235552	42.618679	62.597901	70.69411	15.151090	25.020319	166.602217
## 64	19.724328	16.170619	18.096732	18.50347	10.239295	14.506518	14.635121
## 71	60.897385	87.148645	58.213890	60.41856	39.516020	28.827155	119.352820
## 72	50.429036	62.648515	44.799041	49.04782	54.886130	14.549869	258.563372
## 84	29.907851	25.658848	21.848792	33.97809	12.158648	15.573356	22.344234
## 117	55.698420	35.061948	32.831068	58.58625	13.107811	27.899216	27.176499
## 120	7.845772	2.581028	3.311743	17.66248	0.302999	1.222648	1.046225
## 123	42.087683	43.031580	27.129734	42.68900	13.615904	12.545145	19.247151
## 125	4.478037	404.803343	44.077715	17.84655	72.730750	41.028328	177.214792
## 131	4.160975	265.136463	15.528614	13.54040	130.804062	41.664671	633.844120
## 133	83.900583	259.425512	113.461714	100.58810	53.025576	53.477803	102.248657
##	SAM6D	SAM17D	SAM38D	Callus	Leaf	Root	Nodule
## 5	101.538840	108.035003	57.568952	3.602672	72.181052	53.728970	22.341412
## 12	29.352679	33.917530	17.176745	17.838179	0.000000	0.000000	0.000000
## 15	79.581114	54.407495	42.568039	6.383128	0.000000	0.000000	0.000000
## 20	21.230577	13.709403	9.639602	2.603449	42.803542	126.598709	27.491901
## 23	20.163323	39.919056	73.535631	0.583159	6.250057	14.892682	2.016265
## 25	75.229555	108.926253	87.081914	65.420159	178.440543	98.055423	5.415049
## 29	40.036100	67.500984	33.565553	5.088175	22.636836	12.527492	10.758737
## 32	31.857638	31.063118	16.461606	4.368482	44.030052	25.338768	21.300657
## 33	75.216825	290.336670	100.526085	25.977418	0.000000	0.000000	0.000000
## 37	17.568729	22.163578	11.157726	12.838756	8.101368	7.946719	7.208545
## 39	38.298793	237.526834	76.153152	6.048278	3.674760	1.606524	0.000000
## 59	77.910628	90.255018	63.645599	6.462795	11.030950	6.618096	1.380951
## 64	12.243954	22.316362	15.443247	25.621399	26.106051	5.641934	3.205496
## 71	55.456636	78.452259	42.917887	2.247079	23.957562	65.117084	21.527670
## 72	52.919869	43.482386	62.182188	12.401001	7.425099	12.077243	0.000000
## 84	27.195518	37.963940	21.692550	51.460680	19.141297	21.871309	7.347936
## 117	82.133333	87.939197	38.297183	47.396469	32.138566	17.503017	8.945171
## 120	28.072725	24.136925	7.085973	69.182090	2.954026	4.491654	0.392173
## 123	60.919431	75.317808	31.290701	44.433814	27.410816	21.526502	7.313215
## 125	3.197441	2.314000	2.924909	0.843634	6.919914	6.338750	0.000000
## 131	1.661192	0.821472	1.444420	0.244966	0.972353	62.263900	0.495348
## 133	92.557209	144.677356	86.120903	10.711991	21.058177	104.233361	53.539749
##	PC1	PC2	PC3	PC4	in.ell_TIR	in.ell_IAA	
## 5	7.45961256	-0.60000202	2.0307657	0.445820518	FALSE	FALSE	
## 12	-0.10245780	0.78870349	-0.6333633	-0.606570751	FALSE	TRUE	
## 15	2.27672963	0.75044555	-1.8801490	0.882776840	FALSE	FALSE	
## 20	1.41830620	-2.03622924	5.7800703	1.362986140	FALSE	FALSE	
## 23	3.42533775	-3.40276087	-3.3884443	0.846071762	FALSE	FALSE	
## 25	13.39251285	-1.80641239	2.1698302	-5.419994579	FALSE	FALSE	
## 29	3.12051959	1.14297385	0.1671277	1.320222051	FALSE	TRUE	
## 32	1.33799443	0.01555935	2.7843694	0.468252980	FALSE	FALSE	
## 33	6.70264804	3.57934670	-3.3730284	1.324379018	FALSE	FALSE	
## 37	-0.48862762	0.39871849	0.5694045	-0.135589979	FALSE	TRUE	
## 39	5.27558405	2.30540719	-2.8806692	1.955575677	FALSE	FALSE	

```
## 59 3.83439442 0.79344930 -1.9984688 0.825223890 FALSE FALSE
## 64 0.07775833 0.52435782 0.5022450 -1.514606539 FALSE FALSE
## 71 4.48509992 -1.35142128 1.5097153 1.915691304 FALSE FALSE
## 72 2.77118641 -1.00810982 -2.2925865 -0.002501458 FALSE FALSE
## 84 1.52682137 0.80415967 0.8576155 -2.169356529 FALSE FALSE
## 117 3.95287191 1.62593170 0.5054186 -1.692904946 FALSE FALSE
## 120 -0.27983068 1.61342692 0.2055221 -3.429275402 FALSE FALSE
## 123 2.61523211 1.16403114 0.5868563 -1.585847922 FALSE FALSE
## 125 0.71948030 -4.32442630 -2.0960569 -0.466548138 FALSE FALSE
## 131 1.48447894 -7.91342992 -2.4573024 -0.890649441 FALSE FALSE
## 133 9.87734076 -2.30022562 3.5022307 4.637374786 FALSE FALSE
## in_ell_ARF in_ell
## 5 FALSE FALSE
## 12 TRUE FALSE
## 15 FALSE FALSE
## 20 FALSE FALSE
## 23 FALSE FALSE
## 25 FALSE FALSE
## 29 FALSE FALSE
## 32 FALSE FALSE
## 33 FALSE FALSE
## 37 TRUE FALSE
## 39 FALSE FALSE
## 59 FALSE FALSE
## 64 FALSE FALSE
## 71 FALSE FALSE
## 72 FALSE FALSE
## 84 FALSE FALSE
## 117 FALSE FALSE
## 120 FALSE FALSE
## 123 FALSE FALSE
## 125 FALSE FALSE
## 131 FALSE FALSE
## 133 FALSE FALSE
```

```
PC3_4all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
                                         xend=(PC3*8), yend=(PC4*8)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings$PC3*8.75), #add the tissue names to it manually
           y=(PCA_loadings$PC4*8.75),
           label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw()+
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  ggrepel::geom_text_repel(data = expr_logic_PC3_4all %>%
                           as_tibble(rownames = "name") %>%
                           filter(as.logical(in_ell == FALSE)),
                           aes(PC3, PC4, label=sub(".*\\|", "", heatmap_label)),
                           size=3, max.overlaps = 100, min.segment.length = 0,
                           segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()
```



```
#ggsave("20230927_PC3_4_allTissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20230927_PC3_4_allTissues.pdf", dpi = 1000, width = 10, height = 8)
```

Lets now repeat with the tau values

```
# combine data frames expr_logic with tau values
expr_logic_PC3_4 <- cbind(expr_logic_PC3_4all, tau_df$tau)

# Define the intervals for expr_logic2$tau
expr_logic_PC3_4$tau_interval <- cut(expr_logic_PC3_4$tau,
                                     breaks = c(-Inf, 0.5, 0.8, Inf),
                                     labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC3_4all +
  geom_segment(PCA_loadings, mapping=aes(x=0, y=0, # Change the size of arrows
                                         xend=(PC3*8.75), yend=(PC4*8.75)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings$PC3*8.75), #add the tissue names to it manually
          y=(PCA_loadings$PC4*8.75),
          label=PCA_loadings$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw() +
  geom_point(data = expr_logic_PC3_4 %>% as_tibble(rownames = "name"),
            aes(PC3, PC4, alpha = tau_interval, color = Family),
            size=3) +
  ggrepel::geom_text_repel(data = expr_logic_PC3_4 %>%
```



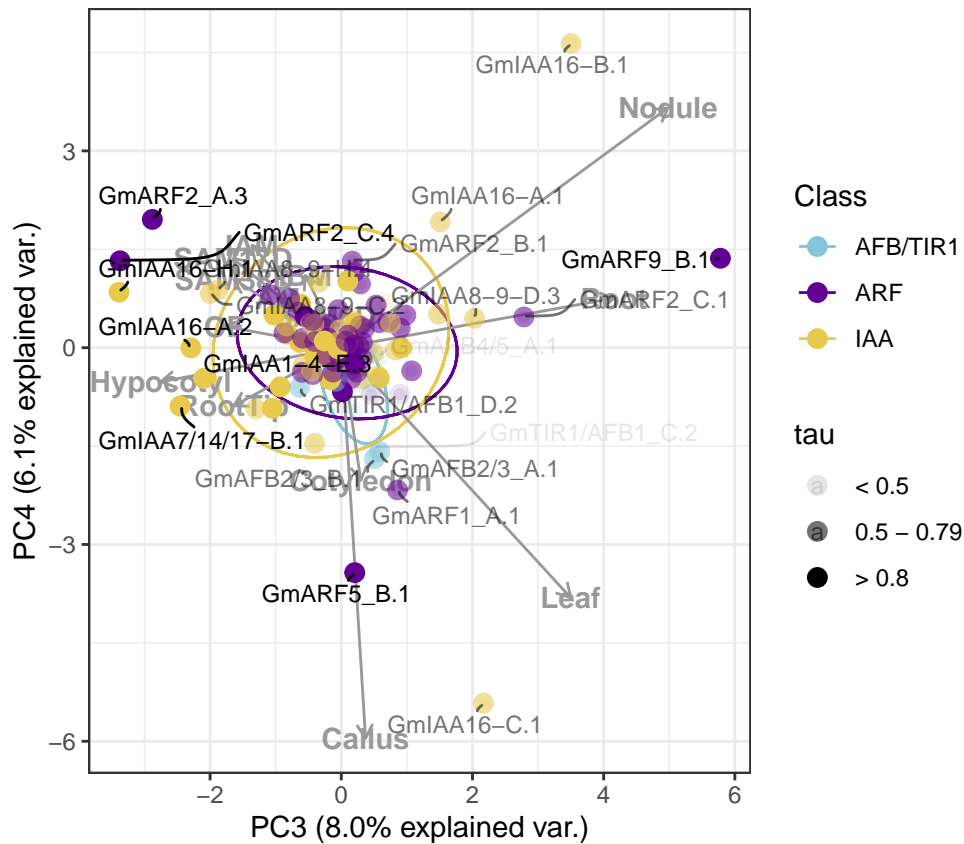
```

as_tibble(rownames = "name") %>%
  filter(as.logical(in_ell == FALSE)),
  aes(PC3, PC4, label=sub(".*\\|", "", heatmap_label),
       alpha = tau_interval),
  size=3, max.overlaps = 100, min.segment.length = 0,
  segment.curvature = -0.1) +

scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C"))+
labs(color = "Class", shape = "tau", alpha = "tau") +
theme_bw()

```

Warning: Using alpha for a discrete variable is not advised.



```

#ggsave("20240117_PC3_4_allTissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20240117_PC3_4_allTissues.pdf", dpi = 1000, width = 10, height = 8)

```

PCA with only tissues of interest

```

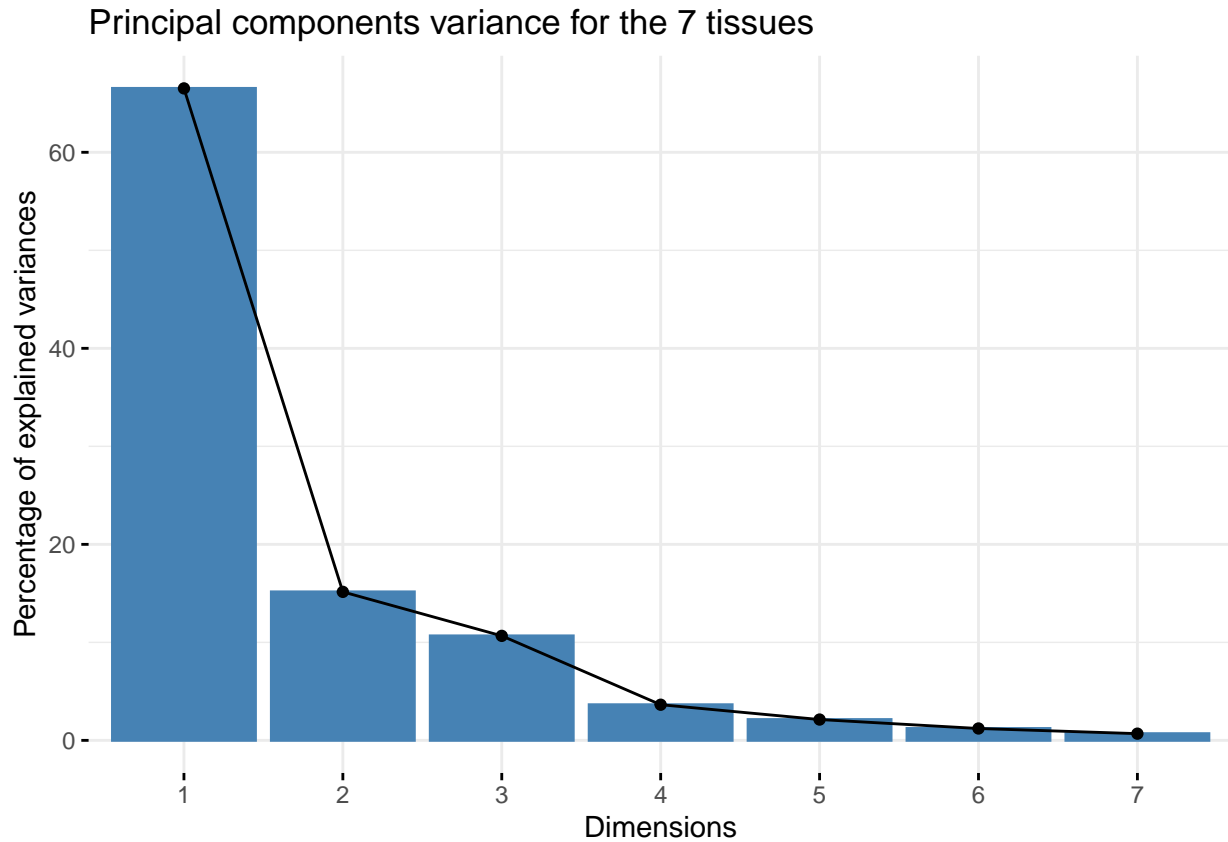
pca2 <- prcomp(pca_data[, -c(1:5, 7:10, 16, 18:19)],
               scale. = T, center = T)
summary(pca2)

