Food Network Analysis

Arpita Jena, Ian Smeenk, Jason Carpenter, Sooraj Subrahamannian, Vinay Patlolla

Data description:

Our dataset is a list of embeddings for ingredients used in different foods. These embeddings are a high dimensional representation of ingredients that can be used to find ingredients with similar flavor properties. The dataset was created by generating a cosine similarity matrix of Jaan Altosaar's food2vec embeddings and then binarizing the similarities.

Vertices and edges:

The vertices represent particular foods (ex. onion, milk, sherry, etc.) and the edges represent whether or not there is a significant cosine similarity between their flavor embeddings. The cut off for significant similarity is set at 0.80.

Number of vertices: 2088
Number of edges: 94751

We opted to make our network unweighted and chose to render our network as an undirected network due to the fact that similarity is a reciprocal relationship.

```
food.adj <- as.matrix(mat)
food.adj2 <- food.adj

thresh <- 0.6
food.adj2[food.adj < thresh] <- 0
food.adj2[food.adj >= thresh] <- 1
diag(food.adj2) <- 0
```

Size of adjacency matrix: 33.5 MB
Size of edge list: 1.6 MB

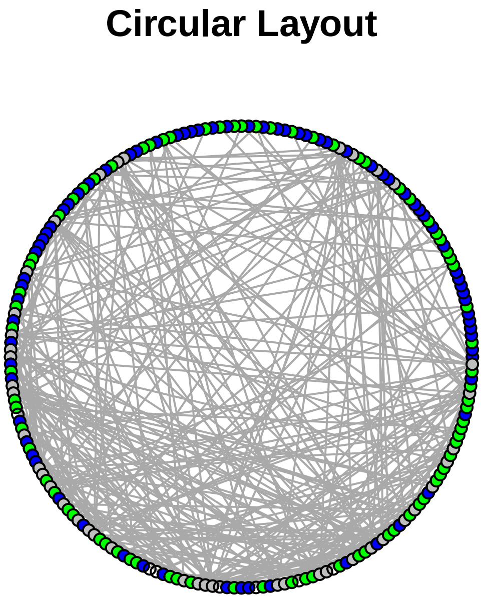
We would like to visualize similar/dissimilar groups of foods and try to categorize them according to their flavors.

```
food.igraph <- graph_from_adjacency_matrix(food.adj2, mode='undirected')

n1 <- sample(length(names), 200, replace = FALSE)
n2 <- sample(length(names), 200, replace = FALSE)
n3 <- sample(length(names), 200, replace = FALSE)

g1 <- induced_subgraph(food.igraph, n1)
g2 <- induced_subgraph(food.igraph, n2)
g3 <- induced_subgraph(food.igraph, n3)
```

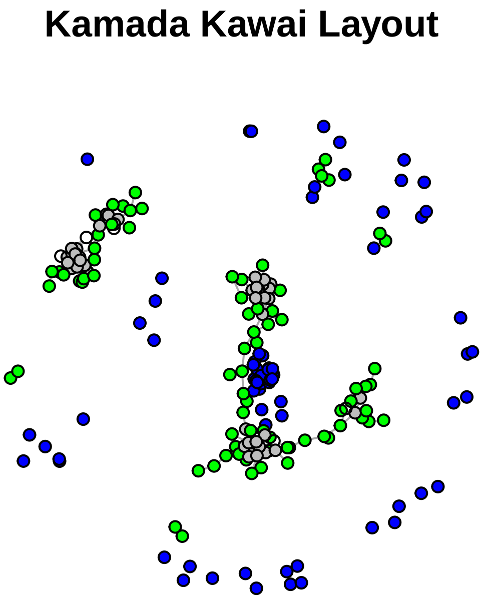
Induced Subgraphs - Part 1

```
num <- 5
V(g1)[centralization.degree(g1)$res == 0]$color <- "blue"
V(g1)[centralization.degree(g1)$res > num]$color <- "grey"
V(g1)[centralization.degree(g1)$res < num &centralization.degree(g1)$res > 0 ]$color <- "green"

#Add Comment
```

```
plot(g1, layout = layout_in_circle(g1), vertex.label = NA, vertex.size=5)
title("Circular Layout")
```

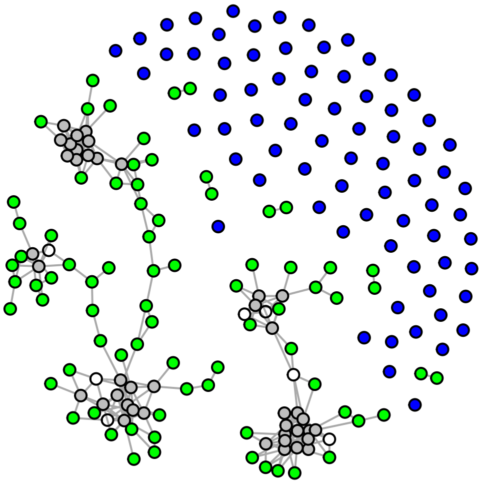


```
plot(g1, layout = layout_with_kk(g1), vertex.label = NA,vertex.size=5)
title("Kamada Kawai Layout")
```



```
plot(g1, layout = layout_with_fr(g1), vertex.label = NA,vertex.size=5)
title("Fruchterman-Reingold Layout")
```

