QCB455/COS551 Homework 4

Sequence Alignment, Sequence Profiles, Machine Learning, Network Analysis, and Proteomics

William Svoboda (wsvoboda)

Last edited December 09, 2021

Contents

Collaboration Statement

I talked with Brendan McManamon (bm18, student) and Sara Schwartz (sarats, student) about this homework.

1 Semi-global Sequence Alignment

1. How would you modify the dynamic programming algorithm we discussed in class?

- **To not penalize gaps at the beginning of S, the first column in the matrix is initialized to zero. To not penalize gaps at the end of T, traceback begins from the last column at the maximum score.**

2. Using the modified algorithm from above, complete the alignment matrix for sequences S and T.

		G	**C**	**A**	**A**	**G**	**T**
	0	−2	−4	−6	−8	−10	−12
A	0	−1	−3	−2	−4	−6	−8
T	0	−1	−2	−4	−3	−5	−4
G	0	2	0	−2	−4	−1	−3
C	0	0	4	2	0	−2	−2
T	0	−1	2	3	1	−1	0
G	0	2	0	1	2	3	1

Figure 1: Alignment matrix of sequences

3. Perform traceback on the alignment matrix from above, and report the final alignment of S and T and their alignment score.

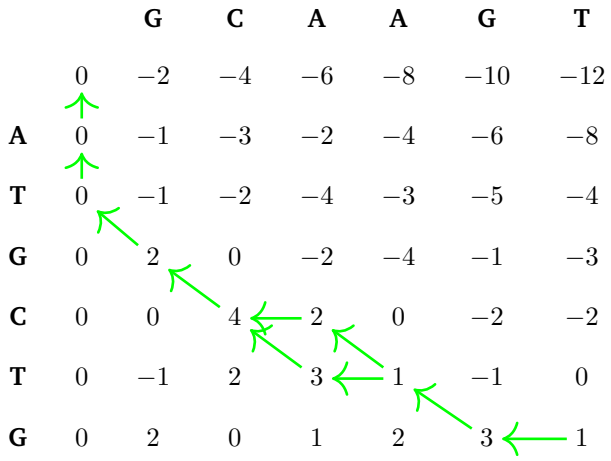


Figure 2: Traceback of alignment matrix

```
s: __GCAAGT
t: ATGCT_G_

s: __GCAAGT
t: ATGC_TG_
```

Figure 3: Final alignments of S and T

2 Sequence Profiles

1. For your model, compute estimates for the probability of observing each of the 20 amino acids for the 5th column. Make sure to correct each estimate by adding a pseudocount ($\frac{1}{20}$) to each observation. Why is this correction necessary?

- **From looking at the fifth column, we know that $b_{m1N} = \frac{3}{5}$ and $b_{m1Q} = \frac{1}{5}$. We then correct each estimate by adding the pseudocount ($\frac{1}{20}$) such that $b_{m1N} = \frac{3+1}{5+20} = \frac{4}{25}$ and $b_{m1Q} = \frac{1+1}{5+20} = \frac{2}{25}$ with all else equal to $\frac{1}{25}$.**

2. For your model, you will need to estimate the probability of having insertions and deletions after each modeled column. Give an estimate for having an insertion, deletion or neither after the 1st column. Make sure to to correct each estimate by adding a pseudocount ($\frac{1}{3}$) to each.

- **I don't know.**

3. You notice that the 3rd and 7th columns of the alignment are correlated. Whenever there is a positively charged amino acid (K or R) in one, there is a negatively charged amino acid (D or E) in the other. Do HMM-profiles effectively capture these correlations? Why or why not?

- **I don't know.**

4. What is the advantage of profile-HMMs over regular profiles/PSSMs?

- **Unlike PSSMs, profile-HMMs allow for position-specific gaps.**

3 Machine Learning to Classify Cancer Type

1. Import the two data files **training.csv** and **testing.csv** as data frames into your working environment. After importing your data, use the as.factor() function to change the **Classification** column in both data frames to a factor. How many breast cancer negative controls are in the training set?

```
# Import data files
training <- read.csv("./hw4data/training.csv", header = TRUE)
testing <- read.csv("./hw4data/testing.csv", header = TRUE)

# Change the Classification column in each data frame to a factor
training$Classification <- as.factor(training$Classification)
testing$Classification <- as.factor(testing$Classification)

# Find number of breast cancer negative controls in training set
num_negative_controls <- training %>%
    filter(Classification == 1) %>%
    nrow()
```