```

Importance of components:

##	PC1	PC2	PC3	PC4	PC5	PC6	PC7
## Standard deviation	2.1579	1.0299	0.8640	0.50467	0.38576	0.29139	0.21916
## Proportion of Variance	0.6652	0.1515	0.1066	0.03638	0.02126	0.01213	0.00686

```
## Cumulative Proportion 0.6652 0.8167 0.9234 0.95975 0.98101 0.99314 1.00000
factoextra::fviz_eig(pca2, main = "Principal components variance for the 7 tissues")
```



```
# ggsave("20230927_PCscreePlot_7Tissues.png", dpi = 1000, width = 10, height = 8)
```

```
joint_pca_df <- cbind(pca_data, pca2$x[, 1:4])
head(joint_pca_df)
```

```
##      Transcript ID      heatmap_label Family Class Clade      AM
## 1 Glyma.01G002100.1 Glyma.01G002100|GmARF7/19_F.1  ARF    A    II 10.166688
## 2 Glyma.01G019400.1 Glyma.01G019400|GmIAA8-9-B.1  IAA    A    I  4.023220
## 3 Glyma.01G019400.2 Glyma.01G019400|GmIAA8-9-B.2  IAA    A    I  3.945653
## 4 Glyma.01G019400.3 Glyma.01G019400|GmIAA8-9-B.3  IAA    A    I 22.514233
## 5 Glyma.01G098000.3 Glyma.01G098000|GmIAA8-9-D.3  IAA    A    I 68.216252
## 6 Glyma.01G103500.1 Glyma.01G103500|GmARF9_A.1    ARF    B    I 23.365836
##      OF      IAM      IBM      RootTip Cotyledon Hypocotyl      SAM6D
## 1 11.093702  9.940766 11.778406  6.055558  6.293037 12.466244  8.948254
## 2  7.572444  0.000000  1.479181  3.679860  1.003818  8.866408  2.983280
## 3  0.000000  6.009114  7.650699  2.217478  2.098637 11.260883  9.399142
## 4 35.850834 20.735453 26.681081 16.122488  4.029864 88.717989 25.081359
## 5 51.350688 53.553189 96.275020 32.628226 78.301786 172.112449 101.538840
## 6  7.195187 14.135920 28.240802  4.918282  2.088168  2.353419 26.594544
##      SAM17D      SAM38D      Callus      Leaf      Root      Nodule      PC1
## 1  8.096833  5.745906  9.251965  4.882454  6.893253  1.926212 -1.0501429
## 2  7.873006  3.443632  6.039812 10.540945  5.246408  0.577947 -1.3485364
## 3  0.000000  2.491927  0.946290 11.828802 22.566715  1.890763 -1.2849928
## 4 26.514627 29.093086 17.719987  6.167892  0.000000  0.000000  0.1258125
```

```
## 5 108.035003 57.568952 3.602672 72.181052 53.728970 22.341412 6.2978744
## 6 33.652046 16.405528 1.091669 0.000000 0.000000 0.000000 -0.3140464
##      PC2      PC3      PC4
## 1 0.10515442 0.1357363 -0.16679473
## 2 0.03516704 0.3910160 0.02172556
## 3 -0.03259973 0.4674483 -0.06753696
## 4 -0.19841417 -0.6187942 0.41436903
## 5 -1.83724237 1.5164566 -1.17162708
## 6 0.76188043 -0.2984882 0.15505685
```

```
Family <- joint_pca_df$Family
```

```
# extract the loading so we can change the arrows more easily later while plotting
PCA_loadings2 <- data.frame(Variables=rownames(pca2$rotation), pca2$rotation)
```

```
# Plot
```

```
library(ggrepel)
```

```
PC1_2_7tissues <- ggbiplot::ggbiplot(pca2, obs.scale = 1, var.scale = .5, choices = c(1, 2),
                                     varname.adjust = 3, alpha = 0.1,
                                     groups = Family, var.axes = F, varname.size = 2,
                                     ellipse = TRUE, circle = FALSE, ellipse.prob = .70)
```

withdraw labels outside ellipses to label outliers (data points located outside the 70% ellipse interval)

```
# Extract components so we can select out
PC1_2_7tissues$data
```

```
##      xvar      yvar  groups
## 1 -1.05014286 0.10515442    ARF
## 2 -1.34853637 0.03516704    IAA
## 3 -1.28499278 -0.03259973    IAA
## 4 0.12581250 -0.19841417    IAA
## 5 6.29787441 -1.83724237    IAA
## 6 -0.31404635 0.76188043    ARF
## 7 0.50539968 -0.34300647    IAA
## 8 -1.12869949 0.37674443 AFB/TIR1
## 9 -0.68221161 -1.06192064    IAA
## 10 -1.32267878 -0.05614018    IAA
## 11 -1.33942303 -0.12354575    IAA
## 12 0.06210206 0.41382610 AFB/TIR1
## 13 -1.32261153 0.16146909 AFB/TIR1
## 14 0.40306612 0.43976369    IAA
## 15 2.07329850 1.01893547    IAA
## 16 -0.05373907 -0.19219653    ARF
## 17 0.81044975 1.11700892    ARF
## 18 -0.78784060 -0.12083822    ARF
## 19 -0.24435712 0.14788376    ARF
## 20 -0.07853023 -0.30909424    ARF
## 21 1.29528965 0.98161007    ARF
## 22 -1.65539096 0.04406573    IAA
## 23 2.50142090 -2.70244068    IAA
## 24 0.70690012 -0.19818565 AFB/TIR1
## 25 9.79977762 -4.26352971    IAA
```

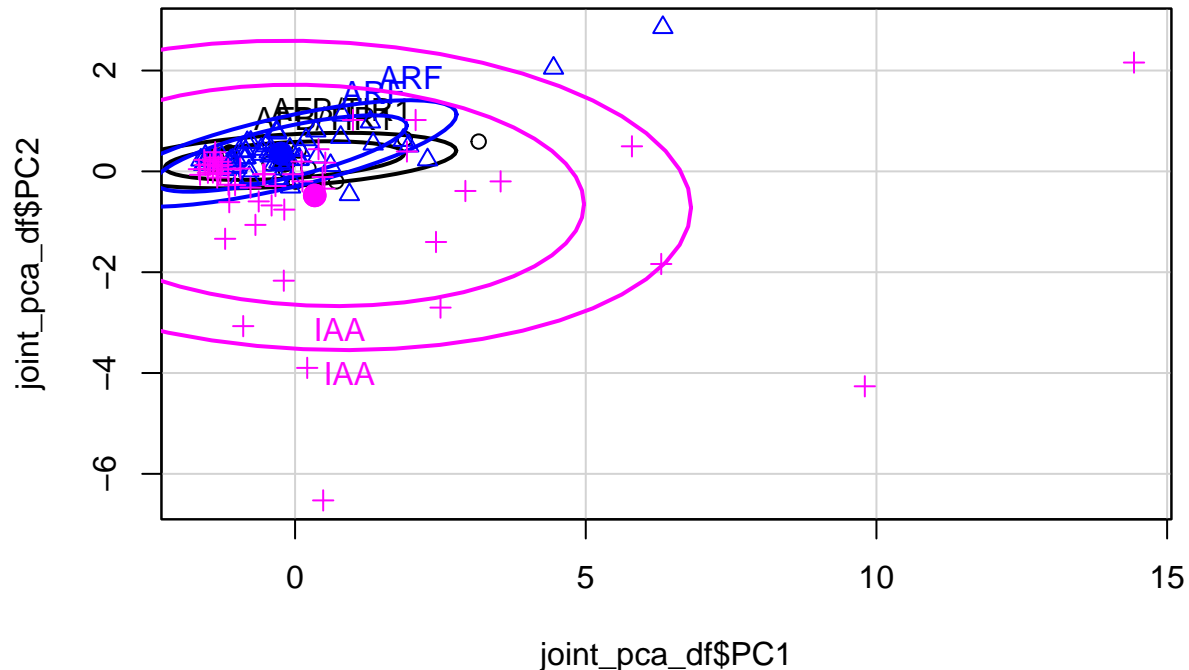
## 26	-0.99159124	-0.23057025	ARF
## 27	-0.03794509	-0.05712991	IAA
## 28	-0.35471399	0.37697535	AFB/TIR1
## 29	1.85976056	0.61742415	ARF
## 30	-0.31860769	0.79593666	ARF
## 31	-1.34000604	0.23346430	ARF
## 32	0.93454507	-0.45975448	ARF
## 33	6.32619943	2.85428889	ARF
## 34	-1.62029102	0.22580682	ARF
## 35	-0.52375557	-0.05145862	IAA
## 36	14.43049528	2.15899012	IAA
## 37	-0.43602126	0.15912727	AFB/TIR1
## 38	1.95569813	0.50580340	ARF
## 39	4.44562780	2.04813145	ARF
## 40	-1.20335584	-1.33745677	IAA
## 41	-0.10578553	0.24206119	ARF
## 42	0.35581526	0.80233487	ARF
## 43	-0.18778513	0.04913286	ARF
## 44	-0.08921436	0.44159097	ARF
## 45	-1.45080932	0.22207205	AFB/TIR1
## 46	-1.14906764	0.28259178	AFB/TIR1
## 47	-0.99845260	0.22602501	ARF
## 48	0.05595252	0.23517727	ARF
## 49	2.27842210	0.23014394	ARF
## 50	1.34307083	0.54381981	ARF
## 51	-1.55156504	0.16357844	ARF
## 52	0.51958352	0.17260396	IAA
## 53	-1.36883544	0.27331654	IAA
## 54	-1.38601667	0.12704175	AFB/TIR1
## 55	-0.47630116	0.55667830	ARF
## 56	-0.84258221	0.27501312	ARF
## 57	-1.56272647	0.22850724	IAA
## 58	-0.94247516	0.05578545	IAA
## 59	3.53507221	-0.19876183	IAA
## 60	-1.07587306	0.22563162	ARF
## 61	-1.46220344	0.13643709	IAA
## 62	0.47369476	-0.11275080	IAA
## 63	-0.18575199	-0.75842052	IAA
## 64	0.01141647	-0.20362431	AFB/TIR1
## 65	-1.42021154	0.20580527	AFB/TIR1
## 66	-0.17447944	0.17662980	AFB/TIR1
## 67	-1.14563503	-0.26383478	IAA
## 68	-1.49981620	-0.02775860	IAA
## 69	-1.54348979	0.12874603	IAA
## 70	-1.39507835	0.16168347	ARF
## 71	2.92945855	-0.38908091	IAA
## 72	2.42394340	-1.40080513	IAA
## 73	-1.15099418	0.04204640	IAA
## 74	-0.62584455	-0.59554073	IAA
## 75	-0.40410313	-0.67555222	IAA
## 76	0.20859279	-3.89840357	IAA
## 77	-0.84786884	0.39304286	ARF
## 78	-0.28107091	-0.14129573	ARF
## 79	-0.79489435	0.01134430	ARF

## 80	-0.42946116	0.30503427	ARF
## 81	-0.86565058	0.26459088	ARF
## 82	-1.57421557	0.13997225	ARF
## 83	-0.58850376	0.33976118	ARF
## 84	0.60982840	0.11755926	ARF
## 85	-0.95270455	0.41027661	ARF
## 86	-1.18400893	0.03642123	ARF
## 87	-1.35845402	0.18832650	ARF
## 88	-1.32330825	0.07623100	ARF
## 89	-1.32211008	0.09616199	IAA
## 90	-1.33857165	0.22631146	ARF
## 91	0.18759733	0.63062446	ARF
## 92	0.37773397	0.28877357	ARF
## 93	-0.28033129	0.18223673	ARF
## 94	-0.49772951	0.50936896	ARF
## 95	0.99582518	1.01924752	IAA
## 96	-0.55223302	-0.06070833	IAA
## 97	-1.37434561	0.37953170	IAA
## 98	-1.28152736	0.19095485	IAA
## 99	-1.13207935	-0.61127542	IAA
## 100	-1.25672736	0.12918250	IAA
## 101	-1.41443268	-0.02383471	IAA
## 102	-1.00237591	0.08053528	ARF
## 103	0.07271378	0.24466668	ARF
## 104	-0.16199629	0.02332871	AFB/TIR1
## 105	0.11133693	0.19479841	IAA
## 106	0.08787857	0.41799993	ARF
## 107	-0.76288272	0.57895723	ARF
## 108	-1.03198092	-0.27146457	IAA
## 109	-1.39231781	0.23943824	IAA
## 110	1.92371297	0.39438069	IAA
## 111	-0.58154082	0.43897284	ARF
## 112	-1.35296511	0.21805555	ARF
## 113	-1.39891628	0.16435444	ARF
## 114	-1.47279337	0.21819497	ARF
## 115	-0.37176747	0.12804991	ARF
## 116	0.78469153	0.67534859	ARF
## 117	3.15794243	0.59153383	AFB/TIR1
## 118	-1.10801710	0.21113642	ARF
## 119	-1.40473569	0.19705245	IAA
## 120	-0.82324378	0.57202079	ARF
## 121	-0.44612445	0.38556844	ARF
## 122	-1.54287005	0.30454921	ARF
## 123	1.88822437	0.64531129	AFB/TIR1
## 124	-1.63443659	-0.06758392	IAA
## 125	-0.19442276	-2.16841690	IAA
## 126	-0.89218238	-3.07007360	IAA
## 127	-1.14683163	0.16640283	ARF
## 128	0.22854712	0.06134928	AFB/TIR1
## 129	-0.33759214	-0.28989506	IAA
## 130	-1.26104706	0.28590134	ARF
## 131	0.48333718	-6.52671369	IAA
## 132	-0.76649931	-0.32242524	IAA
## 133	5.79424460	0.49697177	IAA

```
build_PC1_2_7tissues <- ggplot_build(PC1_2_7tissues)$data
points_PC1_2_7tissues <- build_PC1_2_7tissues[[1]]
```

```
# co-ordinates of the ellipses
```

```
ell_points_PC1_2_7tissues <- car::dataEllipse(joint_pca_df$PC1,
  joint_pca_df$PC2,
  as.factor(joint_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point
```

```
ell_ARF_PC1_2_7tissues <- as.data.frame(ell_points_PC1_2_7tissues$ARF$`0.7`)
```

```
ell_IAA_PC1_2_7tissues <- as.data.frame(ell_points_PC1_2_7tissues$IAA$`0.7`)
```

```
ell_TIR_PC1_2_7tissues <- as.data.frame(ell_points_PC1_2_7tissues$AFB/TIR1$`0.7`)
```

```
# Find which points are outside (!) the ellipse, and add this to the data
```

```
library(sp)
```

```
dat_TIR_PC1_2_7tissues <- data.frame(
  points_PC1_2_7tissues[2:3],
  in.ell_TIR = as.logical(point.in.polygon(points_PC1_2_7tissues$x, points_PC1_2_7tissues$y, ell_TIR_PC1_2_7tissues))
)
```

```
dat_IAA_PC1_2_7tissues <- data.frame(
  points_PC1_2_7tissues[2:3],
  in.ell_IAA = as.logical(point.in.polygon(points_PC1_2_7tissues$x, points_PC1_2_7tissues$y, ell_IAA_PC1_2_7tissues))
)
```

```
dat_ARF_PC1_2_7tissues <- data.frame(
  points_PC1_2_7tissues[2:3],
  in.ell_ARF = as.logical(point.in.polygon(points_PC1_2_7tissues$x, points_PC1_2_7tissues$y, ell_ARF_PC1_2_7tissues))
)
```

```
# as.logical(point..) equals to TRUE indicated points are inside ellipses
```

```
# Combining data points for labeling
```

```
#Combine data points coordinates with PCs and expression data containing gene names will help us to have
```

```
transcript_expr_logical_PC1_2_7tissues <- cbind(joint_pca_df, dat_TIR_PC1_2_7tissues, dat_IAA_PC1_2_7tissues)
```

```
expr_logical_PC1_2_7tissues <- transcript_expr_logical_PC1_2_7tissues %>% mutate(., in_ell = case_when(
  Family=="IAA" & `in_ell_IAA` == TRUE ~ "TRUE",
  Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
  mutate(in_ell = coalesce(in_ell, "FALSE"))
```

