Candidate genes affecting glucosinolate content in *Brassica napus*, by Associative Transcriptomics

Y3856072

***Brassica napus*, commonly oilseed rape, is a widely grown crop, primarily cultivated for its vegetable oil which is extracted leaving a by-product of animal feed. All *Brassica* plants contain a family of secondary metabolites called glucosinolates (Fahey, Zalcmann and Talalay, 2001). This study will identify genes with expression associated to glucosinolate content using the novel association mapping tool Associative Transcriptomics, with an aim of providing a basis for selective breeding or genetic modification for varied glucosinolate levels. Genes with known links were identified and their relationship with glucosinolates investigated. [redacted] genes were found to have expression levels significantly associated with glucosinolate content, [redacted] of which have known links to glucosinolates. Further analysis investigated the correlation between the genes with a known link and the glucosinolate content and identified promising candidates. Analysis also indicated possible clusters of linked genes in the genome.**

1. Introduction

Glucosinolates and their products have been found to have deleterious effects when present in animal feed (Tripathi et al. 2001; Burel et al. 2000). In contrast, studies have shown a link between glucosinolates and a variety of positive attributes; with antibacterial (Johns et al. 1982) and antifungal properties (Drobnica et al. 1967; Manici et al. 1997) alongside a link to cancer chemoprotection (Zhang et al. 1994; Fahey et al. 1997). Whilst a traditional aim was to select *Brassica napus* lines with reduced glucosinolate content, there is now increasing interest in utilising the positive effects of the glucosinolates found in *Brassica* as well.

2. Result and discussion

2.1 Associated genes

It was investigated whether any genes have expression levels significantly associated with glucosinolate content. Using R (R core team, 2016), an ANOVA tested the correlation for all genes. A false discovery rate (FDR) multiple test correction was applied, due to its low stringency allowing the identification of additional results, and any genes passing a 0.05 significance were identified as associated. [redacted] genes were found to be significantly related to glucosinolate content (Figure 1). The correlation between gene expression levels and glucosinolate content, alongside positional information and FDR significance, is detailed in Table 1. These genes are candidate targets for altering glucosinolate content by changing expression levels in *Brassica* lines. This study is limited by only having glucosinolate content data for 53 of the 101 lines investigated, leading to a reduction in power to detect associations.

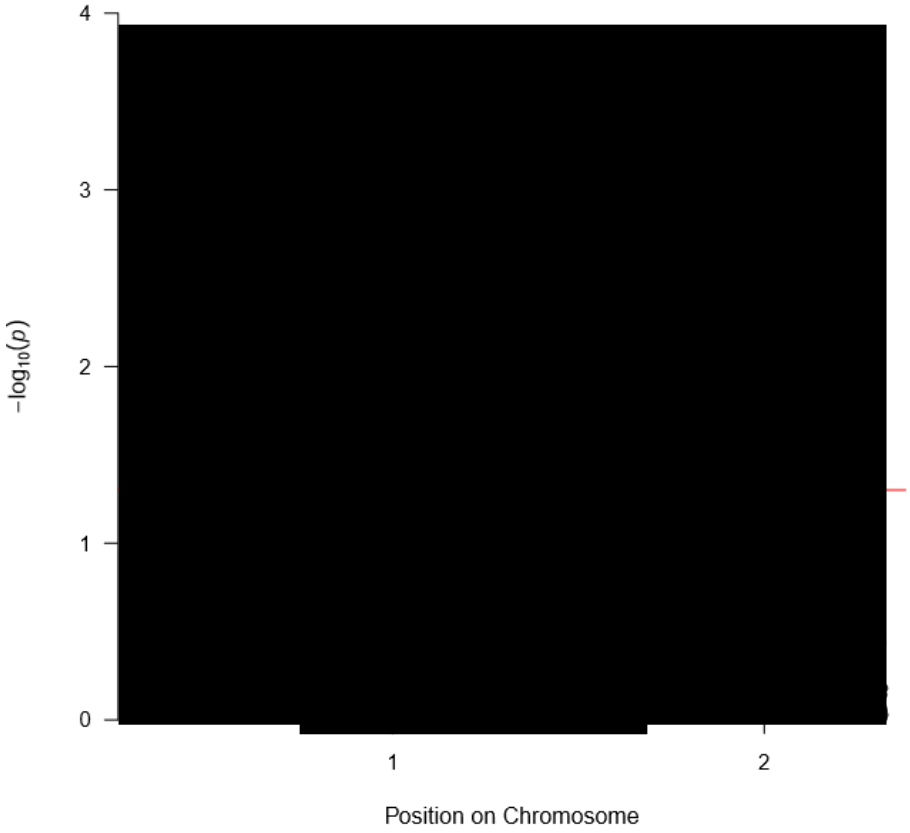

Figure 1. [redacted] genes had significant associations with glucosinolate levels. Manhattan plot shows association between gene expression levels and glucosinolate content. Red line indicates 0.05 significance (manipulated by -log10), with genes above passing significance threshold. Green genes identified as having a known link to glucosinolate.

Table 1. Genes found to be significant. Table shows position along the chromosome (bp) – all genes in the table occurred on chromosome 1. Gradient of correlation between the gene's expression and the glucosinolate content detailed as well as the FDR probability of association. FDR refers to False Discovery Rate multiple test correction used.

Gene	Position (bp)	Gradient	FDR
[redacted]			

2.2. Genes of interest

Whilst this study identified genes with associations, genes with an already understood link to glucosinolates, hereafter genes of interest (GOI), are more likely to be targeted when altering glucosinolate content in *Brassica* lines. This study therefore further analysed only the GOI to better characterise their relationship with glucosinolates.

To identify whether the interaction between these associated genes and glucosinolates are already known, BLAST sequence analysis was performed on all [redacted], often using the closely

related model organism *Arabidopsis thaliana*. 5 GOI were found; usually present in either the biosynthesis or biodegradation pathways.