- **There are 36 breast cancer negative controls in the training set.**

2. Train a classifier on the training data to distinguish breast cancer presence using the Support Vector Machine (SVM) method as discussed in lecture.

```
model <- train(training[, 1:ncol(training) - 1], training$Classification, method =
↪  "svmLinear",
    preProcess = c("center", "scale"))
```

3. Now, test your trained classifier on the testing dataset by predicting the hidden cancer status of the testing data. Generate a confusion matrix and summary statistics (accuracy, precision, and recall) for your classifier. Print out your confusion matrix.

```
# Generate predictions using trained model and testing data
predictions <- predict(model, newdata = testing[, 1:ncol(testing) - 1])

# Generate confusion matrix
confusionMatrix(predictions, testing$Classification)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  1  2
##          1 12  5
##          2  4 14
##
##                Accuracy : 0.7429
##                  95% CI : (0.5674, 0.8751)
##     No Information Rate : 0.5429
##     P-Value [Acc > NIR] : 0.0123
##
##                   Kappa : 0.4845
##
##  Mcnemar's Test P-Value : 1.0000
##
##             Sensitivity : 0.7500
##             Specificity : 0.7368
```

```
##          Pos Pred Value : 0.7059
##          Neg Pred Value : 0.7778
##              Prevalence : 0.4571
##          Detection Rate : 0.3429
##    Detection Prevalence : 0.4857
##       Balanced Accuracy : 0.7434
##
##        'Positive' Class : 1
##
```

4. Do you believe the classifier you trained is a good predictor of breast cancer status? Explain why or why not, citing the summary statistics from the previous problem. Does this conflict with the author's assessment? Name at least one difference between our analysis and the authors' analysis.

- **I believe the classifier is a good predictor of breast cancer status. Generating the summary statistics reveals that the classifier has an accuracy of about 74.3%. Additionally, the P-Value is very low at 0.0123 which greatly favors the alternative hypothesis. This corresponds with the author's assesment that the technique holds promise, as they found the sensitivity to be between 82-88% and the specificity to be between 85-90% (which is higher than my analysis at 75% and 73.7% respectively). One difference between our analyses is that the original paper used fewer features when creating their SVM.**

5. The authors use a process called cross-validation to build 95% confidence intervals for their summary statistics. Briefly describe what cross-validation is and why it is important in machine learning.

- **Cross-validation is a process where resampling takes place in order to evaluate if a machine learning model is over or under-fitting the data. This is important because it means a limited dataset will not unknowingly bias the model as a result of noise.**

6. In a few sentences, explain why neural networks might or might not be a good choice for the dataset and learning task we chose for this problem.

- **One of the biggest disadvantages of neural networks is that they are often a "black-box". In other words, it is often unclear how a neural network arrived at a given output. For this dataset and learning task, interpretability is very important. Neural networks, therefore, might not be a good choice because the input features that would cause a positive classification would not necessarily be clear. Given that we ultimately want to screen for breast cancer given certain metabolic markers, this would not be ideal.**

4 Network Analysis

1. Use www.thebiogrid.org to download the protein-protein interaction network for Stukalov, 2020 "Multilevel proteomics reveals host-perturbation strategies of SARS-CoV-2 and SARS-CoV".

```
biogrid <- read_tsv("./hw4data/BIOGRID-PUBLICATION-222410-4.4.204.DOWNLOADS.zip") %>%
    filter(`Experimental System` == "Affinity Capture-MS") %>%
    select(node_1 = `Official Symbol Interactor A`, node_2 = `Official Symbol Interactor
    ↪  B`,
        spec_1 = `Organism Name Interactor A`, spec_2 = `Organism Name Interactor B`) %>%
    filter(spec_1 != "Severe acute respiratory syndrome-related coronavirus" & spec_2 !=
        "Severe acute respiratory syndrome-related coronavirus") %>%
    filter(node_1 != node_2)  # Remove self edges
edges <- biogrid %>%
    select(node_1, node_2)
g <- as_tbl_graph(edges, directed = TRUE)
```