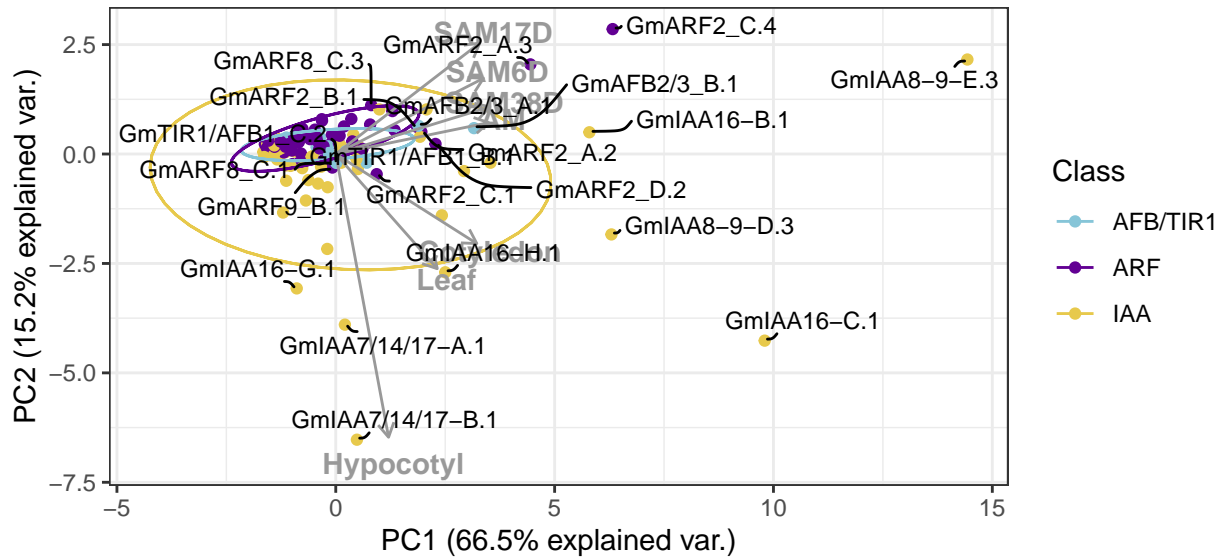
```
expr_logical_PC1_2_7tissues[which(expr_logical_PC1_2_7tissues$in_ell == FALSE),]
```

##	Transcript ID	heatmap_label	Family	Class	Clade	
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I	
## 16	Glyma.02G239600.1	Glyma.02G239600 GmARF8_C.1	ARF	A	II	
## 17	Glyma.02G239600.3	Glyma.02G239600 GmARF8_C.3	ARF	A	II	
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I	
## 23	Glyma.03G158700.1	Glyma.03G158700 GmIAA16-H.1	IAA	C	III	
## 24	Glyma.03G209400.1	Glyma.03G209400 GmTIR1/AFB1_B.1	AFB/TIR1	TIR1/AFB1	I	
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III	
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I	
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I	
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I	
## 36	Glyma.06G091700.3	Glyma.06G091700 GmIAA8-9-E.3	IAA	A	I	
## 38	Glyma.06G164900.2	Glyma.06G164900 GmARF2_A.2	ARF	B	I	
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I	
## 49	Glyma.08G008100.2	Glyma.08G008100 GmARF2_D.2	ARF	B	I	
## 64	Glyma.10G021500.2	Glyma.10G021500 GmTIR1/AFB1_C.2	AFB/TIR1	TIR1/AFB1	I	
## 76	Glyma.10G180100.1	Glyma.10G180100 GmIAA7/14/17-A.1	IAA	C	III	
## 117	Glyma.16G050500.1	Glyma.16G050500 GmAFB2/3_B.1	AFB/TIR1	AFB2/3	II	
## 123	Glyma.19G100200.1	Glyma.19G100200 GmAFB2/3_A.1	AFB/TIR1	AFB2/3	II	
## 126	Glyma.19G161100.1	Glyma.19G161100 GmIAA16-G.1	IAA	C	III	
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III	
## 133	Glyma.20G225000.1	Glyma.20G225000 GmIAA16-B.1	IAA	C	III	
##	AM	OF	IAM	IBM	RootTip	Cotyledon
## 5	68.216252	51.350688	53.553189	96.275020	32.628226	78.301786
## 16	18.062407	8.153791	22.217284	25.144304	6.595787	17.054927
## 17	37.168779	5.439277	41.924483	51.089719	6.308382	0.000000
## 20	14.963834	9.149436	10.014337	18.303006	6.361899	6.335987
## 23	62.978354	188.282417	63.415971	50.273803	93.913232	11.797675
## 24	25.967032	26.377921	14.286116	20.719333	9.445604	22.549691
## 25	123.752976	182.226043	143.672394	136.882838	68.294552	100.483878
## 29	59.132821	34.776950	54.023677	104.014185	5.893899	15.036259
## 32	23.736932	7.300337	13.387625	33.516112	4.197497	21.693739
## 33	106.794815	41.848269	90.206750	89.149798	3.924654	25.580965
## 36	120.096829	73.349658	80.928913	112.799969	14.116324	123.198735
## 38	62.736767	31.362845	56.712867	104.850125	1.621706	14.695636
## 39	78.161251	91.333242	102.461124	95.846621	1.569549	29.653419
## 49	41.952212	22.684099	48.464588	54.600720	2.166574	14.976262
## 64	19.724328	16.170619	18.096732	18.503468	10.239295	14.506518
## 76	7.941917	536.895072	6.035997	7.033665	214.246604	30.729826
## 117	55.698420	35.061948	32.831068	58.586254	13.107811	27.899216

##	123	42.087683	43.031580	27.129734	42.689001	13.615904	12.545145	
##	126	6.156592	108.404702	30.826540	8.523381	43.829775	4.519645	
##	131	4.160975	265.136463	15.528614	13.540402	130.804062	41.664671	
##	133	83.900583	259.425512	113.461714	100.588102	53.025576	53.477803	
##		Hypocotyl	SAM6D	SAM17D	SAM38D	Callus	Leaf	Root
##	5	172.112449	101.538840	108.035003	57.568952	3.602672	72.181052	53.728970
##	16	16.326781	22.517324	12.659333	10.229414	2.716798	21.320212	7.310538
##	17	23.986700	52.872850	68.490459	26.447317	6.139986	0.000000	3.626608
##	20	4.159165	21.230577	13.709403	9.639602	2.603449	42.803542	126.598709
##	23	382.933284	20.163323	39.919056	73.535631	0.583159	6.250057	14.892682
##	24	20.867913	28.476275	25.128377	20.548182	11.542469	26.686917	5.465411
##	25	215.724837	75.229555	108.926253	87.081914	65.420159	178.440543	98.055423
##	29	13.614824	40.036100	67.500984	33.565553	5.088175	22.636836	12.527492
##	32	20.950298	31.857638	31.063118	16.461606	4.368482	44.030052	25.338768
##	33	38.494199	75.216825	290.336670	100.526085	25.977418	0.000000	0.000000
##	36	118.328990	229.139072	393.546142	181.391747	71.756006	45.788590	20.983248
##	38	13.618860	48.102339	51.456817	31.907033	8.665198	27.396630	14.501406
##	39	18.983798	38.298793	237.526834	76.153152	6.048278	3.674760	1.606524
##	49	26.934453	45.899996	104.119336	30.814124	3.509392	48.206033	26.688307
##	64	14.635121	12.243954	22.316362	15.443247	25.621399	26.106051	5.641934
##	76	350.297556	2.600212	4.426392	5.209396	2.744771	20.850299	94.160633
##	117	27.176499	82.133333	87.939197	38.297183	47.396469	32.138566	17.503017
##	123	19.247151	60.919431	75.317808	31.290701	44.433814	27.410816	21.526502
##	126	334.804651	0.817625	0.938549	2.175286	0.045598	1.253888	31.879730
##	131	633.844120	1.661192	0.821472	1.444420	0.244966	0.972353	62.263900
##	133	102.248657	92.557209	144.677356	86.120903	10.711991	21.058177	104.233361
##		Nodule	PC1	PC2	PC3	PC4	in_ell_TIR	
##	5	22.341412	6.29787441	-1.8372424	1.5164566	-1.17162708	FALSE	
##	16	0.657995	-0.05373907	-0.1921965	0.8084196	-0.30166116	FALSE	
##	17	0.000000	0.81044975	1.1170089	-0.7546656	0.42743184	FALSE	
##	20	27.491901	-0.07853023	-0.3090942	1.7082831	0.36298038	FALSE	
##	23	2.016265	2.50142090	-2.7024407	-2.7850869	1.95425693	FALSE	
##	24	1.007309	0.70690012	-0.1981856	0.8944149	-0.24184281	FALSE	
##	25	5.415049	9.79977762	-4.2635297	5.7326293	1.19531413	FALSE	
##	29	10.758737	1.85976056	0.6174242	0.3612860	0.76750320	FALSE	
##	32	21.300657	0.93454507	-0.4597545	1.6269670	-0.09431274	FALSE	
##	33	0.000000	6.32619943	2.8542889	-2.3102058	1.29013983	FALSE	
##	36	22.794609	14.43049528	2.1589901	-1.5310904	-2.57494825	FALSE	
##	38	9.673077	1.95569813	0.5058034	0.6227889	0.85277438	FALSE	
##	39	0.000000	4.44562780	2.0481314	-1.4582800	0.62005161	FALSE	
##	49	8.836310	2.27842210	0.2301439	1.2286204	0.63434759	FALSE	
##	64	3.205496	0.01141647	-0.2036243	0.9369297	0.04368445	FALSE	
##	76	5.269519	0.20859279	-3.8984036	-0.9846724	-0.43448129	FALSE	
##	117	8.945171	3.15794243	0.5915338	0.5657640	-0.06986253	FALSE	
##	123	7.313215	1.88822437	0.6453113	0.4537617	0.33399282	FALSE	
##	126	15.878389	-0.89218238	-3.0700736	-1.9027957	0.34702454	FALSE	
##	131	0.495348	0.48333718	-6.5267137	-3.4003668	-0.81527683	FALSE	
##	133	53.539749	5.79424460	0.4969718	-0.8400912	-0.23361517	FALSE	
##		in_ell_IAA	in_ell_ARF	in_ell				
##	5	FALSE	FALSE	FALSE				
##	16	TRUE	FALSE	FALSE				
##	17	TRUE	FALSE	FALSE				
##	20	TRUE	FALSE	FALSE				
##	23	FALSE	FALSE	FALSE				


```
## 24      TRUE      FALSE FALSE
## 25     FALSE     FALSE FALSE
## 29      TRUE     FALSE FALSE
## 32      TRUE     FALSE FALSE
## 33     FALSE     FALSE FALSE
## 36     FALSE     FALSE FALSE
## 38      TRUE     FALSE FALSE
## 39     FALSE     FALSE FALSE
## 49      TRUE     FALSE FALSE
## 64      TRUE     FALSE FALSE
## 76     FALSE     FALSE FALSE
## 117     TRUE     FALSE FALSE
## 123     TRUE     FALSE FALSE
## 126     FALSE     FALSE FALSE
## 131     FALSE     FALSE FALSE
## 133     FALSE     FALSE FALSE
```

```
ggbiplot::ggbiplot(pca2, obs.scale = 1, var.scale = .5, choices = c(1, 2),
                    varname.adjust = 3,
                    groups = Family, var.axes = F, varname.size = 2,
                    ellipse = TRUE, circle = FALSE, ellipse.prob = .70) +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
                                          xend=(PC1*8), yend=(PC2*8)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC1*8.75), #add the tissue names to it manually
              y=(PCA_loadings2$PC2*8.75),
              label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw()+
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
ggrepel::geom_text_repel(data = expr_logical_PC1_2_7tissues %>%
  as_tibble(rownames = "name") %>%
  filter(as.logical(in_ell == FALSE)),
  aes(PC1, PC2, label=sub(".*\\|", "", heatmap_label)),
  size=3, max.overlaps = 100, min.segment.length = 0,
  segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()
```



```
#ggsave("20230927_PC1_2_7Tissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20230927_PC1_2_7Tissues.pdf", dpi = 1000, width = 10, height = 8)
```

Lets now repeat with the tau values for 7 tissues kept

```
# combine data frames expr_logic with tau values
expr_logic_PC1_2_7tissues <- cbind(expr_logical_PC1_2_7tissues, tau_df$tau)

# Define the intervals for expr_logic2$tau
expr_logic_PC1_2_7tissues$tau_interval <- cut(expr_logic_PC1_2_7tissues$tau,
                                              breaks = c(-Inf, 0.5, 0.8, Inf),
                                              labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC1_2_7tissues +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
                                          xend=(PC1*8.75), yend=(PC2*8.75)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC1*8.75), #add the tissue names to it manually
           y=(PCA_loadings2$PC2*8.75),
           label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1)) +

  geom_point(data = expr_logic_PC1_2_7tissues %>% as_tibble(rownames = "name"),
            aes(PC1, PC2, alpha = tau_interval, color = Family),
            size=3) +
  ggrepel::geom_text_repel(data = expr_logic_PC1_2_7tissues %>%
                           as_tibble(rownames = "name") %>%
                           filter(as.logical(in_ell == FALSE)),
                           aes(PC1, PC2, label=sub(".*\\|", "", heatmap_label),
                               alpha=tau_interval),
                           size=3, max.overlaps = 100, min.segment.length = 0,
                           segment.curvature = -0.1) +

  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  labs(color = "Class") +
  labs(color = "Class", alpha = "tau") +
  ylim(-7.5, 3) +
```

```
## Warning: Using alpha for a discrete variable is not advised.
```

