Prediction of Signal Peptides using Competitive HMMs

Steven A. Gerick
gerick@kth.se

January 2019

Contents

Abstract

Localization of the proteins in an organism's proteome to particular cellular sites can provide information about the proteins' functions. Localization to the endoplasmic reticulum is normally indicated by a signal peptide, which can be difficult to detect from the amino acid sequence due to its similarity with the α-helices present in proteins localized as transmembrane proteins. Here, we present a relatively simple predictor for signal peptides given an amino acid sequence using a linear combination of HMMs, which tends to achieve 93.7% sensitivity and 95.3% specificity for the presence of a signal peptide, with only slightly lower performance on a dataset of transmembrane proteins.

1 Introduction

One of the primary goal of proteomics is to document the functions of proteins. As the organelles of a cell are specialized, the localization of a protein to a specific region of the cell typically indicates that the protein serves some function related to that region's function in cellular metabolism. One primary class of proteins is transmembrane proteins, which typically are used for signal transduction, membrane transport, and energy conversion [4]. A similar class is that of proteins containing signal peptides, which are processed specifically at the intracellular membrane of the endoplasmic reticulum and then possibly transported to other membranes of the cell [3] [1]. The signal peptide and transmembrane regions of the transmembrane proteins have similar structure, namely a long central chain of hydrophobic amino acids [2], which makes differentiation of the two from sequence analysis difficult. In addition, some of the signal peptides behave like transmembrane proteins once they are cleaved from the mature protein (though all of these are viral), indicating that both use some of the same mechanisms for membrane attachment [3]. Development of a better classifier for the presence of a signal peptide would allow automatic identification of the type of membrane a protein is bound for, and therefore insight into the protein's function.

Several models have been used before to predict the presence of signal peptides in a protein given only its amino acid sequence [2], including HMMs such as the Phobius system presented in [2].

Our dataset consists of amino acid sequences and annotations on the signal peptide and transmembrane regions they correspond to.

2 Method

The dataset annotations for the amino acid sequences provide most of the states we will use for the HMMs:

'n' (N-terminus) Positively charged N (front) tail of a signal peptide

'h' (hydrophobic) Hydrophobic interior of a signal peptide

'c' (cleave) non-hydrophobic end of a signal peptide

'C' (cleavage site) last amino acid in a signal peptide

'o' (outside) non-cytoplasmic (exterior) region of a transmembrane protein

'O' (long-looped outside) globular region of a transmembrane protein, which might otherwise prefer the exterior region if its acids were exposed

'i' (inside) cytoplasmic (interior) region of a transmembrane protein

'M' (membrane) region of a transmembrane protein that passes through the membrane

In addition, the following states were not present in the original annotation, but were added in preprocessing as they resulted in better classification results

's' (start) First position of any sequence, regardless of the existing annotation

'S' (second) Second and third positions of any sequence, only for known signal peptides

'g' (arbitrary letter) Position immediately before cleavage site

'v' (arbitrary letter) Position two acids before cleavage site

'r' (arbitrary letter) Position three acids before cleavage site

Better performance was achieved when states 'o', 'O', and 'i' were treated identically. Our final list of 11 states is 'n', 'h', 'c', 'C', 's', 'S', 'g', 'v', 'r', 'x', and 'M', where 'x' is the merged states of 'o', 'O', and 'i'.

We must also define classes for our amino acids. Figure 1 (see miscellaneous figures section at the end of the paper) provides a starting point:

Positively charged side chains (RHK)

Negatively charged side chains (DE)

Polar uncharged side chains (STNQ)

Cysteine (C)

Selenocysteine (U)

Glycine (G)

Proline (P)

Hydrophobic side chains (AVILMFYW)

Better performance was achieved by splitting several of these classes and merging others. The final classifier uses the following classes instead:

Positively charged side chains (RHK)

Negatively charged side chains (DE)

Primary alcohol side chains (ST)

Carboxamide side chains (NQ)

Glycine (G)

Proline (P)

Sulfur-containing side chains (CM)

Small hydrophobic side chains (AV)

Isoleucine (I)

Tyrosine (Y)

All other hydrophobic side chains (LFW)

Anything else (X) No sequences contained Selenocysteine (U) or pyrrolysine (O). X was the only "ambiguous" amino acid seen in any sequences, and it only appeared in four sequences, all signal peptide-containing.

Increasing the number of free parameters by differentiating all amino acids and all state labels resulted in much lower performance, most likely because this reduced the sample size for individual state transitions and amino acid observations. This degraded performance much faster than the increased number of free parameters could model the data, even when validating using the training dataset, indicating that the number of parameters would need to be much higher than this model allows for for overfitting to be a concern. Renormalization of models by the number of parameters to prevent overfitting was therefore deemed unnecessary.