2. Briefly describe the experimental approach in this paper to create the interaction network.

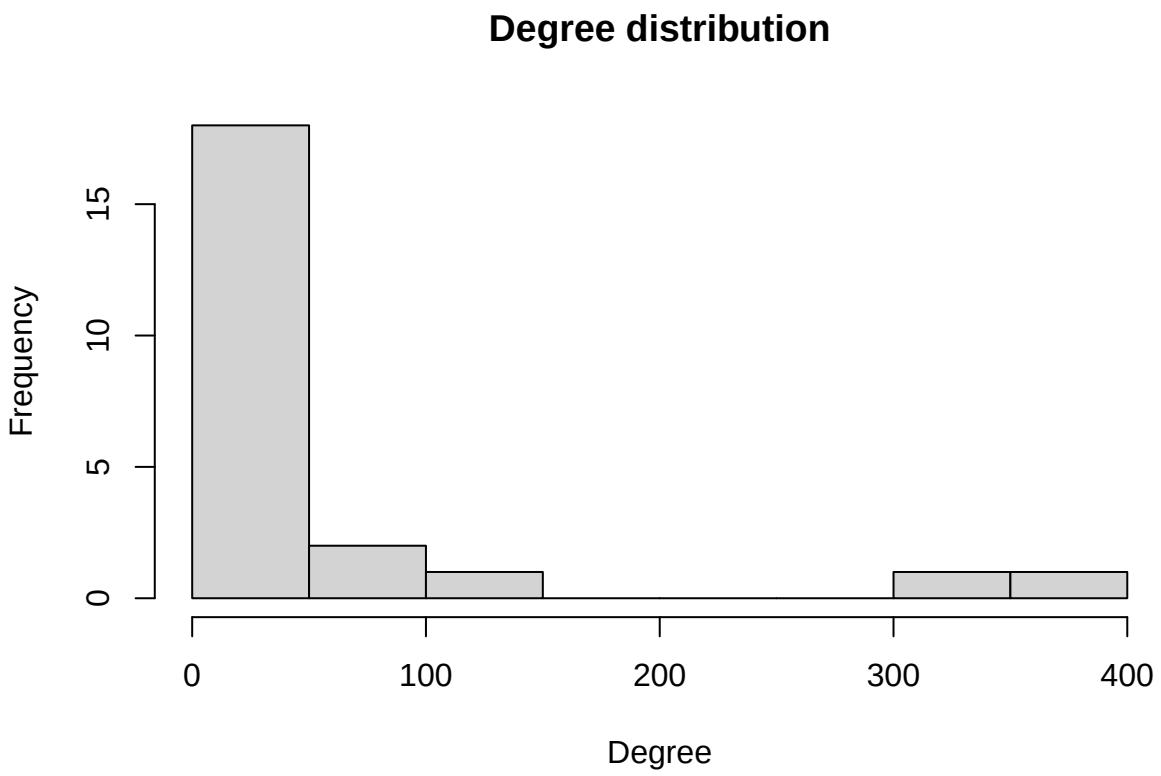
- **In order to create the interaction network, the authors profiled the interactions between coronaviruses and human cells. Unlike previous studies, however, both SARS-CoV-2 and the closely related SARS-CoV were profiled. This allowed for previously hidden interactions to be identified using the additional data from the latter virus.**

3. How many nodes and edges are present in the graph?

- **There are 899 nodes and 1087 edges in the graph.**

4. Plot the degree distribution of the coronavirus proteins. Which two proteins have the highest degree? What, if anything, is known about role of each of these two proteins during infection?

```
# Coronavirus proteins have a nonzero out degree
g <- g %>%
    activate(nodes) %>%
    mutate(out_degree = centrality_degree(mode = "out"))
coronavirus_proteins <- g %>%
    filter(out_degree > 0) %>%
    data.frame()
hist(coronavirus_proteins$out_degree, main = "Degree distribution", xlab = "Degree")
```




```
# Find two proteins with highest degree
highest_degrees <- coronavirus_proteins %>%
    arrange(desc(out_degree)) %>%
    head(2) %>%
    pull(name)
```

- **The two proteins with the highest degree are ORF7b and ORF3a. Currently, neither protein is known to be essential for viral replication.**

5. Which human proteins are found with more than two different SARS-COV-2 proteins?

```
# Find number of incoming edges for each node
g <- g %>%
    activate(nodes) %>%
    mutate(in_degree = centrality_degree(mode = "in"))

# Filter for nodes with more than 2 incoming edges
human_proteins <- g %>%
    filter(in_degree > 2) %>%
    data.frame()
knitr::kable(cbind(human_proteins$name, human_proteins$in_degree) %>%
    `colnames<-`(c("name", "degree")), caption = "Human proteins with $> 2$ different
    ↪  SARS-COV-2 proteins")
```