Figure 1 shows the GOI (in green) are surrounded in the genome by associated genes with no known link. This suggests these genes have yet uncharacterised interactions with glucosinolate pathways. Further investigations may therefore characterise gene expression for these unknown genes and the mechanism by which they interplay with glucosinolates, in doing so revealing more candidates for altering glucosinolate content.

To investigate the relationship between the GOI and glucosinolate content, expression of each gene was plotted against percentage of glucosinolate in seed oil. This, alongside the existing literature for the genes, indicates the relationship each gene has with glucosinolates.

[redacted] is positively correlated with glucosinolate levels. [redacted] This encodes a transcription factor which directly activates genes involved in glucosinolate biosynthesis (Baskar and Park, 2015). [redacted] is therefore a candidate for reducing expression levels, leading to less of the transcription factor and therefore reducing glucosinolate levels. Similarly, increasing expression would likely result in an increased amount of glucosinolates in *Brassica* oil.

[redacted]

In BLAST analysis, [redacted] matched [redacted] which encodes for the [redacted] enzyme. This enzyme is involved in converting between two glucosinolate types (Hansen et al. 2007). The function of converting glucosinolates suggests altering [redacted] expression would increase some glucosinolates and similarly reduce others. The positive correlation shown in figure 2C suggests that increasing or decreasing expression would result in a similar change in overall glucosinolate levels.

Figure 2D found a positive correlation between expression of [redacted] and the amount of glucosinolate. This is supported by the BLAST results, finding [redacted] corresponded to [redacted], which encodes a [redacted]. This protein is involved in glucosinolate biosynthesis (Grubb et al.

2004) and suggests reduction of this protein by lowering gene expression would result in fewer glucosinolates, or vice versa.

BLAST found that [redacted] corresponded to [redacted] a gene involved in the secondary modifications of certain glucosinolates (Neal et al. 2010). The positive correlation shown by Figure 2E indicates this modification results in functional glucosinolates. This suggests that altering the modification of glucosinolates by changing [redacted] expression would result in similarly affected glucosinolate levels.

This study is limited by its ability to prove the speculated change in glucosinolate level resulting from altering expression level. Further investigations are needed to confirm these hypotheses; however this study provides a strong basis. The study is also limited by a lack of specificity of glucosinolates investigated; only quantifying overall glucosinolate levels. This is significant as different glucosinolates may have varied characteristics. This would mean levels of specific glucosinolates may need to be targeted, depending on desired function of the *Brassica* line. Also, this limits the ability of this study to compare to existing findings which refer to specific glucosinolates, as with [redacted]

Correlation values between gene expression levels and glucosinolate content was investigated to reveal the best candidates for altering glucosinolate levels. The gradients of the relationship between GOI and glucosinolate levels are displayed in table 1. [redacted] has the steepest gradient of correlation of [redacted] whilst the other GOI were in the range of 1 to 3. This suggests the bile acid transporter encoded by [redacted] is heavily used in glucosinolate biosynthesis and may have low redundancy from other transporter proteins. Therefore changes in [redacted] expression levels would result in the greatest changes in glucosinolate content, identifying it as a strong candidate for affecting glucosinolate levels. As before, additional studies would be needed to test and confirm these suggestions.

