Exercise 2

Author: Paolo Fabbri
Date: 29 January 2026

Introduction

In this analysis, we explore the application of probabilistic modeling and graph theory focusing on protein sequence analysis and gene function prediction.

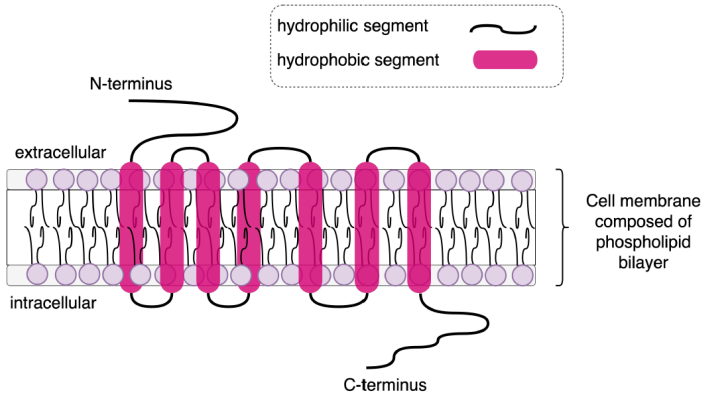
First, we address the challenge of segmenting protein structures, such as G-Protein Coupled Receptors (GPCRs). By constructing a two-state Hidden Markov Model (HMM), we aim to distinguish between hydrophobic (internal) and hydrophilic (external) regions. We utilize the Viterbi algorithm to decode the most probable sequence of hidden states behind an observed amino acid sequence.

The second part of our study shifts focus toward functional annotation using the Gene Ontology (GO). Here, we treat biological functions as nodes within a Directed Acyclic Graph (DAG). Finally, we analyzed how the probabilities produced by a training model can sometimes be inconsistent. In biological terms, this means the model might incorrectly assign a higher probability to a specific function than to its general parent category. To fix this, we applied a Post-hoc Consistency Correction

1. Viterbi Decoding and Two-State HMM

G-Protein coupled receptors (GPCRs) are a family of 7-pass transmembrane proteins that play vital roles such as light sensing (in the retina of the eye) as well as hormone and neurotransmitter signaling.They are defined by a very specific structural signature: they cross the cell membrane exactly seven times (Hydrophobic segments (IN), Hydrophilic loops (OUT).

In the preparation of this report, AI tools were utilized to support the technical analysis and the presentation of results. Specifically, ChatGPT was used to review the R and Python code, clarify the function of specific commands, and refine parameter settings. Gemini was employed to optimize the styling and readability of plots, improve figure legends, and both were used to assist in troubleshooting library compatibility issues encountered on the local machine. Despite the use of these tools for technical optimization and problem-solving, all biological interpretations, final conclusions, and the critical assessment of the models remain the authors' own.



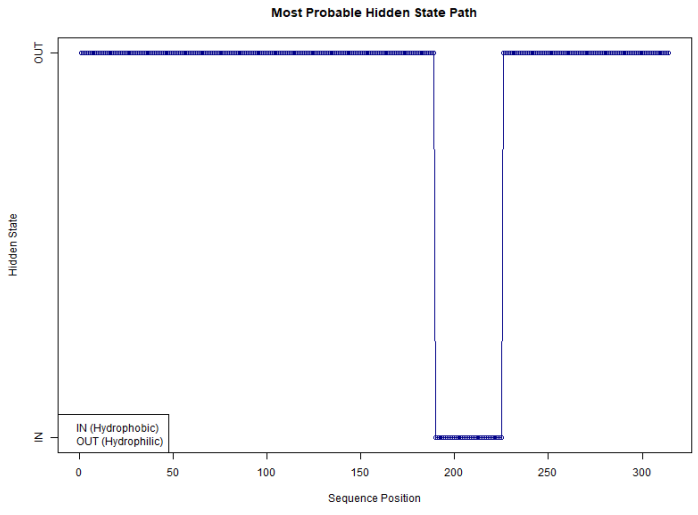
In the first part of the exercise, we implement the Viterbi decoding algorithm from scratch to analyze the structure of a GPCR protein.