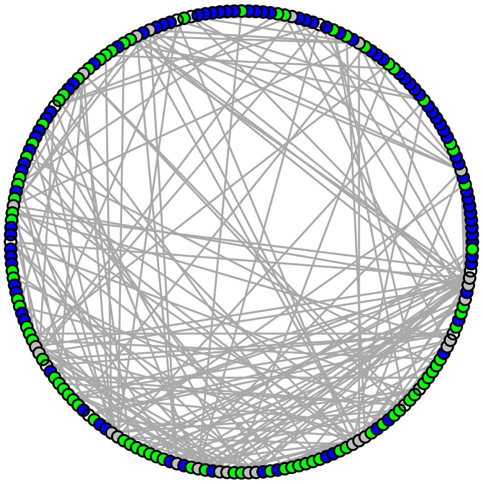
Fruchterman−Reingold Layout



Induced Subgraphs - Part 2

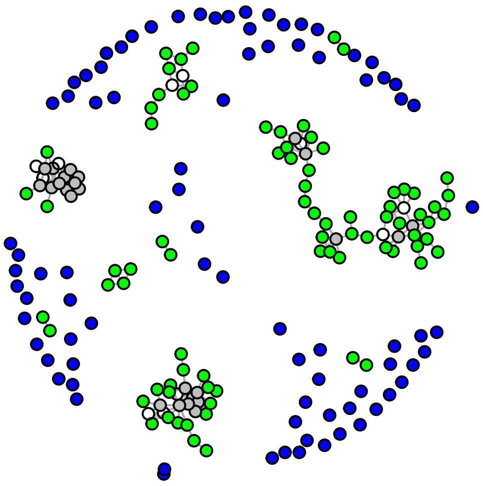
```
num <- 5
V(g2)[centralization.degree(g2)$res == 0]$color <- "blue"
V(g2)[centralization.degree(g2)$res > num]$color <- "grey"
V(g2)[centralization.degree(g2)$res < num &centralization.degree(g2)$res > 0 ]$color <- "green"

plot(g2, layout = layout_in_circle(g2), vertex.label = NA,vertex.size=5)
title("Circular Layout")
```

Circular Layout



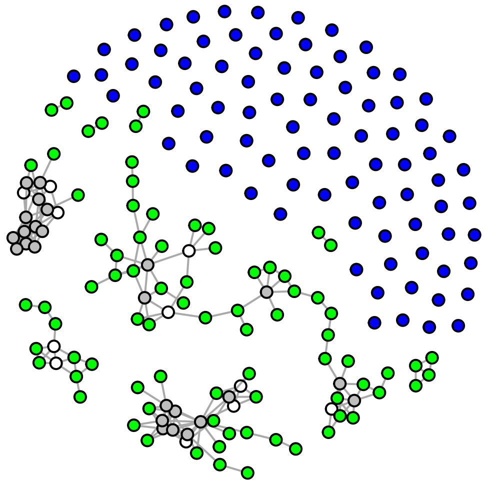
```
plot(g2, layout = layout_with_kk(g2), vertex.label = NA,vertex.size=5)
title("Kamada Kawai Layout")
```

Kamada Kawai Layout



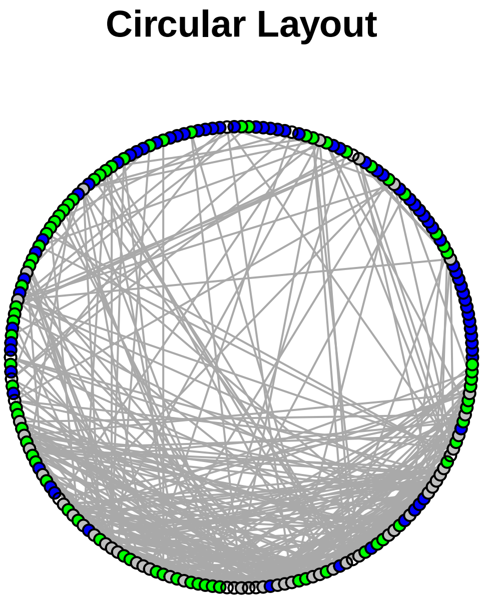
```
plot(g2, layout = layout_with_fr(g2), vertex.label = NA,vertex.size=5)
title("Fruchterman-Reingold Layout")
```

Fruchterman−Reingold Layout

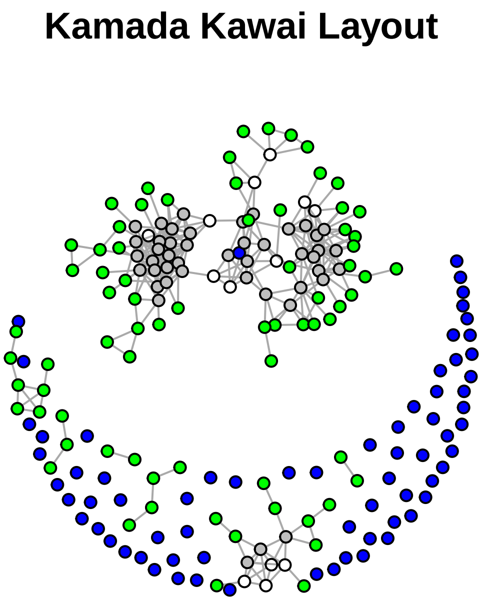


Induced Subgraphs - Part 3

```
num <- 5
V(g3)[centralization.degree(g3)$res == 0]$color <- "blue"
V(g3)[centralization.degree(g3)$res > num]$color <- "grey"
V(g3)[centralization.degree(g3)$res < num &centralization.degree(g3)$res > 0 ]$color <- "green"
plot(g3, layout = layout_in_circle(g3), vertex.label = NA,vertex.size=5)
title("Circular Layout")
```

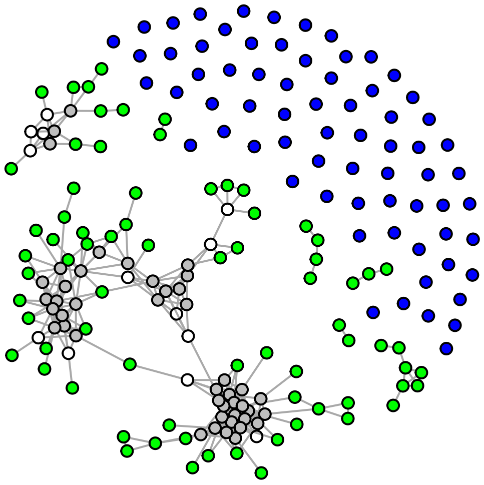


```r
plot(g3, layout = layout_with_kk(g3), vertex.label = NA,vertex.size=5)
title("Kamada Kawai Layout")
```



```r
plot(g3, layout = layout_with_fr(g3), vertex.label = NA,vertex.size=5)
title("Fruchterman-Reingold Layout")
```

Fruchterman−Reingold Layout



Report

Motivation

Data description

The dataset we have used here comes from food2vec. We have tried to visualize the similarity between different foods according to their flavors. Currently, we are working on a project to recommend similar foods across different cuisines considering a person's choice of food. In this scenario, the flavor of the food plays an important role in making a recommendation. Building a network will provide additional information on different food clusters.