## 17	1.11700892	-0.754665609	ARF
## 18	-0.12083822	-0.198053390	ARF
## 19	0.14788376	0.203702144	ARF
## 20	-0.30909424	1.708283097	ARF
## 21	0.98161007	-0.249254037	ARF
## 22	0.04406573	0.048238640	IAA
## 23	-2.70244068	-2.785086871	IAA
## 24	-0.19818565	0.894414856	AFB/TIR1
## 25	-4.26352971	5.732629313	IAA
## 26	-0.23057025	0.800391367	ARF
## 27	-0.05712991	0.024957986	IAA
## 28	0.37697535	-0.074310120	AFB/TIR1
## 29	0.61742415	0.361286016	ARF
## 30	0.79593666	-0.433613208	ARF
## 31	0.23346430	0.101766304	ARF
## 32	-0.45975448	1.626967022	ARF
## 33	2.85428889	-2.310205824	ARF
## 34	0.22580682	0.024556471	ARF
## 35	-0.05145862	0.264364967	IAA
## 36	2.15899012	-1.531090403	IAA
## 37	0.15912727	0.141906335	AFB/TIR1
## 38	0.50580340	0.622788860	ARF
## 39	2.04813145	-1.458279986	ARF
## 40	-1.33745677	-0.750021960	IAA
## 41	0.24206119	0.377262769	ARF
## 42	0.80233487	-0.439075999	ARF
## 43	0.04913286	0.051104600	ARF
## 44	0.44159097	0.071044809	ARF
## 45	0.22207205	-0.047741305	AFB/TIR1
## 46	0.28259178	-0.031576561	AFB/TIR1
## 47	0.22602501	0.161153854	ARF
## 48	0.23517727	0.291896516	ARF
## 49	0.23014394	1.228620351	ARF
## 50	0.54381981	-0.346616692	ARF
## 51	0.16357844	0.061513280	ARF
## 52	0.17260396	0.134356933	IAA
## 53	0.27331654	-0.185967868	IAA
## 54	0.12704175	0.051140263	AFB/TIR1
## 55	0.55667830	-0.286010626	ARF
## 56	0.27501312	0.222705920	ARF
## 57	0.22850724	-0.007314077	IAA
## 58	0.05578545	0.335059268	IAA
## 59	-0.19876183	-1.424143572	IAA
## 60	0.22563162	0.022943597	ARF
## 61	0.13643709	-0.009929315	IAA
## 62	-0.11275080	-1.179610247	IAA
## 63	-0.75842052	0.443021793	IAA
## 64	-0.20362431	0.936929695	AFB/TIR1
## 65	0.20580527	0.214269342	AFB/TIR1
## 66	0.17662980	0.040059182	AFB/TIR1
## 67	-0.26383478	-0.027750940	IAA
## 68	-0.02775860	0.504784255	IAA
## 69	0.12874603	0.067089129	IAA
## 70	0.16168347	0.144718515	ARF

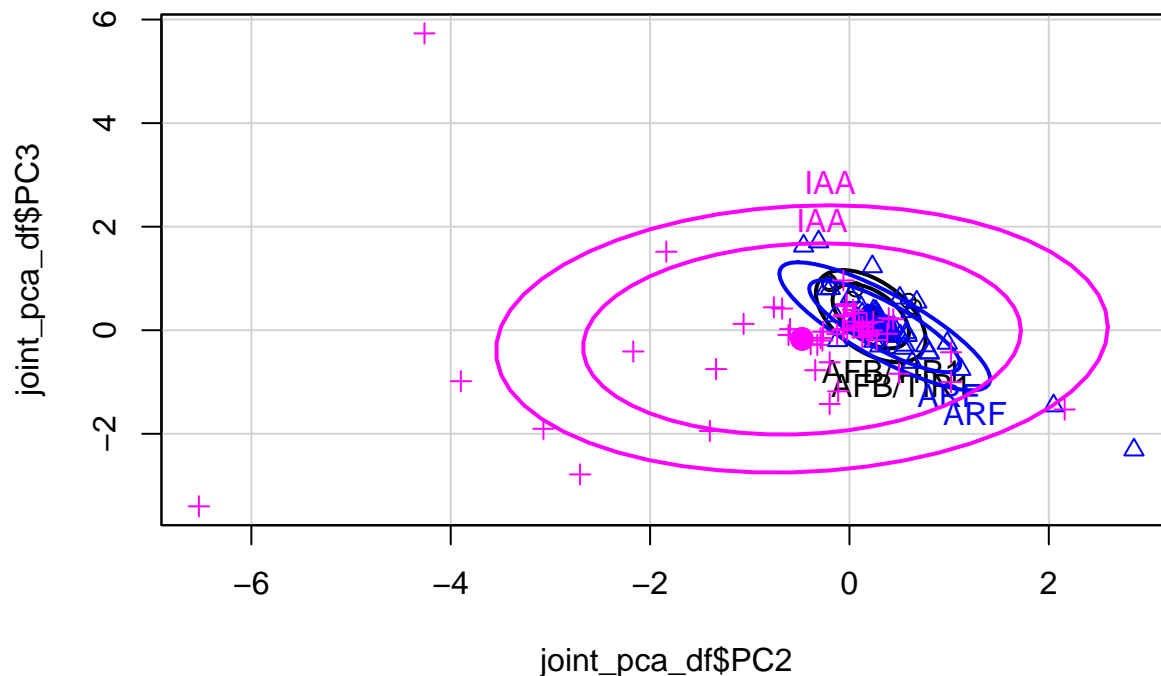
## 71	-0.38908091	-0.270753178	IAA
## 72	-1.40080513	-1.946238594	IAA
## 73	0.04204640	0.165611508	IAA
## 74	-0.59554073	0.021079016	IAA
## 75	-0.67555222	0.418826607	IAA
## 76	-3.89840357	-0.984672374	IAA
## 77	0.39304286	-0.039468673	ARF
## 78	-0.14129573	0.280245925	ARF
## 79	0.01134430	0.650288391	ARF
## 80	0.30503427	-0.302086741	ARF
## 81	0.26459088	0.360448967	ARF
## 82	0.13997225	0.032561259	ARF
## 83	0.33976118	0.035765053	ARF
## 84	0.11755926	0.440653982	ARF
## 85	0.41027661	-0.100032400	ARF
## 86	0.03642123	0.131115667	ARF
## 87	0.18832650	0.112977921	ARF
## 88	0.07623100	0.060351273	ARF
## 89	0.09616199	0.093041365	IAA
## 90	0.22631146	0.075836951	ARF
## 91	0.63062446	-0.618197885	ARF
## 92	0.28877357	0.291349997	ARF
## 93	0.18223673	0.286799671	ARF
## 94	0.50936896	-0.107640571	ARF
## 95	1.01924752	-0.425150473	IAA
## 96	-0.06070833	0.962239600	IAA
## 97	0.37953170	-0.067668928	IAA
## 98	0.19095485	-0.106174638	IAA
## 99	-0.61127542	-0.090989128	IAA
## 100	0.12918250	-0.136450892	IAA
## 101	-0.02383471	-0.002678882	IAA
## 102	0.08053528	0.228044041	ARF
## 103	0.24466668	-0.211148378	ARF
## 104	0.02332871	0.603415593	AFB/TIR1
## 105	0.19479841	-0.088816512	IAA
## 106	0.41799993	0.024403946	ARF
## 107	0.57895723	-0.099986924	ARF
## 108	-0.27146457	-0.198514723	IAA
## 109	0.23943824	0.163541169	IAA
## 110	0.39438069	0.232333327	IAA
## 111	0.43897284	-0.116129991	ARF
## 112	0.21805555	-0.051162129	ARF
## 113	0.16435444	0.183674413	ARF
## 114	0.21819497	0.118174410	ARF
## 115	0.12804991	0.315839653	ARF
## 116	0.67534859	0.544740295	ARF
## 117	0.59153383	0.565764005	AFB/TIR1
## 118	0.21113642	0.022927865	ARF
## 119	0.19705245	-0.047540345	IAA
## 120	0.57202079	-0.045815482	ARF
## 121	0.38556844	-0.100377639	ARF
## 122	0.30454921	-0.025405134	ARF
## 123	0.64531129	0.453761721	AFB/TIR1
## 124	-0.06758392	0.007290830	IAA

```
## 125 -2.16841690 -0.410351711 IAA
## 126 -3.07007360 -1.902795711 IAA
## 127 0.16640283 0.079576472 ARF
## 128 0.06134928 0.790073247 AFB/TIR1
## 129 -0.28989506 -0.151673820 IAA
## 130 0.28590134 0.055831155 ARF
## 131 -6.52671369 -3.400366836 IAA
## 132 -0.32242524 -0.282003528 IAA
## 133 0.49697177 -0.840091171 IAA
```

```
build_PC2_3_7tissues <- ggplot_build(PC2_3_7tissues)$data
points_PC2_3_7tissues <- build_PC2_3_7tissues[[1]]
```

```
# co-ordinates of the ellipses
```

```
ell_points_PC2_3_7tissues <- car::dataEllipse(joint_pca_df$PC2,
  joint_pca_df$PC3,
  as.factor(joint_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point
```

```
ell_ARF_PC2_3_7tissues <- as.data.frame(ell_points_PC2_3_7tissues$ARF$`0.7`)
```

```
ell_IAA_PC2_3_7tissues <- as.data.frame(ell_points_PC2_3_7tissues$IAA$`0.7`)
```

```
ell_TIR_PC2_3_7tissues <- as.data.frame(ell_points_PC2_3_7tissues$`AFB/TIR1`$`0.7`)
```

```
# Find which points are outside (!) the ellipse, and add this to the data
```

```
dat_TIR_PC2_3_7tissues <- data.frame(
  points_PC2_3_7tissues[2:3],
  in.ell_TIR = as.logical(point.in.polygon(points_PC2_3_7tissues$x, points_PC2_3_7tissues$y, ell_TIR_PC2_3_7tissues))
)
```

```

dat_IIA_PC2_3_7tissues <- data.frame(
  points_PC2_3_7tissues[2:3],
  in_ell_IIA = as.logical(point.in.polygon(points_PC2_3_7tissues$x, points_PC2_3_7tissues$y, ell_IIA_PC2_3_7tissues))
)

dat_ARF_PC2_3_7tissues <- data.frame(
  points_PC2_3_7tissues[2:3],
  in_ell_ARF = as.logical(point.in.polygon(points_PC2_3_7tissues$x, points_PC2_3_7tissues$y, ell_ARF_PC2_3_7tissues))
)
# as.logical(point..) equals to TRUE indicated points are inside ellipses

# Combining data points for labeling
transcript_expr_log_PC2_3_7tissues <- cbind(joint_pca_df, dat_TIR_PC2_3_7tissues, dat_IIA_PC2_3_7tissues)

expr_logical_PC2_3_7tissues <- transcript_expr_log_PC2_3_7tissues %>% mutate(., in_ell = case_when(Family=="IAA" & `in_ell_IIA` == TRUE ~ "TRUE",
  Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
  mutate(in_ell = coalesce(in_ell, "FALSE"))

expr_logical_PC2_3_7tissues[which(expr_logical_PC2_3_7tissues$in_ell == FALSE),]

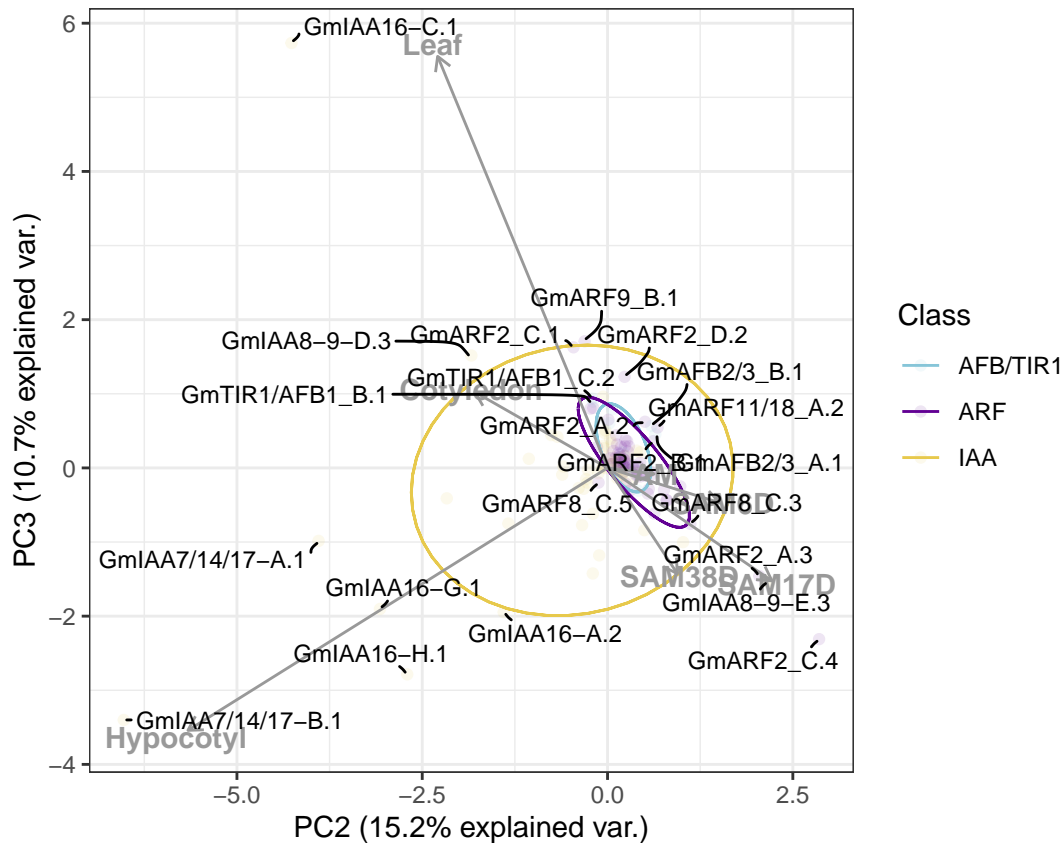
```