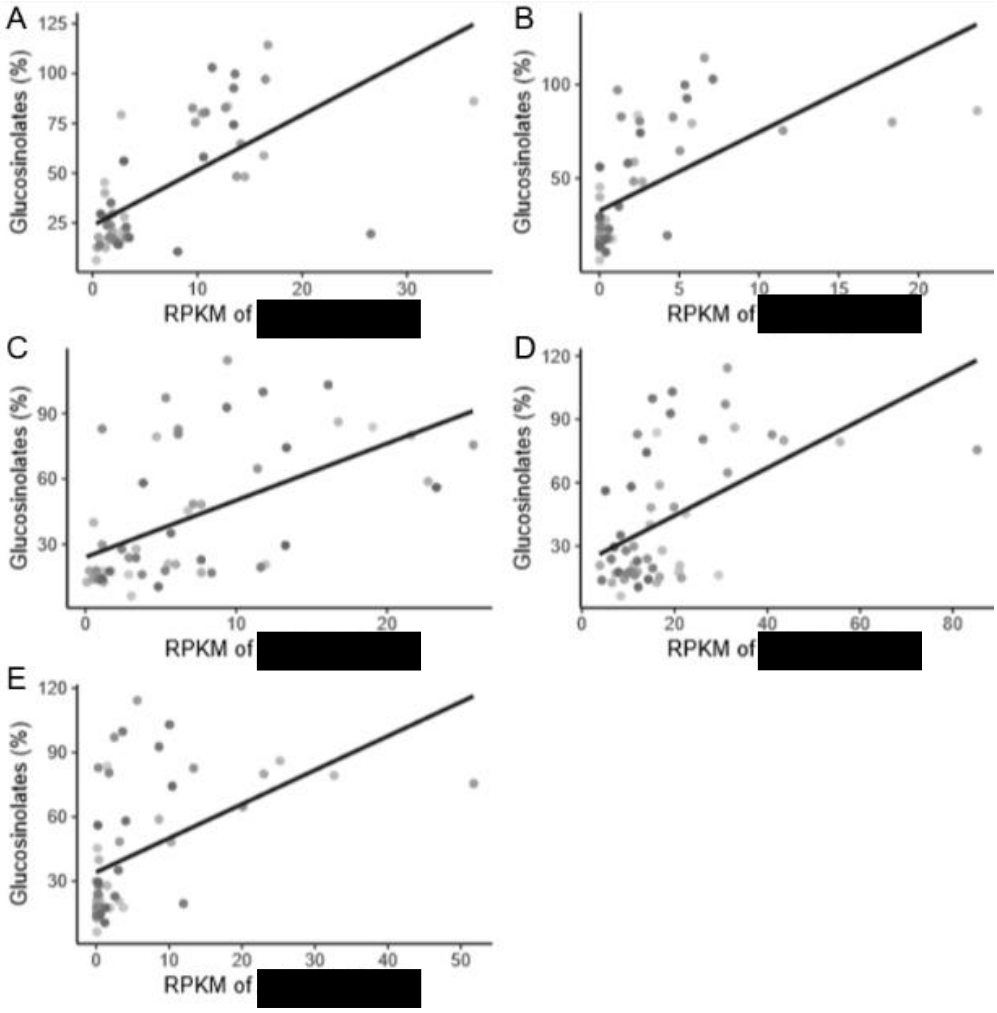

Figure 2.All genes with known links to glucosinolates have a positive correlation with glucosinolate levels. A. Gradient [redacted]

This study also hypothesised that some GOI have significantly higher expression than others. A Kruskal-Wallace test found there was no overall significant difference in expression level between the GOI (Kruskal-Wallis chi-squared = 262.11, df = 245, p-value = 0.2161), however a post-hoc test identified certain genes had significantly different expression values at a 0.05 significance threshold. Figure 3 shows [redacted] has significantly higher expression values than all other GOI, confirming the hypothesis. For gene knockouts, [redacted] would likely show the greatest changes in glucosinolate levels, due to the greatest reduction in expression levels. Further studies are needed to compare glucosinolate levels for knockouts in the GOI to confirm this.

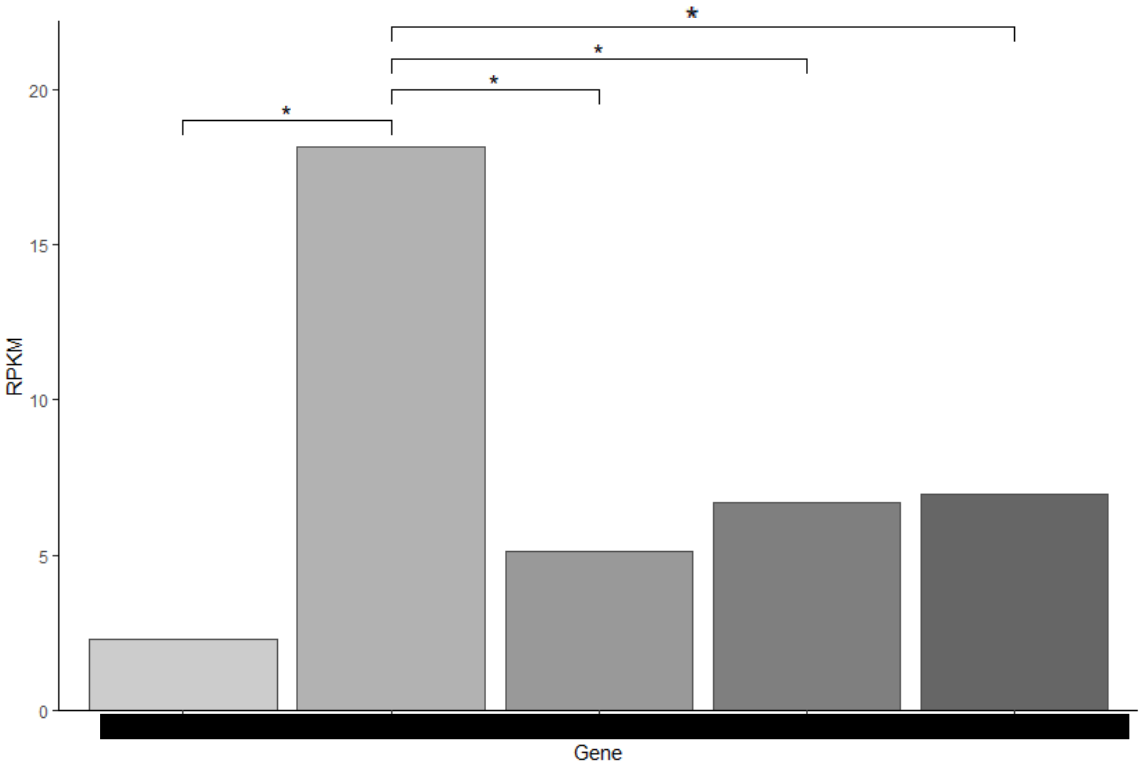

Figure 3 [redacted] has significantly higher expression values than other genes with a known link to glucosinolate levels. * indicates $p < 0.05$. RPKM refers to reads per kilobase per million aligned reads.

2.3. Gene clusters

It was hypothesised that genes relating to glucosinolate content would be clustered on the genome. Figure 4 shows the position of associated genes on the chromosome, with an aggregation marked by the light grey box. This indicates a region in which genes have a high chance of being linked due to proximity. The dark grey area indicates a second high-density region of associated genes, with a higher likelihood of genes within being linked. Most genes within these regions have positive correlation, likely due to the greater number of positively associated genes. Targeting genes within these regions by selective breeding or other means may lead to the shared inheritance of other, linked genes. This could therefore result in unexpected changes in glucosinolate levels when selecting to alter expression levels; either by decreasing expression of a linked gene with the desired correlation or increasing expression of a liked gene with the alternate correlation. The GOI are less common within these regions than elsewhere, with only [redacted] within, therefore focusing on GOI outside of the marked regions may produce results which are in line with expected changes. This study lacks evidence to prove any genetic linkage and so cannot accept or reject the hypothesis. Further inheritance and linkage studies are needed to provide such evidence.