How we obtained it?

We obtained the embeddings of each ingredients and we used cosine similarity to find similar ingredients. We obtained the embeddings from a project names food2vec. Considering a cutoff of 0.707 (cosine of 45°) we found ingredients having significant similarities and joined them with edges.

Why someone should care about analyzing it?

This network contains connections between highly similar foods, estimated purely based on their word embeddings. This is highly relevant for any NLP-related work in the food industry because it allows for a deeper understanding of the food domain. This network analysis can be used for various purposes, such as making food/ingredient recommendations based on similar foods/ingredients. Our project will directly be able to incorporate the network analysis as a means of recommending similar dishes across different cuisines.

What we wish to answer using the data set?

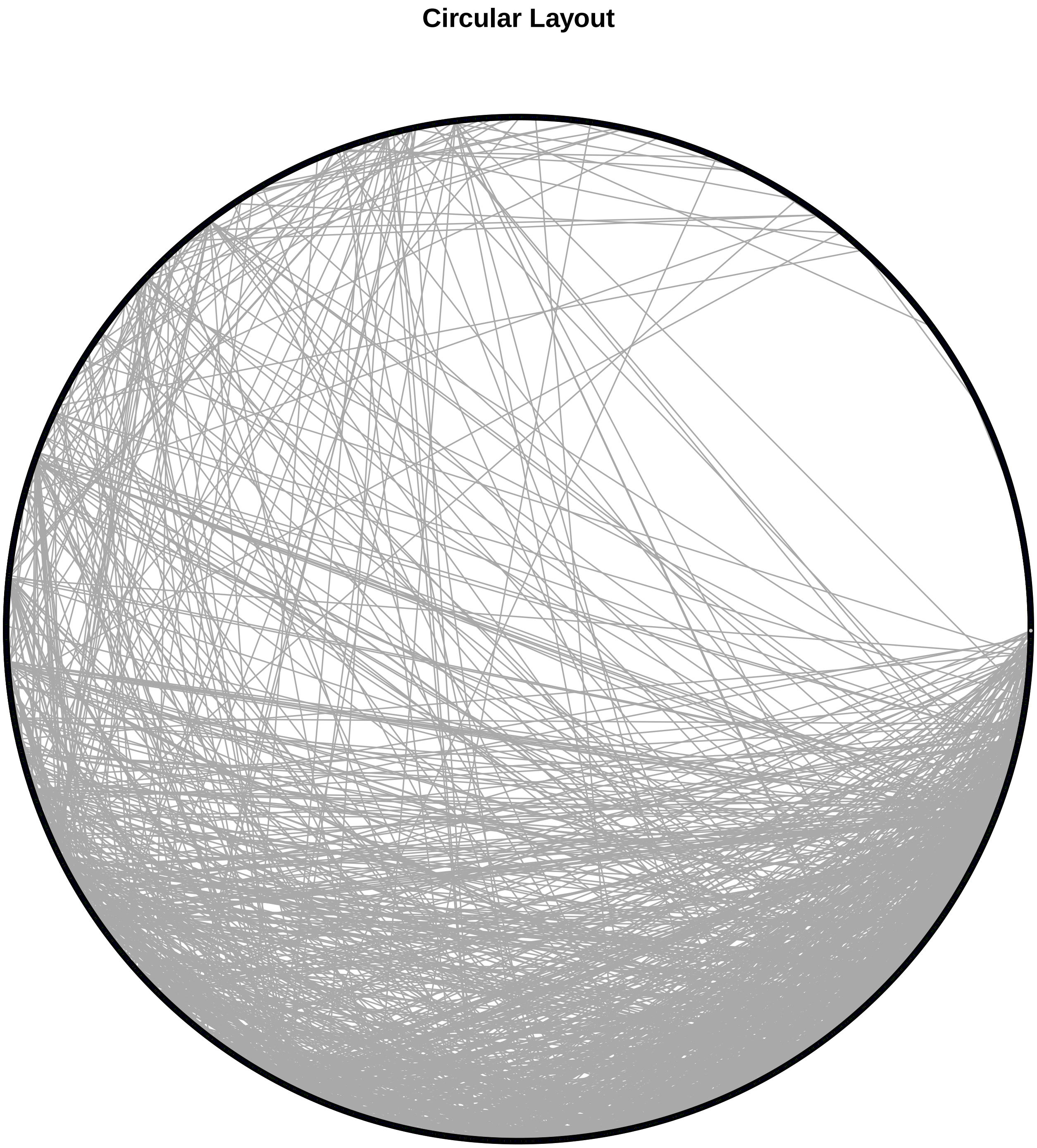
- Which foods/ingredients are highly similar based purely on the similarity of their word embeddings?
- What clusters of foods/ingredients can we identify using network analysis?

Methods

What network techniques you used in exploratory analysis and how they are used to answer your question.

