Unsupervised Learning Techniques In Modularizing Gene Regulatory Networks

Richard Chen

May 8, 2016

0. Dependencies

```
load('FinalProjectData.RData')
library("ggplot2")
```

```
## Warning: package 'ggplot2' was built under R version 3.2.4
```

```
source("http://bioconductor.org/biocLite.R")
library("pracma"); library("MASS"); library("mclust"); library("igraph"); library
("modMax"); library("mixer")
library("ALL"); library("hgu95av2.db"); library("GO.db"); library("annotate"); lib
rary("genefilter")
library("GOstats"); library("RColorBrewer"); library("xtable"); library("Rgraphvi
z"); library("org.EcK12.eg.db")
library("AnnotationForge"); library("glasso"); library("mixer")
p <- function(x,p=5) {x[1:p,1:p]}
options(stringsAsFactors=FALSE)
```

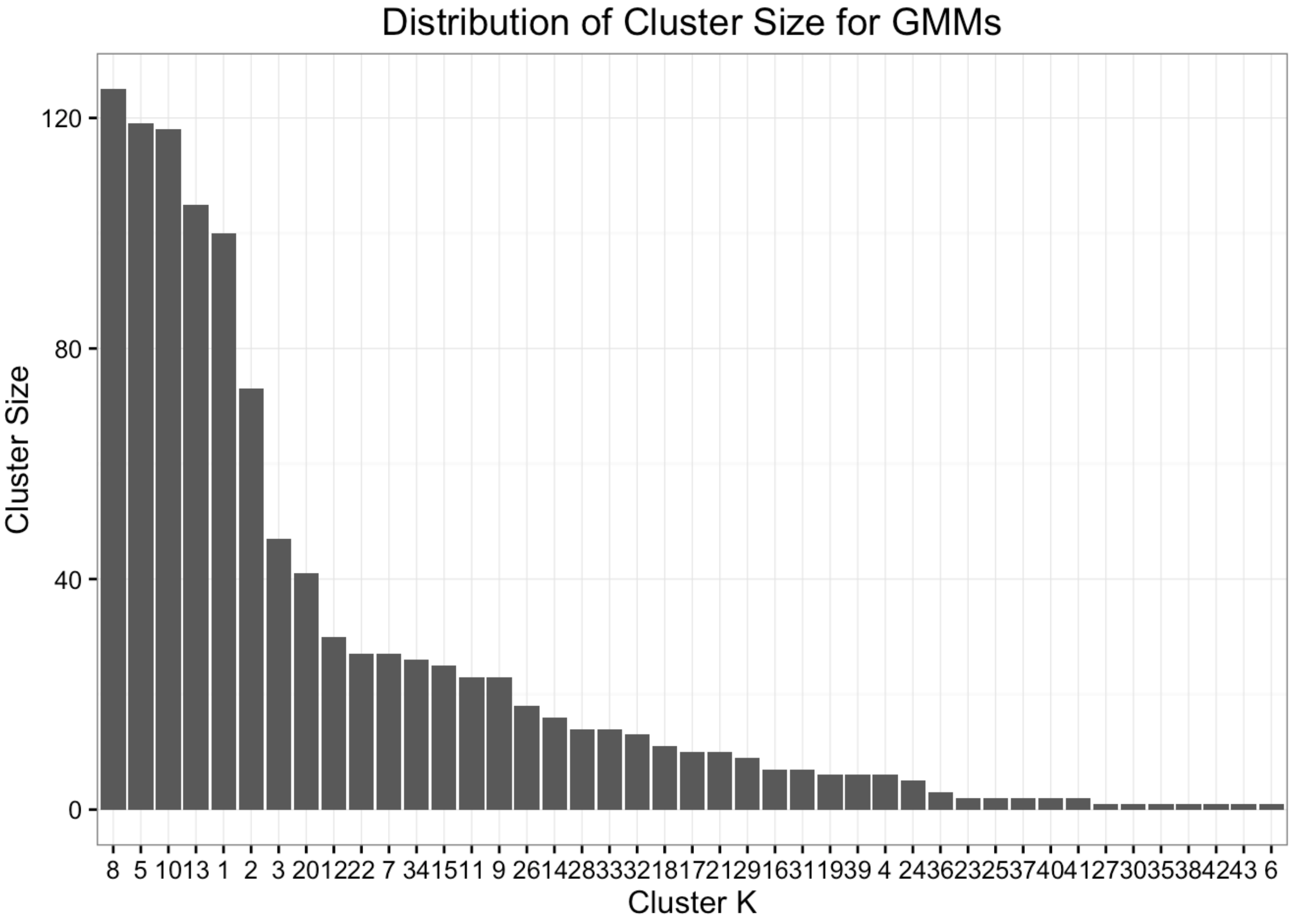
1. Data Preprocessing

```
data3 <- t(read.table("net3_expression_data.tsv", header = TRUE, sep = "\t"))
dict3 <- read.table("net3_gene_ids.tsv", header = FALSE, sep = "\t")
gold3 <- read.table("DREAM5_NetworkInference_GoldStandard_Network3.tsv")
gold3 <- gold3[gold3[,3] == 1,][,c(1,2)]
rownames(dict3) <- dict3[,1]
genes_selected <- dict3[unique(union(gold3[,1], gold3[,2])),]
rownames(data3) <- dict3[,2]
data3 <- data3[genes_selected[,2],]
write.csv(data3, "data3.csv")
```

2. GMM Clustering by EM

```
data3_BIC <- mclustBIC(data = data3, G=10:100)
p1 <- qplot(10:100, data3_BIC[,1], ylab = "Bayesian Information Criterion (BIC)",
xlab = "Mixture Components", main = "BICs for Parameterized GMMs fitted by EM") +
theme_bw()
GMM_optimal <- as.numeric(which(max(data3_BIC[,1]) == data3_BIC[,1]))
p1 <- p1 + geom_point(x = GMM_optimal, y = max(data3_BIC[,1]), aes(colour = "re
d")) + theme(legend.position="none")
data3_BIC_optimal <- mclustBIC(data = data3, G=GMM_optimal, modelNames = "EEI")
GMM_model <- Mclust(data3, x = data3_BIC_optimal)
GMM_label_freq <- data.frame(x = names(sort(table(GMM_model$classification), decre
asing = TRUE)),
                             y = unname(sort(table(GMM_model$classification), decr
easing = TRUE)))
p2 <- ggplot(GMM_label_freq, aes(x = reorder(x,-y), y = y))
p2 <- p2 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p2 <- p2 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of Clu
ster Size for GMMs")
```