##	Transcript ID	heatmap_label	Family	Class	Clade	
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I	
## 17	Glyma.02G239600.3	Glyma.02G239600 GmARF8_C.3	ARF	A	II	
## 18	Glyma.02G239600.5	Glyma.02G239600 GmARF8_C.5	ARF	A	II	
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I	
## 23	Glyma.03G158700.1	Glyma.03G158700 GmIAA16-H.1	IAA	C	III	
## 24	Glyma.03G209400.1	Glyma.03G209400 GmTIR1/AFB1_B.1	AFB/TIR1	TIR1/AFB1	I	
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III	
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I	
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I	
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I	
## 36	Glyma.06G091700.3	Glyma.06G091700 GmIAA8-9-E.3	IAA	A	I	
## 38	Glyma.06G164900.2	Glyma.06G164900 GmARF2_A.2	ARF	B	I	
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I	
## 49	Glyma.08G008100.2	Glyma.08G008100 GmARF2_D.2	ARF	B	I	
## 64	Glyma.10G021500.2	Glyma.10G021500 GmTIR1/AFB1_C.2	AFB/TIR1	TIR1/AFB1	I	
## 72	Glyma.10G162400.2	Glyma.10G162400 GmIAA16-A.2	IAA	C	III	
## 76	Glyma.10G180100.1	Glyma.10G180100 GmIAA7/14/17-A.1	IAA	C	III	
## 116	Glyma.16G023600.2	Glyma.16G023600 GmARF11/18_A.2	ARF	B	I	
## 117	Glyma.16G050500.1	Glyma.16G050500 GmAFB2/3_B.1	AFB/TIR1	AFB2/3	II	
## 123	Glyma.19G100200.1	Glyma.19G100200 GmAFB2/3_A.1	AFB/TIR1	AFB2/3	II	
## 126	Glyma.19G161100.1	Glyma.19G161100 GmIAA16-G.1	IAA	C	III	
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III	
##	AM	OF	IAM	IBM	RootTip	Cotyledon
## 5	68.216252	51.350688	53.553189	96.275020	32.628226	78.301786
## 17	37.168779	5.439277	41.924483	51.089719	6.308382	0.000000
## 18	13.757865	13.859804	8.771196	7.354321	6.099923	14.166313
## 20	14.963834	9.149436	10.014337	18.303006	6.361899	6.335987
## 23	62.978354	188.282417	63.415971	50.273803	93.913232	11.797675
## 24	25.967032	26.377921	14.286116	20.719333	9.445604	22.549691
## 25	123.752976	182.226043	143.672394	136.882838	68.294552	100.483878
## 29	59.132821	34.776950	54.023677	104.014185	5.893899	15.036259

##	32	23.736932	7.300337	13.387625	33.516112	4.197497	21.693739
##	33	106.794815	41.848269	90.206750	89.149798	3.924654	25.580965
##	36	120.096829	73.349658	80.928913	112.799969	14.116324	123.198735
##	38	62.736767	31.362845	56.712867	104.850125	1.621706	14.695636
##	39	78.161251	91.333242	102.461124	95.846621	1.569549	29.653419
##	49	41.952212	22.684099	48.464588	54.600720	2.166574	14.976262
##	64	19.724328	16.170619	18.096732	18.503468	10.239295	14.506518
##	72	50.429036	62.648515	44.799041	49.047824	54.886130	14.549869
##	76	7.941917	536.895072	6.035997	7.033665	214.246604	30.729826
##	116	21.273921	9.543860	18.469447	22.065738	1.482088	5.689943
##	117	55.698420	35.061948	32.831068	58.586254	13.107811	27.899216
##	123	42.087683	43.031580	27.129734	42.689001	13.615904	12.545145
##	126	6.156592	108.404702	30.826540	8.523381	43.829775	4.519645
##	131	4.160975	265.136463	15.528614	13.540402	130.804062	41.664671
##		Hypocotyl	SAM6D	SAM17D	SAM38D	Callus	Leaf
##	5	172.112449	101.538840	108.035003	57.568952	3.602672	72.181052
##	17	23.986700	52.872850	68.490459	26.447317	6.139986	0.000000
##	18	33.826894	0.000000	11.829332	13.561473	1.551810	0.000000
##	20	4.159165	21.230577	13.709403	9.639602	2.603449	42.803542
##	23	382.933284	20.163323	39.919056	73.535631	0.583159	6.250057
##	24	20.867913	28.476275	25.128377	20.548182	11.542469	26.686917
##	25	215.724837	75.229555	108.926253	87.081914	65.420159	178.440543
##	29	13.614824	40.036100	67.500984	33.565553	5.088175	22.636836
##	32	20.950298	31.857638	31.063118	16.461606	4.368482	44.030052
##	33	38.494199	75.216825	290.336670	100.526085	25.977418	0.000000
##	36	118.328990	229.139072	393.546142	181.391747	71.756006	45.788590
##	38	13.618860	48.102339	51.456817	31.907033	8.665198	27.396630
##	39	18.983798	38.298793	237.526834	76.153152	6.048278	3.674760
##	49	26.934453	45.899996	104.119336	30.814124	3.509392	48.206033
##	64	14.635121	12.243954	22.316362	15.443247	25.621399	26.106051
##	72	258.563372	52.919869	43.482386	62.182188	12.401001	7.425099
##	76	350.297556	2.600212	4.426392	5.209396	2.744771	20.850299
##	116	0.823623	51.901613	58.246322	22.000399	0.000000	23.922628
##	117	27.176499	82.133333	87.939197	38.297183	47.396469	32.138566
##	123	19.247151	60.919431	75.317808	31.290701	44.433814	27.410816
##	126	334.804651	0.817625	0.938549	2.175286	0.045598	1.253888
##	131	633.844120	1.661192	0.821472	1.444420	0.244966	0.972353
##		Nodule	PC1	PC2	PC3	PC4	in.ell_TIR
##	5	22.341412	6.29787441	-1.8372424	1.5164566	-1.17162708	FALSE
##	17	0.000000	0.81044975	1.1170089	-0.7546656	0.42743184	FALSE
##	18	0.000000	-0.78784060	-0.1208382	-0.1980534	-0.30047323	FALSE
##	20	27.491901	-0.07853023	-0.3090942	1.7082831	0.36298038	FALSE
##	23	2.016265	2.50142090	-2.7024407	-2.7850869	1.95425693	FALSE
##	24	1.007309	0.70690012	-0.1981856	0.8944149	-0.24184281	FALSE
##	25	5.415049	9.79977762	-4.2635297	5.7326293	1.19531413	FALSE
##	29	10.758737	1.85976056	0.6174242	0.3612860	0.76750320	FALSE
##	32	21.300657	0.93454507	-0.4597545	1.6269670	-0.09431274	FALSE
##	33	0.000000	6.32619943	2.8542889	-2.3102058	1.29013983	FALSE
##	36	22.794609	14.43049528	2.1589901	-1.5310904	-2.57494825	FALSE
##	38	9.673077	1.95569813	0.5058034	0.6227889	0.85277438	FALSE
##	39	0.000000	4.44562780	2.0481314	-1.4582800	0.62005161	FALSE
##	49	8.836310	2.27842210	0.2301439	1.2286204	0.63434759	FALSE
##	64	3.205496	0.01141647	-0.2036243	0.9369297	0.04368445	FALSE
##	72	0.000000	2.42394340	-1.4008051	-1.9462386	0.94710249	FALSE


```
## 76 5.269519 0.20859279 -3.8984036 -0.9846724 -0.43448129 FALSE
## 116 1.267483 0.78469153 0.6753486 0.5447403 0.07576599 FALSE
## 117 8.945171 3.15794243 0.5915338 0.5657640 -0.06986253 FALSE
## 123 7.313215 1.88822437 0.6453113 0.4537617 0.33399282 FALSE
## 126 15.878389 -0.89218238 -3.0700736 -1.9027957 0.34702454 FALSE
## 131 0.495348 0.48333718 -6.5267137 -3.4003668 -0.81527683 FALSE
##      in_ell_IAA in_ell_ARF in_ell
## 5          FALSE      FALSE FALSE
## 17          TRUE      FALSE FALSE
## 18          TRUE      FALSE FALSE
## 20          FALSE      FALSE FALSE
## 23          FALSE      FALSE FALSE
## 24          TRUE       TRUE  FALSE
## 25          FALSE      FALSE FALSE
## 29          TRUE      FALSE FALSE
## 32          TRUE      FALSE FALSE
## 33          FALSE      FALSE FALSE
## 36          FALSE      FALSE FALSE
## 38          TRUE      FALSE FALSE
## 39          FALSE      FALSE FALSE
## 49          TRUE      FALSE FALSE
## 64          TRUE       TRUE  FALSE
## 72          FALSE      FALSE FALSE
## 76          FALSE      FALSE FALSE
## 116         TRUE      FALSE FALSE
## 117         TRUE      FALSE FALSE
## 123         TRUE      FALSE FALSE
## 126         FALSE      FALSE FALSE
## 131         FALSE      FALSE FALSE
```

```
PC2_3_7tissues +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
                                           xend=(PC2*7), yend=(PC3*7)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC2*7.2), #add the tissue names to it manually
              y=(PCA_loadings2$PC3*7.2),
              label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw()+
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  ggrepel::geom_text_repel(data = expr_logical_PC2_3_7tissues %>%
                           as_tibble(rownames = "name") %>%
                           filter(as.logical(in_ell == FALSE)),
                           aes(PC2, PC3, label=sub(".*\\|", "", heatmap_label)),
                           size=3, max.overlaps = 100, min.segment.length = 0,
                           segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()
```



```
# ggsave("20230927_PC2_3_7Tissues.png", dpi = 1000, width = 10, height = 8)
# ggsave("20230927_PC2_3_7Tissues.pdf", dpi = 1000, width = 10, height = 8)
```

Lets now repeat with the tau values for 7 tissues kept

```
# combine data frames expr_logic with tau values
expr_logic_PC2_3_7tissues <- cbind(expr_logical_PC2_3_7tissues, tau_df$tau)

# Define the intervals for expr_logic2$tau
expr_logic_PC2_3_7tissues$tau_interval <- cut(expr_logic_PC2_3_7tissues$tau,
      breaks = c(-Inf, 0.5, 0.8, Inf),
      labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC2_3_7tissues +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
      xend=(PC2*8.75), yend=(PC3*8.75)),
      arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC2*8.75), #add the tissue names to it manually
      y=(PCA_loadings2$PC3*8.75),
      label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1)) +

  geom_point(data = expr_logic_PC2_3_7tissues %>% as_tibble(rownames = "name"),
      aes(PC2, PC3, alpha = tau_interval, color = Family),
      size=3) +
  ggrepel::geom_text_repel(data = expr_logic_PC2_3_7tissues %>%
      as_tibble(rownames = "name") %>%
      filter(as.logical(in_ell == FALSE)),
```

```

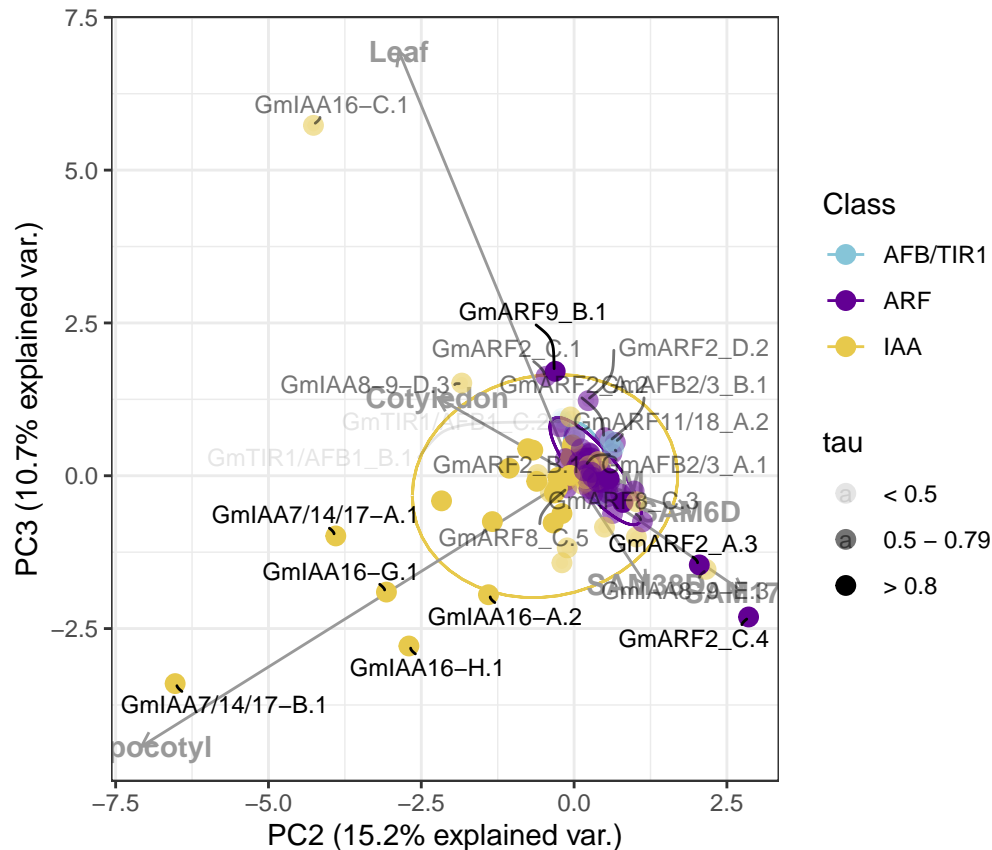
aes(PC2, PC3, label=sub(".*\\|", "", heatmap_label),
     alpha=tau_interval),
size=3, max.overlaps = 100, min.segment.length = 0,
segment.curvature = -0.1) +

scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
labs(color = "Class") +
labs(color = "Class", alpha = "tau") +

theme_bw()

```

Warning: Using alpha for a discrete variable is not advised.



```

#ggsave("20240117_PC2_3_7Tissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20240117_PC2_3_7Tissues.pdf", dpi = 1000, width = 10, height = 8)

```

PC3 and PC4 outside ellipse labels

```

PC3_4_7tissues <- ggbiplot::ggbiplot(pca2, obs.scale = 1, var.scale = .5, choices = c(3,4),
                                     varname.adjust = 3, alpha = .1,
                                     groups = Family, var.axes = F, varname.size = 2,
                                     ellipse = TRUE, circle = FALSE, ellipse.prob = .70)

# Extract components so we can select out
PC3_4_7tissues$data