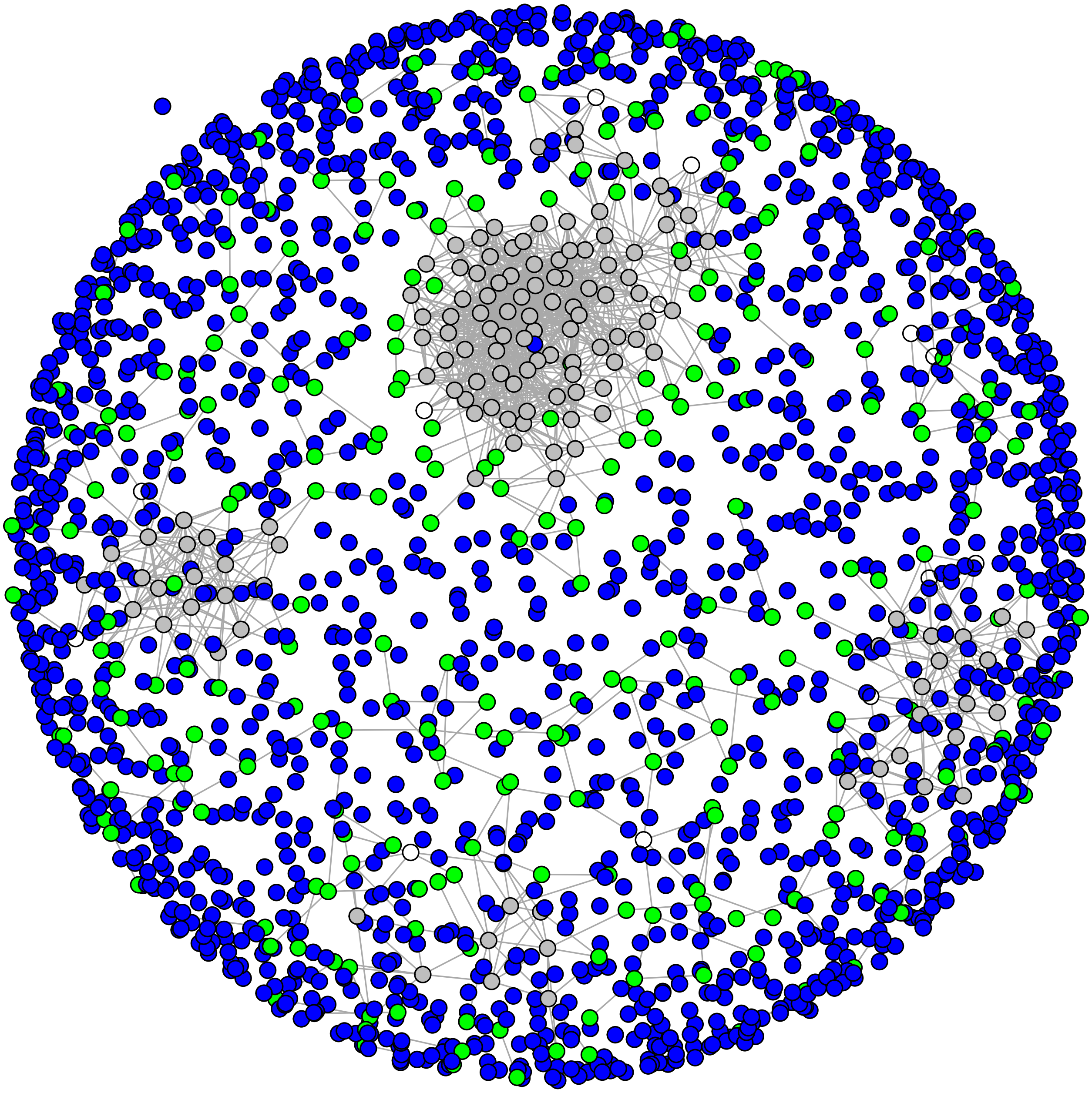
Now let's explore the network properties on the entire dataset.

```r
thresh <- 0.8
food.adj.full <- food.adj
food.adj.full[food.adj < thresh] <- 0
food.adj.full[food.adj >= thresh] <- 1
diag(food.adj.full) <- 0

num <- 5


food.full <- graph_from_adjacency_matrix(food.adj.full, mode='undirected')

V(food.full)[centralization.degree(food.full)$res == 0]$color <- "blue"
V(food.full)[centralization.degree(food.full)$res > num]$color <- "grey"
V(food.full)[centralization.degree(food.full)$res < num &
               centralization.degree(food.full)$res > 0 ]$color <- "green"

plot(food.full, layout = layout_in_circle(food.full), vertex.label = NA,vertex.size=1)
title("Circular Layout")
```



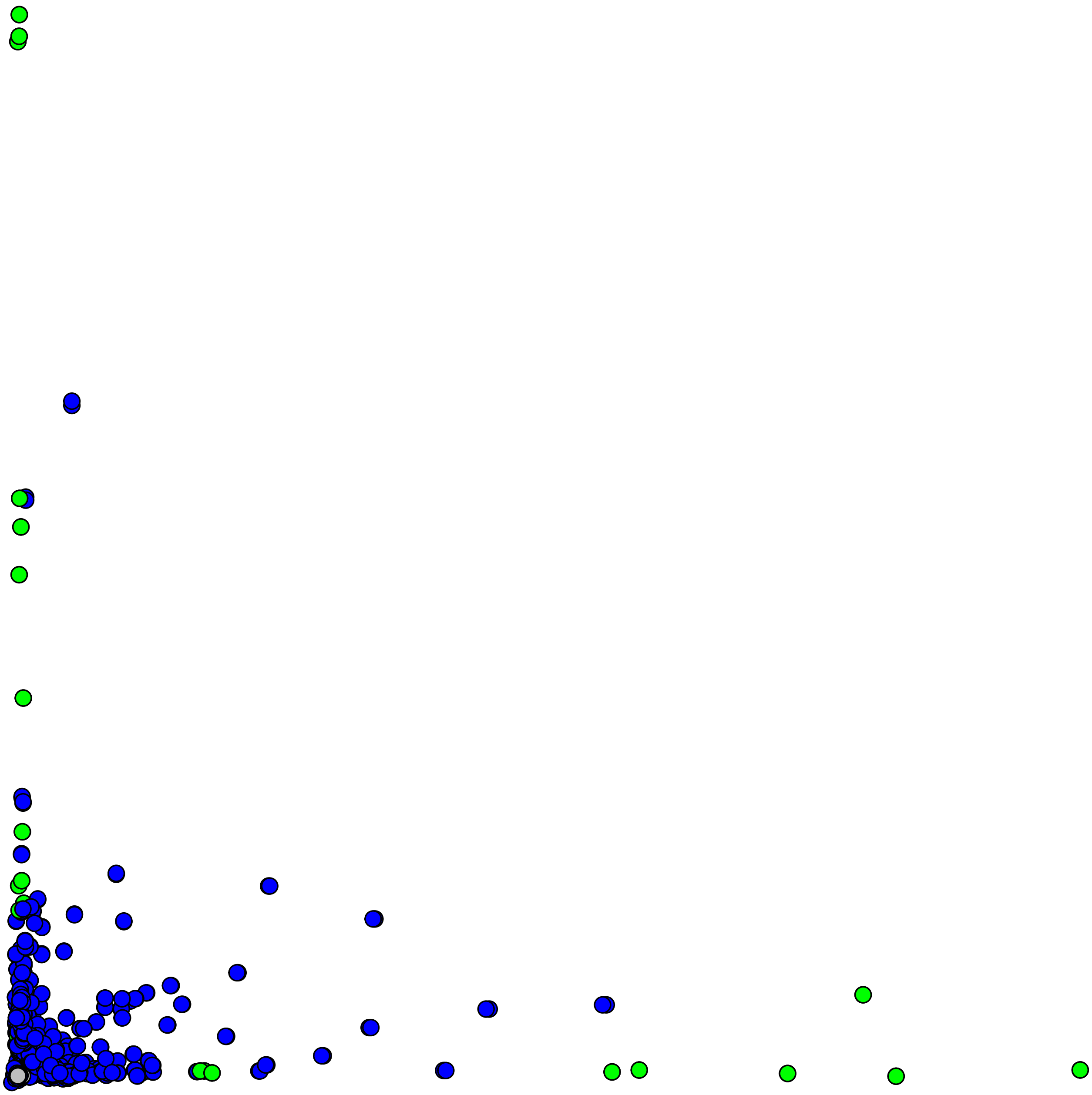
```r
plot(food.full, layout = layout_with_kk(food.full), vertex.label = NA,vertex.size=3)
title("Kamada Kawai Layout")
```

