Restricted Boltzmann Machines
and Sequence Homology Search

A thesis submitted by

Jackson C. Walker

to

Applied Mathematics

in partial fulfillment of the honors requirements

for the degree of

Applied Mathematics

Bachelor of Arts

Harvard College

Cambridge, Massachusetts

March 26, 2021
Abstract

Determining the evolutionary history, structure, and function of newly discovered proteins aids in the task of organizing the protein world around us. However, directly identifying evolutionary relationships, or homology, sequence by sequence is both costly and time-consuming. Creating methods of sequence homology search that can confidently and sensitively infer common ancestry tremendously aids scientists in the journey to classify proteins. This thesis interrogates the use of restricted Boltzmann machines (RBMs) for sequence homology search as their unique structure presents a new approach to profile-based homology search. By comparing the proposed model to established methods for homology in benchmark homology tests, this thesis seeks to see if the information RBMs learn, as illustrated by Tubiana et al. in a recent paper entitled *Learning Protein Constitutive Motifs From Sequence Data*, can be utilized in remote homology search. The data and findings serve as a first step of assessing the use of the RBM in sequence homology searching and a lesson on designing benchmark experiments with models that model sequences differently.
Acknowledgements

This thesis process has been something I never truly thought I’d reach the end of. Every year I heard about seniors submitting their theses, but I never connected the dots that one day that senior would be me. I’ve learned so much from both the actual research I conducted and this process itself that I truly don’t know if fall 2019 Jackson Walker would know the person writing these acknowledgments.

With that said, there are so many people without whom none of this would have been possible. First and foremost I must thank Dr. Sean Eddy for being a wonderful applied math advisor, MCB 112 professor, PI, and thesis advisor during my time at the College. Sean eagerly welcomed me into a corner of academia and STEM both unfamiliar and daunting to me at the time and throughout the thesis process, his support has made the impossible become possible. After Sean, no one else is as directly responsible for this project actually happening and moving forward as my project mentor, (soon to be Dr.) Grey Wilburn. With hardly any statistics and coding under my belt, Grey consistently sat with and literally taught me the information I needed to know to get the project off the ground and consistently progressing over the course of the last two school years. Throughout the thesis process, Grey has only continued to be an invaluable resource for me (also in thanks to Slack). My gratitude extends to everyone in the Eddy/Rivas lab (especially Tom and Nick and Sam for providing direct help in the project) for just welcoming me into the lab community!

I would be remiss to not also thank my wonderful mother for providing me with the foundation for the life I’m able to lead now– I’ll never be able to repay you for the sacrifices you made and continue to make for me, but hey, my 33rd birthday is just over 11 years away. I love you Ma!

Other people that I must also be sure to thank include: my father, for always believing in my abilities, my roommates, Natalie, Veronica, and Abbe for being the best support system I could ask for, and Robyn, for welcoming me into her home and family throughout the pandemic. Sincerely, thank you for helping me get to this wonderful perch of life I get to look out from.
Contents

1 Introduction ......................................................... 4
   1.1 Sequence Homology Search .................................. 4
   1.2 Profile Methods of Sequence Homology Search .......... 5
   1.3 Hidden Markov Models and Homology Search .......... 6
   1.4 Higher Order Models and Sequence Homology Search .. 8
   1.5 Restricted Boltzmann Machines ............................. 9
      1.5.1 Structure and Model ................................. 9
      1.5.2 Training and Determination ......................... 13
   1.6 Learning Protein Constitutive Motifs from Sequence Data; Recreating Results from Tubiana et al. ............... 14

2 Methods and Experimental Design ................................. 21

3 Data and Results .................................................. 26
   3.1 Primary Benchmark Results ................................ 26
   3.2 Experimental Improvements and Secondary Benchmark Results ......................................................... 31

4 Discussion .......................................................... 46
   4.1 Lessons from 1st and 2nd Benchmarks .................... 46
   4.2 Entropy Weighting Does Not Increase HMM Sensitivity in Benchmarks with Aligned Sequences ......................... 47
   4.3 Examination of Selected Data ............................. 50

5 Conclusion and Next Steps ......................................... 50

6 References .......................................................... 53
1 Introduction

Biological sequence comparison between distantly related sequences is paramount to understanding proteins. Pairwise sequence comparison methods can accomplish this task modestly, but begin to break down when attempting to perform remote homology detection, or in other words, for these models to recognize and accurately align the most distantly related sequences. This thesis will investigate profile based methods that help organize data and learn data specific to sequence composition and position that enable more powerful inference. Hidden Markov Models, the current standard for profile-based homology methods, are limited due to their site independence assumption; methods that account for nonindependence between sites, then, should be more effective at distinguishing related from non-related sequences.

Researchers have begun to investigate the use of higher order statistical models, such as Potts models, and neural networks, like restricted Boltzmann machines, in remote homology search. In 2019, Tubiana et al. showed that the features inferred by restricted Boltzmann machines trained on multiple sequence alignments have biological interpretations as a result of its use of hidden and visible layer connections in its structure. Specifically, this thesis will evaluate whether or not the information learned by restricted Boltzmann machines trained on multiple sequence alignments have applications for remote homology detection and alignment creation.

1.1 Sequence Homology Search

Scientists can determine protein structure and function experimentally, and while those methods are improving with respect to speed and cost efficiency, our ability to determine a sequence is improving even more rapidly [16]. Therefore, the need for other means to assess a protein’s structure from its sequence is established within the field.

Sequence homology seeks to answer whether certain genes or proteins share common ancestry. Macroscopically, homology can be used to describe physical features of an organism that share common ancestries in others, like vertebrae and tail bones. Homology at the microscopic level evaluates for common ancestry at the amino acid or nucleic acid sequence level for proteins and genes [2]. Establishing some means to infer homology and discover shared ancestry can allow researchers to denote protein function from sequence alone. Ho-
mology searching then is the process of determining if two sequences are related. Further, remote homology search applies the same methodology to assess whether two sequences could be possibly *distantly* related.

One approach to sequence homology search are pairwise-comparison methods, which compare sequences over the set of possible pairings to detect homologous sequences. The most successful pairwise-comparison methods, BLAST (Basic Local Alignment Search Tool) [9] and FASTA (and its immediate predecessor FASTP) were introduced in the late eighties and nineties [10]. Homology exists as a binary, as sequences either are related or are not. Therefore defining homology from statistical significance is tied to a number of intrinsic and extrinsic factors while searching. BLAST, FASTA and other pairwise-comparison tools are limited in that their information comes from one other sequence; biologically and mathematically, there is simply only so much information that any one sequence can tell you about another as scores generated are not position or family specific [9].

Grouping proteins according to functional and sequential similarity has helped decrease the number of searches required for a given sequence. In all, homology search has been used to classify hundreds of millions of proteins into over eighteen thousand families on the database PFAM [27]. A protein family can be defined as a group of proteins that biologists agree share significant common ancestry regardless of genus or species [3]. Often, this connotes that the set of proteins have a similar function or structure. For example, the HLH domain (PF00010) is a highly studied and well-documented family of helix-loop-helix DNA-binding proteins which can be found in mammals, fungi, plants and insects [27]. As its name conveys, all members of the family contain an identifiable helix-loop-helix domain that dictates its DNA binding function. Most inferred sequence families are made through the process of creating alignments for a set of sequences that properly recognizes structural and sequential motifs. Attempting to align sequences against a well-studied alignment can often provide some basis to identify homology.

### 1.2 Profile Methods of Sequence Homology Search

Profile-based methods for homology search take advantage of the creation of smaller related groups of proteins and represent those groups with statistical models to infer biological information about the given sequences. Profile-based methods take in a given set of sequences (typically consisting of families or subfamilies of proteins) to be trained on. This
training provides position-specific scores that reflect amino acid composition throughout each alignment column with which the profile can use to assess homology [12]. Because of their ability to provide site specific information, profile methods present cutting edge options for remote homology search.

Specific models that can be used for profile-based remote homology search include statistical inference models, such as Hidden Markov Models (HMMs), Potts models [13] (PMs) and covariance models (CMs) [24]. While more work is being done to investigate the potential for more complicated statistical models like PMs and covariance models, HMMs have found a niche within homology searching, with a multitude of programmatic implementations existing.

1.3 Hidden Markov Models and Homology Search

The Hidden Markov Model is a statistical extension of the Markov Chain. Profile HMMs were introduced to sequence homology search by Krogh et al. in the 1990’s [15]. Since then, a plethora of programs and algorithms utilizing profile HMMs have been developed and employed around protein investigation.