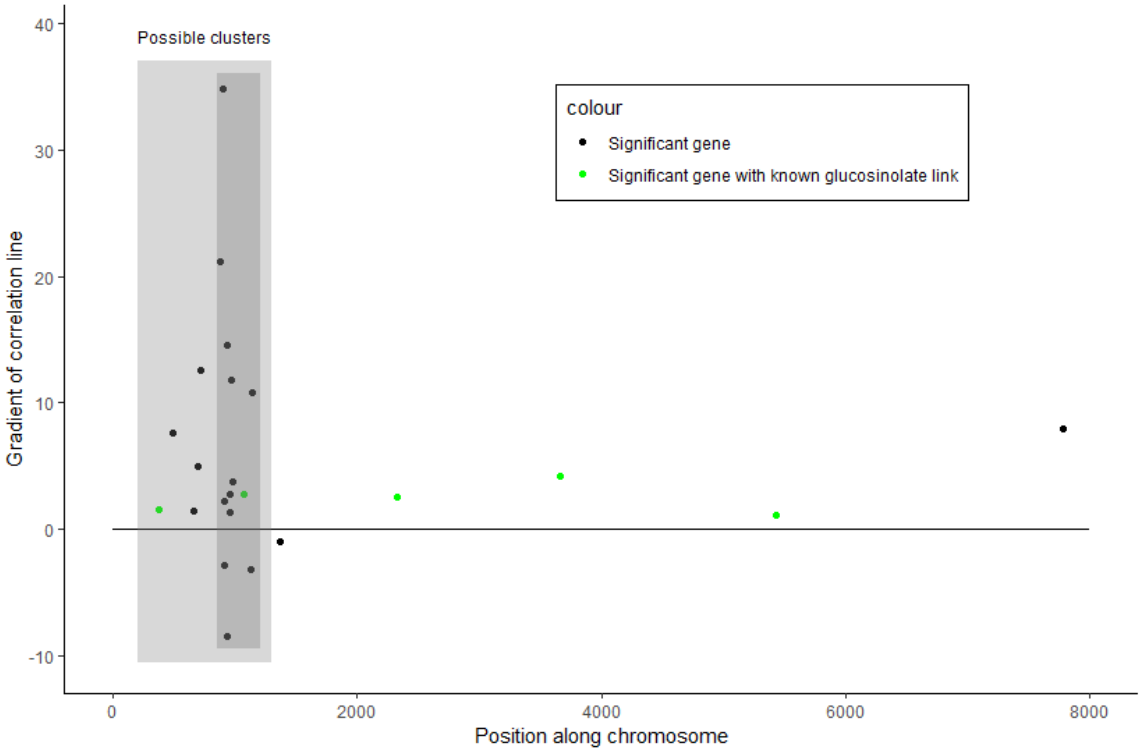

Figure 4. Possible clusters of glucosinolate-related genes in the genome. All data from chromosome 1 as no associated genes on chromosome 2. Gradient of correlation line refers to correlation line between gene expression and glucosinolate content. Every point is an associated gene. Light grey area marker indicates likely clusters within, dark grey indicates location of a possible cluster of associated genes in the genome.

3. Conclusion

Using Associative Transcriptomics [redacted] genes were identified as significantly associated with glucosinolate levels. Of these, 5 were subsequently identified as GOI; genes with a known link to glucosinolates. All GOI were found to have positive correlations between expression and glucosinolate levels. [redacted] was identified as a promising target for altered expression levels due to its high gradient of correlation. [redacted] was identified as a candidate for knockouts due to having significantly larger expression values than other GOI, thus greater reductions. This study found regions suspected to contain linked genes and discussed the implications of this, though had no evidence to support claims of linkage. Overall several candidates were found for affecting glucosinolate levels, the most promising of which were identified through further analysis.

4. References

Baskar, V. and Park, S. W. (2015). Molecular characterization of BrMYB28 and BrMYB29 paralogous transcription factors involved in the regulation of aliphatic glucosinolate profiles in Brassica rapa ssp. pekinensis. *Comptes rendus biologies*, 338 (7), pp.434–442.

Burel, C. et al. (2000). Dietary low-glucosinolate rapeseed meal affects thyroid status and nutrient utilization in rainbow trout (Oncorhynchus mykiss). *The British journal of nutrition*, 83 (6), pp.653–664.

Drobnica, L. et al. (1967). Antifungal activity of isothiocyanates and related compounds. I. Naturally occurring isothiocyanates and their analogues. *Applied microbiology*, 15 (4), pp.701–709.

Fahey, J. W., Zalcmann, A. T. and Talalay, P. (2001). The chemical diversity and distribution of glucosinolates and isothiocyanates among plants. *Phytochemistry*, 56 (1), pp.5–51.

Fahey, J. W., Zhang, Y. and Talalay, P. (1997). Broccoli sprouts: an exceptionally rich source of inducers of enzymes that protect against chemical carcinogens. *Proceedings of the National Academy of Sciences of the United States of America*, 94 (19), pp.10367–10372.

Gigolashvili, T. et al. (2009). The plastidic bile acid transporter 5 is required for the biosynthesis of methionine-derived glucosinolates in Arabidopsis thaliana. *The Plant cell*, 21 (6), pp.1813–1829.

Grubb, C. D. et al. (2004). Arabidopsis glucosyltransferase UGT74B1 functions in glucosinolate biosynthesis and auxin homeostasis. *The Plant journal: for cell and molecular biology*, 40 (6), pp.893–908.

Hansen, B. G., Kliebenstein, D. J. and Halkier, B. A. (2007). Identification of a flavin-monooxygenase as the S-oxygenating enzyme in aliphatic glucosinolate biosynthesis in Arabidopsis. *The Plant journal: for cell and molecular biology*, 50 (5), pp.902–910.