Kamada Kawai Layout



```
plot(food.full, layout = layout_with_fr(food.full), vertex.label = NA,vertex.size=3)
title("Fruchterman-Reingold Layout")
```

Fruchterman−Reingold Layout



```
# Dijkstra's
#---------------
# calculates the pairwise shortest path for each pair of nodes
food.igraph.shortest.dist <- distances(food.igraph, mode="out", algorithm = "dijkstra")
```

```
max(food.igraph.shortest.dist[ which(food.igraph.shortest.dist < Inf)])
```

```
## [1] 13
```

```
#count the number of vertices in each connected component
comps <- decompose.graph(food.igraph)
```

```
table(sapply(comps, vcount))
```

```
##
##    1    2    3    5    6    8 1837
##  190   15    4    1    1    1    1
```

```
#The components are organized in the list by size (in decreasing order)
#Here we just look at the "giant component" (i.e., the component with the largest
# number of nodes) so we will focus on the connected component with 1837 nodes
```

```
food.gc <- comps[[1]]

#average path length
average.path.length(food.gc)
```

```
## [1] NaN
```

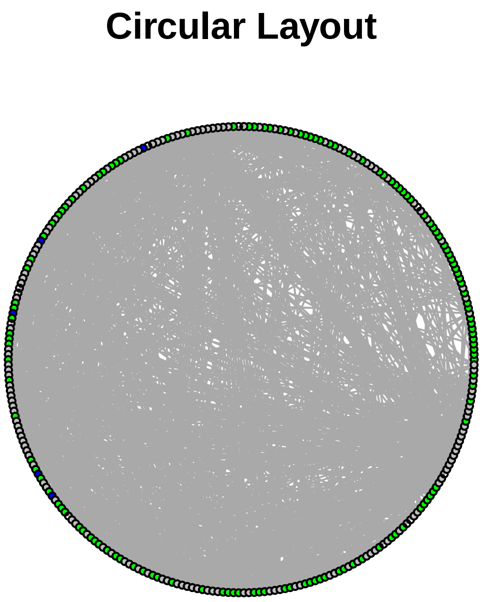
```
#maximum shortest path length / diameter
diameter(food.gc)
```

```
## [1] 0
```

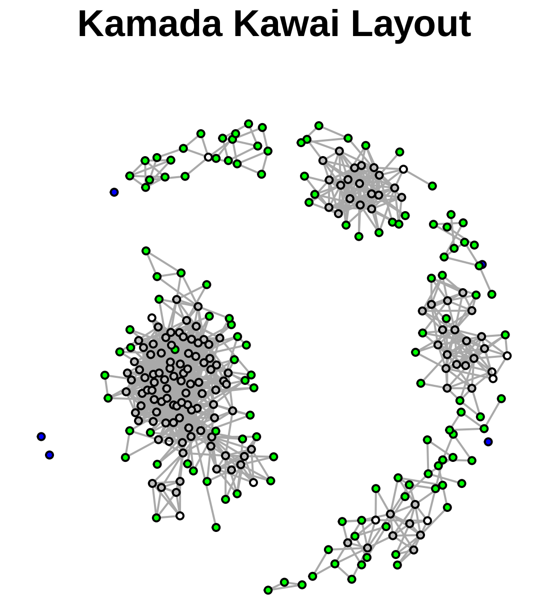
From the experiments above, we noticed that Kamada Kawai is the layout best suited for our analysis; the other two layouts have a high degree of overlap.

Now we will try removing the isolated nodes.

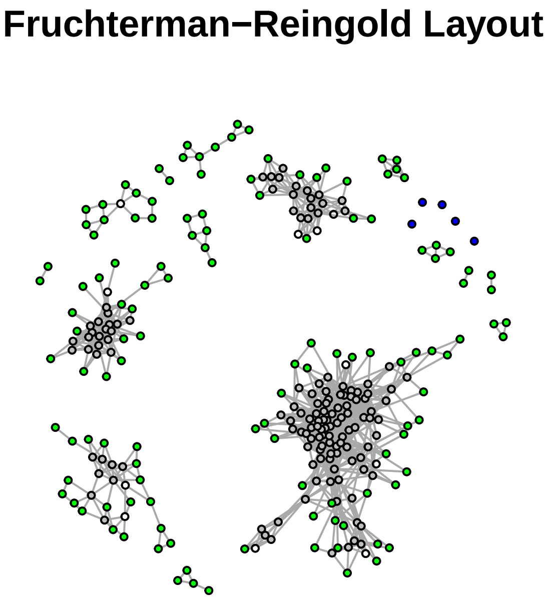
```
food.full.iso.rem <- igraph::induced.subgraph(food.full, vids = which(igraph::degree(food.full) > 1))

V(food.full.iso.rem)[centralization.degree(food.full.iso.rem)$res == 0]$color <- "blue"
V(food.full.iso.rem)[centralization.degree(food.full.iso.rem)$res > num]$color <- "grey"
V(food.full.iso.rem)[centralization.degree(food.full.iso.rem)$res <
                        num &centralization.degree(food.full.iso.rem)$res > 0 ]$color <- "green"

plot(food.full.iso.rem, layout = layout_in_circle(food.full.iso.rem), vertex.label = NA,vertex.size=3)
title("Circular Layout")
```



```
plot(food.full.iso.rem, layout = layout_with_kk(food.full.iso.rem), vertex.label = NA,vertex.size=3)
title("Kamada Kawai Layout")
```



```
plot(food.full.iso.rem, layout = layout_with_fr(food.full.iso.rem), vertex.label = NA,vertex.size=3)
title("Fruchterman-Reingold Layout")
```