Profile HMMs represent columns in an MSA with states in a Markov chain. Transitions between each state k and l are given by transition probabilities $t_{kl}$ and each state emits a character $a$ with emission probability $e_{k}(a)$. An HMM generates a sequence by selecting a state path $\pi_1$ to $\pi_n$ based on the transition probabilities and emitting a character $a$ from each selected state using emission probabilities corresponding to $e_{\pi_i}(a)$. Profile HMMs give a joint distribution over an observed sequence $x$, composed of $x_1, x_2, \ldots, x_n$ which represent the monomer units of a biological sequence, and a hidden state path $\pi$ as follows [16]:

$$P(x, \pi) = t_{0\pi_1} \prod_{i=1}^{L} e_{\pi_i}(x_i) t_{\pi_i,\pi_{i+1}}$$

HMMs have helped to model and answer many problems in biology. Profile HMMs, however, serve as a structural extension of HMMs that have been modified specifically for the purposes of homology search and alignment. Several different implementations of profile HMMs exist for remote homology search including the HHSuite built by Johannes Söding and HMMER written by Sean Eddy [25][14]. The HMMER program suite also includes multiple alignment producing methods that allow for powerful remote homolog
detection and alignment generation. Further, HMMER contains several different algorithms and programs to assess the likelihood of homology between a profile and given sequences [14]. Certain scoring methods have tradeoffs in speed, performance and sensitivity, but all accomplish the same goal of assessing common ancestry.

A constructed profile HMM on a given MSA resembles the following structure:

\[
\begin{align*}
    e_m(C) &= .2 \\
    e_m(A) &= .2 \\
    e_m(P) &= .2 \\
    e_m(H) &= .2 \\
    e_m(V) &= .2
\end{align*}
\]

Figure 1: Example Profile HMM. Match columns are represented by \( M_k \), delete columns are represented by \( D_k \) and insert columns are represented by \( I_k \). B and E correspond the start and stop positions of the alignment and arrows correspond to allowed transitions between states. Emission probabilities for state \( M_1 \) \( e_{m_1}(a) \) and transition probabilities between match states \( t_{m_i,m_{i+1}} \) are illustrated accordingly in the model.

Figure (1) represents a three column-wide alignment represented by \( M_i \) for \( i = [1,3] \) for the three match states. Between each match state there can be insertions, represented by \( I_j \) for \( j = [0,3] \), as insertions can both interrupt match columns and also flank either ends of the alignment. Every match column also runs parallel to a deletion represented by \( D_k \) for \( k = [1,3] \), as each match column in an MSA can either contain an amino acid or lack a character. Finally, B and E represent the start and stop (beginning and end) positions.
Arrows represent which transitions are allowed. Note that self-looping insert transition probabilities allow for any length sequence to be generated.

The structure of profile HMMs allows HMMER to harness the power of modern computer programming for creating alignments. Utilizing the matrix of emission values, dynamic programming allows HMMER to find calculate alignment and sequence probability through two major algorithms. The Viterbi algorithm, while not the default of the HMMER suite, is a form of dynamic programming that optimizes alignment by computing the most likely hidden state path for the given observed sequence. HMMERs primary scoring algorithm, the forward algorithm, calculates the sum of the probabilities over all the possible hidden states for a given visible sequence [16]. For a sequence \(x\) and path of states or alignment \(\pi\), the HMM will express \(P(x, \pi)\). The Viterbi algorithm identifies \(\hat{\pi} = \arg\max_{\pi} P(x, \pi)\), and the forward algorithm computes \(P(x) = \sum_{\pi} P(x, \pi)\). Both algorithms exemplify HMMER’s power with respect to alignment generation and homology detection.

HMMER is, however, not without fault. The largest caveat of HMMs and other similar models is that they cannot model higher order structure and characteristics of proteins and protein families, because their structure assumes independence between sites or columns of an MSA. The above equation considers amino acid identity independent of all other sites, without accounting for any interrelation or covariation between amino acids at certain positions or for higher order structures [14].

1.4 Higher Order Models and Sequence Homology Search

Other higher order models have been introduced to sequence homology search and alignment with varying methods of success. Potts models serve as generalizations of the Ising Model. For a given Potts model, there exist \(N\) interconnected sites with \(q\) symbols/letters; the probabilities then for each site are influenced by the identity at all sites and all pairs of sites. Unlike HMMs, Potts models are distributions over a fixed length and only work with aligned sequences. Hidden Potts models are a cross between the state path-structure of a Hidden Markov Model and a Potts model. For \(v\) a match sequence, \(v_i\) the single position preference at match state \(m_i\), \(h_i(v_i)\) a single position preference for amino acid expressed by \(v_i\), and \(e_{ij}(v_i, v_j)\) representing pairwise energies between given states at alignment columns \(i\) and \(j\), Potts models, then, express pseudo-energy in their corresponding Hamiltonian equation as follows [13]:
\begin{equation} \mathcal{H}(v) = \sum_{i=1}^{N} h_i(v_i) + \sum_{k=1}^{N} \sum_{j=i+1}^{N} e_{ij}(v_i, v_j) \tag{2} \end{equation}

Each of the \( h_k \) and \( e_{ij} \) terms in the Hamiltonian are trained on a given MSA. Then, utilizing its estimated partition function, \( Z \), the probability of a sequence on a Potts model is given by the following:

\[ P(v) = \exp(\mathcal{H}(v)/Z) \tag{3} \]

Potts models show promising results for their applications to remote homology search, and hidden Potts models now have algorithms to generate alignments for MSAs. However, more experimentation is required to confirm their effectiveness in remote homology search.

Covariance models have been used in RNA sequence homology search. Their structure is based on an a binary tree which captures nested pairwise interactions corresponding to most nucleotide base pairs in RNA structure \[38\]. Like HMMs, however, covariance models cannot handle tertiary and non-pairwise structure.

Research has also indicated that some neural networks, while not traditional state-based statistical models, also have potential applications to sequence homology searching. Jerome Tubiana, Simona Cocco, and Remi Monasson published a journal article, \textit{Learning Protein Constitutive Motifs from Sequence Data}, showing that a restricted Boltzmann machine (RBM) has the ability to learn interactions in the structure and phylogeny from protein families in MSAs. This thesis will continue to investigate the results reported by Tubiana et al. and examine whether or not the RBM can be applied to remote homology search.

1.5 Restricted Boltzmann Machines

1.5.1 Structure and Model

A Boltzmann machine is a joint probabilistic network of neuron-like units that stochastically determine whether to be on or off \[17\]. The structure of a Boltzmann machine closely resembles that of a Hopfield network (in fact, Boltzmann Machines are an extension of Hopfield networks) but with the addition of stochastically determined hidden units or layers. In a Boltzmann machine, there exist a series of visible units, which receive information from and represent the environment (or data) the Boltzmann machine sees; in the context of remote homology search those visible units will be the given sequence. Boltzman machines
also contain hidden units, which represent underlying patterns or motifs within the training dataset the Boltzmann Machine learns from. Every unit within a Boltzmann Machine is connected by a weight that determines the cost of activating the specific visible-visible, visible-hidden or hidden-hidden relationship. Given the degree of connectivity between all units and layers, general Boltzmann machines have not yet proved useful for inference in a number of contexts; restricting connection between units, however, has generated Boltzmann machines that are capable of inference [18].

The restricted Boltzmann machine is an alteration on the structure of a Boltzmann machine that limits connections to hidden-visible relationships and every visible unit has a connection to every hidden layer. A restricted Boltzmann machine gives joint probability over two variables: \( v \) and \( h \). The visible layer, \( v \), is composed of sites \( v_1 \ldots v_N \) for \( N \) sites in the visible input. In the context of sequence homology, \( v \) is similar to \( x \) defined by the HMM, but \( v \) represents a fixed length aligned biological sequence including gaps. Each unit of the visible layer can occupy one of the twenty-one discrete values for each of the twenty amino acids and one for an alignment gap. The hidden layer, \( h \), is composed of sites \( h_1 \ldots h_M \) for \( M \) hidden units and contain values over the real numbers that represent various biological information on the set of sequences. The RBM then expresses the joint probability of \( h \) and \( v \):

\[
P(v, h) \propto \exp \left( \sum_{i=1}^{N} g_i(v_i) - \sum_{\mu=1}^{M} U_\mu(h_\mu) + \sum_{i,\mu} h_\mu w_{i\mu}(v_i) \right)
\]

The units of the equation can be interpreted as follows. The local potentials of the visible layer, represented by \( g_i(v_i) \), dictate the likelihood of a certain amino acid (or gap) at position \( i \). For the given visible layer or sequence the first term sums over all of the potentials for all \( v_i \) in \( v \). The local potential of the hidden layer is given by \( U_\mu(h_\mu) \), which similarly expresses a distribution over the values of \( h_\mu \) in \( h \). The second term then sums over all the hidden potentials for the given hidden layer. The third term ties together the visible and hidden layers with the weights matrix, \( w \). Containing entries along \( w_{i\mu}(v_i) \), each entry represents a connection from visible unit and value \( v_i \) to hidden layer site and value \( h_\mu \). Weights exist over the real numbers and are determined over training [19].