```

##	xvar	yvar	groups
## 1	0.135736327	-0.166794726	ARF
## 2	0.391015981	0.021725559	IAA
## 3	0.467448263	-0.067536958	IAA
## 4	-0.618794158	0.414369028	IAA
## 5	1.516456612	-1.171627082	IAA
## 6	-0.298488215	0.155056850	ARF
## 7	-0.771942835	0.596692174	IAA
## 8	-0.019034277	-0.085791352	AFB/TIR1
## 9	0.123263847	0.612851368	IAA
## 10	0.296718025	-0.229885278	IAA
## 11	-0.074216901	-0.327047811	IAA
## 12	-0.307513791	-0.426101201	AFB/TIR1
## 13	0.118751120	-0.092852907	AFB/TIR1
## 14	0.210544795	0.019512657	IAA
## 15	-1.002175172	0.518106927	IAA
## 16	0.808419635	-0.301661161	ARF
## 17	-0.754665609	0.427431840	ARF
## 18	-0.198053390	-0.300473230	ARF
## 19	0.203702144	-0.032318665	ARF
## 20	1.708283097	0.362980383	ARF
## 21	-0.249254037	0.504865466	ARF
## 22	0.048238640	-0.079805461	IAA
## 23	-2.785086871	1.954256930	IAA
## 24	0.894414856	-0.241842809	AFB/TIR1
## 25	5.732629313	1.195314126	IAA
## 26	0.800391367	-0.312065182	ARF
## 27	0.024957986	-0.239320843	IAA
## 28	-0.074310120	0.116710951	AFB/TIR1
## 29	0.361286016	0.767503195	ARF
## 30	-0.433613208	0.518572902	ARF
## 31	0.101766304	-0.077183564	ARF
## 32	1.626967022	-0.094312744	ARF
## 33	-2.310205824	1.290139828	ARF
## 34	0.024556471	-0.195978255	ARF
## 35	0.264364967	0.445650978	IAA
## 36	-1.531090403	-2.574948254	IAA
## 37	0.141906335	-0.156514601	AFB/TIR1
## 38	0.622788860	0.852774381	ARF
## 39	-1.458279986	0.620051612	ARF
## 40	-0.750021960	0.220037571	IAA
## 41	0.377262769	-0.033857714	ARF
## 42	-0.439075999	0.142648481	ARF
## 43	0.051104600	-0.749801388	ARF
## 44	0.071044809	-0.068042055	ARF
## 45	-0.047741305	-0.140250716	AFB/TIR1
## 46	-0.031576561	-0.264358663	AFB/TIR1
## 47	0.161153854	-0.163854323	ARF
## 48	0.291896516	-0.187321330	ARF
## 49	1.228620351	0.634347589	ARF
## 50	-0.346616692	-0.633762872	ARF
## 51	0.061513280	-0.348275075	ARF
## 52	0.134356933	0.691566165	IAA
## 53	-0.185967868	-0.030659800	IAA

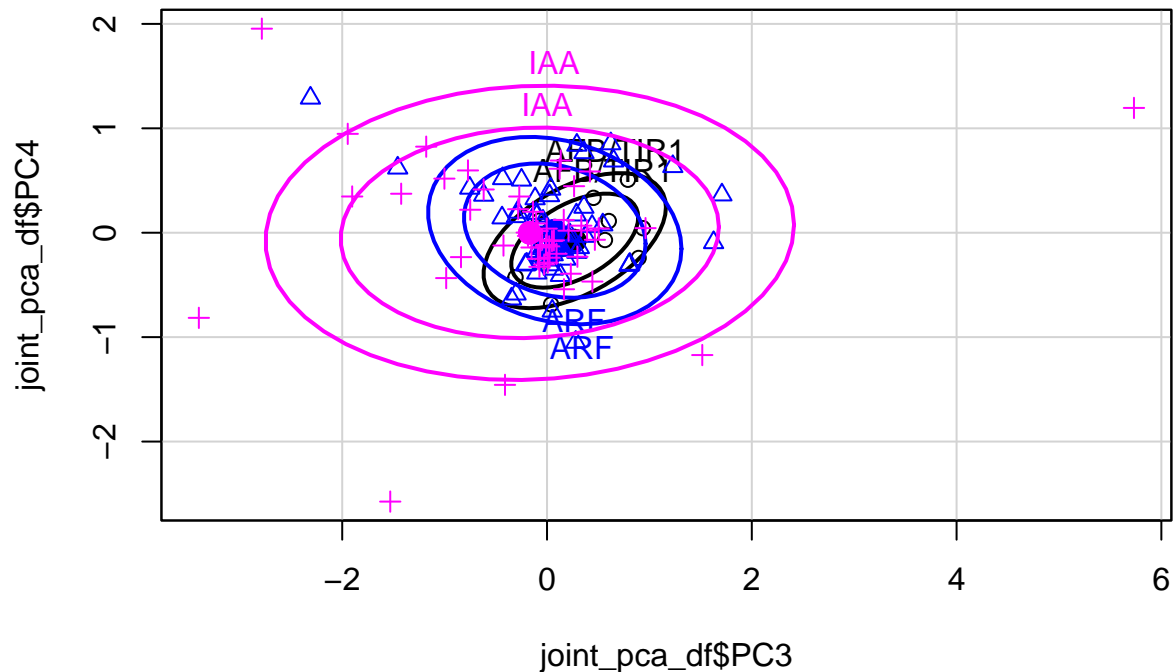
## 54	0.051140263	-0.104071856	AFB/TIR1
## 55	-0.286010626	0.197392945	ARF
## 56	0.222705920	-0.018660135	ARF
## 57	-0.007314077	-0.117126136	IAA
## 58	0.335059268	0.067574118	IAA
## 59	-1.424143572	0.373460488	IAA
## 60	0.022943597	-0.306222560	ARF
## 61	-0.009929315	-0.125112574	IAA
## 62	-1.179610247	0.824300448	IAA
## 63	0.443021793	-0.467603321	IAA
## 64	0.936929695	0.043684449	AFB/TIR1
## 65	0.214269342	-0.060390329	AFB/TIR1
## 66	0.040059182	-0.686082095	AFB/TIR1
## 67	-0.027750940	-0.304216067	IAA
## 68	0.504784255	0.041205637	IAA
## 69	0.067089129	-0.073136425	IAA
## 70	0.144718515	-0.017263227	ARF
## 71	-0.270753178	0.346803341	IAA
## 72	-1.946238594	0.947102490	IAA
## 73	0.165611508	-0.541531009	IAA
## 74	0.021079016	-0.130540709	IAA
## 75	0.418826607	0.586608071	IAA
## 76	-0.984672374	-0.434481291	IAA
## 77	-0.039468673	0.111993163	ARF
## 78	0.280245925	-1.048339856	ARF
## 79	0.650288391	0.687757199	ARF
## 80	-0.302086741	-0.586239367	ARF
## 81	0.360448967	0.244664502	ARF
## 82	0.032561259	-0.154052291	ARF
## 83	0.035765053	0.418059786	ARF
## 84	0.440653982	0.061925950	ARF
## 85	-0.100032400	-0.385496367	ARF
## 86	0.131115667	-0.408555161	ARF
## 87	0.112977921	-0.153389597	ARF
## 88	0.060351273	-0.210476652	ARF
## 89	0.093041365	-0.201897919	IAA
## 90	0.075836951	-0.127619481	ARF
## 91	-0.618197885	0.360031756	ARF
## 92	0.291349997	0.839323391	ARF
## 93	0.286799671	0.170762865	ARF
## 94	-0.107640571	0.185923451	ARF
## 95	-0.425150473	-0.122584160	IAA
## 96	0.962239600	0.043951364	IAA
## 97	-0.067668928	-0.140894757	IAA
## 98	-0.106174638	0.088468230	IAA
## 99	-0.090989128	0.016103825	IAA
## 100	-0.136450892	0.200696132	IAA
## 101	-0.002678882	0.039279304	IAA
## 102	0.228044041	-0.121129223	ARF
## 103	-0.211148378	-0.307775202	ARF
## 104	0.603415593	0.112023704	AFB/TIR1
## 105	-0.088816512	-0.250014529	IAA
## 106	0.024403946	0.355643287	ARF
## 107	-0.099986924	-0.179283342	ARF

```
## 108 -0.198514723 0.001283642 IAA
## 109 0.163541169 0.122506596 IAA
## 110 0.232333327 -0.392657919 IAA
## 111 -0.116129991 0.326223430 ARF
## 112 -0.051162129 -0.151720388 ARF
## 113 0.183674413 -0.252281288 ARF
## 114 0.118174410 -0.100612169 ARF
## 115 0.315839653 -0.141738678 ARF
## 116 0.544740295 0.075765990 ARF
## 117 0.565764005 -0.069862526 AFB/TIR1
## 118 0.022927865 -0.260894094 ARF
## 119 -0.047540345 -0.066857327 IAA
## 120 -0.045815482 -0.223146203 ARF
## 121 -0.100377639 0.160113846 ARF
## 122 -0.025405134 -0.135980283 ARF
## 123 0.453761721 0.333992817 AFB/TIR1
## 124 0.007290830 -0.286482378 IAA
## 125 -0.410351711 -1.457000005 IAA
## 126 -1.902795711 0.347024541 IAA
## 127 0.079576472 -0.140137191 ARF
## 128 0.790073247 0.507465423 AFB/TIR1
## 129 -0.151673820 -0.141425041 IAA
## 130 0.055831155 -0.099674050 ARF
## 131 -3.400366836 -0.815276835 IAA
## 132 -0.282003528 0.225510060 IAA
## 133 -0.840091171 -0.233615166 IAA
```

```
build_PC3_4_7tissues <- ggplot_build(PC3_4_7tissues)$data
points_PC3_4_7tissues <- build_PC3_4_7tissues[[1]]
```

```
# co-ordinates of the ellipses
```

```
ell_points_PC3_4_7tissues <- car::dataEllipse(joint_pca_df$PC3,
                                                joint_pca_df$PC4,
                                                as.factor(joint_pca_df$Family), levels=c(.7, .9))
```



```
# add geom_point with ellipses point

ell_ARF_PC3_4_7tissues <- as.data.frame(ell_points_PC3_4_7tissues$ARF$`0.7`)

ell_IAA_PC3_4_7tissues <- as.data.frame(ell_points_PC3_4_7tissues$IAA$`0.7`)

ell_TIR_PC3_4_7tissues <- as.data.frame(ell_points_PC3_4_7tissues$`AFB/TIR1`$`0.7`)

# Find which points are outside (!) the ellipse, and add this to the data
library(sp)
dat_TIR_PC3_4_7tissues <- data.frame(
  points_PC3_4_7tissues[2:3],
  in_ell_TIR = as.logical(point.in.polygon(points_PC3_4_7tissues$x, points_PC3_4_7tissues$y, ell_TIR_PC3_4_7tissues))
)

dat_IAA_PC3_4_7tissues <- data.frame(
  points_PC3_4_7tissues[2:3],
  in_ell_IAA = as.logical(point.in.polygon(points_PC3_4_7tissues$x, points_PC3_4_7tissues$y, ell_IAA_PC3_4_7tissues))
)

dat_ARF_PC3_4_7tissues <- data.frame(
  points_PC3_4_7tissues[2:3],
  in_ell_ARF = as.logical(point.in.polygon(points_PC3_4_7tissues$x, points_PC3_4_7tissues$y, ell_ARF_PC3_4_7tissues))
)

# as.logical(point..) equals to TRUE indicated points are inside ellipses

# Combining data points for labeling
transcript_expr_logical_PC3_4_7tissues <- cbind(joint_pca_df, dat_TIR_PC3_4_7tissues, dat_IAA_PC3_4_7tissues, dat_ARF_PC3_4_7tissues)

expr_logical_PC3_4_7tissues <- transcript_expr_logical_PC3_4_7tissues %>% mutate(., in_ell = case_when(
  Family=="IAA" & `in_ell_IAA` == TRUE ~ "TRUE",
  Family=="ARF" & `in_ell_ARF` == TRUE ~ "TRUE",
  Family=="TIR1" & `in_ell_TIR` == TRUE ~ "TRUE",
  TRUE ~ "FALSE"))
```

```

                                Family=="AFB/TIR1" & `in_ell_TIR` == TRUE ~ "TRUE")) %>%
mutate(in_ell = coalesce(in_ell, "FALSE"))

expr_logical_PC3_4_7tissues[which(expr_logical_PC3_4_7tissues$in_ell == FALSE),]