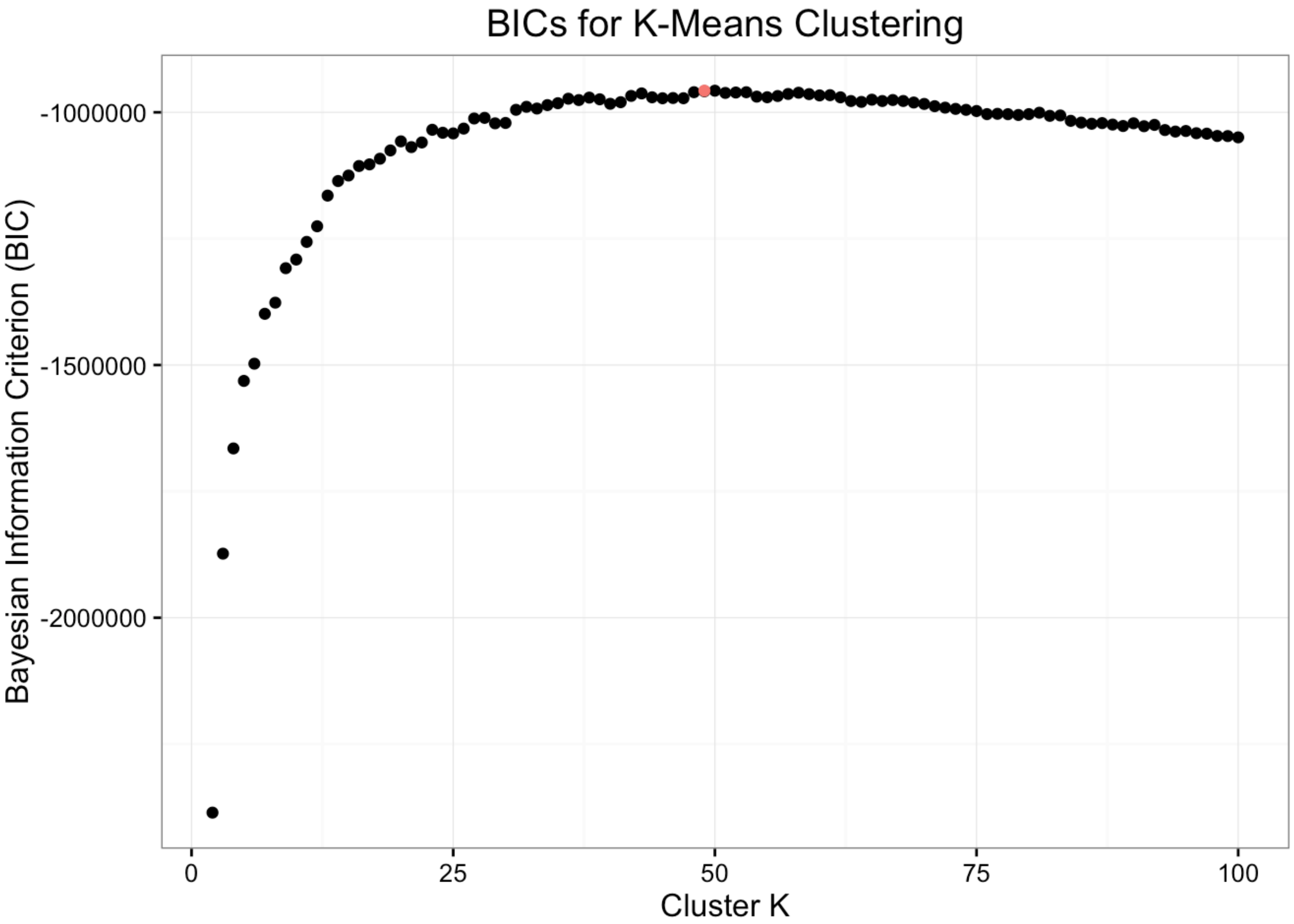
```
print(p2)
```



3. K-Means Clustering

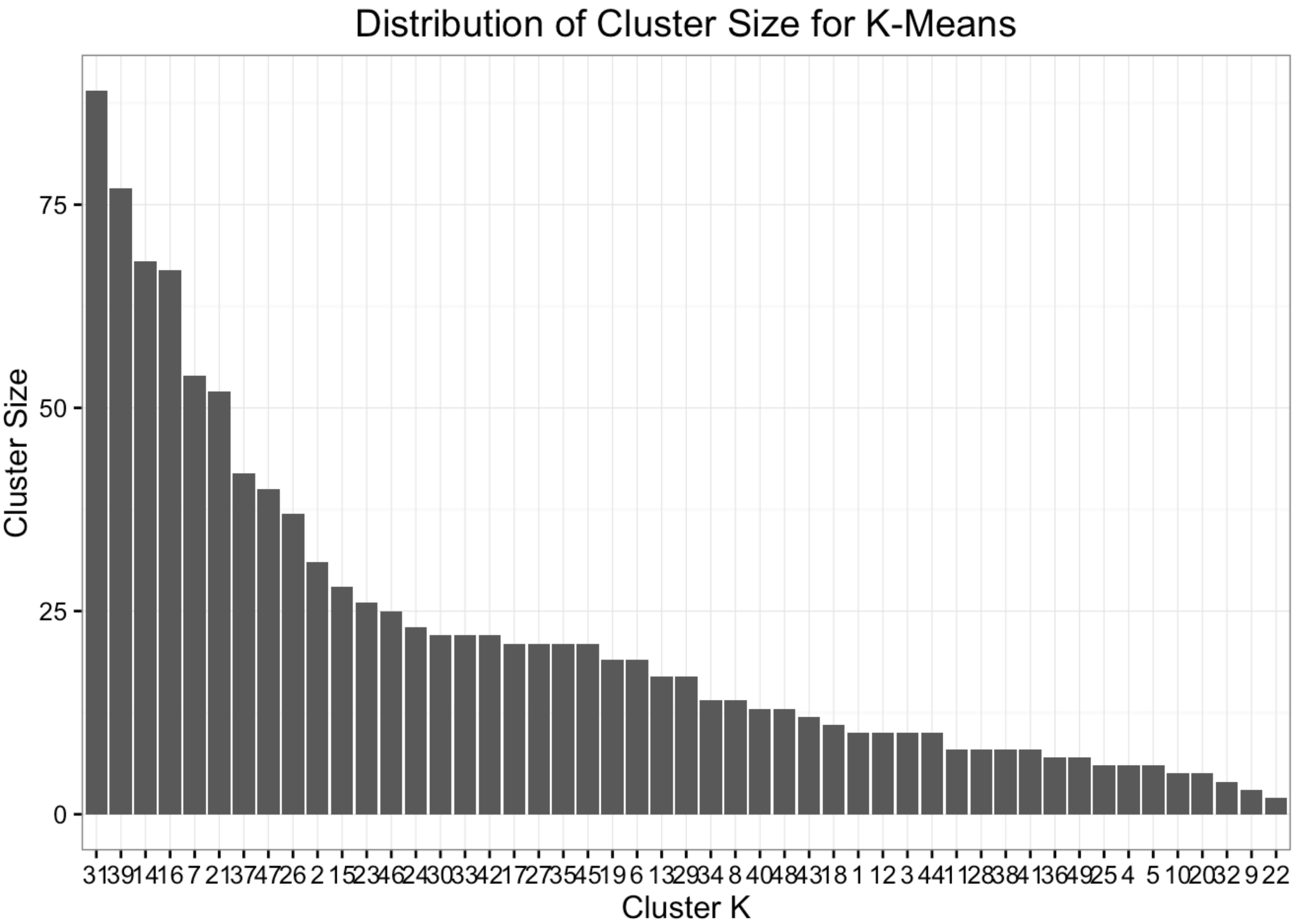
```
BIC <- function(modelFit, expr) {
  cluster_labels = modelFit$cluster
  expr_residuals = expr
  centers = modelFit$centers
  for (s in names(cluster_labels)) {
    curr_k <- as.numeric(cluster_labels[s])
    curr_center <- centers[curr_k,]
    expr_residuals[s,] = (expr_residuals[s,]-curr_center)^2
  }
  loglik_approx = sum(expr_residuals)
  k = length(levels(as.factor(cluster_labels)))
  d = length(colnames(expr))
  n = length(rownames(expr))
  -2*loglik_approx+k*d*log10(n)
}
BICrange_kmeans <- sapply(2:100, function(k) BIC(kmeans(data3, centers = data3[1:k,], iter.max = 20), data3))
p3 <- qplot(2:100, BICrange_kmeans, ylab = "Bayesian Information Criterion (BIC)", xlab = "Cluster K", main = "BICs for K-Means Clustering") + theme_bw()
kmeans_optimal <- which(max(BICrange_kmeans) == BICrange_kmeans)
p3 <- p3 + geom_point(x = kmeans_optimal, y = max(BICrange_kmeans), aes(colour = "red")) + theme(legend.position="none")
```

```
print(p3)
```



```
kmeans_model <- kmeans(data3, centers = data3[1:kmeans_optimal,], iter.max = 20)
kmeans_label_freq <- data.frame(x = names(sort(table(kmeans_model$cluster), decreasing = TRUE)),
                                y = unname(sort(table(kmeans_model$cluster), decreasing = TRUE)))
p4 <- ggplot(kmeans_label_freq, aes(x = reorder(x,-y), y = y))
p4 <- p4 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p4 <- p4 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of Cluster Size for K-Means")
```

```
print(p4)
```

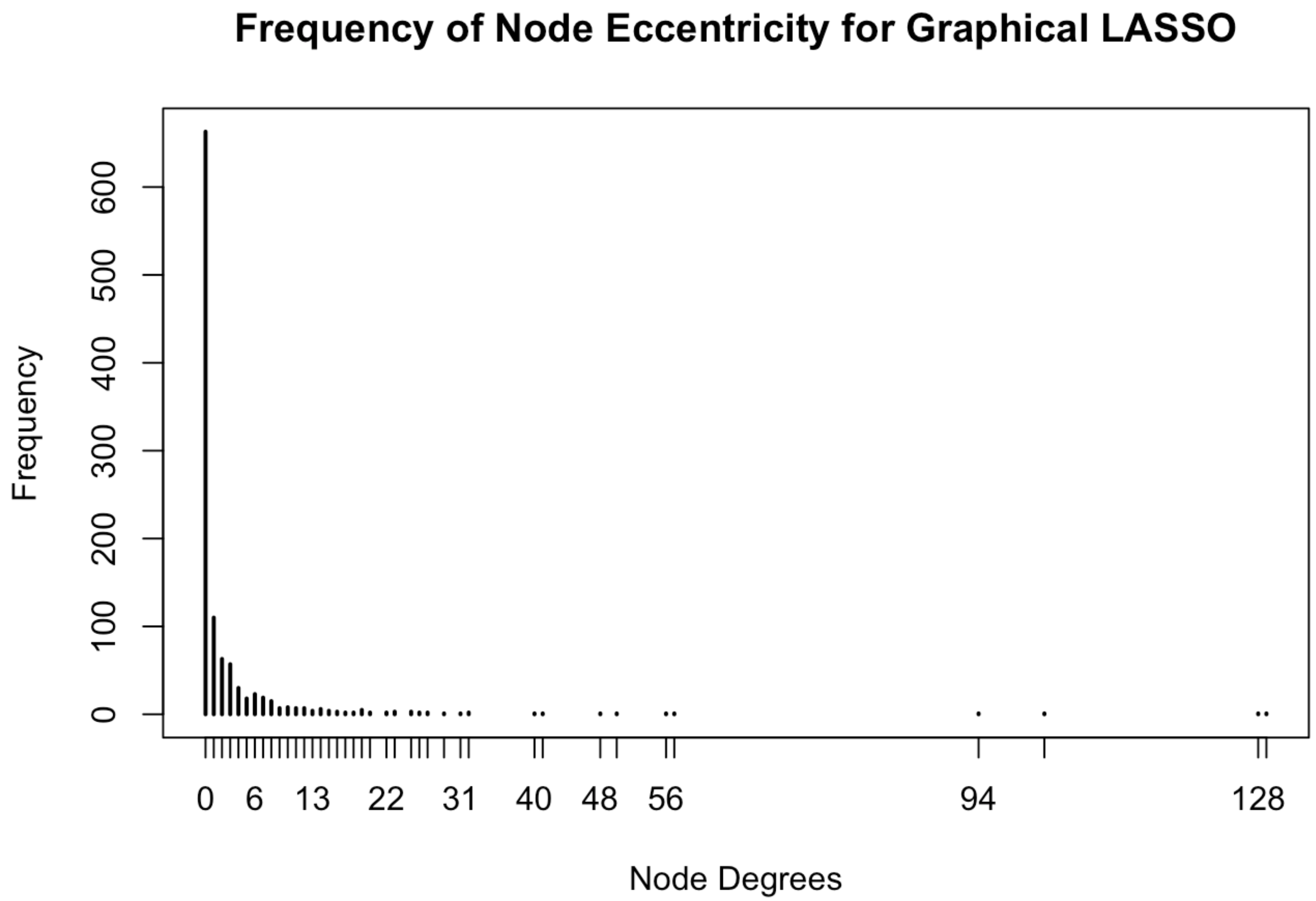


4. GLASSO

4.1 GLASSO Construction

```
data3_cov <- cov(t(data3))
glasso_model <- glasso(data3_cov, rho = 0.50, thr = 10e-4, penalize.diagonal=FALS
E)
glasso_model_Q <- glasso_model$wi
diag(glasso_model_Q) <- 0
glasso_model_Q[glasso_model_Q != 0] <- 1
glasso_net <- graph.adjacency(data.matrix(glasso_model_Q), mode = "undirected", di
ag = FALSE)
degrees_glasso <- table(as.factor(degree(glasso_net)))
```