Johns, T. et al. (1982). Anti-reproductive and other medicinal effects of Tropaeolum tuberos um. *Journal of ethnopharmacology*, 5 (2), pp.149–161.

Manici, L. M., Lazzeri, L. and Palmieri, S. (1997). In Vitro Fungitoxic Activity of Some Glucosinolates and Their Enzyme-Derived Products toward Plant Pathogenic Fungi. *Journal of agricultural and food chemistry*, 45 (7), pp.2768–2773.

Neal, C. S. et al. (2010). The characterisation of AOP2: a gene associated with the biosynthesis of aliphatic alkenyl glucosinolates in Arabidopsis thaliana. *BMC plant biology*, 10, p.170.

RStudio Team (2016). RStudio: Integrated Development for R. RStudio, Inc., Boston, MA U R. http://www.rstudio.com/.

Tripathi, M. K. et al. (2001). Effect of substitution of groundnut with high glucosinolate mustard (Brassica juncea) meal on nutrient utilization, growth, vital organ weight and blood composition of lambs. *Small ruminant research: the journal of the International Goat Association*, 39 (3), pp.261–267.

Zhang, Y. et al. (1994). Anticarcinogenic activities of sulforaphane and structurally related synthetic norbornyl isothiocyanates. *Proceedings of the National Academy of Sciences of the United States of America*, 91 (8), pp.3147–3150.