We define the four HMMs as follows:

HMM1, trained only on non-signal-peptide non-transmembrane proteins

HMM2, trained only on signal peptide non-transmembrane proteins

HMM3, trained only on non-signal-peptide transmembrane proteins

HMM4, trained only on signal peptide transmembrane proteins

We do not use the classical HMM training method, where the parameters are unknown and must be estimated using only the observations. This method resulted in much lower performance because we were unable to exploit information in the state label during training. Each model instead has its starting distribution, transition matrix, and observation matrix set using statistics drawn from all of the relevant samples as follows:

1. The signal peptide and membrane state annotation is transformed into a state vector using the state representation described above
2. The amino acid sequence is transformed into an observation vector using the classes of amino acids described above
3. For every state, the observation matrix has its entry for the observation of the corresponding amino acid class in that state incremented by one
4. The starting distribution has its entry for the first state incremented by one
5. For every state after the first state, the transition matrix has its entry for the transition from the previous state to the current state incremented by one
6. The starting distribution is normalized to sum to one
7. For every state in the transition matrix, the outgoing state transition counts are normalized to sum to one. If there are no outgoing state transitions, the vector is set to be uniform with a sum of one.
8. For every state in the observation matrix, the emission counts are normalized to sum to one. If there are no emissions for this state (meaning this state was never encountered), the vector is set to be uniform with a sum of one.

At the end of this process, we have four trained HMMs. To classify a new amino acid sequence, we apply the Viterbi algorithm to each given the sequence, and extract the log probabilities of the resulting state sequences from each machine: p_1 from HMM1, p_2 from HMM2, p_3 from HMM3, and p_4 from HMM4. Our prediction is that a signal peptide exists if $p_1 + p_3 < p_2 + p_4$

Note that the state annotation is *only available during the training phase.* During validation or prediction, the prediction is made using *only the amino acid sequence.*

3 Results

Before final analysis of results, note that dataset sizes were the following:

non-signal, non-transmembrane 1087

non-signal, transmembrane 1275

signal, non-transmembrane 247

signal, transmembrane 45

Testing was run 100 times. During each test run, a different seed was used to partition the dataset into 80% training samples and 20% validation samples. Each run recorded its specificity across both negative datasets and sensitivity across both positive datasets, as well as accuracy for all four datasets separately. We can estimate a 95% confidence interval for all metrics using a two-tail student t test, as the number of samples is large enough that the standard deviation is much smaller than the distance between the measurements and the edges of the range(0 or 100%). The 95% confidence intervals for all metrics in percent are as follows:

Sensitivity 93.74 ± 0.30

Transmembrane only 92.22 ± 1.87

Non-transmembrane only 93.80 ± 0.29

Specificity 95.28 ± 0.23

Transmembrane only 89.55 ± 0.76

Non-transmembrane only 96.59 ± 0.22

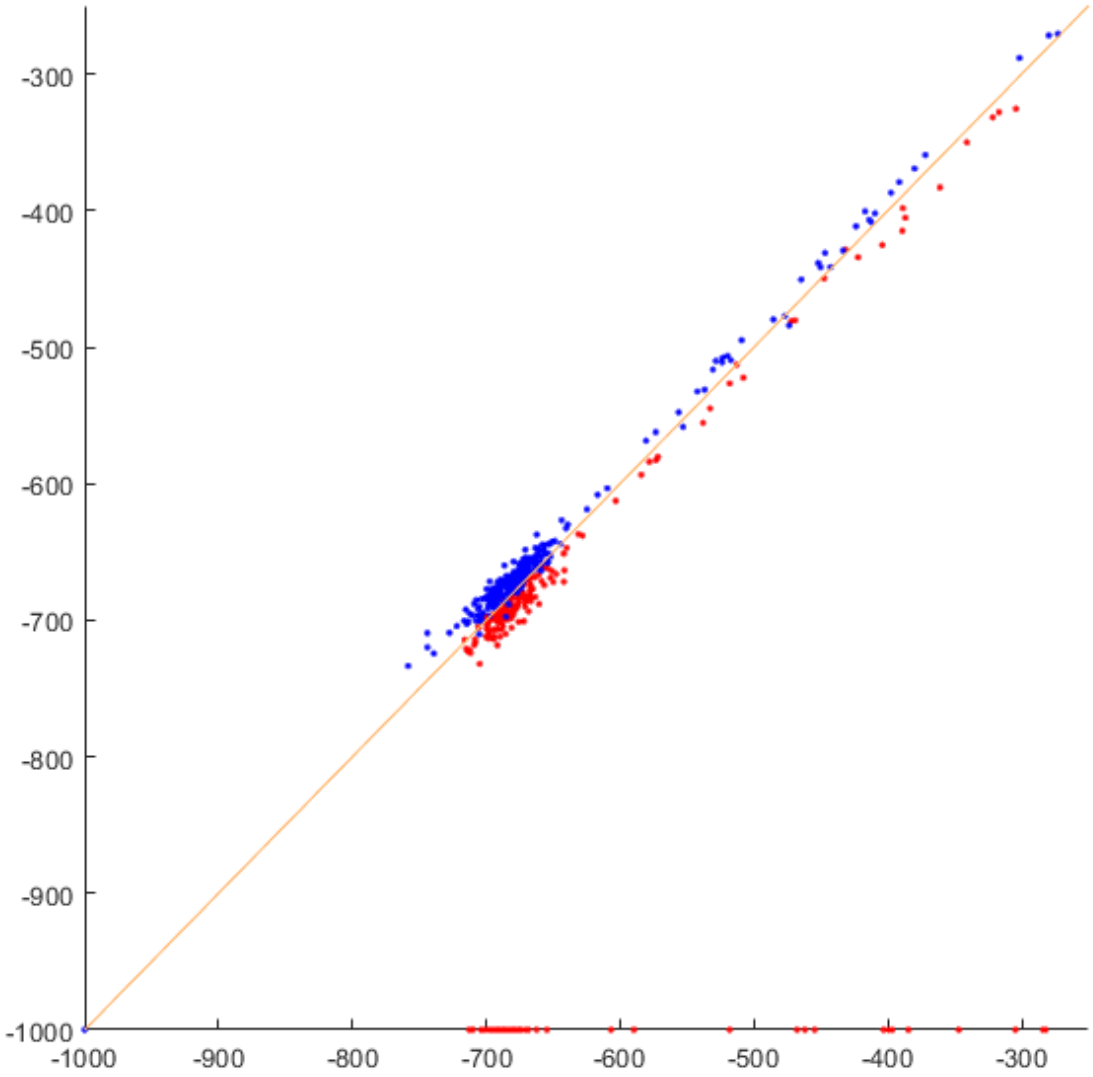

Figure 1: Points in one validation set, with the sum of the log-probabilities for the non-signal machines on the X axis, and the sum of the log probabilities for the signal machines on the Y axis. Sequences known to contain signal peptides are in blue, and sequences known not to contain signal peptides are in red. Sequences that are more likely to be produced by the negative machines have larger X coordinates, and sequences that are less likely to be produced by the positive machines have smaller Y coordinates. Sequences for which the positive machines are more certain than the negative machines (i.e. larger Y coordinate than X coordinate, the orange line) are classified as containing a signal peptide. Note that some of the negative examples produce very large negative Y values; this occurs when the signal tm machine produces a probability of zero that the sequence could have been a signal-peptide tm protein, which would normally produce an undefined log probability, which we instead limit to -1000 for display purposes. In the real classifier, this limit is instead -10 000, which means that a sample will always be classified as negative if that machine produces a probability of 0. The samples at (-1000, -1000) correspond to cases where all machines except the non-tm non-signal predict probability of 0, so these are always predicted to be negative examples.

I downloaded two entire proteomes (all gene transcripts that are protein coding) from BioMart (https://www.ensembl.org/biomar that for *Homo sapiens* and that for *Drosophila melanogaster*. The *Homo sapiens* dataset contains peptide sequences from chromosomes 1 to 22, X, and Y. The *Drosophila melanogaster* dataset contains peptide sequences from chromosomes 1L/R, 2L/R, 3L/R, and 4.

Some of the downloaded sequences contained selenocysteine, and had to be ignored because none of the training samples contained it. For *Homo* the percentage of such sequences was about 0.11%For *Drosophila* only about 1.6% of sequences were unavailable, but . The number of valid sequences were