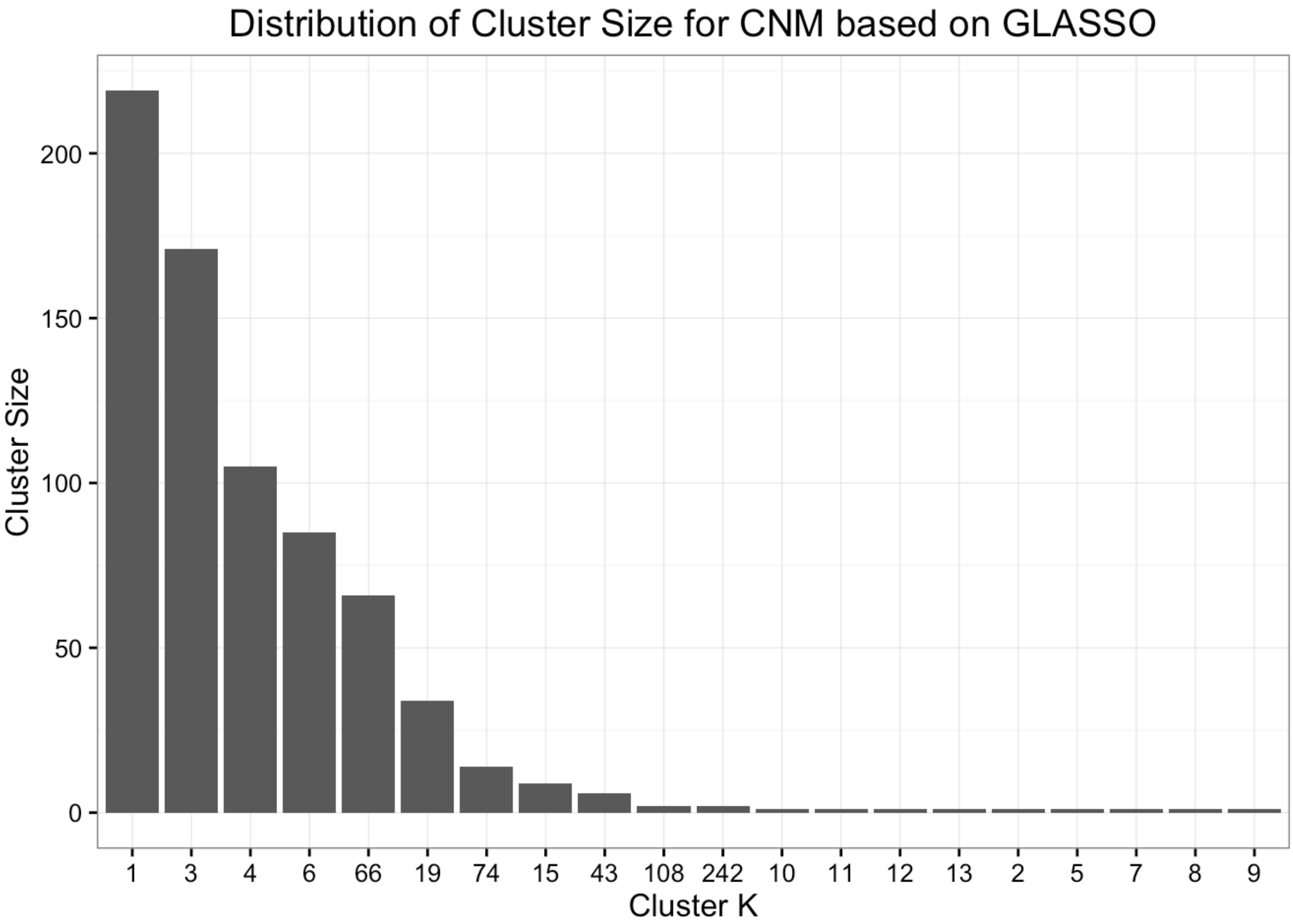
```
plot(degrees_glasso, xlab = "Node Degrees", ylab = "Frequency", main = "Frequency
of Node Eccentricity for Graphical LASSO")
```



4.2 GLASSO-based CNM

```
glasso_CNM <- greedy(glasso_model_Q)
print(glasso_CNM$`number of communities`)
glasso_CNM_label_freq <- data.frame(x = names(sort(table(glasso_CNM$`community str
ucture`), decreasing = TRUE)),
                                    y = unname(sort(table(glasso_CNM$`community st
ructure`), decreasing = TRUE)))
p6 <- ggplot(head(glasso_CNM_label_freq,20), aes(x = reorder(x,-y), y = y))
p6 <- p6 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p6 <- p6 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of Clu
ster Size for CNM based on GLASSO")
```

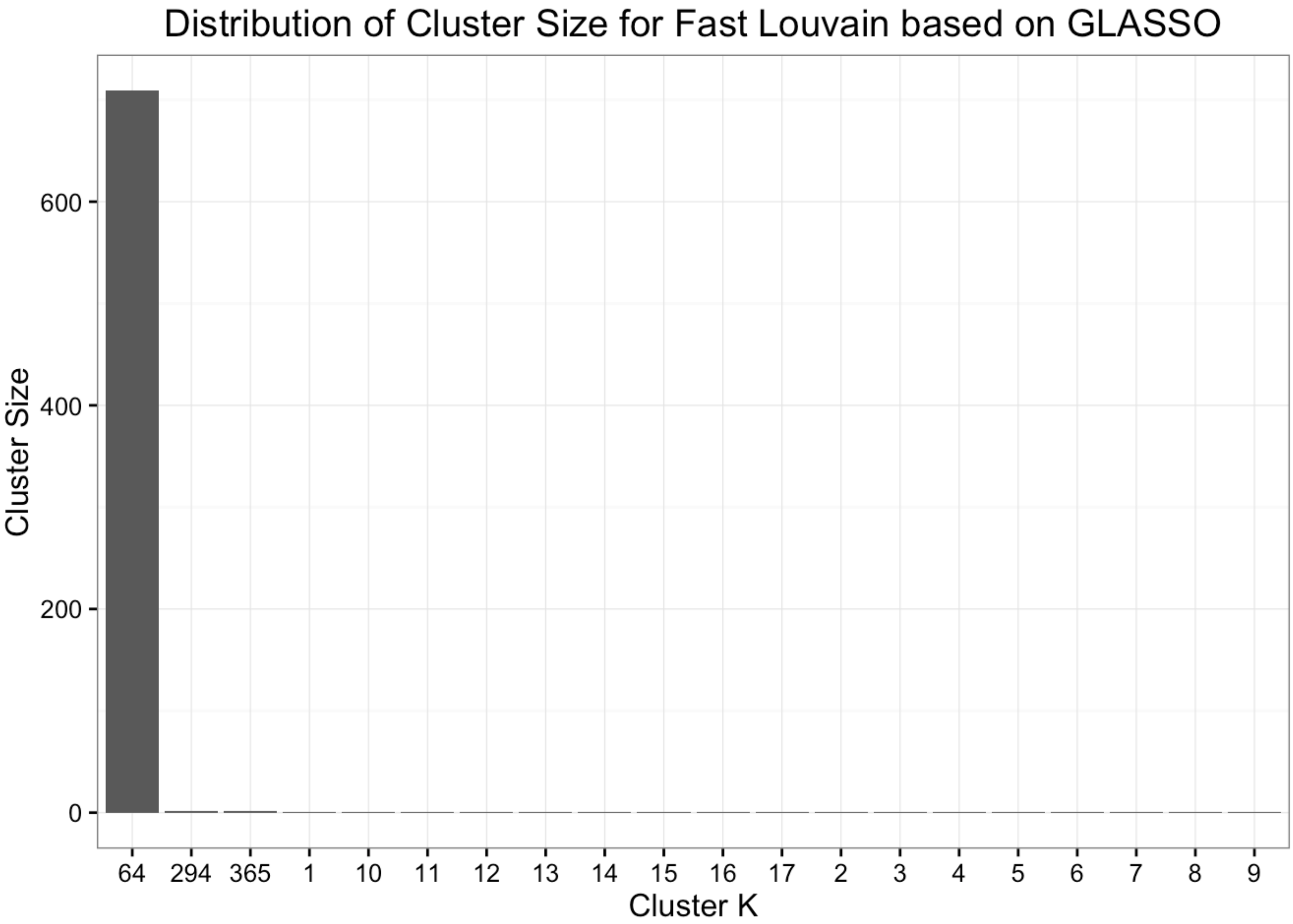
```
p6
```



4.3 GLASSO-based Fast Louvain

```
glasso_louvain <- louvain(glasso_model_Q)
print(glasso_louvain$`number of communities`)
glasso_louvain_label_freq <- data.frame(x = names(sort(table(glasso_louvain$`commu
nity structure`), decreasing = TRUE)),
                                        y = unname(sort(table(glasso_louvain$`comm
unity structure`), decreasing = TRUE)))
p7 <- ggplot(head(glasso_louvain_label_freq,20), aes(x = reorder(x,-y), y = y))
p7 <- p7 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p7 <- p7 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of Clu
ster Size for Fast Louvain based on GLASSO")
```