```

##	Transcript ID	heatmap_label	Family	Class	Clade		
## 5	Glyma.01G098000.3	Glyma.01G098000 GmIAA8-9-D.3	IAA	A	I		
## 12	Glyma.02G152800.2	Glyma.02G152800 GmTIR1/AFB1_D.2	AFB/TIR1	TIR1/AFB1	I		
## 17	Glyma.02G239600.3	Glyma.02G239600 GmARF8_C.3	ARF	A	II		
## 20	Glyma.03G070500.1	Glyma.03G070500 GmARF9_B.1	ARF	B	I		
## 23	Glyma.03G158700.1	Glyma.03G158700 GmIAA16-H.1	IAA	C	III		
## 24	Glyma.03G209400.1	Glyma.03G209400 GmTIR1/AFB1_B.1	AFB/TIR1	TIR1/AFB1	I		
## 25	Glyma.03G247400.1	Glyma.03G247400 GmIAA16-C.1	IAA	C	III		
## 29	Glyma.04G200600.1	Glyma.04G200600 GmARF2_B.1	ARF	B	I		
## 32	Glyma.05G200800.1	Glyma.05G200800 GmARF2_C.1	ARF	B	I		
## 33	Glyma.05G200800.4	Glyma.05G200800 GmARF2_C.4	ARF	B	I		
## 36	Glyma.06G091700.3	Glyma.06G091700 GmIAA8-9-E.3	IAA	A	I		
## 38	Glyma.06G164900.2	Glyma.06G164900 GmARF2_A.2	ARF	B	I		
## 39	Glyma.06G164900.3	Glyma.06G164900 GmARF2_A.3	ARF	B	I		
## 43	Glyma.07G130400.1	Glyma.07G130400 GmARF7/19_E.1	ARF	A	II		
## 49	Glyma.08G008100.2	Glyma.08G008100 GmARF2_D.2	ARF	B	I		
## 50	Glyma.08G008100.3	Glyma.08G008100 GmARF2_D.3	ARF	B	I		
## 62	Glyma.09G203300.3	Glyma.09G203300 GmIAA8-9-A.3	IAA	A	I		
## 64	Glyma.10G021500.2	Glyma.10G021500 GmTIR1/AFB1_C.2	AFB/TIR1	TIR1/AFB1	I		
## 66	Glyma.10G021500.1	Glyma.10G021500 GmTIR1/AFB1_C.1	AFB/TIR1	TIR1/AFB1	I		
## 72	Glyma.10G162400.2	Glyma.10G162400 GmIAA16-A.2	IAA	C	III		
## 78	Glyma.11G145500.1	Glyma.11G145500 GmARF10/16_D.1	ARF	C	III		
## 79	Glyma.11G204200.1	Glyma.11G204200 GmARF8_A.1	ARF	A	II		
## 80	Glyma.11G204200.2	Glyma.11G204200 GmARF8_A.2	ARF	A	II		
## 92	Glyma.13G221400.2	Glyma.13G221400 GmARF6_C.2	ARF	A	II		
## 125	Glyma.19G161000.3	Glyma.19G161000 GmIAA1-4-E.3	IAA	B	II		
## 126	Glyma.19G161100.1	Glyma.19G161100 GmIAA16-G.1	IAA	C	III		
## 128	Glyma.19G206800.1	Glyma.19G206800 GmTIR1/AFB1_A.1	AFB/TIR1	TIR1/AFB1	I		
## 131	Glyma.20G210400.1	Glyma.20G210400 GmIAA7/14/17-B.1	IAA	C	III		
##	AM	OF	IAM	IBM	RootTip	Cotyledon	Hypocotyl
## 5	68.216252	51.350688	53.55319	96.275020	32.628226	78.301786	172.112449
## 12	22.069192	16.181094	21.00405	22.849022	12.164769	15.200032	20.202937
## 17	37.168779	5.439277	41.92448	51.089719	6.308382	0.000000	23.986700
## 20	14.963834	9.149436	10.01434	18.303006	6.361899	6.335987	4.159165
## 23	62.978354	188.282417	63.41597	50.273803	93.913232	11.797675	382.933284
## 24	25.967032	26.377921	14.28612	20.719333	9.445604	22.549691	20.867913
## 25	123.752976	182.226043	143.67239	136.882838	68.294552	100.483878	215.724837
## 29	59.132821	34.776950	54.02368	104.014185	5.893899	15.036259	13.614824
## 32	23.736932	7.300337	13.38762	33.516112	4.197497	21.693739	20.950298
## 33	106.794815	41.848269	90.20675	89.149798	3.924654	25.580965	38.494199
## 36	120.096829	73.349658	80.92891	112.799969	14.116324	123.198735	118.328990
## 38	62.736767	31.362845	56.71287	104.850125	1.621706	14.695636	13.618860
## 39	78.161251	91.333242	102.46112	95.846621	1.569549	29.653419	18.983798
## 43	18.550414	8.914698	15.95095	18.729053	15.689098	23.181993	16.710000
## 49	41.952212	22.684099	48.46459	54.600720	2.166574	14.976262	26.934453
## 50	41.903106	15.801812	23.41788	55.329663	4.236413	28.260708	18.724678
## 62	36.472556	37.373970	38.15859	43.541833	11.505870	0.000000	117.295863
## 64	19.724328	16.170619	18.09673	18.503468	10.239295	14.506518	14.635121
## 66	14.419899	9.741531	6.79071	12.735183	6.157738	17.688606	17.203871

## 72	50.429036	62.648515	44.79904	49.047824	54.886130	14.549869	258.563372
## 78	15.909573	17.106814	12.58611	12.520897	6.758486	30.401493	8.543969
## 79	19.241075	21.103199	0.00000	3.055660	3.172375	0.000000	5.721453
## 80	13.373647	0.000000	35.09998	44.159391	4.581945	10.261290	31.902014
## 92	34.943626	18.239549	27.73661	43.916594	1.658267	1.279513	28.990865
## 125	4.478037	404.803343	44.07771	17.846549	72.730750	41.028328	177.214792
## 126	6.156592	108.404702	30.82654	8.523381	43.829775	4.519645	334.804651
## 128	29.377480	13.907445	22.73580	26.728168	6.489989	7.571060	13.364493
## 131	4.160975	265.136463	15.52861	13.540402	130.804062	41.664671	633.844120
##	SAM6D	SAM17D	SAM38D	Callus	Leaf	Root	Nodule
## 5	101.538840	108.035003	57.568952	3.602672	72.181052	53.728970	22.341412
## 12	29.352679	33.917530	17.176745	17.838179	0.000000	0.000000	0.000000
## 17	52.872850	68.490459	26.447317	6.139986	0.000000	3.626608	0.000000
## 20	21.230577	13.709403	9.639602	2.603449	42.803542	126.598709	27.491901
## 23	20.163323	39.919056	73.535631	0.583159	6.250057	14.892682	2.016265
## 24	28.476275	25.128377	20.548182	11.542469	26.686917	5.465411	1.007309
## 25	75.229555	108.926253	87.081914	65.420159	178.440543	98.055423	5.415049
## 29	40.036100	67.500984	33.565553	5.088175	22.636836	12.527492	10.758737
## 32	31.857638	31.063118	16.461606	4.368482	44.030052	25.338768	21.300657
## 33	75.216825	290.336670	100.526085	25.977418	0.000000	0.000000	0.000000
## 36	229.139072	393.546142	181.391747	71.756006	45.788590	20.983248	22.794609
## 38	48.102339	51.456817	31.907033	8.665198	27.396630	14.501406	9.673077
## 39	38.298793	237.526834	76.153152	6.048278	3.674760	1.606524	0.000000
## 43	16.764956	21.488396	10.756881	24.155359	2.997797	25.928328	7.688516
## 49	45.899996	104.119336	30.814124	3.509392	48.206033	26.688307	8.836310
## 50	55.230644	26.660841	30.862812	6.303453	0.000000	0.000000	0.216233
## 62	30.426647	41.114157	30.034268	6.427166	0.000000	30.080385	5.435140
## 64	12.243954	22.316362	15.443247	25.621399	26.106051	5.641934	3.205496
## 66	28.663983	21.325627	11.737596	3.718783	4.726063	2.093318	0.000000
## 72	52.919869	43.482386	62.182188	12.401001	7.425099	12.077243	0.000000
## 78	9.310248	10.078082	10.215456	0.759723	4.047713	2.691333	0.596545
## 79	0.000000	0.000000	17.832069	0.000000	18.572984	3.488127	1.310676
## 80	32.214786	40.620313	0.000000	17.181705	0.000000	0.000000	0.000000
## 92	22.677736	46.861420	20.641988	2.961649	19.960475	4.115777	1.477684
## 125	3.197441	2.314000	2.924909	0.843634	6.919914	6.338750	0.000000
## 126	0.817625	0.938549	2.175286	0.045598	1.253888	31.879730	15.878389
## 128	20.598762	24.312187	18.931187	13.373494	25.380331	5.994108	3.091861
## 131	1.661192	0.821472	1.444420	0.244966	0.972353	62.263900	0.495348
##	PC1	PC2	PC3	PC4	in.ell_TIR	in.ell_IAA	
## 5	6.29787441	-1.83724237	1.51645661	-1.17162708	FALSE	FALSE	
## 12	0.06210206	0.41382610	-0.30751379	-0.42610120	FALSE	TRUE	
## 17	0.81044975	1.11700892	-0.75466561	0.42743184	FALSE	TRUE	
## 20	-0.07853023	-0.30909424	1.70828310	0.36298038	FALSE	FALSE	
## 23	2.50142090	-2.70244068	-2.78508687	1.95425693	FALSE	FALSE	
## 24	0.70690012	-0.19818565	0.89441486	-0.24184281	FALSE	TRUE	
## 25	9.79977762	-4.26352971	5.73262931	1.19531413	FALSE	FALSE	
## 29	1.85976056	0.61742415	0.36128602	0.76750320	FALSE	TRUE	
## 32	0.93454507	-0.45975448	1.62696702	-0.09431274	FALSE	TRUE	
## 33	6.32619943	2.85428889	-2.31020582	1.29013983	FALSE	FALSE	
## 36	14.43049528	2.15899012	-1.53109040	-2.57494825	FALSE	FALSE	
## 38	1.95569813	0.50580340	0.62278886	0.85277438	FALSE	TRUE	
## 39	4.44562780	2.04813145	-1.45827999	0.62005161	FALSE	TRUE	
## 43	-0.18778513	0.04913286	0.05110460	-0.74980139	FALSE	TRUE	
## 49	2.27842210	0.23014394	1.22862035	0.63434759	FALSE	TRUE	

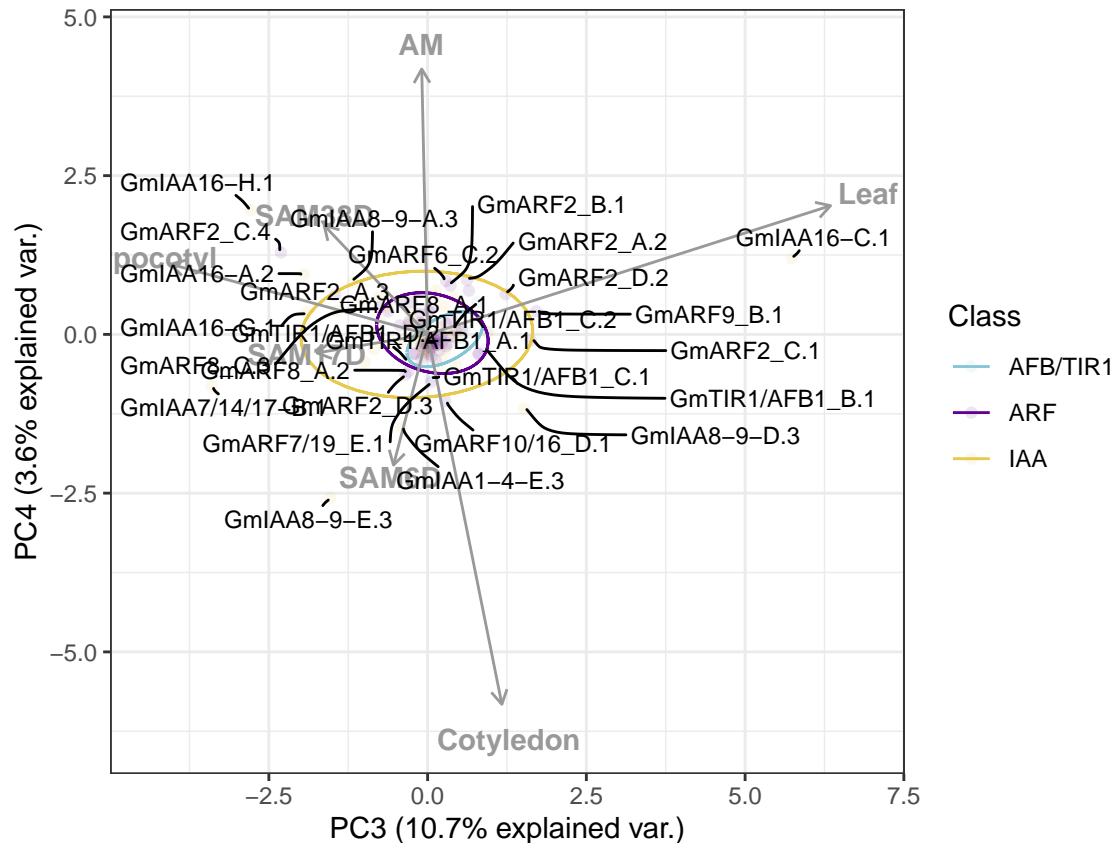
```
## 50  1.34307083  0.54381981 -0.34661669 -0.63376287      FALSE      TRUE
## 62  0.47369476 -0.11275080 -1.17961025  0.82430045      FALSE      FALSE
## 64  0.01141647 -0.20362431  0.93692969  0.04368445      FALSE      TRUE
## 66 -0.17447944  0.17662980  0.04005918 -0.68608210      FALSE      TRUE
## 72  2.42394340 -1.40080513 -1.94623859  0.94710249      FALSE      FALSE
## 78 -0.28107091 -0.14129573  0.28024593 -1.04833986      FALSE      FALSE
## 79 -0.79489435  0.01134430  0.65028839  0.68775720      FALSE      TRUE
## 80 -0.42946116  0.30503427 -0.30208674 -0.58623937      FALSE      TRUE
## 92  0.37773397  0.28877357  0.29135000  0.83932339      FALSE      TRUE
## 125 -0.19442276 -2.16841690 -0.41035171 -1.45700001      FALSE      FALSE
## 126 -0.89218238 -3.07007360 -1.90279571  0.34702454      FALSE      FALSE
## 128  0.22854712  0.06134928  0.79007325  0.50746542      FALSE      TRUE
## 131  0.48333718 -6.52671369 -3.40036684 -0.81527683      FALSE      FALSE
##      in_ell_ARF in_ell
## 5          FALSE FALSE
## 12         TRUE  FALSE
## 17          FALSE FALSE
## 20          FALSE FALSE
## 23          FALSE FALSE
## 24         TRUE  FALSE
## 25          FALSE FALSE
## 29          FALSE FALSE
## 32          FALSE FALSE
## 33          FALSE FALSE
## 36          FALSE FALSE
## 38          FALSE FALSE
## 39          FALSE FALSE
## 43          FALSE FALSE
## 49          FALSE FALSE
## 50          FALSE FALSE
## 62          FALSE FALSE
## 64         TRUE  FALSE
## 66          FALSE FALSE
## 72          FALSE FALSE
## 78          FALSE FALSE
## 79          FALSE FALSE
## 80          FALSE FALSE
## 92          FALSE FALSE
## 125         FALSE FALSE
## 126         FALSE FALSE
## 128         FALSE FALSE
## 131         FALSE FALSE
```

```
PC3_4_7tissues +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
                                           xend=(PC3*8), yend=(PC4*8)),
              arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC3*8.75), #add the tissue names to it manually
             y=(PCA_loadings2$PC4*8.75),
             label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw()+
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  ggrepel::geom_text_repel(data = expr_logical_PC3_4_7tissues %>%
```

```

as_tibble(rownames = "name") %>%
  filter(as.logical(in_ell == FALSE)),
  aes(PC3, PC4, label=sub(".*\\|", "", heatmap_label)),
  size=3, max.overlaps = 100, min.segment.length = 0,
  segment.curvature = -0.1) +
  labs(color = "Class") +
  theme_bw()

```



```

#ggsave("20230927_PC3_4_7Tissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20230927_PC3_4_7Tissues.pdf", dpi = 1000, width = 10, height = 8)