The number of weights and hidden units for a given RBM is a predetermined hyperparameter; Tubiana et al. defined \( M = 100 \) as the number of hidden units for their data. Like
the number of hidden sites, the form and distribution of the local potentials of a hidden unit was determined experimentally by Tubiana et al. $U_\mu$ can take on any form, or equation and set of parameters to train, according to the desired interactions between the hidden and visible layers. For example, a quadratic form of $U_\mu$ introduces two parameters that need to be trained and the resulting RBM distribution is Gaussian; this Gaussian RBM is actually equivalent to a Potts model [44]. In order to properly introduce higher order (and site dependent) couplings between the visible units, non-quadratic forms of $U_\mu$ were chosen. Specifically, double Rectified Linear Unit (dReLU) potentials were chosen as they allow for 4-parameter potentials at each hidden unit; these non-quadratic 4-parameter potentials are what offer the ability for RBM to move beyond the site independence assumption. $U_\mu(h)$ is dictated by the four parameters $\gamma_{\mu,a}$, $\gamma_{\mu,b}$, $\theta_{\mu,a}$, and $\theta_{\mu,b}$ with the following equation:

$$U_\mu(h) = \frac{1}{2}\gamma_{\mu,a}h_a^2 + \frac{1}{2}\gamma_{\mu,b}h_b^2 + \frac{1}{2}\theta_{\mu,a}h_a^2 + \frac{1}{2}\theta_{\mu,b}h_b^2$$  

(5)

With $h_a = \max(h, 0)$ and $h_b = \min(h, 0)$. These parameters and the overall hidden layer local potentials are determined through training.

A visual representation of the network structure as described is as follows:
Figure 2: Restricted Boltzmann Machine Model Structure. The visible layer $v$ and its units $v_1, v_2, \ldots, v_N$ appear below and the hidden layer $h$ and its units $h_1, \ldots, h_M$ appear above. The two layers are connected by weights $w_{i\mu}(v_i)$ which dictate the degree of connectivity between the visible and hidden unit. The local potentials of the visible and hidden layer $g_i(v_i)$ and $U_{\mu}(h_{\mu})$, respectively, represent the inputs the RBM receives from a given visible and hidden input.

Given a visible layer, $v$, and one hidden unit, $\mu$, the hidden layer will interpret the following input as what the RBM "sees" when handed an MSA:

$$I_{\mu}(v) = \sum_i w_{i\mu}(v_i) \quad (6)$$

Which effectively sums all of the weights connecting the visible sequence to a single hidden unit. Visible sequences that receive scores with large magnitudes (both positive or negative in sign) correlate to sequences that have a strong match to a given hidden unit. Conversely, sequences that receive low scores (close to zero) in magnitude indicate a poor match between to the hidden unit. This input value then sets the conditional probability of the activity of the given hidden unit, which is also influenced by the choice of form of $U_{\mu}(h_{\mu})$ [19].

The probability of a visible sequence $P(v)$, which is related to the effective energy of a
visible sequence \( E_{\text{eff}}(v) \), is then found through marginalizing over all the possible representations of the hidden layer. That marginalization reveals the following relations:

\[
P(v) \propto \exp \left( \sum_{i=1}^{N} g_i(v_i) + \sum_{\mu=1}^{M} \Gamma_{\mu}(J_{\mu}(v)) \right) = \exp \left( E_{\text{eff}}(v) \right) \tag{7}
\]

The equation consists of two terms: The first of which returns the sum over the local potentials of each visible site and the second, taking advantage of equation (6), marginalizes over all \( M \) hidden sites. \( \Gamma_{\mu} \) is the cumulant-generating function of the corresponding \( \mathcal{U} \) form as chosen in the model. The general form of \( \Gamma_{\mu} \) takes on the following form:

\[
\Gamma_{\mu}(I) = \log \int dh \exp \left( -U_{\mu}(h) + hI \right) \tag{8}
\]

The probability, or energy, of a sequence generated by the RBM trained on an MSA can be normalized by its corresponding partition function (denoted \( Z \)) through the following:

\[
P(v) = e^{-E_{\text{eff}}(v)} / \sum_{v'} e^{-E_{\text{eff}}(v')} \tag{9}
\]

In which the denominator can be defined as the partition function \( Z \). \( Z \) normalizes probabilities produced by the RBM such that these normalized probabilities can be compared to scores and output from other RBMs and models with other \( Z \)s. Calculating \( Z \) is both computationally expensive and mathematically difficult, therefore estimating \( Z \) is important to generate normalized log odds scores. The RBM can estimate \( Z \) using Annealed Importance Sampling (AIS), which estimates the partition function ratios of each sequence’s respective partition function value [20].

### 1.5.2 Training and Determination

The RBM training procedure uses the principle of gradient ascent. The given training algorithm seeks to stochastically optimize (maximize) the probability of the MSA parameters given the parameters. For \( \theta \), the set of RBM variables in \( P(h, v) \), and assuming independence between each of the \( N \) sequences in the MSA, the log probability of the sequences can be given as follows:

\[
\log(P(\text{MSA}|\theta)) = \sum_{v \in \text{MSA}} \log P(v|\theta) \tag{10}
\]
Equation (10) then simply returns the sum of the log probabilities for each sequence over an MSA given a set of parameters [19]. By taking the partial derivative with respect to $\psi$, representing any one variable from the RBM (i.e. $g_i$, $w_{ij}$, etc.), the gradient can then be represented as:

$$
\frac{\partial}{\partial \psi} \log(P(\text{MSA}|\theta)) = N \sum_{v \in \text{MSA}} P(v) \frac{\partial E_{\text{eff}}(v)}{\partial \psi} - \sum_{v \in \text{MSA}} \frac{\partial E_{\text{eff}}(v)}{\partial \psi}
$$

Stochastic evaluation of the gradient uses Persistent Contrastive Divergence (PCD), which utilizes Markov Chain Monte Carlo (MCMC) simulation to estimate the derivative over small batches of approximation [39] [19]. With weights randomly drawn from a Gaussian distribution, $g_i$ potentials from the independent site model and hidden unit potentials $\gamma_{\mu,a}, \gamma_{\mu,b} = 1$, and $\theta_{\mu,a}, \theta_{\mu,b} = 0$, training iterations perform 100 batches to best optimize parameter values.

Higher order statistical models like RBMs are starting to make headway in homology searching because they are theoretically capable of representing complex higher-order relationships correlation between sites. However, the RBM’s structure presents its own set of limitations. Connections between hidden layers and visible sites make RBMs incompatible with dynamic programming methods that make developing an efficient alignment algorithm difficult. Additionally, the RBM assumes that all inputted sequences are fixed-length aligned sequences. If we want the RBM to train on unaligned sequences, it must then be able to be given an alignment and treat gaps as characters [19]. For the RBM to serve as a tool for remote homology search, it needs to be able to examine unaligned sequences utilizing its own alignment algorithm. RBMs have yet to be introduced to remote homology search, and the aforementioned limitations make benchmarking the RBM to other existing homology search tools challenging; such obstacles must be accounted for in experimental design.

1.6 Learning Protein Constitutive Motifs from Sequence Data; Recreating Results from Tubiana et al.

Turning to Tubiana’s et al.’s results, they outline that their RBM implementation uses the following settings and specifications. With dReLU hidden potentials and 100 hidden units, they trained their RBMs on MSAs from the following four families: the Kunitz domain (PF00014), the WW domain (PF00397), the HSP70 domain(PF00012), and a synthetic
lattice protein domain originally created by Mirny and Shakhnovich in *Protein Folding Theory: From Lattice to All-Atom Models* to model protein folding [27] [40]. Tubiana et al. developed an RBM using Python 2.7 that takes in training MSAs in aligned FASTA format and generates RBM objects that can be saved and utilized later. In order to support their result that the RBM can successfully learn biologically interpretable motifs from sequence data, Tubiana et al. utilized the RBMs to generate sequence logo plots for the weights from a given MSA and predict contacts for a given protein structure [19]. The sequence logos interpret site to site values for each amino acid at each weight [21]. Effectively, it represents $w_{ij}$ for a held $j$ over all possible $i$ in $N$. The x-axis represents each visible site $i$ and the y-axis displays the magnitude of the weight $w_{ij}(v_i)$. Like above, strong correlation between a weight and a single site is indicated by large (positive and negative) values, but amino acids that occur on the same side of the y-axis, tend to covary with one another [19]. This can be seen with the following logo example from a selected weight from Tubiana’s results for the Kunitz domain:

![Sequence Logo Plot Result From Tubiana et al.](image)

**Figure 3:** Sequence Logo Plot Result From Tubiana et al. [19]. Displaying $w_{i1}(v_i)$ for all $i$ within the sequence length $N$. The x-axis represents position $i$ along the sequence and the y-axis represents the magnitude of the weight of a given $v_i$, connecting to $h_1$. The position of a letter with respect to $y = 0$ indicates which residues are likely to present in the sequence. This weight demonstrates a salt bridge interaction between ionic residues at position forty-five and forty-nine within the alignment.