Word count: 1347

```
##########################################################################
########## #                  Supplementary methods section                  #
##########################################################################
##########

##########################################################################
##########
#                  Section 1: Initial setup                  #
##########################################################################
##########

# Initial setup of working directory.
setwd("~/Uni/year 2/Big data")

# Install the packages needed throughout the data analysis and load them in.
# References for each package below.

# Install and load the Car package.
install.packages("car")
library("car")
# Reference:
# John Fox and Sanford Weisberg (2011). An {R} Companion to Applied
# Regression, Second Edition. Thousand Oaks CA: Sage. URL:
# http://socserv.socsci.mcmaster.ca/jfox/Books/Companion

# Install and load the qqman package
install.packages("qqman")
```

```
library(qqman)
# Reference:
# Stephen Turner (2017). qqman: Q-Q and Manhattan Plots for GWAS Data. R
# package version 0.1.4. https://CRAN.R-project.org/package=qqman

# Install and load the ggplot2 package
install.packages("ggplot2")
library(ggplot2)
# Reference:
# H. Wickham. ggplot2: Elegant Graphics for Data Analysis. Springer-Verlag
# New York, 2016.

# Install and load the tidyr package
install.packages("tidyr")
library(tidyr)
# Reference:
# Hadley Wickham and Lionel Henry (2019). tidyr: Easily Tidy Data with
# 'spread()' and 'gather()' Functions. R package version 0.8.3.
# https://CRAN.R-project.org/package=tidyr

#install and load the pgirmess package
install.packages("pgirmess")
library(pgirmess)
# Reference:
# Patrick Giraudoux (2018). pgirmess: Spatial Analysis and Data Mining for
# Field Ecologists. R package version 1.6.9.
```

```
# https://CRAN.R-project.org/package=pgirmess

##############################################################################
##########
#                    Section 2: Data preporation                    #
##############################################################################
##########

# Read in the datafile containing expression data for different genes for different
# Brassica napus lines and format it ready for processing and analysis.

# Read in the Brasicus data file for gene expression.
Bn <-read.delim("http://www-users.york.ac.uk/~ah1309/BigData/data/OSR101_RPKM.txt")
# Make the genes the row names and delete the column containing genes as it is now
# duplicated.
rownames(Bn) <- Bn$Gene
# Delete duplicated column, the column duplicated is the 1st column.
Bn_2 <- Bn[,-1]

# Check if all the data is in numeric format.
is.numeric(Bn_2)
# [1] FALSE
# Not all the data is numeric so must convert the dataset into numerical form.
Bn_numeric <- as.matrix(sapply(Bn_2,as.numeric))
# Check all data is now numeric and the above manipulation worked.
is.numeric(Bn_numeric)
```

```
# [1] TRUE
# The data is now all numeric.
# New dataset has no rownames so add row names to numeric matrix from previous
# dataset.
row.names(Bn_numeric) <- row.names(Bn_2)


# Visualise all the genes in a histogram to check and compare expression levels
# of genes and compare them.
# Open PDF writer to export as PDF.
pdf("Histogram of Gene expression.pdf")
# plot the histogram
hist(Bn_numeric, main= "Histogram of Bn expression levels", col= "grey55",
    xlab = "Expression levels")
# stop writing PDF
dev.off()
# Many genes with low levels of expression so filter to remove low expression level
# genes. Keep genes with mean expression values equal to or greater than 1.
# Create a parameter with the rowmeans in which will be used to filter.
rowmeans <- rowMeans(Bn_numeric)
# Filter out any with a rowmean of less than 1.
Bn_filtered <- subset(Bn_numeric,  rowmeans >= 1)
# Check results
dim(Bn_filtered)
# [1] 5251 101
dim(Bn)
# [1] 8015 102
```

```
# 2764 genes rows filtered out, the loss of a column is because of the removal of
# the gene row as it was made row name.


# Plot the mean RPKM for the whole dataset of lines to compare expression for
# each line. Expected to be similar. Differences may be interesting.
# Create a dataframe of the column means and the genes.
Bn_col_means <- data.frame(colMeans(Bn_filtered))
# Make the rownames (genes) a column in the table.
Bn_col_means$Gene <- rownames(Bn_col_means)
# Rename the column header of the first row.
colnames(Bn_col_means)[1] <- "colMeans"
# Use color ramp Pallette to create a scale colour for the graph. Begin by
# setting the colors for the graph in the parameter Pal.
pal<-colorRampPalette(c("grey80","grey40"))
# Give each mean a color in the table. Start by finding data length to
# split the scale by.
dim(Bn_col_means)
# [1] 101  3
# 101 long - give every point a colour value (range split 101 ways).
Bn_col_means$Col <- pal(101)
# Open PDF writer.
pdf("mean RPKM of whole dataset lines.pdf")
# plot the barplot
ggplot(Bn_col_means, aes(y=colMeans, x=Gene))+
  geom_bar(stat="identity", fill=Bn_col_means$Col, color= "grey31")+
  theme_classic()+
```

```
  xlab("Line")+
  ylab("RPKM")+
  theme(axis.text.x=element_blank(),axis.ticks.x=element_blank())+
  scale_y_continuous(expand = c(0,0))
# stop writing PDF
dev.off()

# To investigate whether expression corelates to glucosinolate data need to combine
# the datasets ready for investigation.
# Load in glucosinolate trait data.
Bn_gluc <- read.delim("http://www-
users.york.ac.uk/~ah1309/BigData/data/Glucosinolates.txt", header=T)
# Set line column as rowname.
rownames(Bn_gluc) <- Bn_gluc$Line
# Transpose gene expression dataset to match the orientation of the trait data.
Bn_t <- t(Bn_filtered)
# Merge the two datasets into one.
Bn_merge <- merge(Bn_gluc, Bn_t, by="row.names")
# Set gene name as row name
rownames(Bn_merge) <- Bn_merge$Line
# Remove duplicated gene name columns.
Bn_merge <- Bn_merge[,-c(1:2)]

##############################################################################
##########
#                    Section 3: Analysis of data                    #
```

```
##########################################################################
##########

# Test whether each genes expression values correlate to the trait data. To test
# for all genes a for loop will be used to run the test over the whole dataset.
# Begin by telling loop where to stop - the last value in the dataset.
numcol <- ncol(Bn_merge)
# Create a results table for for loop to feed into.
Bn_results <- as.data.frame(matrix(nrow = 0, ncol = 8))
# Run a for loop which runs an Anova test for a Lm for each genes expression
# correlated to glucosinolate content and write all data outputted into a table
# (Bn_results).
for (i in 2:numcol){
  lm1 <- lm(Bn_merge$Trait~Bn_merge[,i])
  anova <- as.data.frame(anova(lm1)[1,])
  intercept = as.data.frame(coefficients(lm1))[1,1]