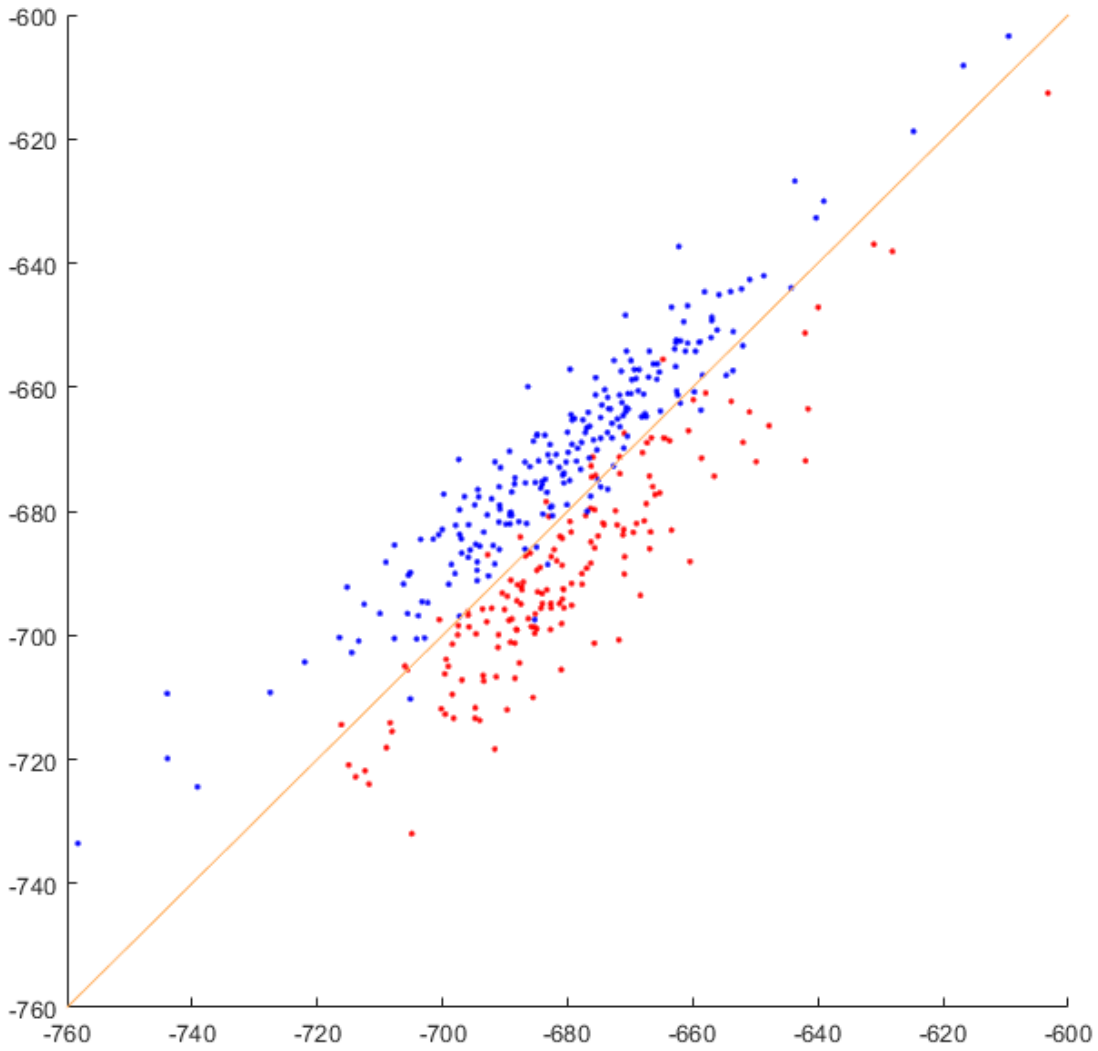

Figure 2: The previous plot, but showing only the dense inner region of points. Sensitivity is lower than specificity, meaning blue (positive) examples are more likely to be located on the wrong side of the classifier curve than red (negative) examples.

Counts of protein sequences expected to contain a signal peptide were 31632 of 82944 for *Homo sapiens* and 13818 of 24687 for *Drosophila melanogaster*. The human genome contained around 3.4 times more protein sequences but only 2.3 times as many predicted signal peptides, indicating that the proportion of genes coding for a signal peptide may be correlated with genome size.

For *Homo sapiens*, [2] calculated approximately 15.9% of the genes in the dataset were predicted to have a signal peptide, while 34.1% were predicted to have either a signal peptide or be a transmembrane protein. The classifier in this paper predicts about 38.1% of the proteins in a much larger dataset for the same species have a signal peptide. This indicates that the statistics on the provided labelled dataset provide nowhere near a good estimate of specificity or sensitivity for the full proteome, that the proportion of signal-peptide containing proteins being added to the database has risen sharply since that paper made their count, or both.

4 Discussion

Sensitivity was lower than specificity overall (there are more false negatives than false positives). Both measures were better when measured only on the non-transmembrane proteins compared to the entire dataset. Oddly, sensitivity was higher than specificity when considering only transmembrane proteins, but this could be due to overfitting in the the small size of the transmembrane signal protein dataset.

Sensitivity and specificity are high enough to accurately estimate signal peptide counts in a proteome, but not quite high enough for this method to be used in isolation to conclusively identify a protein containing a signal peptide. For that, it should be combined with other measures if possible to increase the strength of the classifier.

Because the HMMs are memoryless, they cannot count the number of times they have stayed in a particular state and use it to judge the emission probabilities. This does not allow the models for example to consider sequences with exactly 7 'c' states as more likely than sequences with exactly 6, which might be expected otherwise from looking at Figure 12 (see the following sequence logos section). Similarly, a sequence of at least 8 'c' states seems extraordinarily unlikely according to figure 12, but the HMMs would penalize the additional 'c' from 7 to 8 as much as an addition c from 7 to 8, as long as the predicted emissions are consistent. A model that can fully incorporate a short term memory model, such as an LSTM neural network, would be able to model such a timed state model better, especially considering than the peptidases that cleave the signal peptides presumably also operate as a sort of LSTM across the amino acid sequences.

Although the HMMs are unlikely to overfit the data due to the large dataset size, it can still be extremely difficult to determine which parameter combinations are better than others in a statistically significant way. With the complex trade-off between specificity and sensitivity, even adding a simple bias to the linear classifier can often increase performance by shifting error in one metric to the other. By the time the model reaches 90% performance in any metric, absolute number of misclassifications on the training set drops so low (to around 5-10 for each category) that even small gains in performance are undetectable or masked by noise. All of these concerns have a straightforward but somewhat costly solution: increase the dataset size and run testing more times to increase the significance of any positive findings. The first is impossible with the provided fixed dataset, and the second provides performance precision that scales only with the square root of the number of tests. Eventually, I decided that any new attempted combination of state and amino acid differentiations would be kept only if the sum of specificity and sensitivity increased and both remained above some floor (80% in early stages and progressing to 90% in later stages). Number of tests for every parameter change naturally increased as the performance improved, to account for the problems described above.