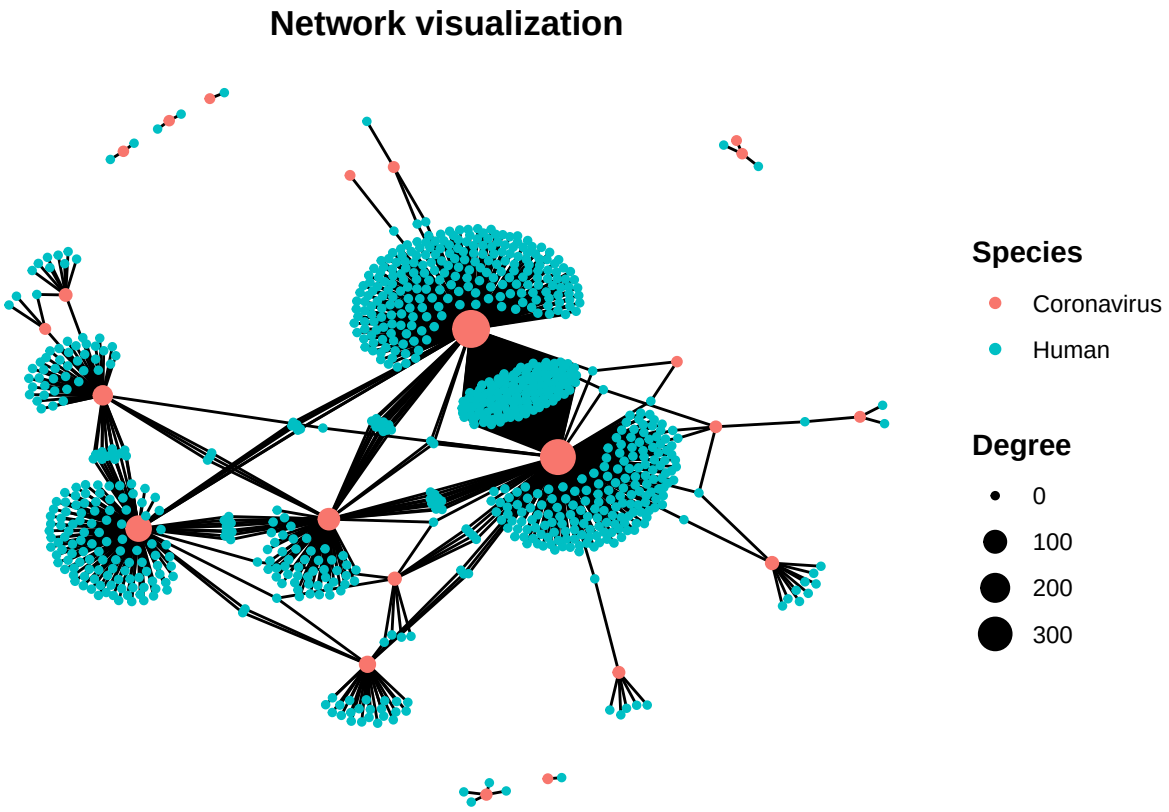
Table 1: Human proteins with > 2 different SARS-COV-2 proteins

name	degree
ATP6V1B1	3
GGH	3
FSCN1	3

name	degree
LRP12	3
ATP6AP1	3
CA12	3
ZDHHC5	3

6. Create a network visualization of this network, sizing nodes by their degree, and coloring by species.

```
layout <- create_layout(g, layout = "igraph", algorithm = "nicely")
ggraph(layout) + geom_edge_link() + geom_node_point(aes(size = out_degree, color =
↪  ifelse(out_degree ==
    0, "Human", "Coronavirus"))) + theme(plot.title = element_text(face = "bold",
    hjust = 0.5), legend.title = element_text(face = "bold"), legend.key =
    ↪  element_blank(),
    panel.grid.major = element_blank(), panel.grid.minor = element_blank(), panel.border
    ↪  = element_blank(),
    panel.background = element_blank()) + ggtitle("Network visualization") + labs(color =
    ↪  "Species",
    size = "Degree")
```



7. What is one way you could use this interaction network to learn about the biology of SARS-COVID-2 infection in humans?

- **The interaction network reveals that a relatively small number of SARS-COV-2 proteins are responsible for the majority of interactions with human proteins. This might give insight into what parts of the virus actually drive the infection process.**

5 Proteomics

1. Determine the amino acid sequence of the following spectrum using the provided monoisotopic mass table on the next page. The spectrum contains the full series of B and Y ions and a partial series of A and Z ions.