At positions forty-five and forty-nine, you see strong positive values for glutamic acid (E) and lysine (K) (and arginine (R)), respectively and strong negative weights for K (and R) and E respectively. The displayed weights most likely represent a salt bridge motif between the two positions from the Kunitz domain, as the placement of glutamic acid
and lysine residues close together indicates ionic interaction between the negatively and positively charged residues, respectively. This result suggests that the RBM is able to capture and recognize structural interactions within an MSA. Co-variation between amino acids is witnessed in the mirrored presence of K (and R) and E in the negative Y-axis; essentially, the sequence logo plot demonstrates that the RBM infers that if the residue at position forty-five mutates from a negatively charged glutamic acid to a positively charged lysine, there is a high likelihood that a paired mutation would occur at site forty-nine from a previous positively charged lysine to a negatively charged glutamic acid to maintain the ionic bonding between the two residues. This salt bridge can be viewed on the 3D protein structure for the 1BZ5 structure from PDB [45]:

![Figure 4: 1BZ5 3D Structure [45]. 1BZ5 is a sample structure taken from the Kunitz domain. The salt bridge motif suggested by Figure 3 can be seen between interactions (yellow dashed lines) between the glutamic acid at position 49 in cyan and arginine at position 53 in magenta.](image)

One of the other means Tubiana et al. use to test the RBM’s ability to capture structural
motifs is contact prediction of a protein structure. Their algorithm for the RBM utilizes
the concept of pairwise epistasis, or non-independence between mutations within the same
sequence. Consider a sequence \( v \) with \( a \) and \( b \) at sites \( i \) and \( j \), respectively. For the possible
mutations from \( a \) to \( c \) and \( b \) to \( d \), the RBM evaluates four mutations on the sequence: the
wild type \( v^{a,b} \), the two single mutants \( v^{c,b} \) and \( v^{a,d} \) and the double mutant \( v^{c,d} \). It then
evaluates the differential likelihood ratio of the four mutants, \( R(v;a,b,c,d) \) as follows:

\[
R(v;a,b,c,d) \equiv \log \frac{P(v^{a,b})P(v^{c,d})}{P(v^{c,b})P(v^{a,d})} \tag{12}
\]

The two sites are said to be non-independent if the value of \( R(v;a,b,c,d) \) is non-zero.
The RBM then generates effective coupling pseudo-energy \( J_{ij}^{\text{eff}}(a,b) \) by summing over all
possible amino acid options for \( c \) and \( d \) and averaging that value over \( q^2 \) for \( q \) equal to the
number of potential amino acid for \( c \) and \( d \). The resulting value of the effective coupling
pseudo-energy is then large and positive if the RBM likes contact at the site coupling,
negative if the RBM does not like contact and zero if it neither prefers or dislikes contact
at the coupling. By taking the frobenius norm of \( J_{ij}^{\text{eff}}(a,b) \) over all possible \( a \) and \( b \) at \( v_i \)
and \( v_j \), the RBM then generates contact scores for sites \( i \) and \( j \).

Their RBM result from contact prediction on the 5PTI structure from the kunitz domain
are pictured below, with the predicted contacts shown in Figure 5a and the positive predicted
value (PPV) plot quantifying those results in Figure 5b [22]:

\[
R(v;a,b,c,d) \equiv \log \frac{P(v^{a,b})P(v^{c,d})}{P(v^{c,b})P(v^{a,d})}
\]
Figure 5: Tubiana et al. Contact Prediction Results [19]. Figure 5a displays the true contacts for Kunitz structure 5PTI below the diagonal in green and sites the RBM predicted contact above the diagonal in red. Figure 5b plots the positive predicted value (PPV), defined as TP/P, for the pseudo-likelihood maximization with a Potts model (PLM), Boltzmann machine (BM) (Potts model with Boltzmann machine learning), Gaussian RBM (RBM that assumes site independence) and a dReLU RBM (4-parameter RBM). Predicted sites with PPVs close to 1 are correct predictions, indicating strong results across all methods.

Below the diagonal in green are the correct contacts from the biological crystal structure from the online Protein Data Bank (PDB). Above the diagonal in red are sites that the RBM determined are likely to be in contact. The PPV plot displays the positive predicted value, defined as the number of true positive contacts over the number of total positive reported against the rank of the contact score for a pair of sites. In their experiment, they compared the dReLU RBM against a Gaussian RBM, which only creates pairwise relations, pseudo-likelihood maximization (PLM) with a Potts model, and a general Boltzmann machine (BM) (a Potts model trained with Boltzmann machine learning) [19]. As many high-ranking site pairings had PPVs close to 1 for the dReLU RBM and the other methods, the dReLU RBM successfully predicted distant contacts involved in structure. As with logos, the above underscores that the RBM is able to interpret and produce information about family structure and higher order motifs solely from aligned sequence data.

Prior to thinking how to utilize RBM’s motif learning ability in other contexts, we sought to verify the numerical and empirical evidence reported by Tubiana et al. by utilizing their code to recreate their data. We trained a dReLU RBM on the the provided Kunitz training
MSA in aligned FASTA with 100 hidden units. Generated contact predictions were created for the 5PTI structure and Tubiana et al.'s approximation for $J_{\text{eff}}$ as calculating $J_{\text{eff}}$ is time consuming. Pictured below are the resulting sequence logo plots for $w_{i61}$ and the contact prediction for 5PTI [43].

Figure 6: Recreated Sequence Logo Result from Tubiana et al. See Figure 4 for definition of axes and interpretation. Results from RBM code developed by Tubiana et al. trained on Kunitz domain testing MSA. Weight $w_{i61}$ is plotted to display similar result to $w_{i1}$ from Tubiana et al. The corresponding glutamic acid and arginine at sites 45 and 49, respectively, show the same salt bridge motif for the Kunitz domain. The numbering of the weights do not match as the order of hidden units is random upon initialization of training.
Figure 7: Recreated Contact Prediction Result from Tubiana et al. Results from RBM code developed by Tubiana et al. trained on Kunitz domain testing MSA. Figure 7a shows the true contact map for Kunitz structure 5PTI in the left-hand figure and the contact prediction map created by the RBM with a colormap corresponding to the contact score. See Figure 5b for definition of axes and interpretation for 7b. The resulting PPV plot roughly matches the result from Tubiana et al., further showing that the RBM is able to predict contacts when trained on an MSA.

Save ordering (as there exists some randomness in the ordering of the RBMs hidden units, $W_{i\alpha}$ in RBM$_a$ might be $W_{i\alpha}$ in RBM$_b$), the RBM was able to reproduce almost identical results when implemented on our own system, indicating that the RBM is perhaps capable of learning constitutive family motifs.

The experiments and data presented in this paper will evaluate the following questions: How well can the RBM detect remote homologs in benchmark assessments across the PFAM database? Does the latent structure of the RBM help in detecting remote homologs? How does that detection compare to other leading models, particularly profile HMMs implemented with HMMER? What exactly are the familial motifs that the RBM is learning?
Most saliently, do RBMs have the potential to be a powerful tool to detect remote homology among protein sequences?

2 Methods and Experimental Design

In order to assess the effectiveness of Restricted Boltzmann Machines in remote homology search, we needed to construct an experiment that would both measure how well the RBM could distinguish remote homologs from unrelated sequences and compare how well the RBM did to other standard methods for homology search. Given their speed, the number of programmatic implementations and accompanying suite of tools and programs, we chose an HMM-based method provided for the best means of comparison for the RBM methods. Further, we decided that homology benchmarks with a known set of remote homologs and synthetic non-homologs, or decoys, between the models would provide the data required to answer and evaluate the research questions proposed by this thesis. At a high level, our benchmark procedure takes a known protein family and splits the MSA into two groups, one that the model will be shown for training (the training set) and one that contains sequences that are homologous to, but not very similar to, the set of sequences in the training set (the testing set). Then, the model is fed sequences from the testing set and the randomly generated non-homologous decoy sequences (the decoy set) to examine whether or not the model can detect true remote homologs from noise. This mimics the context of a real homology search problem but the ground truth of the sequences are known.