5 Materials: Sequence logos

Sequence logos generated from various reference points in the transmembrane, non-transmembrane, signal, and non-signal amino acid sequences reveal semi-conserved sequence points or relative abundances of certain types of amino acids.

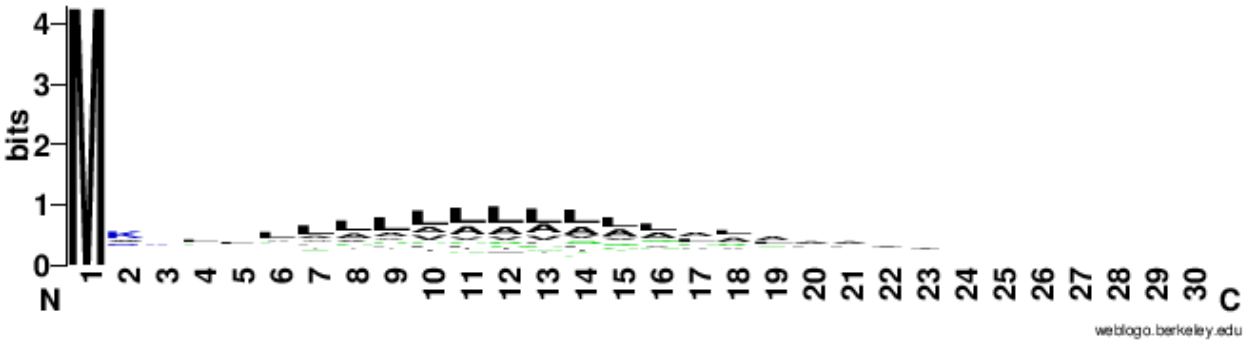

Figure 3: Sequence logo for start of non-tm signal peptide protein. Shows a clear preference for starting with M, and a relative abundance of amino acids L and A in positions 5 to 15

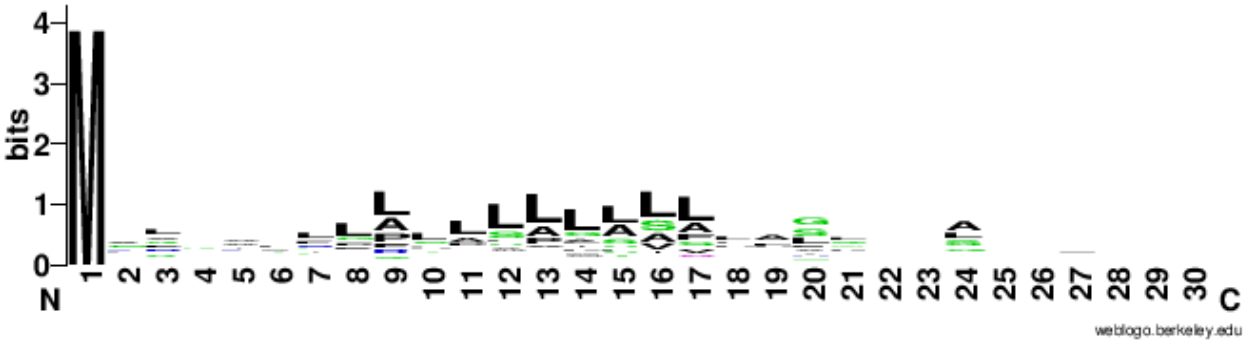

Figure 4: Sequence logo for start of tm signal peptide protein. Shows a clear preference for starting with M, and a relative abundance of amino acids L and A in positions 5 to 15. The bins are noisy due to small sample size (45)

Figure 5: Sequence logo for start of non-tm non-signal peptide protein. Shows a clear preference for starting with M, though it is much less pronounced than for any protein that is either a tm protein or has a signal peptide.

Figure 6: Sequence logo for start of tm non-signal peptide protein. Shows a clear preference for starting with M.

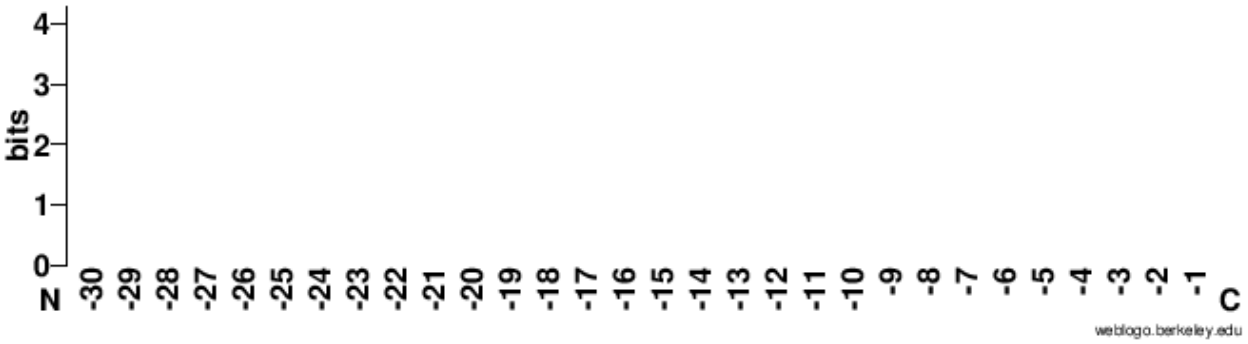

Figure 7: Sequence logo for last 30 amino acids in a non-signal non-tm protein. No sequence information can be gained from these for this class.