- **If the total mass of the peptide 1105.55 amu and the range of amino acid masses are 57-186 amu, we can estimate the length of the peptide:**

$$\begin{aligned} &= \frac{1105.55}{\frac{57+186}{2}} \\ &= \frac{1105.55}{121.5} \\ &\approx 9 \end{aligned} \tag{1}$$

The amino acid sequence is `(I/L)WSVCDQR`**.**

2. Describe briefly how you figured out the sequence.

- **Starting from the left side of the spectrum, I sequentially subtracted residue masses until reaching the other end. Each time, I compared the difference with the provided monoisotopic mass table to determine if there was a match. If there wasn't, I calculated the difference using the next reading to the right.**

3. How would a Post-Translation Modification of a peptide be identified from a mass spectrum?

- **Because a Post-Translation Modification affects the molecular weight of the peptide, this would be picked up from a mass spectrum which by definition is sensitive to the sample mass.**

6 Project Update

1. Submit a separate R file containing all new code that has been written by you (as an individual) for your final project since the project update that was due November 19th.

- **Please see the attached files** `svoboda.pdf` **and** `svoboda.rmd`**.**

```
sessionInfo(package = NULL)
```

```
## R version 4.1.1 (2021-08-10)
## Platform: x86_64-apple-darwin17.0 (64-bit)
## Running under: macOS Big Sur 10.16
##
## Matrix products: default
## BLAS:   /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRblas.0.dylib
## LAPACK: /Library/Frameworks/R.framework/Versions/4.1/Resources/lib/libRlapack.dylib
##
## locale:
## [1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] ggraph_2.0.5    igraph_1.2.9    forcats_0.5.1   stringr_1.4.0
##  [5] purrr_0.3.4     readr_2.1.1     tidyr_1.1.4     tibble_3.1.5
##  [9] tidyverse_1.3.1 tidygraph_1.2.0 caret_6.0-90    lattice_0.20-45
## [13] ggplot2_3.3.5   formatR_1.11    knitr_1.36      dplyr_1.0.7
##
## loaded via a namespace (and not attached):
##  [1] colorspace_2.0-2     ellipsis_0.3.2       class_7.3-19
##  [4] fs_1.5.0             rstudioapi_0.13      proxy_0.4-26
##  [7] listenv_0.8.0        farver_2.1.0         graphlayouts_0.7.2
## [10] ggrepel_0.9.1        bit64_4.0.5          prodlim_2019.11.13
## [13] fansi_0.5.0          lubridate_1.8.0      xml2_1.3.2
## [16] codetools_0.2-18     splines_4.1.1        polyclip_1.10-0
## [19] jsonlite_1.7.2       pROC_1.18.0          broom_0.7.10
## [22] kernlab_0.9-29       dbplyr_2.1.1         ggforce_0.3.3
## [25] compiler_4.1.1       httr_1.4.2           backports_1.4.0
## [28] assertthat_0.2.1     Matrix_1.3-4         fastmap_1.1.0
## [31] cli_3.1.0            tweenr_1.0.2         htmltools_0.5.2
## [34] tools_4.1.1          gtable_0.3.0         glue_1.4.2
## [37] reshape2_1.4.4       Rcpp_1.0.7           cellranger_1.1.0
## [40] vctrs_0.3.8          nlme_3.1-153         iterators_1.0.13
## [43] timeDate_3043.102    gower_0.2.2          xfun_0.27
## [46] globals_0.14.0       rvest_1.0.2          lifecycle_1.0.1
## [49] future_1.23.0        MASS_7.3-54          scales_1.1.1
## [52] ipred_0.9-12         vroom_1.5.7          hms_1.1.1
## [55] parallel_4.1.1       yaml_2.2.1           gridExtra_2.3
## [58] rpart_4.1-15         stringi_1.7.5        highr_0.9
## [61] foreach_1.5.1        e1071_1.7-9          lava_1.6.10
## [64] rlang_0.4.12         pkgconfig_2.0.3      evaluate_0.14
## [67] recipes_0.1.17       labeling_0.4.2       bit_4.0.4
## [70] tidyselect_1.1.1     parallelly_1.29.0    plyr_1.8.6
## [73] magrittr_2.0.1       R6_2.5.1             generics_0.1.0
## [76] DBI_1.1.1            pillar_1.6.4         haven_2.4.3
## [79] withr_2.4.2          survival_3.2-13      nnet_7.3-16
## [82] future.apply_1.8.1   modelr_0.1.8         crayon_1.4.1
## [85] utf8_1.2.2           tzdb_0.2.0           rmarkdown_2.11
## [88] viridis_0.6.2        grid_4.1.1           readxl_1.3.1
```

```
## [91] data.table_1.14.2    ModelMetrics_1.2.2.2 reprex_2.0.1
## [94] digest_0.6.28        stats4_4.1.1         munsell_0.5.0
## [97] viridisLite_0.4.0
```