Our initial implementations of RBMs in homology search involved unpacking the code provided by Tubiana et al. Utilizing the functions and classes provided, we ran their scripts with a few well-documented families, most namely the globins (PF00042), to better understand the output of Tubiana et al.’s code such that the output could be reasonably compared to the output of HMM methods. In order to test the RBM’s ability to apply learned motifs to homology, we utilized the profmark benchmark construction procedure [46]. The profmark program takes in an MSA of known homologs and seeks to cluster them by pairwise percent identity. The largest of the two sets is then determined to be the training set. The smaller of the two clusters is then assigned to be the testing set. To distinguish which sequences were “different” enough to be included in the testing set, profmark utilized a 25% ID threshold, meaning any sequences in the testing set could share, at most, 25% identity
with any sequence in the training set.

Prior to being run through profmark, all families were screened for and eliminated of fragmented sequences, which can be defined as sequences inside of MSAs that are only partially aligned to the overall alignment, and sequences with less than 50% average sequence length. Due to the RBMs lack of alignment methods, fragmented and gappy sequences introduce aspects of alignment and homology that we determined are unfair in comparison to HMMs and HMMER.

Following profmark clustering results on PF00042, we created an RBM file using the software provided by Tubiana et al. trained on the training set. In order to complete the benchmarking procedure, we constructed two hundred thousand non-homologous decoy sequences to serve as negative controls against the testing set. In generating decoys, we acknowledged that sequences inputted to the RBM needed to be aligned and removed of non-consensus columns, where we define consensus columns as those having fewer than 50% gap characters. Comparatively, the HMM methods view full (with consensus and non-consensus columns), unaligned sequences. Thus decoys needed to be constructed in a way that reflected the alignment for a given family without providing an artifactual way for the RBM to distinguish homologs from decoys in comparison to the HMM methods. In short, we sought that decoys for the RBM matched gap patterns and amino acid frequencies and that decoys for the HMM matched amino acid frequencies and overall sequence length. The algorithm for decoy generation can be described as follows (note that a and b denote different branches of the algorithm not the order of steps):

1. Pick random test homolog from the testing MSA
2. Remove non-consensus columns from selected homolog
3a. Randomly replace non-gap characters using AA frequencies from testing MSA
4a. Score sequence with RBM code
3b. Dealign sequence and randomly replace non-gap characters using AA frequencies from testing MSA
4b. Add characters to the end of the dealigned decoy to match original sequence length
5. Score with uniglobal and local HMM
A visual example of the decoy process is given by Figure (8):

![Diagram of decoy process]

Figure 8: Original Decoy Algorithm. Consensus columns are indicated by capital letter residues or the gap character, -. Following steps in the table above, a random sequence is chosen from the testing MSA and then non-consensus columns, indicated by lowercase letter residues, are removed. Non-gap characters are then replaced with residues according to frequencies from the testing alignment. The decoy is then passed through the RBM code to be scored. For the HMM methods, the sequences is dealigned and then residues are added to the match the original sequence length. Then, the decoy is passed through the HMM code to be scored.

From there, we sought to implement the scoring function provided by Tubiana et al. in a way that produced normalized log odds (in base 2 such that they can be compared to HMMER’s bit scores) scores for the sequences fed in through the given MSA. Then the RBM was fed the three alignments containing the training sequences, the testing sequences and the decoy sequences for scoring. The data, stored in arrays, then allowed for analysis and comparison to determine how well the RBM assessed homology.

Initial results from PF00042 indicated positive results for the RBM’s ability to detect remote homologs from a testing set as compared to randomly generated decoys. Thus, the process for creating RBMs on each family within the Pfam database followed the same general procedure of training on an aligned, FASTA dataset with non-consensus columns removed, and then scoring the training (positive control), decoy (negative control) and testing sets. Results for initial implementation on the globins are shown below:
Figure 9: Histogram of Initial Results for the Globins Domain (PF00042). The y-axis represents the frequency of a given score value, which is represented by the x-axis. The score distributions for the training MSA, testing MSA and decoy MSA are plotted in blue, green, and red, respectively. Results were produced with an RBM trained on globin domain testing MSA produced with profmark. The resulting histogram indicates that the RBM was able to perfectly detect remote homologs from non-homologous decoy sequences, as the tails of the green and red curves do not overlap.

The methods and procedures for implementing HMMs were developed as follows. Because of its functionality with respect to profmark, we decided that Sean Eddy’s HMMER suite provided the most effective means to create a profile HMM with which to perform remote homology search. HMMER can take in aligned and unaligned sequences for training and scoring, respectively, thus introducing variables between the methods which will be discussed in further sections. Unlike the RBMs, HMMs were trained using HMMER on the full (non-consensus and consensus columns) aligned trimmed alignments and with decoy generation as described above. By training on full alignments, the HMM sees and learns from the insert columns (columns with less than 50% identity) whereas the RBM will only train and learn from the match columns on an alignment, thus missing any biological information contained in the insert columns. Generally, profile HMMs built with HMMER establish their own set of consensus columns and emission and transition probabilities for each site. For consistency between methods, we constructed profile HMMs with the same consensus columns as the RBM.
Profile HMMs built in HMMER by default contain entropy weighting measures that are used to alter the emission values of residues across an alignment profile. In most instances, entropy weighting works to flatten emission values across all sites and amino acids; therefore, extremely high and low probabilities are brought closer to zero [41][42]. RBMs, however, do not currently have access to any entropy weighting algorithms. We believed that because entropy weighting was developed to optimize HMMER-3 and its search methods, that this would create a large source of variation where HMMER would outperform the RBM. Therefore, two HMMs were built on each training set, one with entropy weighting on to best simulate the default for remote homology search and one with entropy weighting off to better represent results the RBM might be able to produce. This allows us to better examine whether or not the weights and hidden layer of the RBM add power to distinguishing homologs from decoys. The scoring procedure for the HMMs matches that of the RBM, such that the trained HMM scores the corresponding training (positive control), decoy (negative control) and testing sets.

The default homology search method built into HMMER uses local alignment, which allows part of the model to match only part of the given sequence. As the RBM only sees the alignment without non-consensus columns and then matches that against the corresponding MSA profile, its scoring algorithm more closely resembles a global alignment process, which aligns all of the sequence to all of the model. Therefore, we found that a truer comparison for the RBM would be an HMM that used a glocal scoring algorithm, which aligns part of the sequence to all of the model (a cross between global and local alignment). When implementing a glocal search method, we utilized a uniglocal method, which searches for single hits as opposed to looking for multi hits in biological repeats. To distinguish between the two HMM methods, the two global HMMs will be referred to as eSHMM, corresponding to entropy weighting on, and nSHMM, corresponding to entropy weighting off, and the two uniglocal HMMs will be referred to as eUHMM, corresponding to entropy weighting on, and nUHMM, corresponding to entropy weighting off.

Visual and numerical analytical methods were used to present and compare how well each method separated the true homologs from non-homologous decoys with the log odds score they produced for each sequence. A perfectly performing method would score each sequence in the testing MSA higher than each decoy sequence. For each family and model, a histogram was generated to show the visual separation between the training, testing, and
decoy sets. When methods aren’t perfect, however, it becomes more challenging to assess model performance. Scatterplots were used as another means to visualize how effective one method was compared to another, with each axis representing one model and or method used (thus many pairs of scatterplots existed for each family). Additionally, receiver operating characteristic (ROC) plots were used to quantify the success of each method at distinguishing fake sequences from biological sequences. ROC plots plot the True Positive Rate (TPR), defined as the number of true positives over the number of true positives and false negatives, against False Positive Rate (FPR), defined as the number of false positives over the number of false positives and true negatives for each model and method used. For direct application to homology search, ROC plots were generated by iterating through the unique scores generated for the decoy and testing sets and setting the current score as the threshold for positivity; false positives are then defined as decoy sequences that were above the given threshold and false negatives are testing sequences that were below the given threshold.

For direct numerical analysis, the lowest number of minimum errors across all possible score thresholds (taken from iterating through each score reported for a sequence in the training, testing and decoy sets) was recorded for each method on each family. The minimum error rate (MER) then provides an interpretable metric for a model’s performance in a single family’s benchmark such that we can compare it to other methods and models for a single family. We also quantified the number of families each model achieved perfect separation (achieving a minimum error rate of zero) on to serve as an overall indicator of how often the method achieved the desired goal.

Utilizing the aforementioned methods and procedures, remote homology benchmarks were performed on the available Pfam database [27]; the number of families in the benchmark is limited by profmark’s ability to generate unique testing clusters within the dataset.

