From Bayesian Networks To Priors

The following network represents one of several possible Bayesian networks given Lina's droughted data. Several networks may exhibit similar statistical properties–thus, we want a network that is simple and makes sense. Prior knowledge about the data and future information about the dependencies between the variables will help us continue to improve the network.

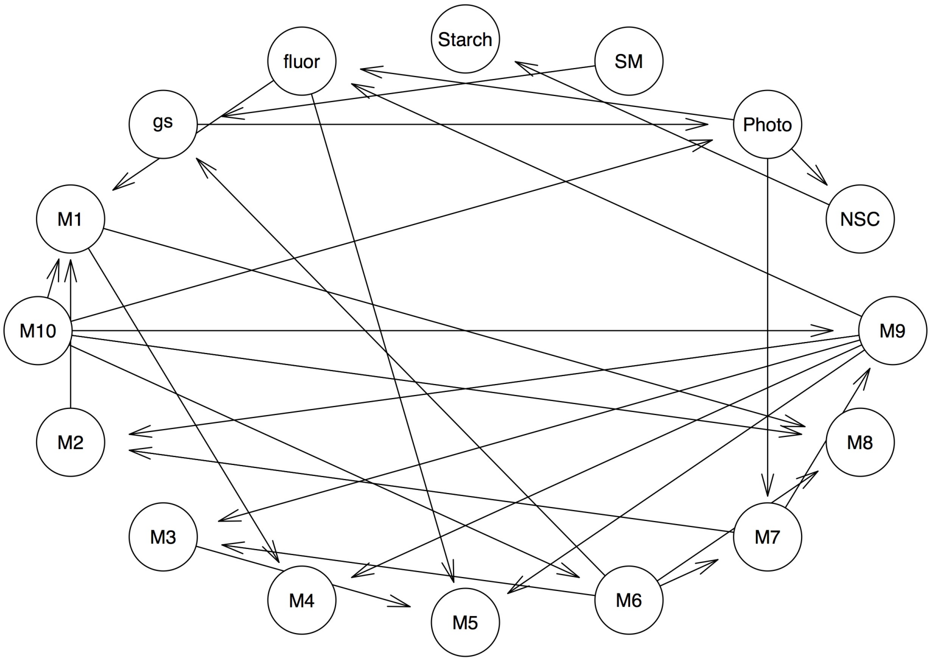
Since Bayesian networks are computationally expensive, genes were subset into gene modules to reduce the number of variables in the network. Gene modules should show similar behavior across all time points and treatments. Hierarchical clustering was used to find genes with similar expression. We haven't found a good way to quantify our uncertainty here and other clustering methods still need to be tried.

Note: This following example is not the final network. At this point it is a **candidate**. I think that a simpler network can still be found.

```
# Load Bayesian network package bnlearn.
suppressMessages(library(bnlearn))

# Load network structure learned in clusteredRNApheno3.R.
dag <- model2network("[SM][M10][M6|M10][gs|SM:M6][Photo|gs:M10]
                      [NSC|Photo][M7|Photo:M6][Starch|NSC][M9|M7:M10]
                      [fluor|Photo:M9][M2|M7:M9][M3|M6:M9]
                      [M1|fluor:M2:M10][M5|fluor:M3:M9][M4|M1:M9]
                      [M8|M1:M6:M10]")

# The network is too small with plot(dag) so the plot will be saved
# as a jpeg and attached.
```


Starch
fluor
SM
gs
Photo
M1
NSC
M10
M9
M2
M8
M3
M7
M4
M6
M5

The process of finding the dependencies within the network is called *structure learning* and its output is a network structure. The entire network represents a joint distribution and each node in the network represents a conditional distribution given its parents (any node pointing to it). *Parameter learning* involves learning the local, conditional distributions given by the structure. It may make sense to be consistent with the methods used for structure learning and parameter learning, but it is not necessary. Once the structure is learned, parameter learning can be performed using any method desired. **For example, I can use bnlearn to learn the structure and rjags to perform parameter learning.**