Figure 8: Sequence logo for last 30 amino acids in a signal non-tm protein. No sequence information can be gained from these for this class.

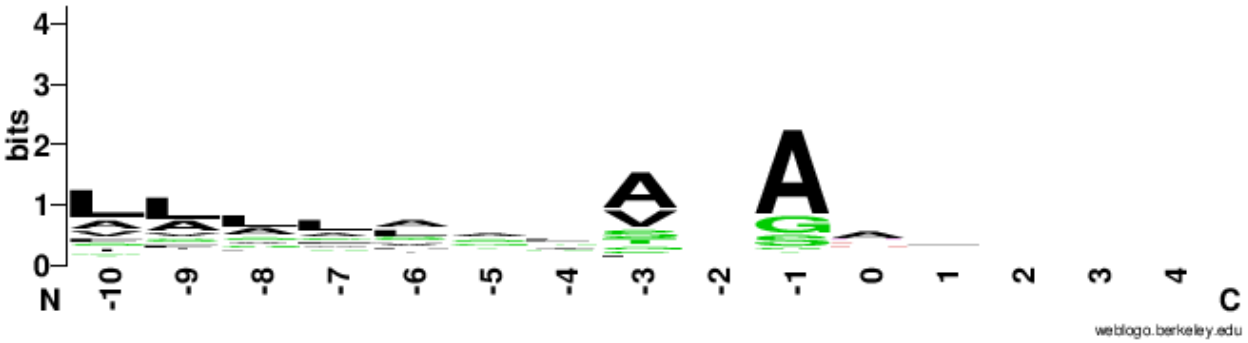

Figure 9: Sequence logo for the amino acids immediately before and after the cleavage point (0) for non-tm proteins with a signal peptide. Shows a clear preference for A or G in the position immediately before, no preference for two positions before, clear preference for A and V three positions before, and the hydrophobic A and L overabundances in the h region earlier in the sequence.

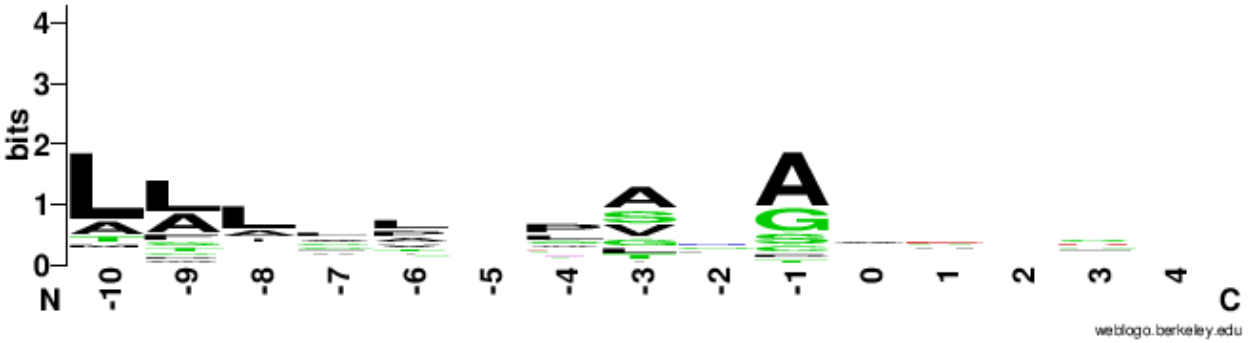

Figure 10: Sequence logo for the amino acids immediately before and after the cleavage point (0) for tm proteins with a signal peptide. Shows a clear preference for A or G in the position immediately before, no preference for two positions before, clear preference for A and V three positions before, and the hydrophobic A and L overabundances in the h region earlier in the sequence. This logo is noisy due to small sample size (45)