3 Data and Results

3.1 Primary Benchmark Results

Following the methods and procedures previously outlined, data was collected on several iterations of benchmark experiments. In all, two full iterations were required due updating certain steps of the procedure. The first iteration produced significant data and helped
identify aspects of the benchmarking procedure that unfairly skewed results to the RBM. The second provided a truer comparison between methods over the Pfam online database.

A change made prior to the first iteration of the benchmarking procedure was the addition of a new profmark algorithm by Samantha Petti, a postdoc in the Eddy Lab. Older versions of profmark required all of the sequences from the original MSA to be put into either the training or testing set, thus occasionally failing to separate groups due to the ID threshold. The updated profmark allows for sequences that do not meet the ID threshold, however, to be discarded or not placed into either the testing or training set [26]; this tweak increased the number of families that could be split and tested on in the benchmark. In the first iteration of the profmark, only around 1,100 families were successfully split; in the final attempt, 4,052 families were successfully clustered; a 362% increase in the overall size of the dataset for the benchmark.

Ultimately 3,941 families were successfully split, trained on and tested by the RBM and the four HMM methods. The remaining 111 families failed at various points of the decoy generation process either due to time, as longer alignments take several cpu-days to generate, or other bugs and inconsistencies in the alignments. A model achieved perfect separation on a family when there were zero testing sequences that score lower than the highest scoring decoy sequence; in other words, the model detects testing sequences as homologous and decoy sequences as noise with a minimum error of zero. Table (1) summarizes how well each model performed overall:
Table 1: Primary Benchmark Results Summary. RBM and non-entropy weighted uniglobal HMM (nUHMM) methods are highlighted as the non-entropy weighted uniglobal HMM was the closest comparison for the RBM in the first benchmark. # perfect corresponds to the number of families each method achieved perfect separation on, defined as having a minimum error rate of 0 for the family. % perfect is the percent of the total number of families in the benchmark (3,941) that the method was able to achieve a minimum error rate of 0 over. The RBM had the strongest results in the benchmark, followed by the entropy weighted uniglobal HMM (eUHMM), the non-entropy weighted uniglobal HMM, the entropy weighted local HMM (eSHMM), and the non-entropy weighted HMM (nSHMM), sequentially.

<table>
<thead>
<tr>
<th>Method</th>
<th># Perfect</th>
<th>% Perfect</th>
</tr>
</thead>
<tbody>
<tr>
<td>RBM</td>
<td>3065</td>
<td>77.7</td>
</tr>
<tr>
<td>eUHMM</td>
<td>2839</td>
<td>72.0</td>
</tr>
<tr>
<td>nUHMM</td>
<td>2035</td>
<td>51.6</td>
</tr>
<tr>
<td>eSHMM</td>
<td>2274</td>
<td>57.7</td>
</tr>
<tr>
<td>nSHMM</td>
<td>580</td>
<td>14.7</td>
</tr>
</tbody>
</table>

The RBM achieved perfect separation more times than the uniglobal or global HMM with and without entropy weighting. For a more direct comparison between models, Table 2 quantifies the number of times each model’s reported minimum error rate was less than and greater than the RBM:
Table 2: Primary Benchmark Minimum Error Comparison. All method abbreviations are identical to those defined in Table 1. The non-entropy weighted uniglobal HMM (nUHMM) method is highlighted as it was the closest comparison for the RBM in the first benchmark. The RBM better column corresponds to the number of families where the minimum error rate for the RBM was lower than the given method. Conversely, the method better column corresponds to the number of families where the minimum error rate for the given method was lower than the RBM. As every value in the RBM better column is greater than every value in the method better column, the data supports the RBM’s strong results in the primary benchmark test.

<table>
<thead>
<tr>
<th>Method</th>
<th>RBM Better</th>
<th>Method Better</th>
</tr>
</thead>
<tbody>
<tr>
<td>eUHMM</td>
<td>848</td>
<td>365</td>
</tr>
<tr>
<td>nUHMM</td>
<td>1798</td>
<td>110</td>
</tr>
<tr>
<td>eSHMM</td>
<td>1512</td>
<td>171</td>
</tr>
<tr>
<td>nSHMM</td>
<td>3330</td>
<td>16</td>
</tr>
</tbody>
</table>

Therefore, the data indicated that the RBM was able to achieve fewer errors on a significant portion of families for all models, even against the uniglobal HMM with entropy weighting. Visual results fell into one of two major categories: one where the RBM succeeded and another where the RBM struggled. The cytochrome b$_5$ family (PF00173) shows strong results for the RBM in comparison to the HMM methods, as the RBM was able to achieve perfect separation between tests and decoys while the other methods struggled. PF00173 contains electron transport hemoproteins found across animals, plants, fungi and bacteria [27]. Its ROC plot result is shown below:
Figure 10: PF00173 Primary ROC Plot. The results for the RBM, entropy weighted uniglobal HMM (eUHMM), non-entropy weighted uniglobal HMM (nuHMM), entropy weighted local HMM (eSHMM), and non-entropy weighted local HMM (nSHMM) are plotted in blue, orange, green, red, and purple, respectively. The y-axis represents the true positive rate for a given score threshold, and the x-axis represents the base 10 log of the false positive rate for a given score threshold. The horizontal blue line at the y = 1.0 position indicates that the RBM was able to achieve a minimum error rate of zero. The other methods struggled to achieve separation.

Even in the set of negative results for the RBM, data indicated generally positive trends; where the RBM struggled to separate the sets the other methods also struggled, if not more so. In the armadillo repeat family (PF00514) both the RBM and various HMM methods struggled to separate decoys from homologs. PF00514 contains sequences that contain a repetitive amino acid sequence found primarily in plants, animals and fungi [27]. Its ROC plot result is shown below:
Figure 11: PF00514 Primary ROC Plot. With legend abbreviations, plotting colors, and axes as defined in Figure 10, the resulting plot indicates that the RBM struggled to achieve separation between the testing and decoy sets. Although, it struggled more than the entropy weighted uniglocal HMM, it performed better than the target comparison model (the non-entropy weighted uniglocal HMM).

The strong results from the RBM on its first benchmark presented opportunities to further interrogate the testing procedure and model.

3.2 Experimental Improvements and Secondary Benchmark Results

Intrigued by apparently strong results, we questioned ways in which the experiment may or may not have been necessarily fair to the HMM methods. Upon reevaluation of how the training alignment format differed between the methods, as the RBM only views match columns in an alignment directly given to it, we realized that an additional HMM method was needed for a true apples to apples comparison to the RBM’s ability to detect remote homology. We decided that an HMM that was fed in an alignment directly would serve as the best means of comparison to the RBM. Then, the HMM scoring method would see the same sequences as the RBM. Therefore there were six HMM methods and models that scored each family: entropy weighted and non-entropy weighted local search HMMs, entropy weighted and non-entropy weighted glocal search HMMs and entropy-weighted and non-entropy weighted glocal HMMs that were directly fed in an alignment. For notation,
the entropy weighted glocal method fed in an alignment will be referred to as eIHMM and the non-entropy weighted glocal method fed in an alignment will be referred to as nIHMM. Given that the RBM still did not have entropy weighting measures, the closest comparison for the RBMs results will be the inputted-alignment HMM without entropy weighting.

We also acknowledged that the previous decoy method did not adequately account for the copied test alignment for full length (with consensus and non-consensus columns) decoys. While the original algorithm outlined in Figure (8) maintained gap patterns effective for the alignment with non-consensus columns removed, we were concerned about how the previous method matched the length of testing sequences. Concerned that this would produce artifacts in searching for homology, we updated the procedure to insert residues where the original test sequence had insert columns. The new decoy algorithm was then as follows:

1. Pick random test homolog from the testing MSA
2. Replace all non-gap characters using AA frequencies from testing MSA
3. Remove all non-consensus columns from generated decoy
4. Score sequence with RBM code
5. Dealign decoy sequence
6. Score with uniglobal and local HMM code
7. Score with inputted-alignment uniglobal HMM code

A visualization of the new decoy process is offered in Figure (12):
Figure 12: New Decoy Algorithm. Consensus and non-consensus are defined as in figure 8. As described in the proceeding list, a test decoy is chosen randomly from the testing MSA and then all non-gap characters are replaced according to residue frequencies from the testing MSA. That decoy is then saved for the inputted-alignment HMMs. For the RBM, the non-match columns are removed from the decoy. For the local and uniglobal HMMs, the decoy is dealigned and then saved accordingly.