After removing vertices with degree less than two, we can clearly visualise the communities. Among the three layouts, Fruchterman-Reingold is the most informative.

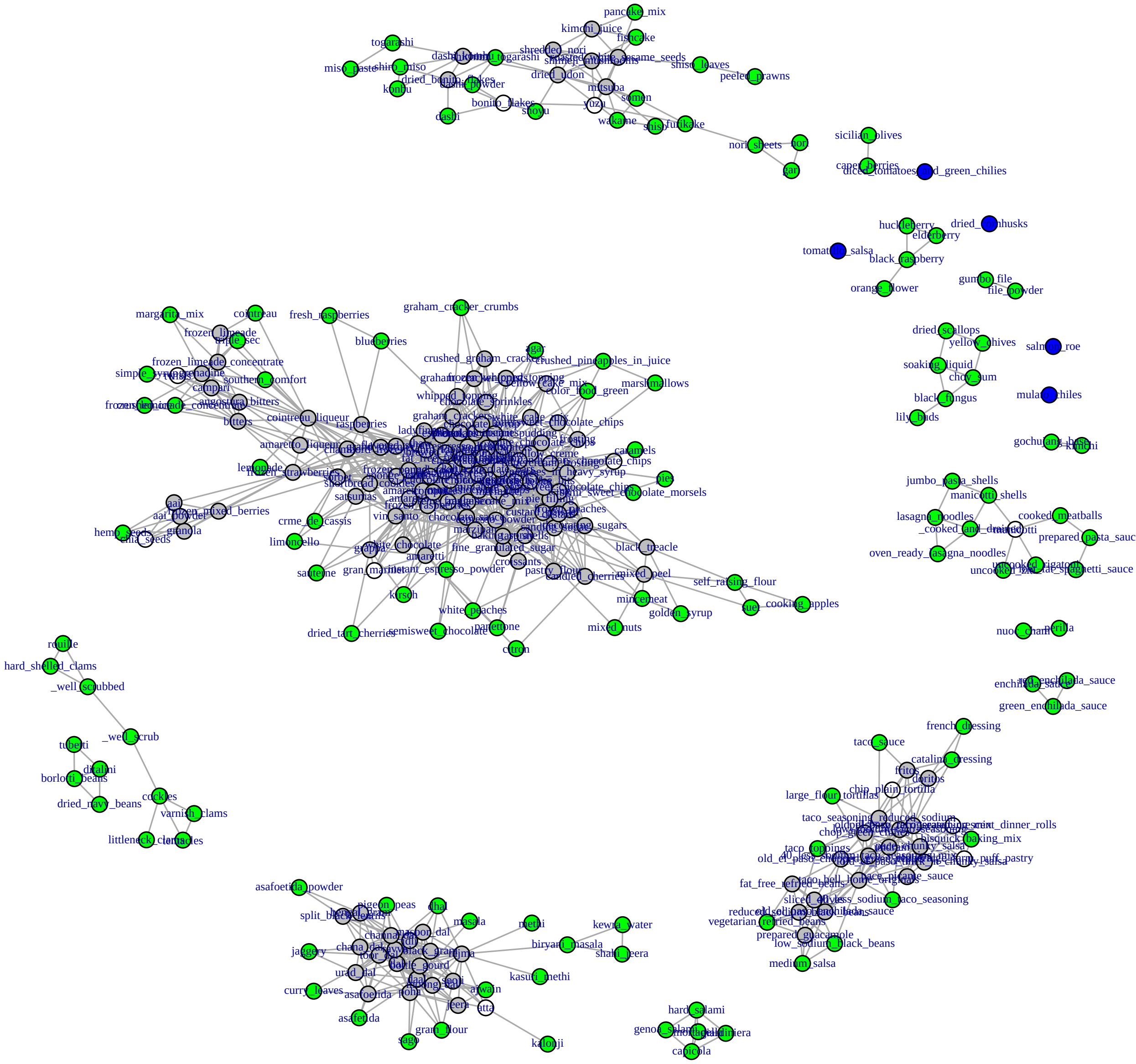
Now let us see how each of the vertices are connected.

```
f_g <- cluster_fast_greedy(food.full.iso.rem)
table(f_g$membership)
```