```

Repeat the same to add tau measurement

```

# combine data frames expr_logic with tau values
expr_logical_PC3_4_7tissues <- cbind(expr_logical_PC3_4_7tissues, tau_df$tau)

# Define the intervals for expr_logic2$tau
expr_logical_PC3_4_7tissues$tau_interval <- cut(expr_logical_PC3_4_7tissues$tau,
  breaks = c(-Inf, 0.5, 0.8, Inf),
  labels = c("< 0.5", "0.5 - 0.79", "> 0.8"))

PC3_4_7tissues +
  geom_segment(PCA_loadings2, mapping=aes(x=0, y=0, # Change the size of arrows
    xend=(PC3*8), yend=(PC4*8)),
    arrow = arrow(length = unit(1/2, "picas")), color="gray60") +
  annotate("text", x=(PCA_loadings2$PC3*8.75), #add the tissue names to it manually
    y=(PCA_loadings2$PC4*8.75),

```

```

    label=PCA_loadings2$Variables, size=4, color="gray60", fontface="bold") +
  theme(panel.background = element_rect(fill = "white", linewidth = 1))+
  theme_bw() +

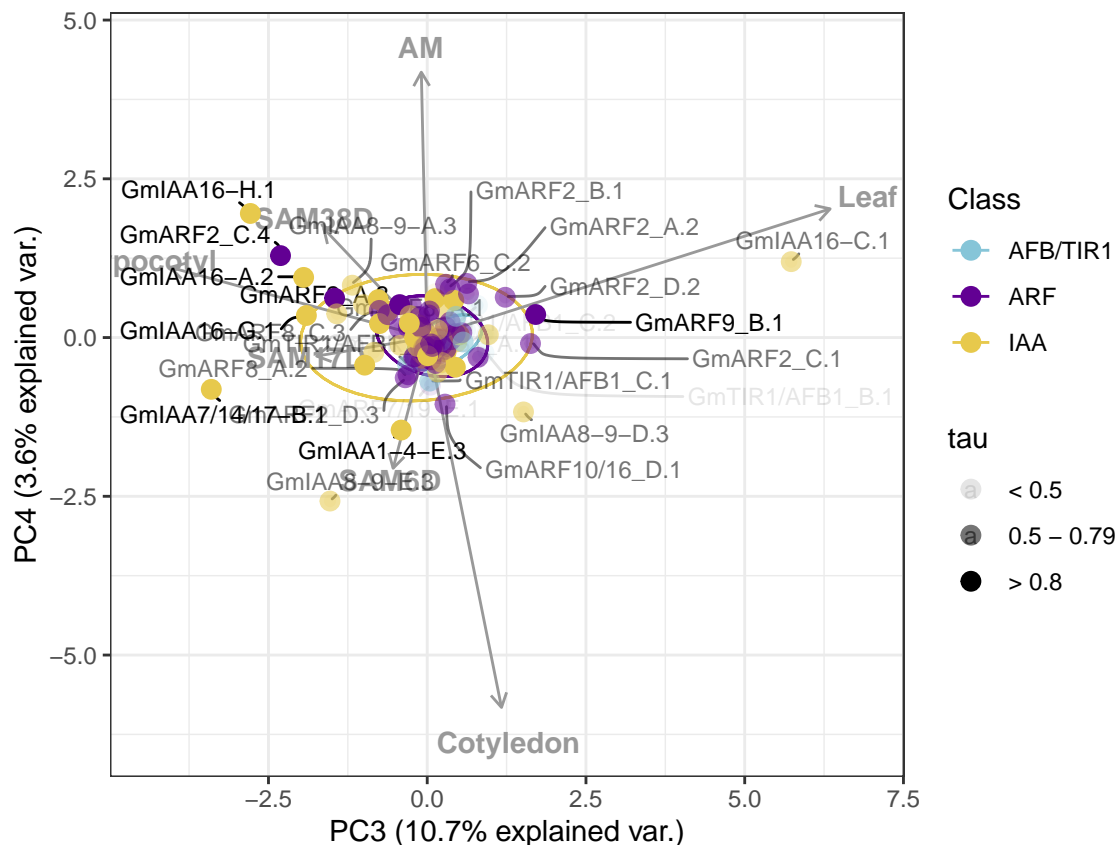
  ggrepel::geom_text_repel(data = expr_logical_PC3_4_7tissues %>%
    as_tibble(rownames = "name") %>%
    filter(as.logical(in_e11 == FALSE)),
    aes(PC3, PC4, label=sub(".*\\|", "", heatmap_label),
      alpha = tau_interval),
    size=3, max.overlaps = 100, min.segment.length = 0,
    segment.curvature = -0.1) +

  labs(color = "Class") +

  geom_point(data = expr_logical_PC3_4_7tissues %>% as_tibble(rownames = "name"),
    aes(PC3, PC4, alpha = tau_interval, color = Family),
    size=3) +
  scale_color_manual(values=c("#86C5D8", "#620093", "#E7C94C")) +
  labs(color = "Class") +
  labs(color = "Class", alpha = "tau") +
  theme_bw()

```

Warning: Using alpha for a discrete variable is not advised.



```

#ggsave("20240117_PC3_4_7Tissues.png", dpi = 1000, width = 10, height = 8)
#ggsave("20240117_PC3_4_7Tissues.pdf", dpi = 1000, width = 10, height = 8)

```

Principal component comparisons.

We observe here the differences between our principal component analysis with all 14 tissues from databases, and with only the 7 tissues that are part of aerial architecture. It is visible that our principal component analysis is minimally affected by the exclusion of tissues herein analysed, hence we are confident to display only the aerial tissues we are interested in. We also observe that root, hypocotyl, nodule, and open flower fall along the same direction, implying that these tissues are correlated. As we move to principal components that accounts for smaller amounts of variation, we notice an improvement in the discrimination of the correlation between root, nodule and hypocotyl tissues. On the other hand root tips, and open flower correlation to hypocotyl shifts more when looking at PC3 and PC4, with a very small discrimination between these tissues. We can speculate that this correlation could be due to these tissues being at approximate similar developmental stages, and therefore exhibit similar patterns of gene expression as they respond to common developmental cues. It also shows the redundancy of auxin response genes as they have overlap in functional roles. Further investigation of these genes are important to gain more insightful information of which genes are important in growth and development of these tissues, and if different pair of auxin regulatory genes are important during this process.

add tau correlation to Geom distance from origin

- Tau does provide some additional information
- highlight some original candidates with high | low tau identified in PCA
- family wise tau distribution: are ARFs or IAAs more tissue specific?
- Discussion:
 - other means of calculating tau, or better suited datasets
 - machine learning model of candidate genes

```
geom_dist_calc_df <- cbind(comb_pca_df, tau_df$tau)

geom_dist_calc_df2 <- geom_dist_calc_df %>% dplyr::select(c(heatmap_label, Family, PC1, PC2, PC3, PC4,

# square PC values
geom_dist_calc_df2[,3:6] <- (geom_dist_calc_df2[, 3:6])^2

# sum all PCs then take its square root
geom_dist_calc_df2$Geom_dist <- NA

for (i in 1:nrow(geom_dist_calc_df2)) {
  geom_dist_calc_df2[i, 8] <- sum(geom_dist_calc_df2[i, 3:6], na.rm = TRUE)
}

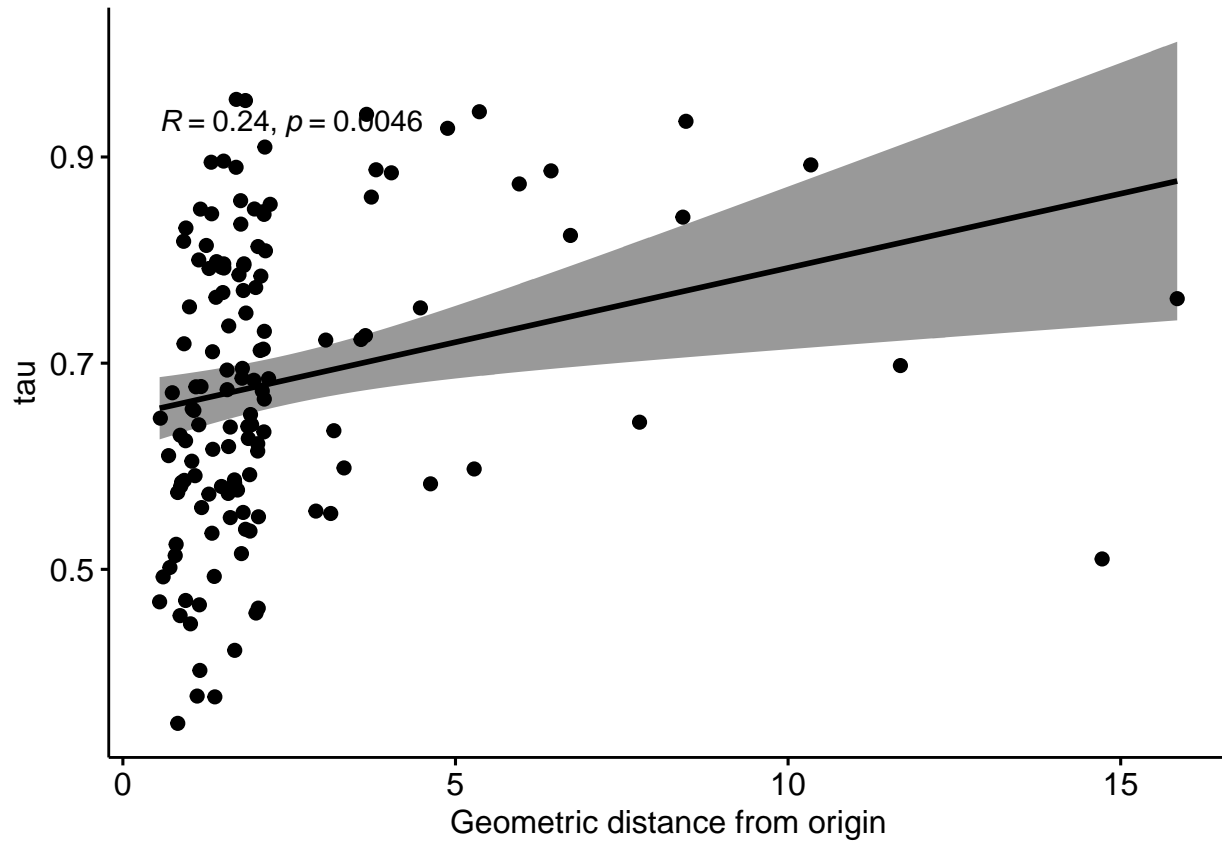
geom_dist_calc_df2$Geom_Dist_sqrt <-
  sqrt(geom_dist_calc_df2$Geom_dist)

library("ggpubr")

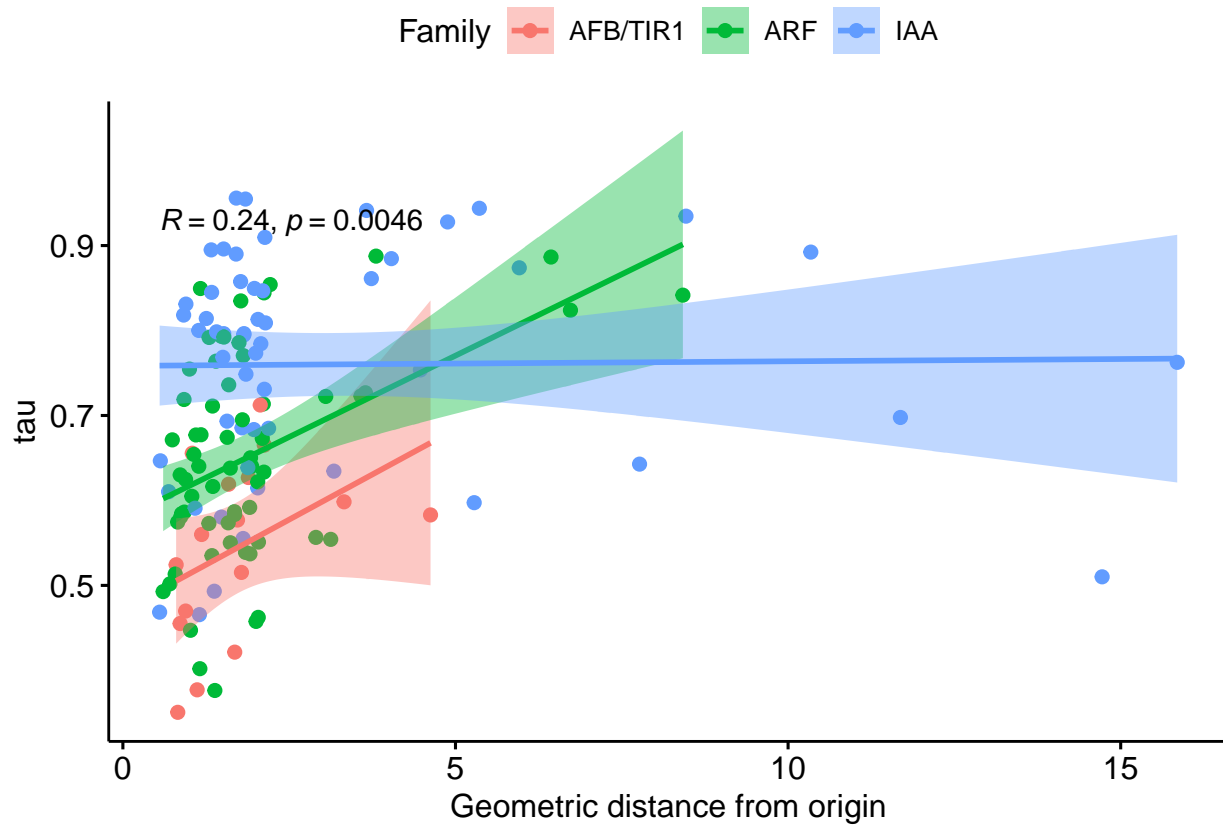
##
## Attaching package: 'ggpubr'

## The following object is masked from 'package:plyr':
##
##      mutate
```

```
ggscatter(geom_dist_calc_df2, x = "Geom_Dist_sqrt",
  y = "tau_df$tau",
  add = "reg.line", conf.int = TRUE,
  cor.coef = TRUE, cor.method = "pearson",
  xlab = "Geometric distance from origin", ylab = "tau")
```



```
ggscatter(geom_dist_calc_df2, x = "Geom_Dist_sqrt",
  y = "tau_df$tau",
  add = "reg.line", conf.int = TRUE,
  cor.coef = TRUE, cor.method = "pearson",
  color = "Family",
  xlab = "Geometric distance from origin", ylab = "tau")
```



From the correlation plots above we can see that there isn't a strong correlation between Geometric distance and tau values, supporting the importance of tau values in this analysis. We believe a larger dataset may be important in order to in this calculations and we recommend that that should be pursued in future analysis.