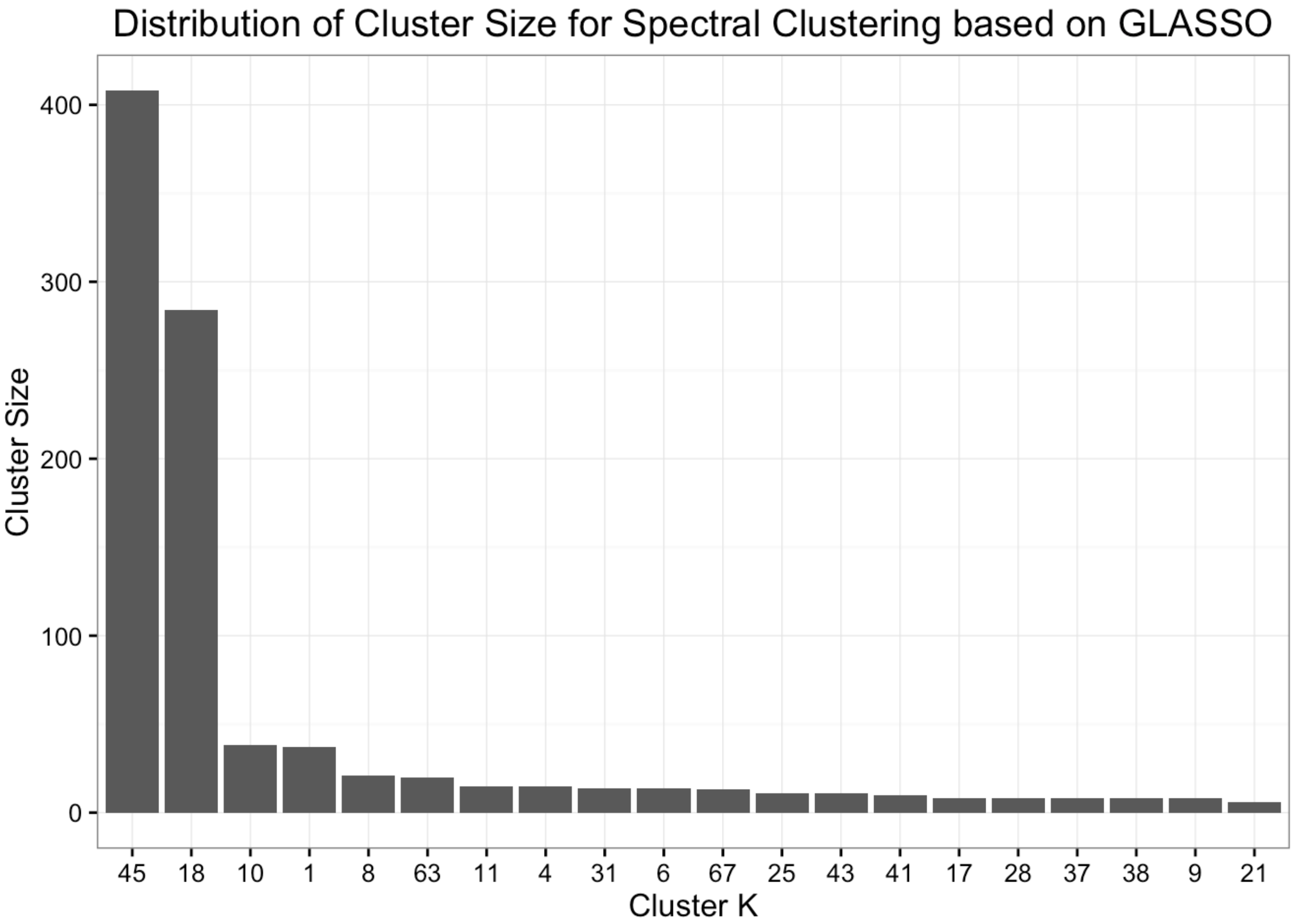
```
p7
```



4.4 GLASSO-based Spectral Clustering

```
glasso_spectral <- spectralOptimization(glasso_model_Q)
print(glasso_spectral$`number of communities`)
glasso_spectral_label_freq <- data.frame(x = names(sort(table(glasso_spectral$`com
munity structure`), decreasing = TRUE)),
                                         y = unname(sort(table(glasso_spectral$`co
mmunity structure`), decreasing = TRUE)))
p8 <- ggplot(head(glasso_spectral_label_freq,20), aes(x = reorder(x,-y), y = y))
p8 <- p8 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p8 <- p8 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of Clu
ster Size for Spectral Clustering based on GLASSO")
```

```
p8
```

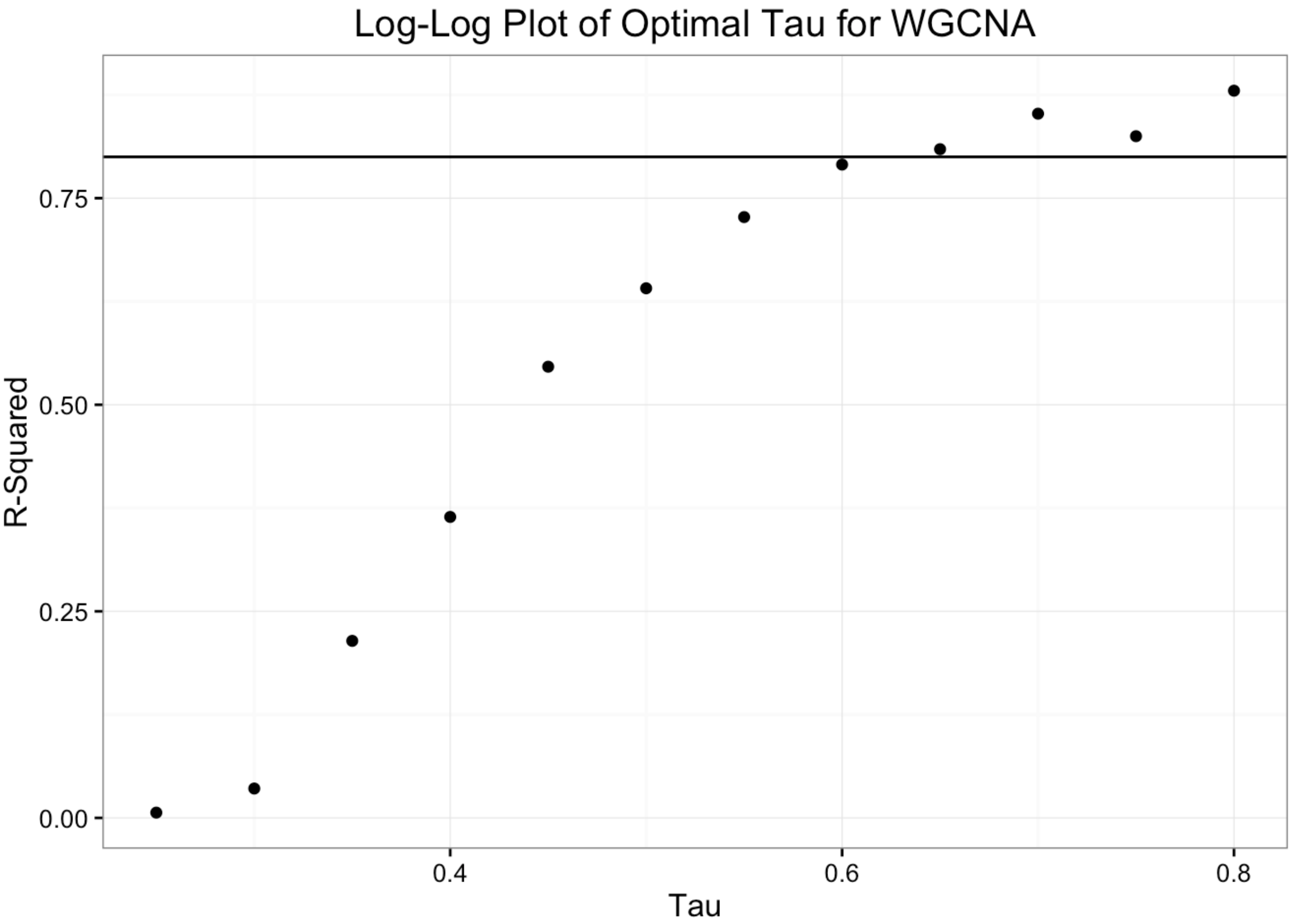


5. WGCNA

5.1 WGCNA Construction

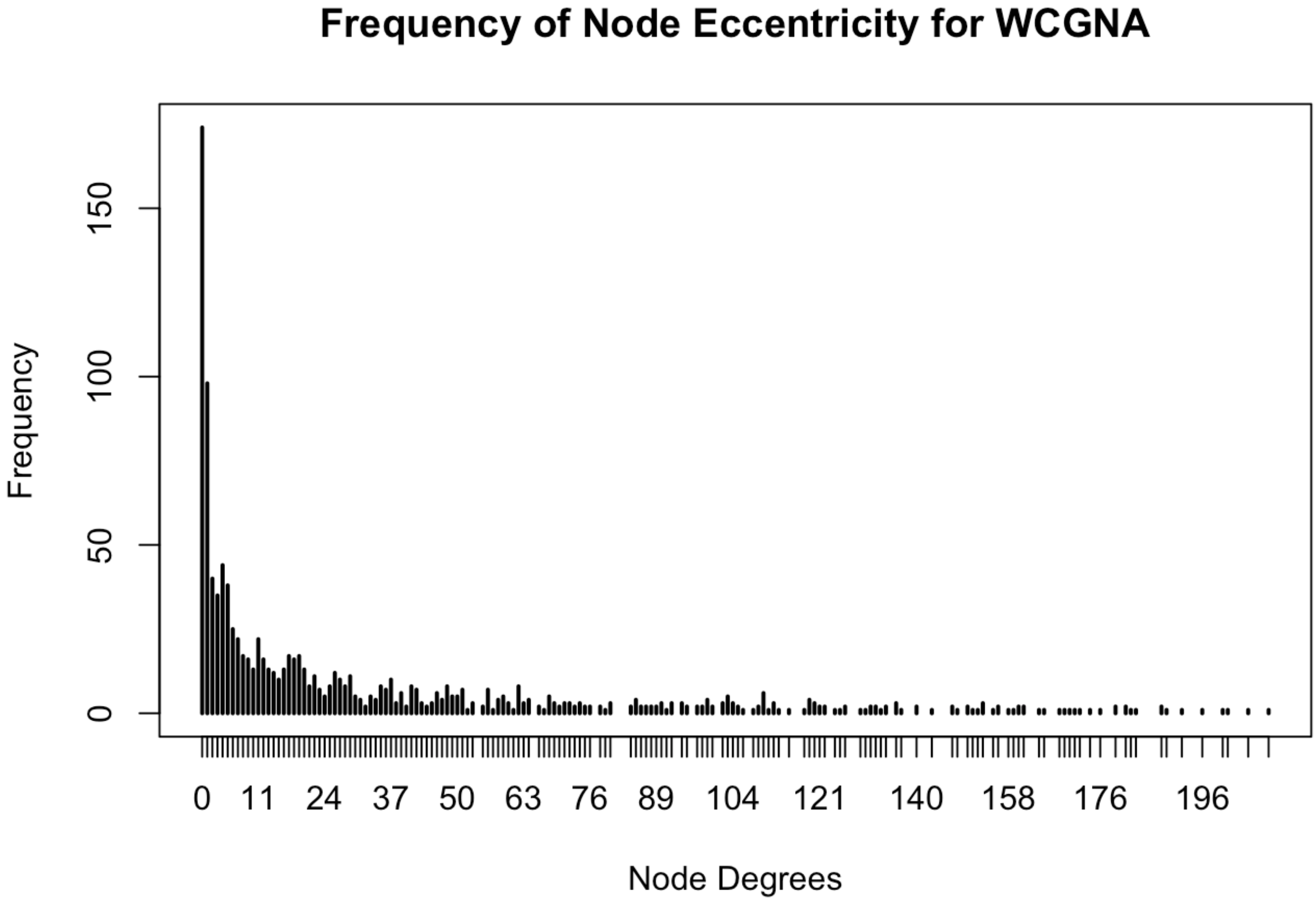
```
data3_cor <- cor(t(data3))
logPlot <- function(adjDF, tau) {
  adjDF[adjDF >= tau] = 1
  adjDF[adjDF < tau] = 0
  g = graph.adjacency(data.matrix(adjDF))
  degrees <- table(as.factor(degree(g)))
  summarylm = summary(lm(log(as.numeric(names(degrees))) ~ log(as.numeric(degrees/
1000))))
  summarylm$r.squared
}

Rsquareds <- sapply(seq(0.25,0.80, 0.05), function(tau) logPlot(data3_cor, tau))
p9 <- qplot(seq(0.25,0.80, 0.05), Rsquareds, xlab = "Tau", ylab = "R-Squared", mai
n = "Log-Log Plot of Optimal Tau for WGCNA") + theme_bw()
p9 <- p9 + geom_hline(yintercept = 0.80)
```

```
p9
```



```
tau_s <- 0.65
adjDF_0.65 <- abs(data3_cor)
adjDF_0.65[adjDF_0.65 >= tau_s] <- 1
adjDF_0.65[adjDF_0.65 < tau_s] <- 0
diag(adjDF_0.65) <- 0
wgcna_net_0.65 <- graph.adjacency(data.matrix(adjDF_0.65), mode = "undirected", di
ag = FALSE)
degrees_wgcna <- table(as.factor(degree(wgcna_net_0.65)))
```