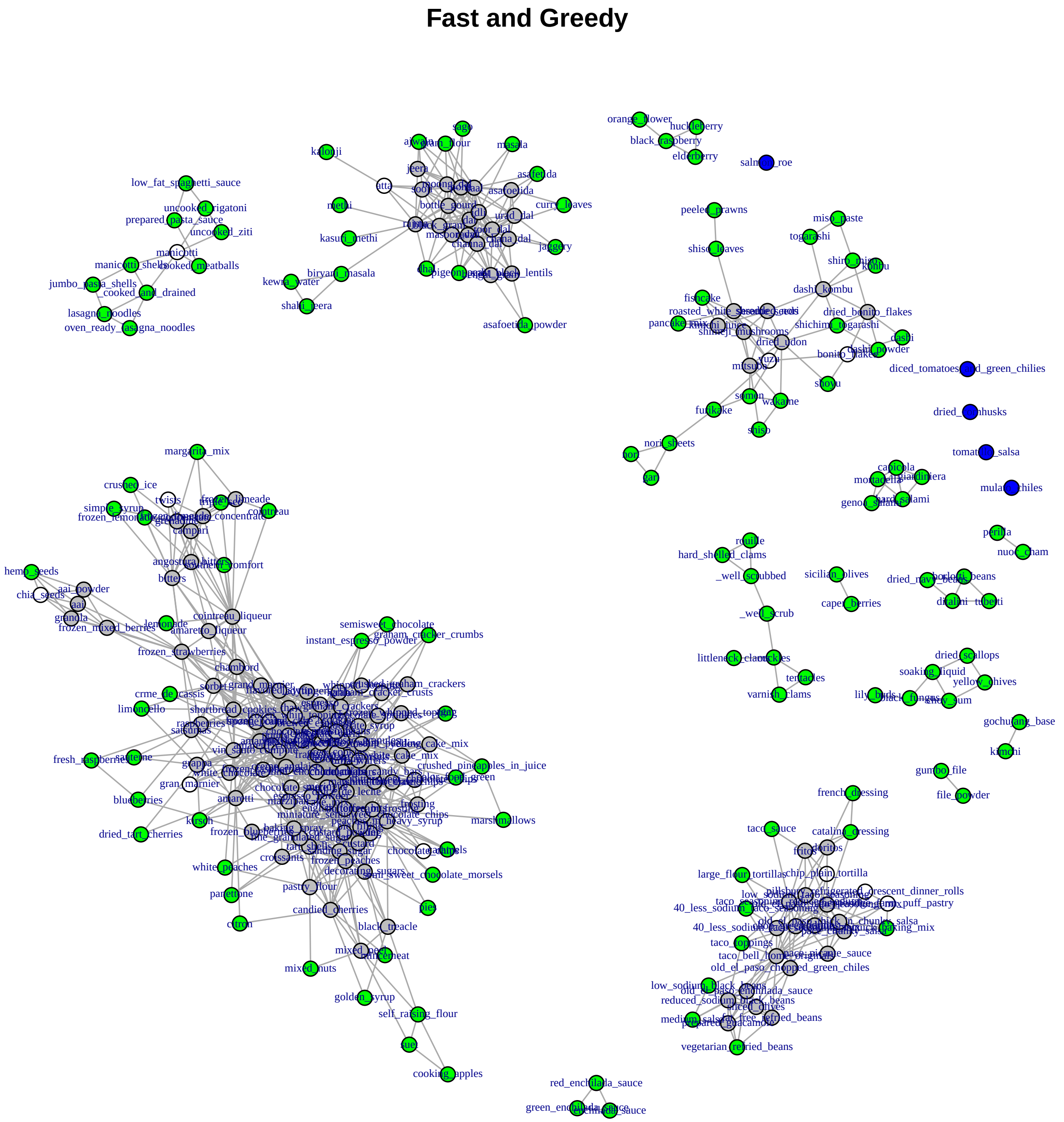
```
##
##  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22
## 36 80 44 29  8  8  5  6 31 11  4  3  4  2  2  2  2  1  1  1  1  1
```

Here, we can see that 36 ingredients are connected with just one other ingredient. There are also five which are connected with over 18 other ingredients.

```r
#plotting the network according to community label
#first plot the original graph
V(food.full.iso.rem)$label.cex = 0.5
x <- plot(food.full.iso.rem, vertex.size=3)
```



```r
colors <- c("blue", "red", "yellow", "purple", "pink")
#par(mfrow = c(1,2))
#plotting the community structure using our own defined colors
plot(food.full.iso.rem, coord = x, main = "Fast and Greedy",
     vertex.col = colors[f_g$membership], vertex.size=3)
```

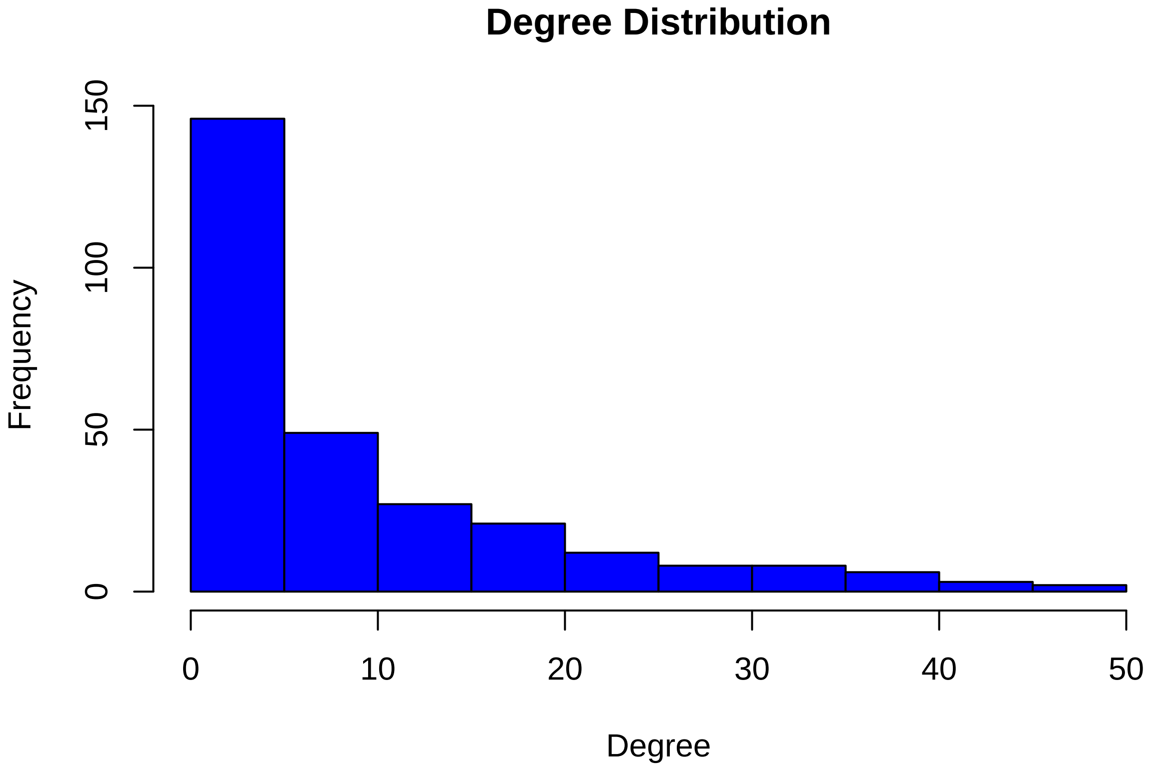


```
##Running 5 different community detection methods and compare
info.clusters <- cluster_infomap(food.full.iso.rem)
l_p <- cluster_label_prop(food.full.iso.rem)
louvain <- cluster_louvain(food.full.iso.rem)
walktrap <- cluster_walktrap(food.full.iso.rem)
table(walktrap$membership)
```

```
##
##  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
## 30 19  4  5  3 13  2  6 42  3  6  3  3  2  3  3  7 38  4  3 11  4  3  4  6
## 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
##  6  8  2  2  2  4  5  2  2  2  3  2  2  1  1  1  1  1  1  1  1  1  1  1  1
## 51
##  1
```