We use the Viterbi algorithm to translate a raw sequence of amino acids into a sequence of biological states. The algorithm works by evaluating two main components simultaneously: Emission Probabilities and Transition Probabilities. By combining these, Viterbi inferred the most probable hidden state sequence that generated a given protein sequence at each position.

We began by building the Transition Matrix defined in the text, which represents the probability of switching between the IN and OUT states. Next, we transformed the raw amino acid counts (frequency table) into an Emission Matrix by normalizing the counts so that each row represents the probability of finding a specific amino acid in a given state.

We then implemented a custom function to execute the Viterbi algorithm (we assumed both states are equally probable at the initial stage and we mapped the IN and OUT states respectively to 0 and 1). To handle the very small probabilities associated with long protein sequences, we converted all values into base-2 logarithms, preventing numerical underflow.

Finally, we mapped the predicted states to numerical values and generated a piecewise plot.



Through the graph we can see that it is identified only one hydrophobic segment instead of the seven typically expected for a GPCR protein. This suggests that the initial transition and emission matrices are not yet optimized to capture the specific structural complexity of this receptor.

After that we score the 3 given sequences using the log probabilities from the Viterbi algorithm, assessing which of the sequences is most likely to have been generated by the 2-state HMM defined above.

Most probable sequence according to the 2-state HMM:
Most likely sequence: NP 149420.4
With a log-probability score of: -1336.655

In the second part of the first exercise we used the previous Viterbi algorithm to train a 2-state HMM to infer the optimal transition and emission probability matrices for the sequence seq1 (NP 001001957.2). We train the model for 100 iterations and use a tolerance of 10−9 for checking for convergence. We assume that both states are equally probable at the initial stage.

Here below is the comparison between the two transition matrices after the application of the HMM model.

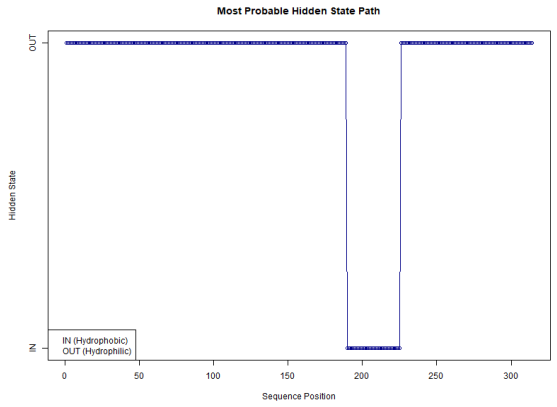
Initial Transition Matrix

Current State	IN	OUT
IN	0.80	0.20
OUT	0.05	0.95

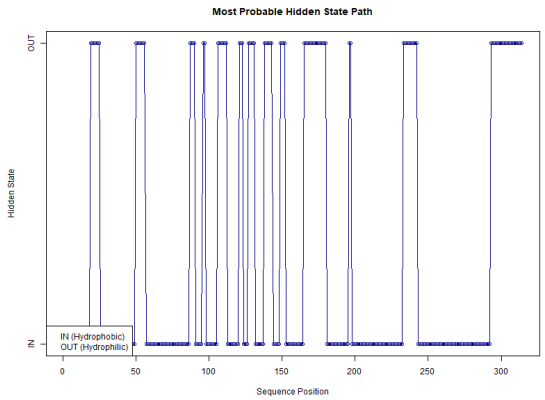
Final Transition Matrix

Current State	IN	OUT
IN	0.9011	0.0989
OUT	0.2009	0.7991

Initially, the HMM predicted only a single transmembrane domain. As observed in the first graph, the Viterbi Training failed to improve this result. This occurred because Viterbi updates parameters based solely on the single most likely path; if the initial parameters are not sensitive enough, the algorithm gets stuck in a local optimum. It essentially ignores other hydrophobic regions because they don't reach a high enough probability threshold in the first pass.



By applying the Baum-Welch algorithm, the model escaped this local optimum. Unlike Viterbi, Baum-Welch considers all possible paths (Forward-Backward probabilities). This allowed the model to "notice" the other hydrophobic segments that were previously below the threshold. However, as seen in the second plot, the model now predicts too many transitions. While we expect exactly 7 transmembrane passes, the trained HMM shows frequent "flickering" between states.