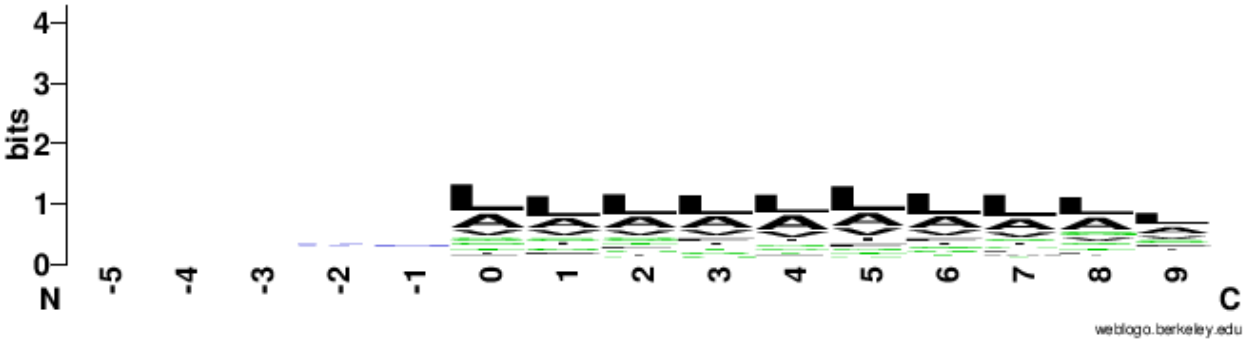

Figure 11: Sequence logo for amino acids immediately before and after the first marked 'h' state for non-tm proteins with a signal peptide. Shows a clear and almost constant distribution of peptides for several acids, indicating knowledge of the h region gives more information about where these hydrophobic acids will be compared to the start of the sequence or the cleavage site. Shows almost no information present in the prior n region, indicating that the n and h region mechanisms are not related.

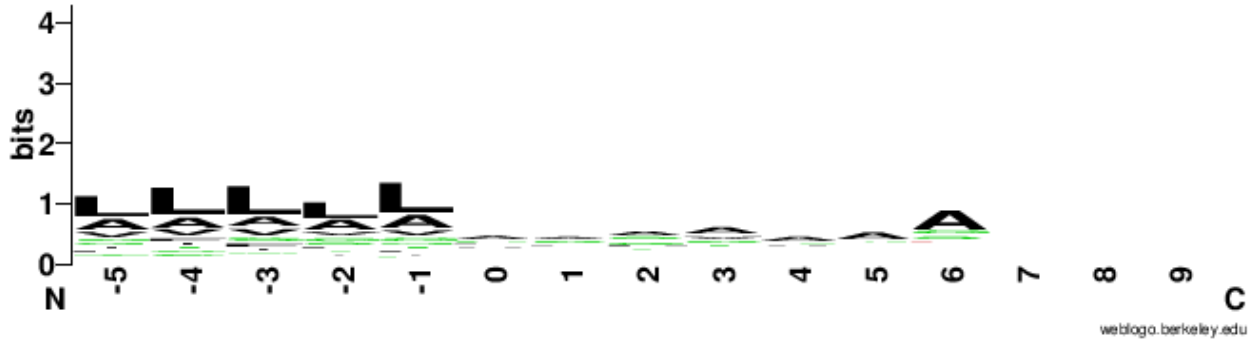

Figure 12: Sequence logo for amino acids immediately before and after the first marked 'c' for non-tm proteins with a signal peptide. Shows the same constant preference for L and A as the previous figure (because this guarantees that the previous acids are in the h region). Information in the positive indices indicates that the content at the cleavage site is independent of distance to the cleavage site, and that the cleavage site is almost never more than 6 peptides after the first 'c'.

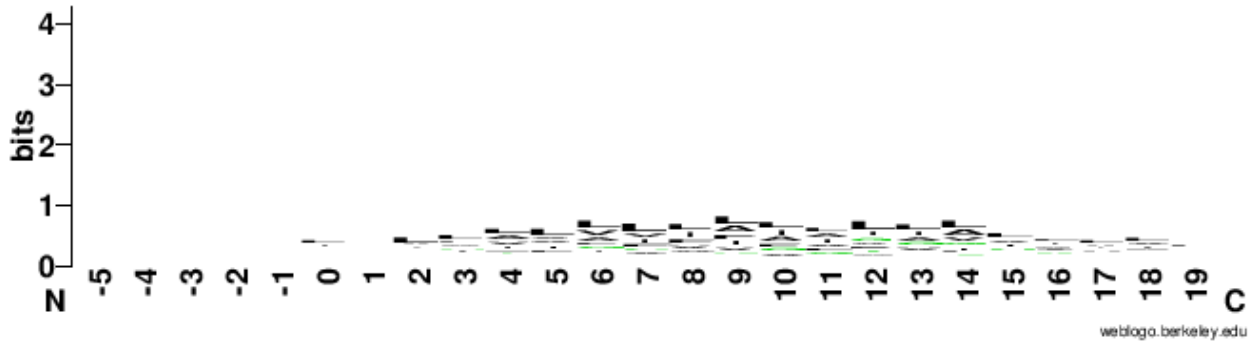

Figure 13: Sequence logo for amino acids immediately before and after the first marked 'M' for tm non-signal proteins. Shows a superficially similar information content to the h regions of signal peptides, but the signal is weaker and prefers L and I instead of L and A.