In this second iteration, the same 3,941 families from the original benchmark. Perfect results across all seven methods are defined identically to the first benchmark. The seven methods achieved the following degrees of perfect separation in the second benchmark:
Table 3: Secondary Benchmark Results Summary. RBM and inputted-alignment HMM without entropy weighting (nIHMM) methods are highlighted as the inputted-alignment HMM without entropy weighting was the closest comparison for the RBM in the second benchmark. # perfect and % perfect are defined as in Table 1. While the RBM performed about as well as the first benchmark, the inputted-alignment HMM with entropy weighting (eIHMM) and inputted alignment HMM without entropy weighting achieved perfect results more frequently than the RBM. The performance of the other methods in descending order is as follows: the entropy weighted uniglocal HMM (eUHMM), the entropy weighted local HMM (eSHMM), the non-entropy weighted uniglocal HMM (nUHMM) and the non-entropy weighted local HMM (nSHMM).

Indicating that comparatively, the RBM’s performance appears strong when compared to methods in the previous benchmark, but markedly less so when put in the context of the inputted-alignment HMMS; this suggests that previous results might have been an artifact of the benchmark’s design. Comparing the minimum error achieved by of the six HMM methods to the RBM showed the number of families where the minimum error rate of a given method was lower than the RBM and vice versa:
Table 4: Secondary Benchmark Minimum Error Comparison. All method abbreviations are identical to those defined in Table 2. The inputted-alignment HMM without entropy weighting is highlighted as it was the closest comparison for the RBM in the second benchmark. The RBM better column corresponds to the number of families where the minimum error rate for the RBM was lower than the given method. Conversely, the method better column corresponds to the number of families where the minimum error rate for the given method was lower than the RBM. While the RBM performed better than the local and uniglocal HMMs like the first benchmark, the opposite is true for the inputted-alignment HMMs.

<table>
<thead>
<tr>
<th>Method</th>
<th>RBM Better</th>
<th>Method Better</th>
</tr>
</thead>
<tbody>
<tr>
<td>eUHMM</td>
<td>829</td>
<td>367</td>
</tr>
<tr>
<td>nUHMM</td>
<td>1775</td>
<td>112</td>
</tr>
<tr>
<td>eSHMM</td>
<td>1506</td>
<td>169</td>
</tr>
<tr>
<td>nSHMM</td>
<td>3326</td>
<td>19</td>
</tr>
<tr>
<td>eIHMM</td>
<td>393</td>
<td>588</td>
</tr>
<tr>
<td>nIHMM</td>
<td>76</td>
<td>753</td>
</tr>
</tbody>
</table>

Given that the inputted-alignment HMM without entropy weighting outperformed the RBM by around 12% of the families tested, initially positive results from previous iterations might not have been as strong as originally expected. Specifically, examining the ROC plots of PF00173 and PF00514 further highlights how past results might not be a true indication of the RBM’s ability to detect remote homology in a benchmark test and that ensuring the proper comparison model is established.
Figure 13: Secondary Benchmark ROC plots. The results for the RBM, inputted-alignment HMM with entropy weighting (eIHMM), inputted-alignment HMM without entropy weighting (nIHMM), entropy weighted uniglobal HMM (eUHMM), non-entropy weighted uniglobal HMM (nuHMM), entropy weighted local HMM (eSHMM), and non-entropy weighted local HMM (nSHMM) are plotted in blue, brown, pink orange, green, red, and purple, respectively. The y-axis represents the true positive rate for a given score threshold, and the x-axis represents the base 10 log of the false positive rate for a given score threshold. For PF00173, the originally RBM result is matched exactly (and covered) by the inputted-alignment HMM without entropy weighting. For PF00514, the originally poor RBM result is made weaker by both inputted-alignment methods performing better than the RBM.

To best summarize the results, visual representations of data can be broken down into three categories for the RBM’s performance with respect to the HMM methods based on their minimum error rate for each family. Most frequently, the RBM and the inputted-alignment HMMs achieved perfect separation and the remaining methods struggled to repeat the same. Exemplar families include PF00141, PF00591 and PF00703. PF00141 is the peroxidase family, containing heme peroxidases found across prokaryotes and numerous genii of eukaryotes [27]. PF00591 is the glycol transferase family of proteins responsible for ribose substrate transfer found primarily in bacteria and some fungi [27]. PF00703 is the second glycoside hydrolase family, which contain proteins involved in hydrolyzing glycosidic bonds in predominantly fungi and bacteria [27]. Their ROC plots, scatterplots, and histograms of RBM scores are pictured below.
Figure 14: Strong RBM Exemplar Families. Legend abbreviations, plotting colors, and axes defined in Figure 13 are the same for figures 14a, 14d and 14e. Figures 14b, 14f, and 14g show scatterplots of the scores produced by the inputted-alignment HMM with entropy weighting on the y-axis against the inputted-alignment HMM without entropy weighting on the x-axis for each sequence with the training set in blue, the testing set in green, and the decoy set in red. Figures 14c, 14h and 14i show histograms of the distribution of the RBM scores for the training set, the testing set and the decoy set for one of the given families in blue, red, and green, respectively. PF00591 is the glycol transferse family, PF00141 is the peroxidase family, and PF00703 is the second glycoside hydrolase family. Horizontal lines at $y = 1.0$ for the RBM and inputted-alignment HMMs show strong results in the ROC plots. The scatterplots and histogram support the takeaways from the ROC plots.
Figure 14: Strong RBM Exemplar Families.
PF00703 represented one of the most common general trends where all families succeeded at achieving complete separation between testing and decoy sets. Generally, though, results resembled that of PF00141 more, where the other HMM models struggled to achieve separation while the RBM and the inputted-alignment HMMs succeeded. In all, the RBM was able to keep up with IHMM methods in this class of results even when other methods were not.

The second most frequent class of results showed cases where the RBM failed to achieve perfect separation when the inputted-alignment HMMs were able to achieve perfect separation, or generally just performed more strongly. Generally in cases when the RBM failed the number of minimum errors were relatively low. The exemplar families for this set are PF00512, PF01127, and PF00672. PF00512 is the Histidine Kinase A domain, which contains proteins involved in two-component regulatory systems in prokaryotes and bacteria [27]. PF01127 is the succinate dehydrogenase/fumarate reductase transmembrane subunit domain, containing proteins that act as enzymes catalyzing the reaction between succinate and fumarate in bacteria and fungi [27]. PF00672 is the HAMP domain containing classes of signaling proteins contained primarily in bacteria [27]. Their corresponding ROC plots, scatterplots, and histogram of RBM scores are pictured below.
Figure 15: Imperfect RBM Exemplar Families
Figure 15: Imperfect RBM Exemplar Families. Legend abbreviations, plotting colors, and axes are defined as in Figure 13 for figures 15a, 15b and 14g. Figures 15c, 15d, and 15h show scatterplots with plotting colors and axes as defined in Figure 14. Figures 15e, 15f and 15i show histograms also as defined in Figure 14. PF01127 is the furmate reductase subunit domain, PF00672 is the HAMP family and PF00512 is the histidine kinase A family. Imperfect RBM results are seen across all ROC plots where the inputted-alignment HMMs outperform or match RBM performance. Scatterplots with more red-green overlap in the x-axis and less so in the y-axis support these results, as do histograms with green and red distributions overlapping slightly.

PF00512 indicates a common result where both the RBM and inputted-alignment HMM
without entropy weighting struggle to achieve zero errors at the optimal threshold, but have few errors such that their ROC plot lines plateau quickly. The RBM slightly outperforms the inputted-alignment HMM without entropy weighting here. PF00672 displays a similar trend, however the ROC indicates that the inputted-alignment HMM without entropy weighting, while also struggling, was able to achieve stronger separation than the RBM. Akin to PF00672, PF01127 shows a result where the RBM lagged behind the inputted-alignment HMM without entropy weighting, but where all methods were later able to achieve decent separation.