Here, I will use the package bnlearn for parameter learning where each variable has been discretized and represented by a multinomial distribution and each prior by a uniform Dirichlet distribution.

```r
# Load phenotype and transcriptome data for parameter learning.
rnaPheno <- read.csv(file = "RNAphenoBN.csv", row.names = 1)

# Convert modules to factors.
rnaPheno[, 7:dim(rnaPheno)[2]] <- lapply(rnaPheno[, 7:dim(rnaPheno)[2]], factor)

# Perform parameter learning for data using network structure from first R chunk.
bn <- bn.fit(dag, rnaPheno, method = "bayes")
```

We can now identify the distributions of each node. For example, module 6 (M6) has one parent, module 10 (M10). Thus, the probability of any value from M6 is dependent on the value of M10: $P(M6|M10)$. This network is discrete, so we can see the probabilities in a barchart. However, if using rjags, you can use continuous distributions.

```r
# Since the default graphics don't show intervals well, the data
# will be converted for importation into Tableau.

# Create a dataframe storing the values and frequencies for M6 and M10.
m6 <- data.frame(bn$M6$prob)

# Write csv.
write.csv(m6, file = "m6CPD.csv")
```

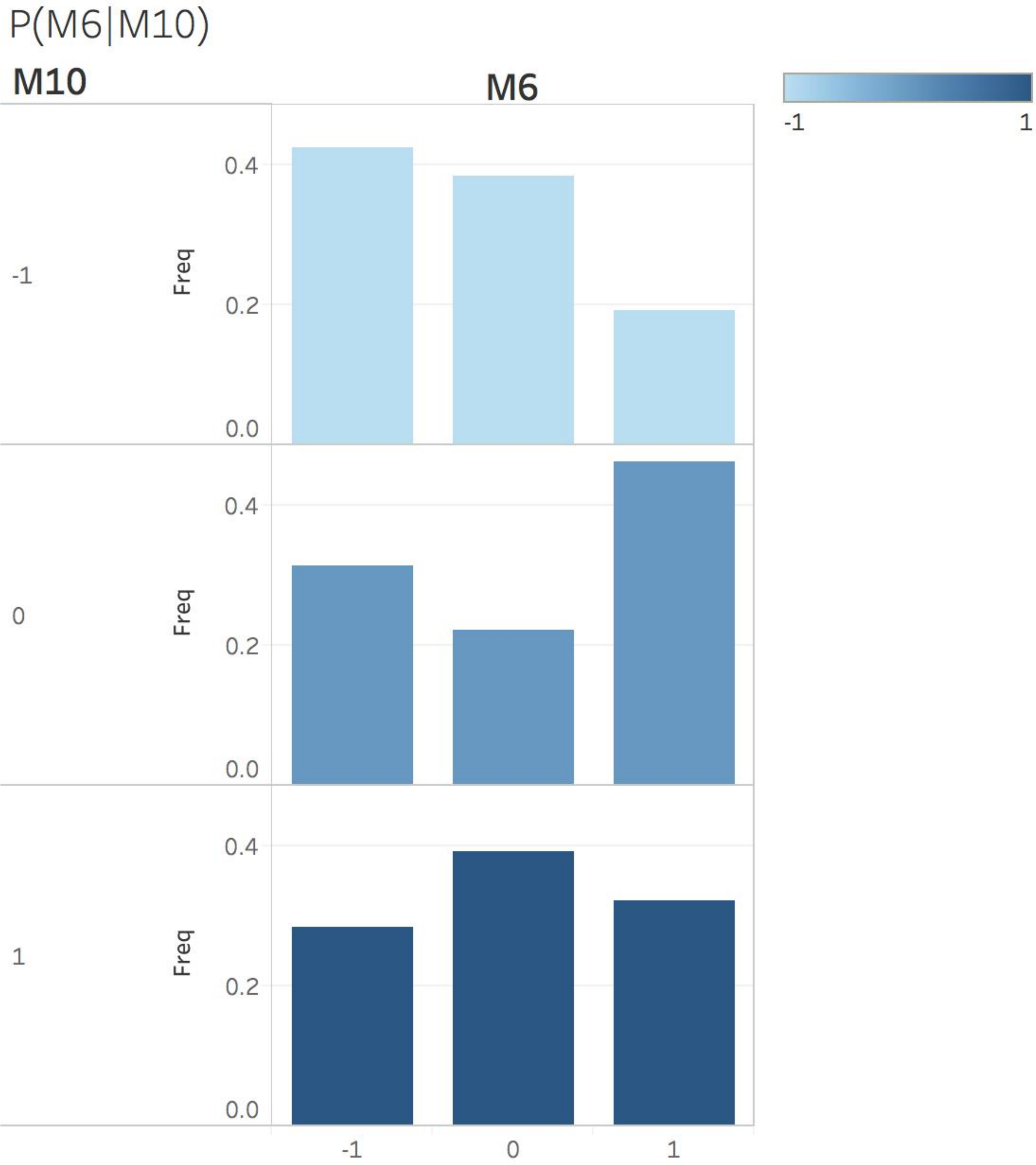


Figure 1:

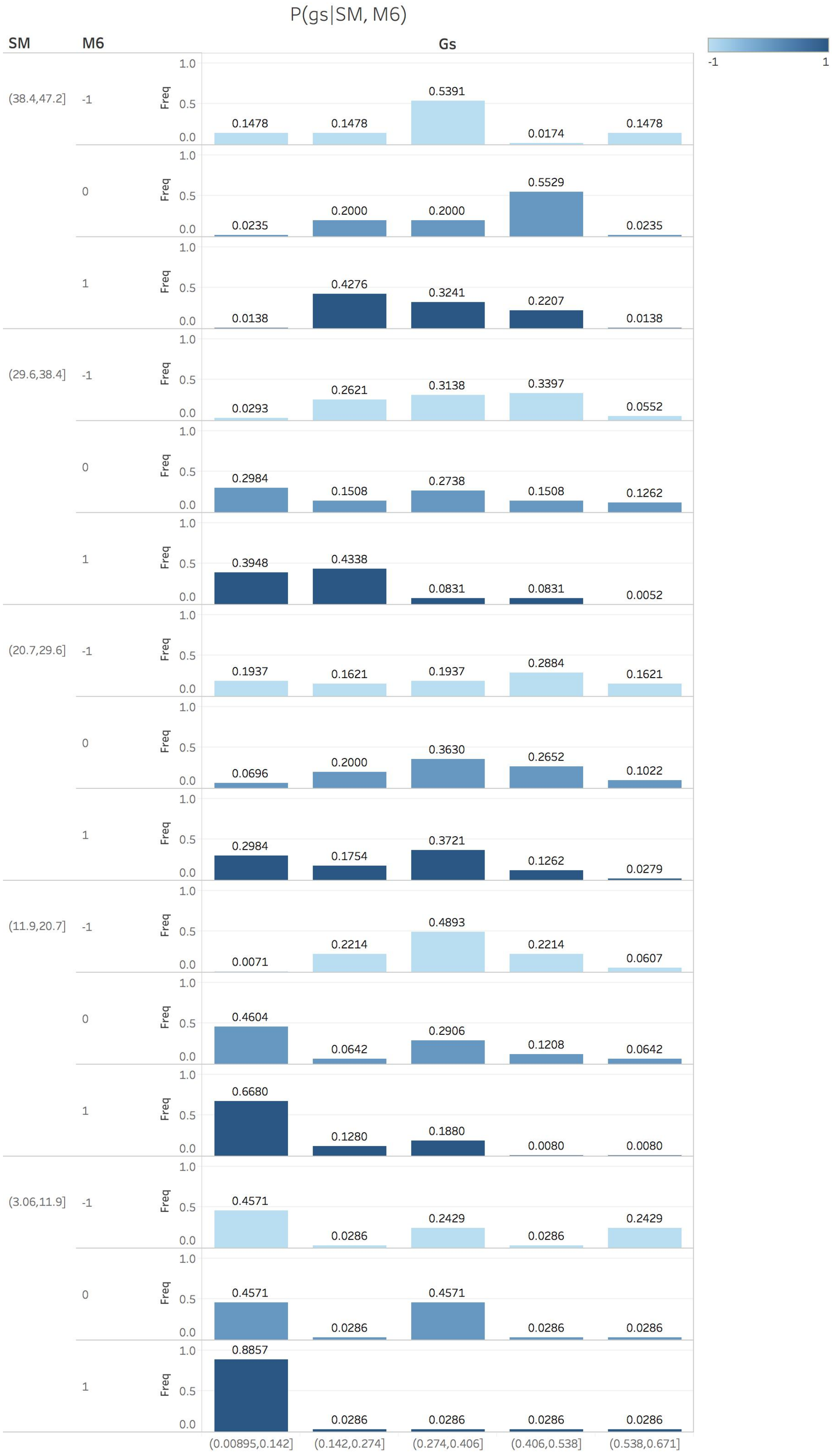
Mathematically, $P(M6, M10) = P(M6|M10)P(M10)$. Therefore, the distribution of M10 represents the prior distribution and the distribution of M6 the conditional probability distribution. If we want P(M6), we can simply marginalize out M10 by finding the probability of M6 given all possible values of M10.