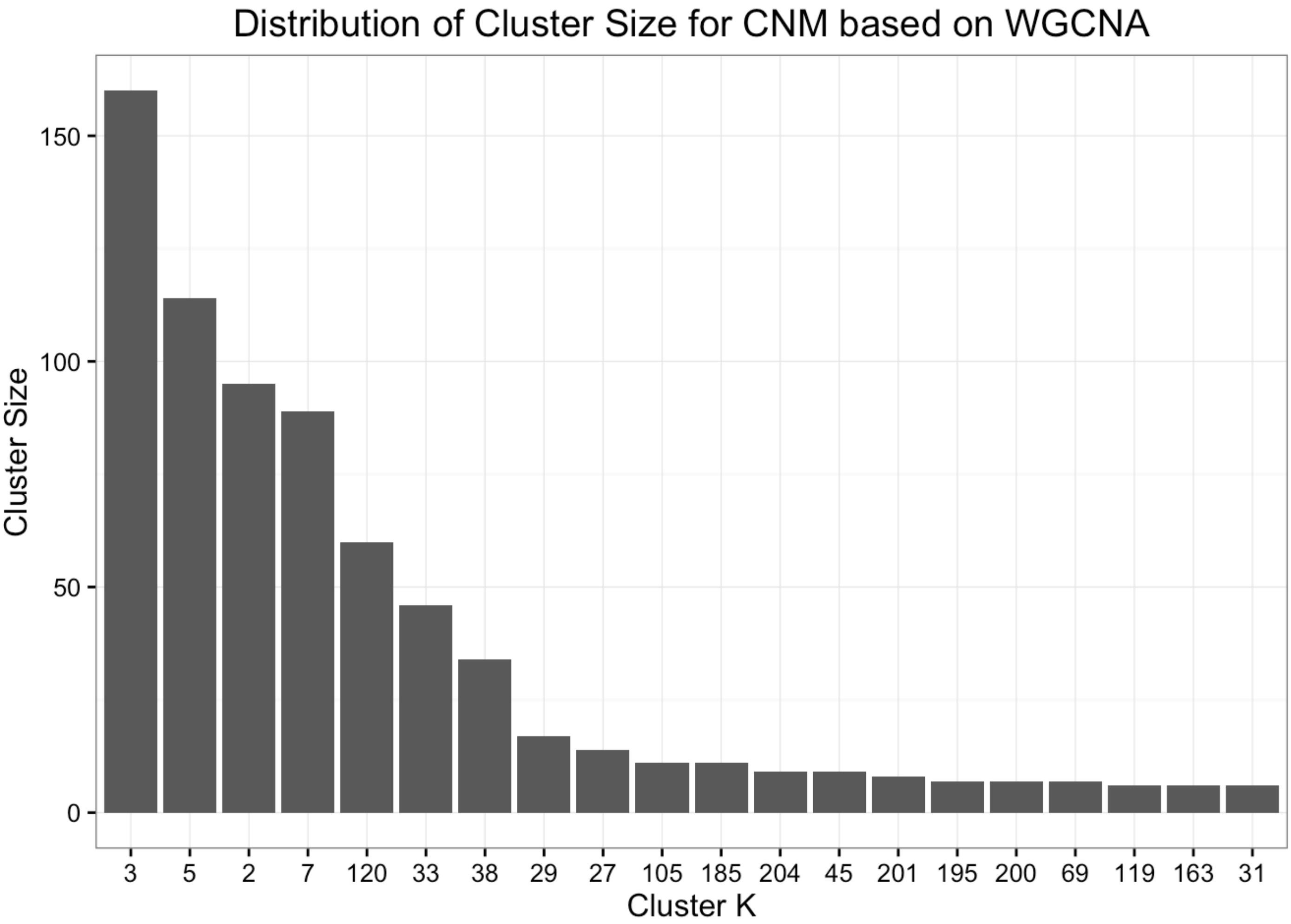
```
plot(degrees_wgcna, xlab = "Node Degrees", ylab = "Frequency", main = "Frequency o
f Node Eccentricity for WCGNA")
```



5.2 WGCNA-based CNM

```
WGCNA_CNM <- greedy(adjDF_0.65)
print(WGCNA_CNM$`number of communities`)
WGCNA_CNM_label_freq <- data.frame(x = names(sort(table(WGCNA_CNM$`community struc
ture`), decreasing = TRUE)),
                                    y = unname(sort(table(WGCNA_CNM$`community str
ucture`), decreasing = TRUE)))
p11 <- ggplot(head(WGCNA_CNM_label_freq,20), aes(x = reorder(x,-y), y = y))
p11 <- p11 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p11 <- p11 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of C
luster Size for CNM based on WGCNA")
```

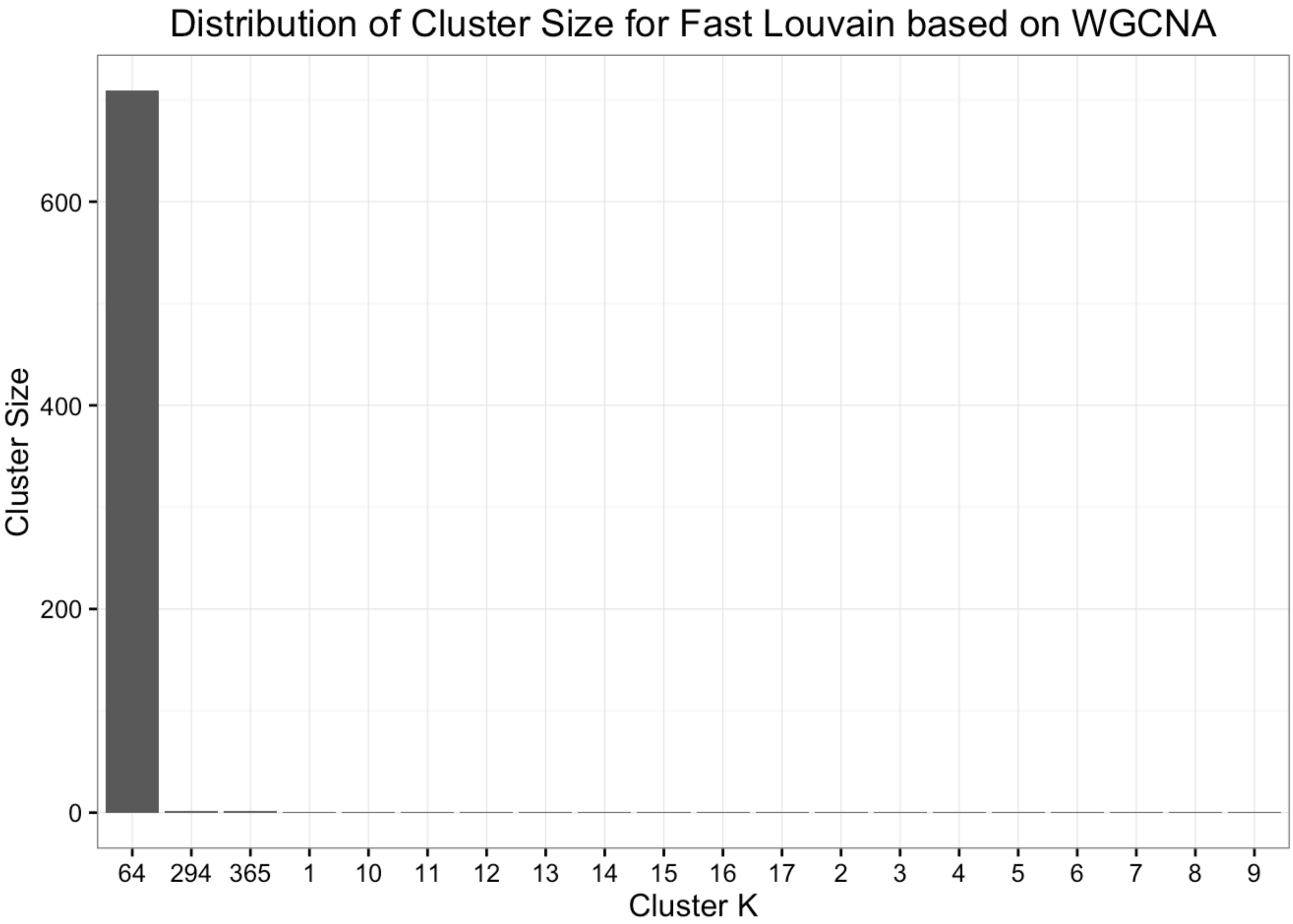
```
p11
```



5.2 WGCNA-based Fast Louvain

```
WGCNA_louvain <- louvain(adjDF_0.65)
print(WGCNA_louvain$`number of communities`)
WGCNA_louvain_label_freq <- data.frame(x = names(sort(table(WGCNA_louvain$`communi
ty structure`), decreasing = TRUE)),
                                        y = unname(sort(table(WGCNA_louvain$`commu
nity structure`), decreasing = TRUE)))
p12 <- ggplot(head(glasso_louvain_label_freq,20), aes(x = reorder(x,-y), y = y))
p12 <- p12 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p12 <- p12 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of C
luster Size for Fast Louvain based on WGCNA")
```