Let's have a look at the degree, transitivity and density of our final graph.

```
hist(centralization.degree(food.full.iso.rem)$res,col = "blue",
     xlab = "Degree", ylab = "Frequency",
     main = "Degree Distribution")
```



```
print(paste("Transitivity = ", transitivity(food.full.iso.rem)))
```

```
## [1] "Transitivity =  0.512973744072175"
```

```
print(paste("Density = ", graph.density(food.full.iso.rem)))
```

```
## [1] "Density =  0.0338709270336438"
```

Results

Were you able to answer your question?

Yes, we were able to identify some clusters of highly similar foods/ingredients based purely on their word embeddings. In particular, we were able to detect communities of connected components which consisted of particular cuisines or food types. For example, using visualization of the fast and greedy and walktrap community detections algorithms, we discovered a few communities. Some examples of easily visible communities are Indian foods, Mexican foods, Japanese foods, Italian foods, cocktail ingredients, and desserts.

What did you learn about the data?

We learned that the community structure of the data is well-represented by cuisines and even food types that transcend cuisine, such as cocktails and desserts. From this network analysis, we were able to validate the efficacy of the word embeddings in food2vec for representing the similarity of many foods. This is valuable to us because now we have confirmed that can effectively use the food2vec embeddings in our recommendation system for developing our web application.

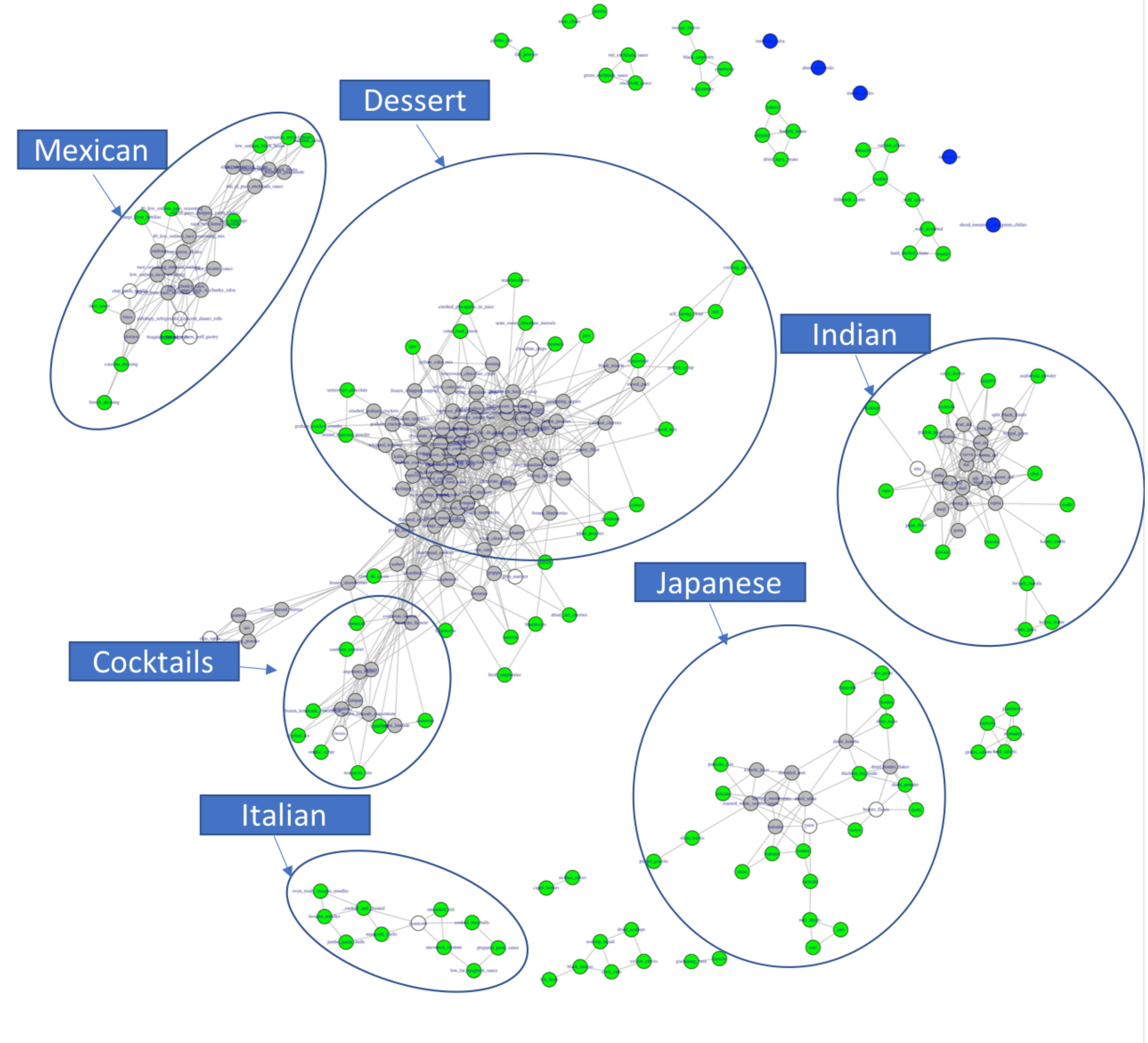


Figure 1: food2vec clusters