Despite the optimizations, the results obtained by applying the Viterbi decoding on the trained HMM did not yield the expected biological accuracy. While the Baum-Welch algorithm successfully identified the presence of multiple hydrophobic regions, it introduced excessive 'noise', failing to clearly define the canonical 7-pass transmembrane topology.

2. Probabilistic modeling of protein families with profile HMMs

We used a profile HMM derived from experimentally-verified members of the family to recognize (potential) new members of the GPCR family. We used a dataset about GPCR proteins responsible for the sense of smell in rats.

To build a representative model of the rat olfactory receptors, we first performed a Multiple Sequence Alignment (MSA) using the ClustalW heuristic. we observed that no columns consisted of a single symbol.

Then, we used the derived PHMM function to generate a profile HMM using the result of the full MSA previous values (we setted pseudocount = Laplace).

“Unfortunately, I cannot show the results here due to compatibility issues between the aphid library and my PC architecture. I am using a Windows PC with ARM architecture, and aphid does not currently run properly on it.”

3. Exploring the Gene Ontology (GO)

In this section, we used the goatools library in python to inspect the gene ontology graph.

We downloaded the basic version of the GO, using the wget command and we loaded the file as a dictionary.

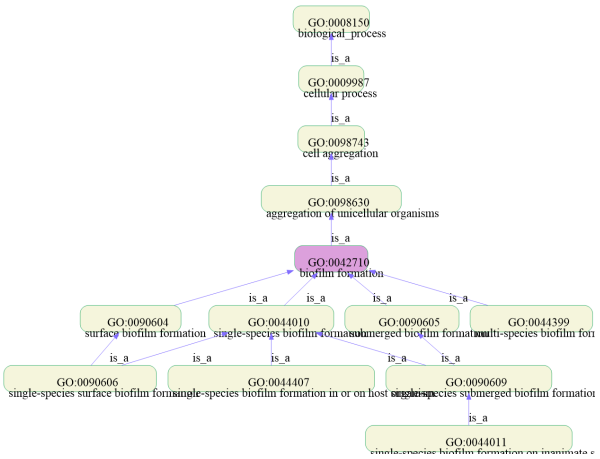
We first analyze some statistics about the total number of active and obsolete functional terms.

Total number of functional active terms: 42666
Total number of functional active + obsolete terms: 51842
Total number of functional obsolete terms: 9176

After we analyze some statistics of a specific term: GO:0042710.

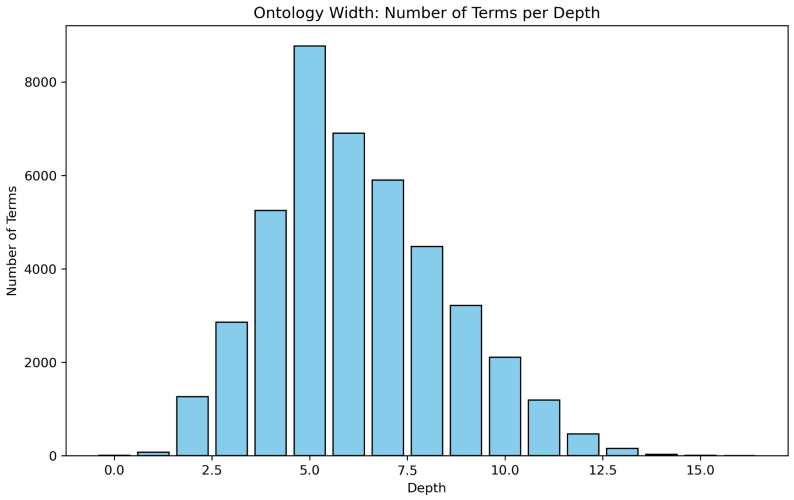
Name of term GO:0042710: biofilm formation
Depth of term GO:0042710: 4
Namespace of term GO:0042710: biological_process

Finally we plotted the lineage chart of the term showing its ancestors and children and counted the number of functional terms containing ligase in their textual definition.