  gradient = as.data.frame(coefficients(lm1))[2,1]
  R2 <- as.data.frame(summary(lm1)$r.squared)
  result1 <- as.data.frame(c(anova, R2, intercept, gradient))
  colnames(Bn_results) <- colnames(result1)
  Bn_results <-rbind(Bn_results, result1)
}
# change Bn_results rownames to gene names.
rownames(Bn_results) <- colnames(Bn_merge[,2:numcol])
# Label column names with what each column contains.
colnames(Bn_results) <- c("Df", "Sum.Sq", "Mean.Sq", "F.value", "P.value",
```

```
                    "R2", "Intercept", "Gradient")

# Determine which p-values are significantly different from the expected outcomes
# using Car to make a qqplot.
# Open PDF creater.
pdf("qqplot of data.pdf")
# Plot the QQ plot.
qqPlot(Bn_results$P.value, cex= 0.05, xlab="Norm quantiles", ylab = "P value")
# Close PDF plotter.
dev.off()

# Combine this dataset with the position data.
# Begin by installing the position data.
Bn_position <- read.delim("http://www-
users.york.ac.uk/~ah1309/BigData/data/osr_dir.txt",row.names=1)
# Merge the position data with the results data.
Bn_results_position <- merge(Bn_position,Bn_results, by="row.names")
# Make gene names column names.
colnames(Bn_results_position)[1] <- c("Gene")

# Apply False Discovery Rate multiple test correction. FDR used as it is less
# stringent so may reveal more interesting relationships with the glucosinolate
# trait.
# Begin by sorting the dataset so low P.values are at the top.
Bn_res_pos_sort <- Bn_results_position[order(Bn_results_position$P.value),]
# Create a parameter with rank of the p-value - smallest P value being 1.
```

```
R=nrow(Bn_res_pos_sort)
# Add a column to the dataset with the rank of that row in.
Bn_res_pos_sort$Rank <- 1:R
# Apply FDR multiple test correction and insert the new adjusted P value in a
# column - FDR.
Bn_res_pos_sort$FDR <- Bn_res_pos_sort$P.value*(R/Bn_res_pos_sort$Rank)

# Make q manhatten plot using the package qqman and using FDR that shows position
# along the chromosome and significance of relationship with glucosinolate trait.
# Genes which pass the significance threshold have a significant relationship.
# Open PDF writer.
pdf("manhatten of both chromosomes.pdf")
# plot the manhatten plot.
manhattan(Bn_res_pos_sort, chr="Graph", bp="Position", snp="Gene",p="FDR",
          ylim = c(0, 4), cex = 1.1, cex.axis = 0.9,
          col = c("grey55", "grey10"), suggestiveline = F,
          genomewideline = -log10(0.05),
          chrlabs = c("1", "2"))
#close PDF tool.
dev.off()

##############################################################################
##########
#                    Section 4: Genes of interest                    #
##############################################################################
##########
```

```
# Take the significantly associated genes and combine with the nucleotide sequences
# of these genes.

# Begin by setting strings as factors as false.
options(stringsAsFactos = FALSE)
# Read in the sequence data for the genes.
Bn_seqs <- read.delim("http://www-users.york.ac.uk/~ah1309/BigData/data/genes.txt")
# Make the genes the row names.
rownames(Bn_seqs) <-Bn_seqs$Gene
# Subset the significant genes.
Bn_sig <- subset(Bn_res_pos_sort$Gene, Bn_res_pos_sort$FDR<0.05)
# Select only significant sequences and put into a new dataset.
Bn_sig_genes<- Bn_seqs[Bn_sig,]
# Write table with genomic information for the significant genes.
write.table(Bn_sig_genes,"Significant genes.txt",quote = F, sep="\t",
            row.names = FALSE)

# Write a table containing all the information needed to further investigate the
# genes associated to glucosinolate content. Create a dataset of gene name, positon
# along chromosome,FDR and correlation between gene expression levels and
# glucosinolate content.
# Start by making genes row names
rownames(Bn_res_pos_sort)<- Bn_seqs$Gene
# Save only the columns wanted into a new dataset.
Bn_sig_info <- Bn_res_pos_sort[Bn_sig,c(1,3,11,13)]
```

```
# Write a table from the dataset.
write.table(Bn_sig_info, "Information on the signifiant genes.txt", quote=F,
            sep="\t", row.names = FALSE)

##############################################################################
##########
#                    Section 5: Genes linked to trait                    #
##############################################################################
##########

# Further investigation found that of the genes identified, some genes have a well
# known
# link with glucosinolates. This section will analyse these genes which are known
# to have a link.

# Subset the linked genes. For accuracy  and due to difficulties with R this is
# done by manually finding each gene in the already sorted dataset and subsetting.
Bn_link <- Bn_res_pos_sort[c(1,2,10,16,22),]
# Highlight the genes with an already known link to glucosinolate content on the
# manhatten plot. Do this by highlighting the previously made subset.
# Open PDF viewer.
pdf("Manhatten plot with highlight.pdf")
# Plot the manhatten.
manhattan(Bn_res_pos_sort, chr="Graph", bp="Position",
          snp="Gene",p="FDR", ylim = c(0, 4),
          cex = 1.1, cex.axis = 0.9, col = c("grey55", "grey10"),
```

```
        suggestiveline = F,genomewideline = -log10(0.05),
        xlab = "Position on Chromosome",
        highlight = Bn_link$Gene)
# Close PDF writer.
dev.off()

# Create a dataframe of the genes found to be linked  which is already a dataset
# and save as a table.
# Make the dataframe.
Bn_link_frame <- data.frame(Bn_link)
# Write the dataframe into a table.
write.table(Bn_link_frame,"Interesting genes.txt",quote = F, sep="\t",
          row.names = FALSE)

# Create a graphs investigating the correlation between expression levels and
# glucosinolate content for each of the genes found to have a link. This is to
# visualise and better understand the # relationship between the Gene and the
# glucosinolate content.
# Create a subset of the linked genes.
Bn_merge_link_col <- subset.data.frame(Bn_merge, select =
                          (c("Trait","A_JCVI_40613","A_JCVI_16890",
                            "A_JCVI_5227","A_JCVI_31290","A_JCVI_33047")))
# Give each Gene a color in the table, again using the color range earlier made.
# Start by finding data length.
dim(Bn_merge_link_col)
# [1] 53  6
```

```
# 53 long - give every point a colour value (range split 53 ways).
Bn_merge_link_col$Col <- pal(53)
# Define where the for loop will stop
n_col <- ncol(Bn_merge_link_col)
# Run a for loop to create and save the graphs as PDFs, each with unique names.
for (i in 2:(n_col-1)){
  pdf(paste("RPKMof",Bn_link$Gene[(i-1)],".pdf", sep=""))
  print(ggplot(Bn_merge_link_col, aes(Bn_merge_link_col[,i],Trait))+
    geom_point(col=Bn_merge_link_col$Col)+
    stat_smooth(method = "lm", se= FALSE, col="grey10", size=1)+
    theme_classic()+
    ylab("Glucosinolates (%)")+
    xlab(paste("RPKM of", Bn_link$Gene[(i-1)])))
  dev.off()
}