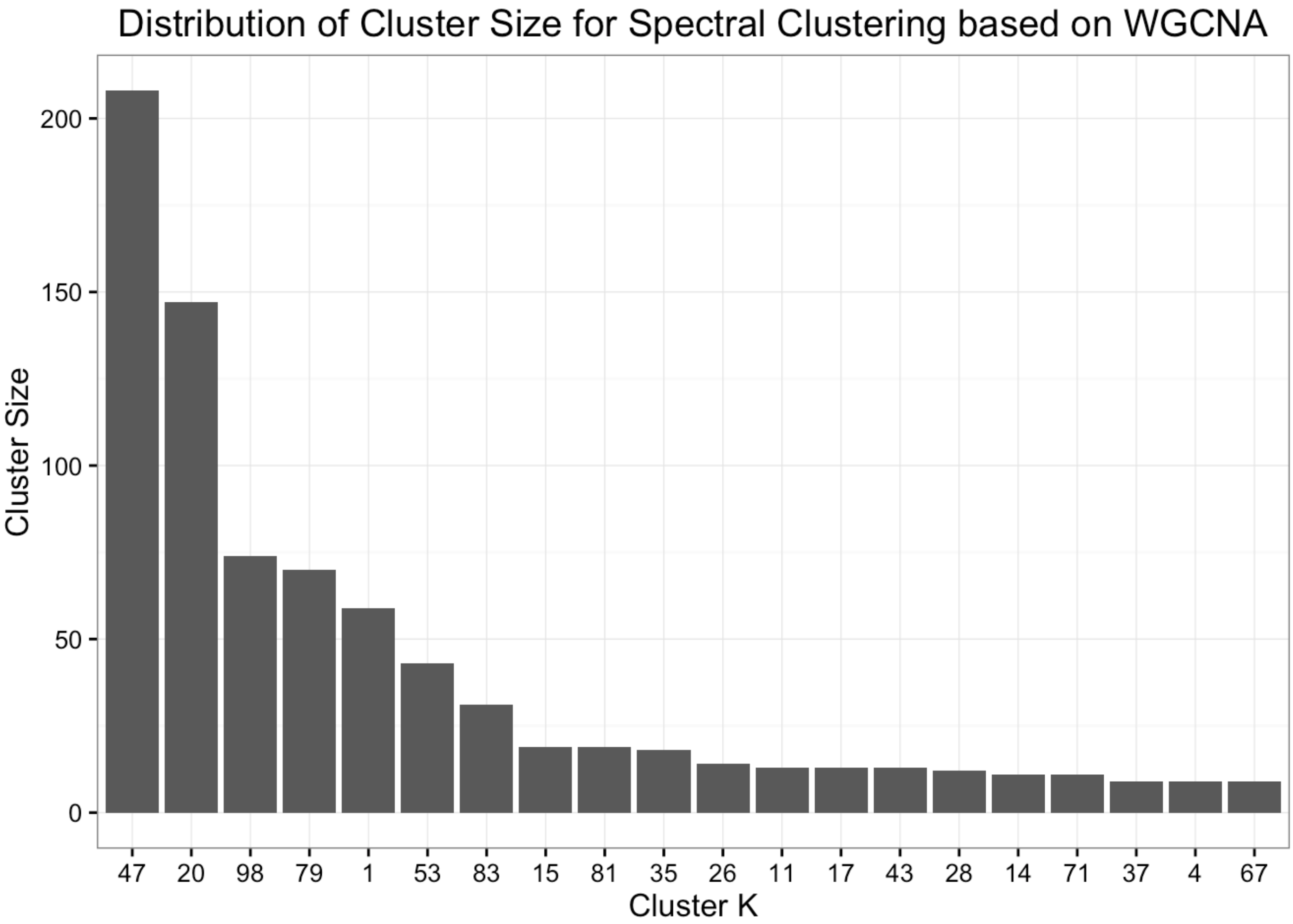
```
p12
```



5.3 WGCNA-based Spectral Clustering

```
WGCNA_spectral <- spectralOptimization(adjDF_0.65)
print(WGCNA_spectral$`number of communities`)
WGCNA_spectral_label_freq <- data.frame(x = names(sort(table(WGCNA_spectral$`commu
nity structure`), decreasing = TRUE)),
                                        y = unname(sort(table(WGCNA_spectral$`com
munity structure`), decreasing = TRUE)))
p13 <- ggplot(head(WGCNA_spectral_label_freq,20), aes(x = reorder(x,-y), y = y))
p13 <- p13 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p13 <- p13 + xlab("Cluster K") + ylab("Cluster Size") + ggtitle("Distribution of C
luster Size for Spectral Clustering based on WGCNA")
```

```
p13
```

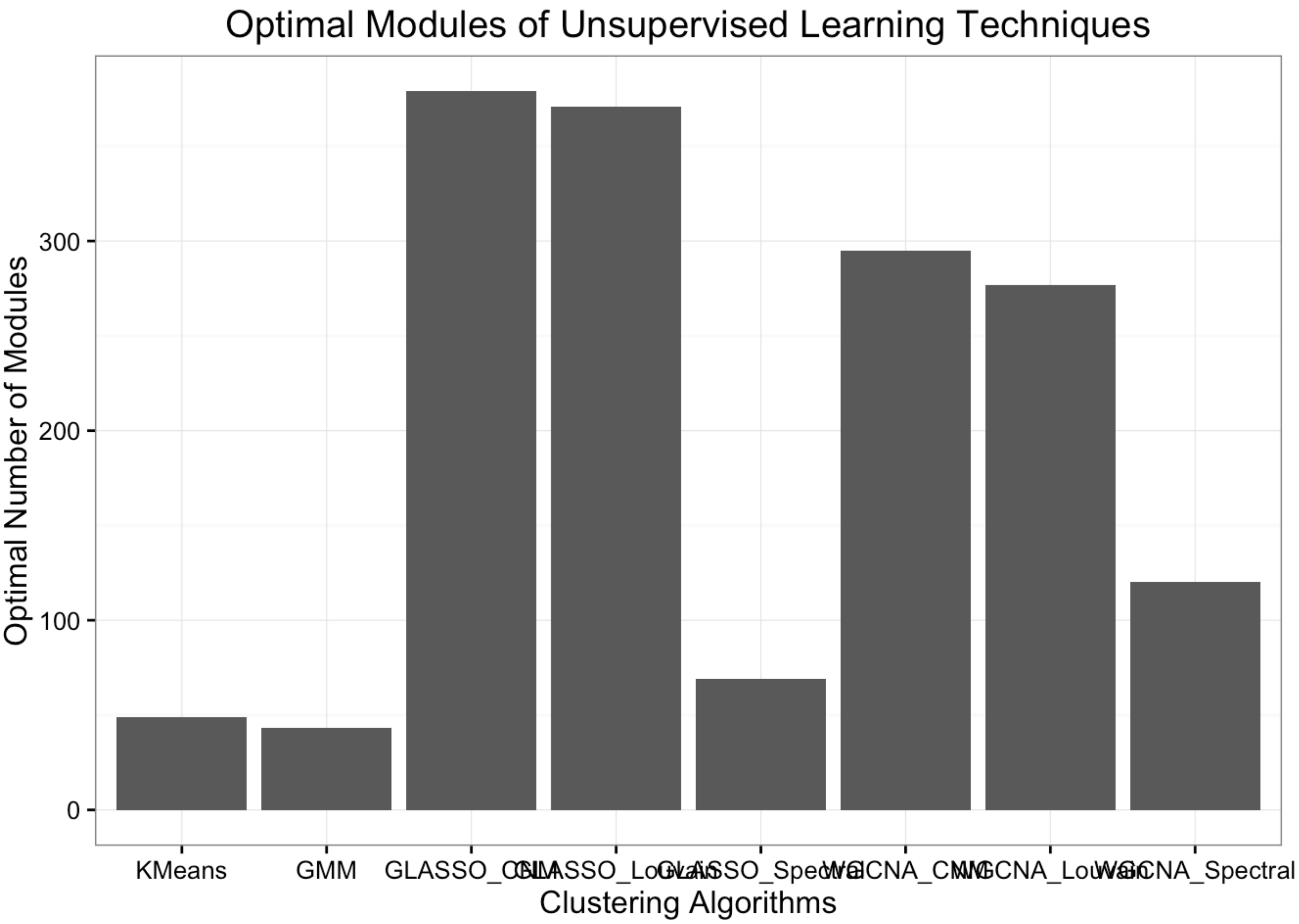


6 Analysis

6.1 Optimal Modules/# of Communities

```
Optimal_Modules <- c(kmeans_optimal,
                     GMM_optimal,
                     glasso_CNM$`number of communities`,
                     glasso_louvain$`number of communities`,
                     glasso_spectral$`number of communities`,
                     WGCNA_CNM$`number of communities`,
                     WGCNA_louvain$`number of communities`,
                     WGCNA_spectral$`number of communities`)
names(Optimal_Modules) <- c("KMeans", "GMM",
                            "GLASSO_CNM", "GLASSO_Louvain", "GLASSO_Spectral",
                            "WGCNA_CNM", "WGCNA_Louvain", "WGCNA_Spectral")
Optimal_Modules_DF <- data.frame(x = names(Optimal_Modules),
                                 y = unname(Optimal_Modules))
Optimal_Modules_DF$x <- factor(Optimal_Modules_DF$x, levels = Optimal_Modules_DF
$x)
p14 <- ggplot(Optimal_Modules_DF, aes(x = x, y = y))
p14 <- p14 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p14 <- p14 + xlab("Clustering Algorithms") + ylab("Optimal Number of Modules") + g
gtitle("Optimal Modules of Unsupervised Learning Techniques")
```

```
p14
```