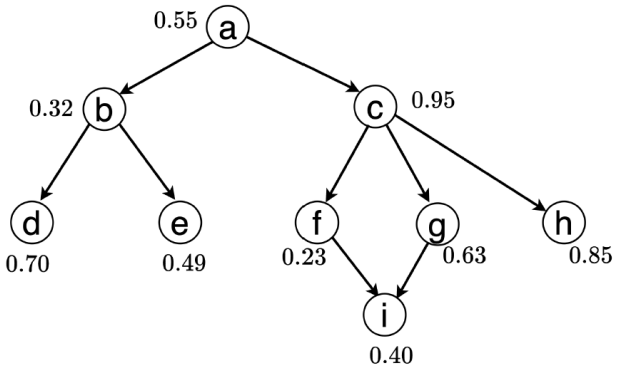
Number of functional terms containing 'ligase': 293

Then, I plotted a bar chart of the number of terms found at each depth of the full ontology, excluding obsolete terms.



4. Post-hoc Consistency Correction

In this section, we analyze the performance of a multi-label classifier applied to a Directed Acyclic Graph (DAG) representing a Gene Ontology subcategory (MF, CC, or BP). Unlike standard classification, GO terms are subject to the True Path Rule: if a protein is associated with a specific term, it must automatically be associated with all its ancestor terms.



To find the most plausible functional annotation, we enumerated all consistent subgraphs from the global graph provided in Figure above. A subgraph is considered consistent if, for every node included in the subgraph, all of its parent nodes are also included.

Consistent Subgraphs
Number of consistent subgraphs: 55
Number of subgraphs having 7 nodes: 8

In this section, we assumed that the raw probabilities assigned by the classifier (M1) were logically inconsistent. Since the classifier treated each node independently, it ignored the

hierarchical rules of the Gene Ontology, leading to "impossible" predictions where a child node could have a higher probability than its parent.

To fix this, we applied a post-hoc correction procedure. We recalculated the scores using a bottom-up approach, propagating the maximum probabilities from the leaves up to the root. This ensured that the hierarchy was respected.

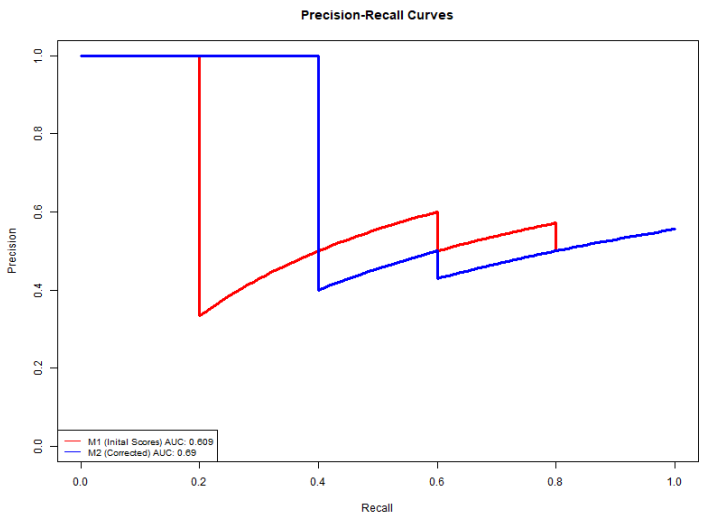
After we applied the correction with a threshold of >= 0.5, these are the results:

Final Scores after Correction:
0.95 0.7 0.95 0.7 0.49 0.4 0.63 0.85 0.4
Nodes in Corrected Subgraph (Threshold >= 0.5): a, b, c, d, g, h

---

Finally we reported the AUC values and we plotted the precision-recall curves corresponding to the classifier’s original predictions and the consistency correction model’s output.

AUC M1 (Originale): 0.6089529
AUC M2 (Corretto): 0.6895555



The results demonstrate that the consistency correction applied to create Model with the Post-hoc Consistency Correction was essential for achieving biologically sound predictions. By recalculating the probabilities post-hoc, we aligned the classifier's output with the fundamental consistency property of the Gene Ontology.

While the initial model (M1) produced "impossible" scenarios, the corrected model (M2) restored the logical structure of the graph. The improvement is quantitatively confirmed by the Precision-Recall curves. The AUC increased from 0.609 (M1) to 0.690 (M2), proving that a model respecting hierarchical constraints is inherently more accurate.

In conclusion, the post-hoc correction successfully transformed a series of independent, inconsistent probabilities into a coherent functional lineage. This confirms that for complex biological structures like the Gene Ontology, the True Path Rule is a necessary constraint to ensure that computational predictions remain both statistically valid and biologically meaningful.