# Plot the gradient and the position on the chromosome to attempt to identify any
# clusters in the genome and whether the clusters tend to contain positively
# affecting genes or negatively affecting genes.
# Subset all the information needed from the significant genes.
Bn_sig_full <- subset(Bn_res_pos_sort, Bn_res_pos_sort$FDR<0.05)
# Open PDF saver.
pdf("Possible clusters in the genome.pdf")
# Plot the significant genes in black and the Significant genes with known link in
# green and highlight possible clustered regions. Horizontal line makes
# distinguishing positive and negative easier.
```

```
ggplot(Bn_sig_full, aes(Position, Gradient))+
  geom_point(data = Bn_sig_full, aes( col="Significant gene")) +
  annotate("segment", x=0, xend=8000, y=0, yend=0)+
  geom_point(data =Bn_link,
             aes(col="Significant gene with known glucosinolate link"))+
  annotate("rect", xmin = 860, xmax = 1205, ymin = -9.5, ymax = 36, alpha = .3,
           fill ="grey40")+
  annotate("text", x=750, y=39, label="Possible clusters", size=3.2)+
  theme_classic()+
  scale_color_manual(values=c("Significant gene"="black",
                    "Significant gene with known glucosinolate link"="green"))+
  theme(legend.position = c(0.65,0.8),
        legend.background = element_rect(linetype = 1, size = 0.5, colour = 1))+
  annotate("rect", xmin = 200, xmax = 1300, ymin = -10.5,
           ymax = 37, alpha = .3, fill ="grey50")+
  xlab("Position along chromosome")+
  ylab("Gradient of correlation line")
# Close the PDF writer.
dev.off()


# Write a table containing just the smaller hypothetical cluster for durther
# analysis.Begin by removing all other genes too large.
Bn_semi_clust <- subset(Bn_sig_full,Bn_sig_full$Position < 1205)
# Then filter out the genes too small.
Bn_clust <- subset(Bn_semi_clust$Gene, Bn_semi_clust$Position > 860)
# Combine with sequence information for the genes in this region.
```

```
Bn_clust_seq<- Bn_seqs[Bn_clust,]
# Write the cluster and sequence dataset into a table.
write.table(Bn_clust_seq,"Possible cluster.txt",quote = F, sep="\t",
            row.names = FALSE)

# Plot the mean RPKM for each significant gene with a known link so as to compare
# expression
# levels between these genes and identify if any are significantly different.
# Create a dataframe of the column means and the genes of these significant genes
# with a known
# link. Begin by again subsetting the genes with a link.
Bn_merge_link_2 <- subset.data.frame(Bn_merge, select =
                                  (c("Trait","A_JCVI_40613","A_JCVI_16890",
                                    "A_JCVI_5227","A_JCVI_31290",
                                    "A_JCVI_33047")))
# Remove the Trait data as not needed.
Bn_link_col_means <- data.frame(colMeans(Bn_merge_link_2[,-1]))
# Make a column of all the gene names and read this information in.
Bn_link_col_means$Gene <- rownames(Bn_link_col_means)
# Rename the first column to make data analysis easier.
colnames(Bn_link_col_means)[1] <- "colMeans"
# Give each mean a color in the table. Need to know dimentions first.
dim(Bn_link_col_means)
# [1]  5  2
# 5 rows in the table so split scale by 6 and give every mean a color.
Bn_link_col_means$Col <- pal(5)
```

```
# Open PDF writer.
pdf("mean RPKM of linked genes.pdf")
# Plot the barplot.
ggplot(Bn_link_col_means, aes(y=colMeans, x=Gene))+
  geom_bar(stat="identity", fill=Bn_link_col_means$Col, color= "grey31")+
  theme_classic()+
  xlab("Gene")+
  ylab("RPKM")+
  scale_y_continuous(expand = c(0,0))
# Stop writing PDF.
dev.off()

##############################################################################
##########
#                    Section 6: Statistical tests                    #
##############################################################################
##########

# Test if the significant genes with a known link have significantly different RPKM

# create a dataset ready to be made tidy, remove glucosinolate data. Data needs
# tidying so statistical tests can be carried out on it.
# Start by subsetting the useful genes into a fresh dataset.
Bn_merge_link <- subset.data.frame(Bn_merge, select =
                                   (c("Trait","A_JCVI_40613","A_JCVI_16890",
                                     "A_JCVI_5227","A_JCVI_31290",
```

```
                                    "A_JCVI_33047")))
Bn_tidy_prep <- Bn_merge_link[,-1]
# Create a row with the line information.
Bn_tidy_prep$Line <- rownames(Bn_tidy_prep)
# Make the dataset tidy - columns of gene and line with the expression data sorted
# into these columns. Tidr package used to do this.
Bn_tidy <- gather(Bn_tidy_prep,key = Gene, value = Expression, -Line)
# Remove the Line data as not needed.
Bn_tidy <- Bn_tidy[,-1]


# Test if the dataset is normal.
tapply(Bn_tidy$Expression,Bn_tidy$Gene, shapiro.test)
# Only 1 gene was normal so test non-parametrically.
# Apply a kruskal-wallis test to the dataset.
kruskal.test(Bn_tidy$Gene,Bn_tidy$Expression)
# Kruskal-Wallis rank sum test
# data:  Bn_tidy$Gene and Bn_tidy$Expression
# Kruskal-Wallis chi-squared = 262.11, df = 245, p-value = 0.2161
# no signicant affect of gene on expression levels, post hoc test to see if any
# specific genes had significant differences. Use pgirmess package for this.
kruskalmc( Bn_tidy$Expression,Bn_tidy$Gene, probs =0.05)
# The significant differences are:
# A_JCVI_16890-A_JCVI_31290, A_JCVI_16890-A_JCVI_40613,
# A_JCVI_16890-A_JCVI_5227, A_JCVI_31290-A_JCVI_33047,
# A_JCVI_31290-A_JCVI_40613,A_JCVI_31290-A_JCVI_5227,
# A_JCVI_33047-A_JCVI_5227
```

```
# Plot graph showing which genes are significantly different.
# Open PDF writer.
pdf("mean RPKM of linked genes with significance labelled.pdf")
# Plot the barplot with NS labelled.
ggplot(Bn_link_col_means, aes(y=colMeans, x=Gene))+
  geom_bar(stat="identity", fill=Bn_link_col_means$Col, color= "grey31")+
  theme_classic()+
  xlab("Gene")+
  ylab("RPKM")+
  scale_y_continuous(expand = c(0,0))+
  annotate("segment", x=1, xend=2, y = 19, yend = 19)+
  annotate("segment", x=2, xend=2, y = 19, yend = 18.5)+
  annotate("segment", x=1, xend=1, y = 19, yend = 18.5)+
  annotate("text", x=1.5, y = 19.2, label ="*", size=5)+
  annotate("segment", x=2, xend=3, y = 20, yend = 20)+
  annotate("segment", x=3, xend=3, y = 19.5, yend = 20)+
  annotate("segment", x=2, xend=2, y = 19.5, yend = 20)+
  annotate("text", x=2.5, y = 20.2, label ="*", size=5)+
  annotate("segment", x=2, xend=4, y = 21, yend = 21)+
  annotate("segment", x=4, xend=4, y = 20.5, yend = 21)+
  annotate("segment", x=2, xend=2, y = 20.5, yend = 21)+
  annotate("text", x=3, y = 21.2, label ="*", size=5)+
  annotate("segment", x=2, xend=5, y = 22, yend = 22)+
  annotate("segment", x=5, xend=5, y = 21.5, yend = 22)+
  annotate("segment", x=2, xend=2, y = 21.5, yend = 22)+
```

```
  annotate("text", x=3.5, y = 22.2, label ="*", size=5)

# stop writing PDF

dev.off()
```