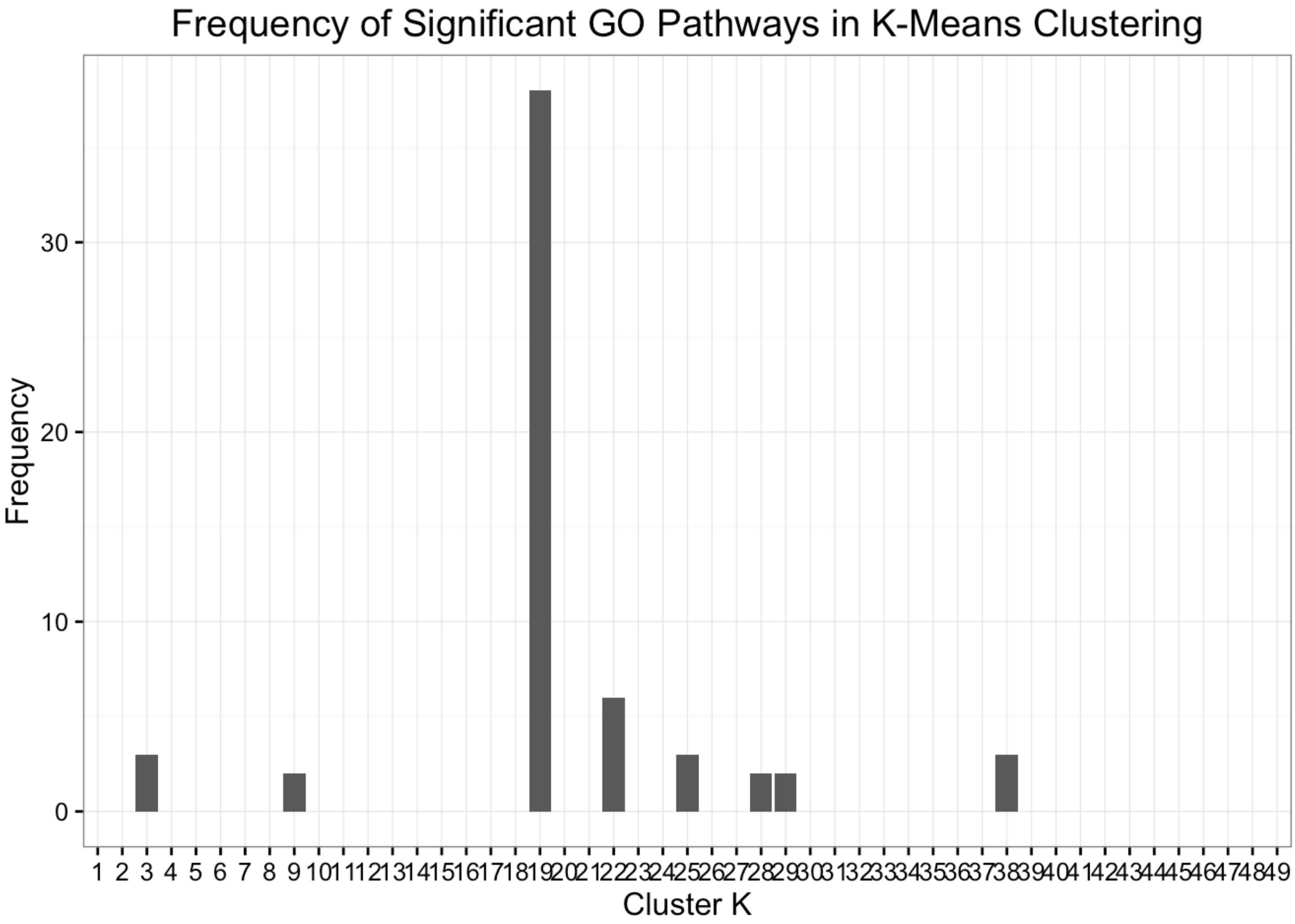


6.2 Functional Coherence

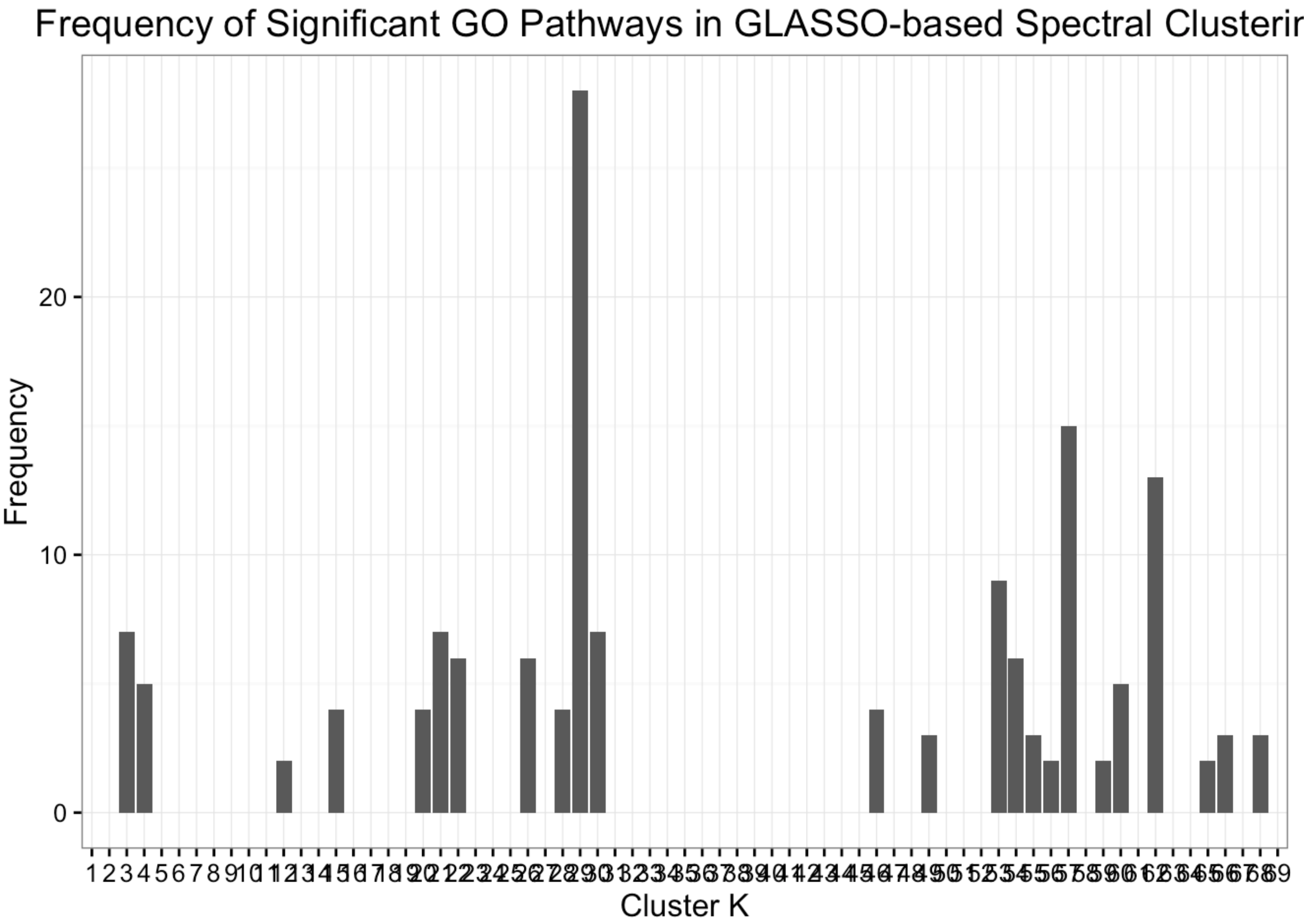
```
ClusterResults <- data.frame(KMeans = kmeans_model$cluster,
                             GMM = GMM_model$classification,
                             GLASSO_CNM = glasso_CNM$`community structure`,
                             GLASSO_Louvain = glasso_louvain$`community structure
`,
                             GLASSO_Spectral = glasso_spectral$`community structur
e`,
                             WGCNA_CNM = WGCNA_CNM$`community structure`,
                             WGCNA_Louvain = WGCNA_louvain$`community structure`,
                             WGCNA_Spectral = WGCNA_spectral$`community structure
`)
ecoli_genes <- rownames(data3)
mapped_genes_ecoli <- mappedkeys(org.EcK12.egSYMBOL2EG)
gene2entrez_ecoli <- as.list(org.EcK12.egSYMBOL2EG[mapped_genes_ecoli])
ClusterResults <- data.frame(CNM = glasso_CNM$`community structure`,
                             Louvain = glasso_louvain$`community structure`,
                             Spectral = glasso_spectral$`community structure`)
ecoli_genes <- rownames(data3)
ecoli_mapping <- gene2entrez_ecoli[ecoli_genes]
ecoli_mapping <- data.frame("Symbol" = as.character(names(ecoli_mapping)),
                       "Entrez" = as.character(unname(ecoli_mapping)))
unmapped_indices <- which(is.na(ecoli_mapping[,1]))
ClusterResults <- ClusterResults[-unmapped_indices,]
ecoli_mapping <- ecoli_mapping[-unmapped_indices,]
ecoli_mapping$Entrez <- as.numeric(ecoli_mapping$Entrez)
rownames(ClusterResults) <- ecoli_mapping$Entrez

features_OI <- c("KMeans", "GMM", "GLASSO_Spectral", "WGCNA_Spectral")
numsignificant <- sapply(features_OI, function(foi)
  sapply(seq(1,length(levels(as.factor(ClusterResults[,foi])))), function(coi) Hyp
erGeometricTest(foi, ClusterResults[,foi], coi))
)

significantModules_kmean <- data.frame(x = seq(1,length(levels(as.factor(ClusterRe
sults[,"KMeans"])))),
                                       y = numsignificant$KMeans)
significantModules_kmean$x <- factor(significantModules_kmean$x, levels = signific
antModules_kmean$x)
p15 <- ggplot(significantModules_kmean, aes(x = x, y = y))
p15 <- p15 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p15 <- p15 + xlab("Cluster K") + ylab("Frequency") + ggtitle("Frequency of Signifi
cant GO Pathways in K-Means Clustering")

significantModules_GMM <- data.frame(x = seq(1,length(levels(as.factor(ClusterResu
lts[,"GMM"])))),
                                       y = numsignificant$GMM)
significantModules_GMM$x <- factor(significantModules_GMM$x, levels = significantM
odules_GMM$x)
p16 <- ggplot(significantModules_GMM, aes(x = x, y = y))
p16 <- p16 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
```

```r
p16 <- p16 + xlab("Cluster K") + ylab("Frequency") + ggtitle("Frequency of Significant GO Pathways in GMM")

significantModules_GLASSO_Spectral <- data.frame(x = seq(1,length(levels(as.factor(ClusterResults[,"GLASSO_Spectral"])))),
                                                 y = numsignificant$GLASSO_Spectral)
significantModules_GLASSO_Spectral$x <- factor(significantModules_GLASSO_Spectral$x, levels = significantModules_GLASSO_Spectral$x)
p17 <- ggplot(significantModules_GLASSO_Spectral, aes(x = x, y = y))
p17 <- p17 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p17 <- p17 + xlab("Cluster K") + ylab("Frequency") + ggtitle("Frequency of Significant GO Pathways in GLASSO-based Spectral Clustering")

significantModules_WGCNA_Spectral <- data.frame(x = seq(1,length(levels(as.factor(ClusterResults[,"WGCNA_Spectral"])))),
                                                y = numsignificant$WGCNA_Spectral)
significantModules_WGCNA_Spectral$x <- factor(significantModules_WGCNA_Spectral$x, levels = significantModules_WGCNA_Spectral$x)
p18 <- ggplot(significantModules_WGCNA_Spectral, aes(x = x, y = y))
p18 <- p18 + geom_bar(stat = "identity", position = 'dodge') + theme_bw()
p18 <- p18 + xlab("Cluster K") + ylab("Frequency") + ggtitle("Frequency of Significant GO Pathways in WGCNA-based Spectral Clustering")


HyperGeometricTest <- function(foi, foi_labels, coi) {
  goi <- ecoli_mapping$Entrez[which(foi_labels == coi)]
  print(coi)
  if (length(goi) != 1) {
    params <- new("GOHyperGParams",
                  geneIds=goi,
                  universeGeneIds=ecoli_mapping$Entrez,
                  annotation="org.EcK12.eg.db",
                  ontology="BP",
                  pvalueCutoff=0.05/length(goi),
                  conditional=FALSE,
                  testDirection="over")
    mfhyper = tryCatch({
        hyperGTest(params)

      }, error = function(e) {
        NA
      }
    )

    if (!is.na(mfhyper)) {
      #write.csv(summary(mfhyper), paste(foi,"_",coi,".csv", sep = ""))
      return(nrow(summary(mfhyper)))
    } else {
      return(0)
    }
  } else {
    return(0)
  }
}
```