6 Miscellaneous Figures

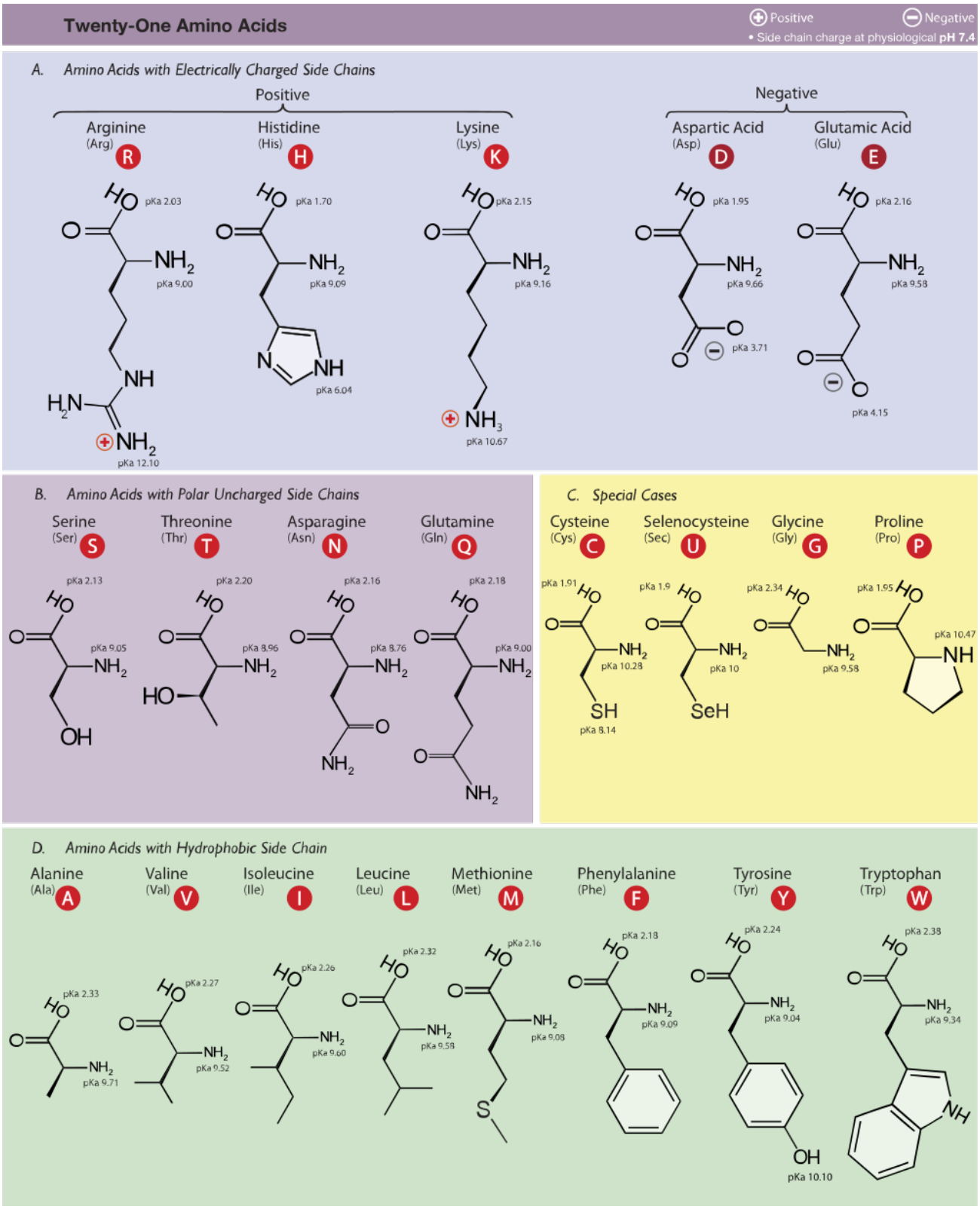

Figure 14: Classes of eukaryotic amino acids, made by Wikipedia user Dancojocari

References

[1] G Blobel and B Dobberstein. "Transfer of proteins across membranes. I. Presence of proteolytically processed and unprocessed nascent immunoglobulin light chains on membrane-bound ribosomes of murine myeloma." In: *The Journal of Cell Biology* 67.3 (1975), pp. 835–851. ISSN: 0021-9525. DOI: 10.1083/jcb.67.3.835. eprint: http://jcb.rupress.org/content/67/3/835.full.pdf. URL: http://jcb.rupress.org/content/67/3/835.

[2] Lukas Käll, Anders Krogh, and Erik L.L Sonnhammer. "A Combined Transmembrane Topology and Signal Peptide Prediction Method". In: *Journal of Molecular Biology* 338.5 (2004), pp. 1027–1036. ISSN: 0022-2836. DOI: https://doi.org/10.1016/j.jmb.2004.03.016. URL: http://www.sciencedirect.com/science/article/pii/S0022283604002943.

[3] Katja Kapp et al. *Post-Targeting Functions of Signal Peptides*. Available at https://www.ncbi.nlm.nih.gov/books/NBK6322/. 2009.

[4] Jin Xiong. *Essential bioinformatics*. Cambridge University Press, 2006, pp. 208–209. ISBN: 978-0-521-84098-9.