Now we'll look at stomatal conductance (g_s), which has two parents: M6 and soil moisture (SM). Notice that for low SM, g_s is generally low. However, when M6 is 0, the distribution of values is split between the lowest interval and the middle interval. This could indicate that the M6 cluster isn't behaving similarly across all timepoints and should be divided further.

```r
# Create a dataframe storing the values and frequencies for
# stomatal conductance and its parents.
gs <- data.frame(bn$gs$prob)

# Tableau has a difficult time ordering the intervals properly with
# closed brackets. The first closed bracket will be changed to an
# open bracket.
levels(gs$gs)[5] <- "(0.00895,0.142]"
levels(gs$SM)[5] <- "(3.06,11.9]"

# Write csv.
write.csv(gs, file = "gsCPD.csv")
```


P(gs|SM, M6)
SM
M6
Gs
-1
1
(38.4,47.2]
-1
Freq
1.0
0.5
0.0
0.1478
0.1478
0.5391
0.0174
0.1478
0
0.0235
0.2000
0.2000
0.5529
0.0235
1
0.0138
0.4276
0.3241
0.2207
0.0138
(29.6,38.4]
-1
0.0293
0.2621
0.3138
0.3397
0.0552
0
0.2984
0.1508
0.2738
0.1508
0.1262
1
0.3948
0.4338
0.0831
0.0831
0.0052
(20.7,29.6]
-1
0.1937
0.1621
0.1937
0.2884
0.1621
0
0.0696
0.2000
0.3630
0.2652
0.1022
1
0.2984
0.1754
0.3721
0.1262
0.0279
(11.9,20.7]
-1
0.0071
0.2214
0.4893
0.2214
0.0607
0
0.4604
0.0642
0.2906
0.1208
0.0642
1
0.6680
0.1280
0.1880
0.0080
0.0080
(3.06,11.9]
-1
0.4571
0.0286
0.2429
0.0286
0.2429
0
0.4571
0.0286
0.4571
0.0286
0.0286
1
0.8857
0.0286
0.0286
0.0286
0.0286
(0.00895,0.142]
(0.142,0.274]
(0.274,0.406]
(0.406,0.538]
(0.538,0.671]

If we want to use M6 in TREES as a prior, we have two options:

1) Use P(M6) directly by marginalizing out M10 (its parent). In this version, M6 would be the only prior.
2) Use the conditional probability distribution of M6 in conjunction with the distribution of M10. In this version, M6 would be a prior for g_s with M10 as a prior for M6.

Because there are several candidate networks that may share similar statistical properties, it is important to know if a prior helps improve the predictions in TREES. If not, that can help narrow down candidate networks and get us closer to understanding the relationships between transcripts and physiological data.