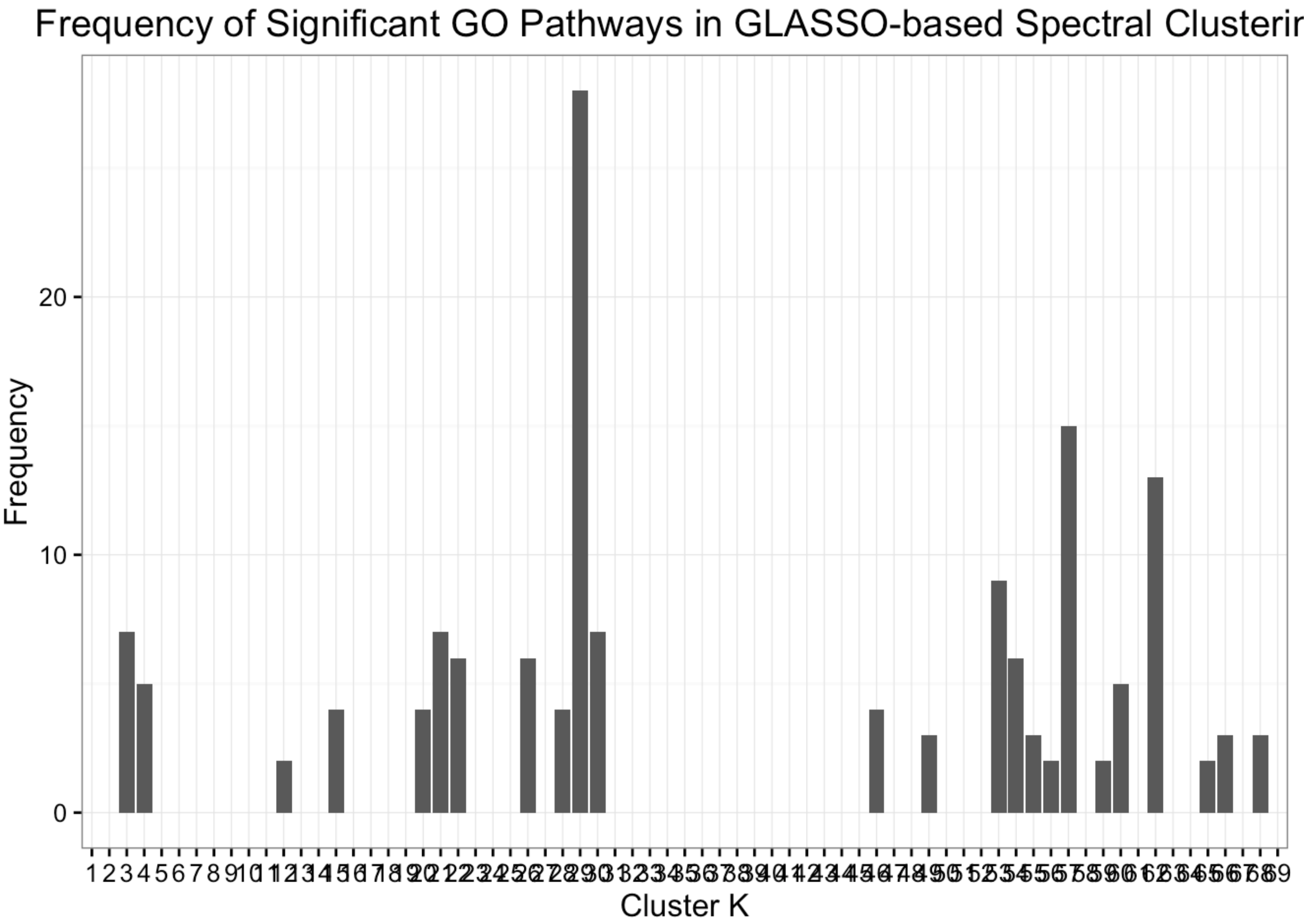
```
p15
```



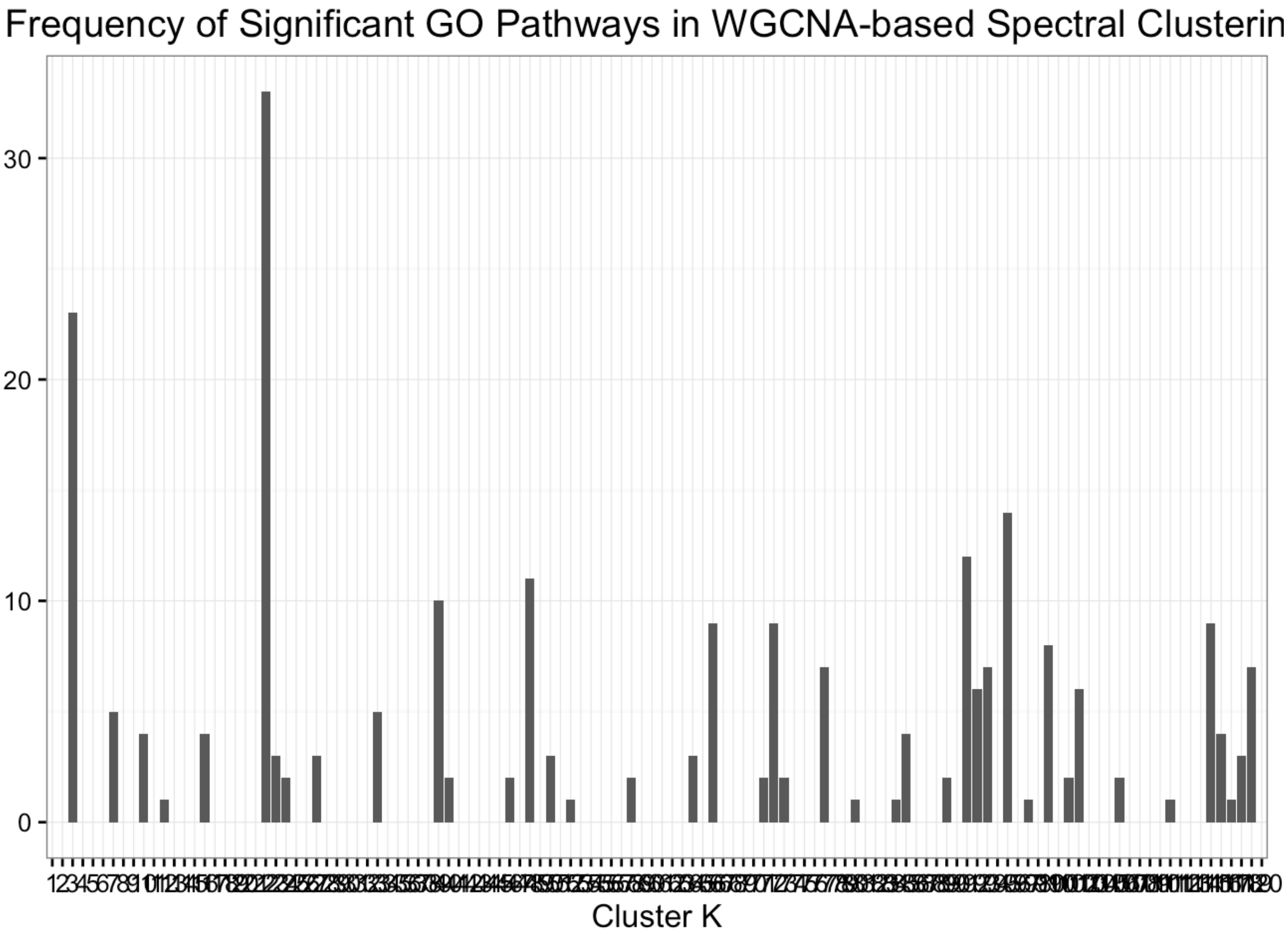
```
p16
```


Frequency of Significant GO Pathways in GLASSO-based Spectral Clusterir
Frequency
0
10
20
Cluster K
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

```
p17
```


Frequency of Significant GO Pathways in GLASSO-based Spectral Clusterir
Frequency
0
10
20
Cluster K
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69

```
p18
```


Frequency of Significant GO Pathways in WGCNA-based Spectral Clusterin
30
20
10
0
Frequency
Cluster K