The final class of results showed cases where all seven methods struggled to detect remote homologs. The exemplar families from this category are PF00036, PF00347 and PF00549. PF00036 is the EF hand motif which is a helix-loop-helix structural domain found in many calcium binding proteins found predominantly in eukaryotes from insects to primates [27]. PF00347 is the ribosomal L6 domain of ribosomal proteins found in bacteria, fungi and other eukaryotes [27]. PF00549 is the Ligase CoA domain, representing various CoA ligases found across bacteria and various eukaryotes [27]. Their corresponding ROC plots, scatterplots, and RBM histogram plots are below:
Figure 16: Weak RBM Exemplar Families
Figure 16: Weak RBM Exemplar Families. Legend abbreviations, plotting colors, and axes are defined as in Figure 13 for figures 16a, 16b and 16g. Figures 16c, 16d, and 16h show scatterplots with plotting colors and axes as defined in Figure 14. Figures 16e, 16f and 16i show histograms also as defined in Figure 14. PF00347 is the ribosomal_L6 family, PF00549 is the ligase_CoA domain and PF00036 is the EF hand motif domain. All ROC plot results show that the RBM performed significantly worse than the inputted-alignment HMM without entropy weighting (and even the unglocal HMM with entropy weighting in the case of PF00036). The scatterplots reflect these trends as the data achieves some separability between red and green groups from the y-axis and less, if any, in the x direction. Histograms show the degree of overlap between the decoy and testing score distributions.
PF00036 represents a result pattern where ultimately all seven methods across both models struggled to achieve separation between datasets. In this family, the inputted-alignment HMM with entropy weighting outperforms the inputted-alignment HMM without entropy weighting against previously established trends. PF00347 displays a more common trend for the benchmark where, while all methods struggle, the inputted-alignment HMM without entropy weighting ultimately performs the best and the RBM trails behind it and the other HMM methods follow suit. PF00549 was selected as an exemplar due to its atypical result, namely in the RBM lagging behind the uniglobal HMM methods. Discussion will investigate these abnormalities in greater detail.

While not a direct result of the benchmark, the testing and training alignment size, as well as the alignment lengths of the nine exemplar families are featured in Table (5), as alignment size and width are worth discussion in the next section:

<table>
<thead>
<tr>
<th>Family/Domain Name</th>
<th>Train Al. Size</th>
<th>Test Al. Size</th>
<th>Alignment Length</th>
</tr>
</thead>
<tbody>
<tr>
<td>Peroxidase (PF00141)</td>
<td>5499</td>
<td>51</td>
<td>229</td>
</tr>
<tr>
<td>Glycol transferase (PF00591)</td>
<td>6397</td>
<td>70</td>
<td>253</td>
</tr>
<tr>
<td>Glycoside hydrolase II (PF00703)</td>
<td>1401</td>
<td>236</td>
<td>110</td>
</tr>
<tr>
<td>Histidine kinase A (PF00512)</td>
<td>17111</td>
<td>269</td>
<td>67</td>
</tr>
<tr>
<td>Fumarate reductase subunit (PF01127)</td>
<td>2664</td>
<td>148</td>
<td>121</td>
</tr>
<tr>
<td>HAMP (PF00672)</td>
<td>5053</td>
<td>815</td>
<td>53</td>
</tr>
<tr>
<td>EF hand (PF00036)</td>
<td>417</td>
<td>20</td>
<td>29</td>
</tr>
<tr>
<td>L₆ (PF00347)</td>
<td>7381</td>
<td>373</td>
<td>76</td>
</tr>
<tr>
<td>Ligase,CoA (PF00549)</td>
<td>6832</td>
<td>102</td>
<td>153</td>
</tr>
</tbody>
</table>

Table 5: Exemplar Families Alignment Data. The training alignment size column reports the number of sequences in the training alignment that profmark produced for the given family. Similarly, the testing alignment size column reports the number of sequences in the testing alignment that profmark produced for the given family. The alignment length column shows the number of consensus columns in the alignment for each family. Generally, no strong trends between size and the ability of the RBM to detect remote homologs from decoy sequences were noticed.

Overall, the second set of data indicates potentially positive results for the RBM, but
shows that the model struggles in comparison to methods in HMMs. As I discuss in the next section, these results could be studied further given experimental parameters and other variables.

4 Discussion

4.1 Lessons from 1st and 2nd Benchmarks

The results from the previous section present various lessons and topics for discussion. The first test, which indicated strong results for the RBM, suggested that the latent structure of the RBM aided in its ability to detect remote homology. We expected that alignment methods (global vs. glocal) and entropy weighting would have a strong effect on the data for comparison. We did not, however, think about how the difference in how the models see those alignments would affect those benchmark results. Repeating the benchmark experiment with an HMM method that is fed in an alignment, like the RBM, illustrated that initially strong results were instead a factor of experimental design and not indicative of the RBM’s ability to learn information applicable to homology search. Seeing that the RBM achieved perfect separation on 78% of presented families appears strong in comparison to the uniglobal HMM with entropy weighting that achieved separation on 72% families, but noticeably weaker when examined with the HMM that was fed in an alignment without entropy weighting, which achieved separation on 90% of families. These results align with choices made about how to manipulate the MSAs given to the RBM and HMM methods. Removing fragmented sequences and screening for sequences that were extremely gappy severely biased our results towards the RBM and inputted-alignment HMMs. When comparing any test model without an alignment algorithm to another model like HMMER with alignment algorithms in a remote homology benchmark, ensuring that the models are a true apples to apples comparison with respect to what columns and sequences they see should be of paramount importance.

Returning to the central questions of this thesis, we sought to learn whether or not the added latent layer of the RBM helps with detecting remote homology. While a 78% success rate could be interpreted as a strong result for the RBMs first major application to homology search, further experimentation is required to directly answer the question of whether the structure offers the model additional ability to detect homology. In applying the previous
lesson, an apples to apples comparison of the RBM without any weights and hidden units and the RBM with hidden units as established in these experiments provide a direct apples to apples comparison to answer the question.

4.2 Entropy Weighting Does Not Increase HMM Sensitivity in Benchmarks with Aligned Sequences

The inputted alignment HMM without entropy weighting achieving perfect separation more frequently than the inputted alignment HMM with entropy weighting was initially counterintuitive to our understanding of entropy weighting and sequence scoring. Entropy weighting normally is used to prevent overfitting the emission probabilities, which proves helpful when searching for remote homologs. A reexamination of the function of entropy weighting indicates that the result actually confirms the intended design of entropy weighting. Under entropy weighting, the process works to flatten emission probabilities across the sequence such that there are fewer extremely probable or improbable emissions. An example of this flattening can be seen by comparing the emission probabilities of the 36th column of the alignment of the globin family (PF00042) \( e_{x_{36}}(a) \) over all 20 possible amino acids for \( a \) of an HMM with entropy weighting to an HMM without entropy weighting. Additionally, comparing the total informational entropy \( H(x) \) informs us about the overall uncertainty of a distribution. Given a single HMM match column \( m_i \), we express the entropy of the emission values \( H(e_{m_i}(x_j)) \) over \( q \) possible amino acids as:

\[
H(e_{m_i}(x_j)) = - \sum_{j=1}^{q} e_{m_i}(x_j) \log(e_{m_i}(x_j))
\]  

(13)

A distribution with high levels of uncertainty, like a uniform distribution, has \( H(x) = \log_2(20) \) whereas a distribution containing a single character would have \( H(x) = 0 \). Directly comparing the emission probabilities from an HMM with entropy weighting against emission probabilities from an HMM without entropy weighting produces the following plot:
A visual comparison indicates the flattening of emission probabilities for the given column of the globin alignment; the probability of emitting an asparagine at site 36 without entropy weighting is around .14 but only around .1 with entropy weighting. Comparing $H(e_{m_{36}}(x_j))$ from each model reveals that the HMM with entropy contains a higher total entropy across the distribution of emission probabilities; this indicates that the distribution of $H(e_{m_{36}}(x_j))$ contains a higher degree of informational uncertainty than without entropy weighting.

Therefore, in a given test or training sequence it is likely that there are a couple very high likelihood residues, some less likely and some unlikely residues which contribute to the overall score. When comparing decoy sequences that copy the underlying alignment from the given testing sequences, the transition probabilities of the sequence remain identical as they only differ in the given emission probabilities. Because decoy residues were selected at random, they have a significantly higher proportion of sites that have extremely low scoring
residues. With entropy weighting the low probabilities in the testing sequence will be raised and the highly scoring positions will also be lowered. In comparison, the decoys with large groups of poorly scoring residues will be pushed up with the entropy weighting. Therefore testing sequences whose scores are only marginally raised, if not lowered, struggle to be detected from decoys whose scores are improved at a greater degree than test sequences. This can be viewed with a scatterplot of the scores produced by an inputted-alignment HMM with entropy weighting against the scores produced by an inputted-alignment HMM without entropy weighting. That scatterplot for PF00591 is below, where we see the red cloud of decoy sequences is raised relatively higher than the green cloud of testing sequences. The line for x=y has been added to help show how entropy weighting raises or lowers scores relative to the non-entropy weighted HMM.

Figure 18: The Effect of Entropy Weighting on Decoy Scores. The plot compares the score assigned to a sequence by the inputted-alignment HMM with entropy weighting on the y-axis against the score assigned to a sequence by the inputted-alignment HMM with entropy weighting on the x-axis. The data is taken from the glycol transferase family (PF00591). Its training set, testing set, and decoy set are plotted in blue, green, and red, respectively. Entropy weighting raised the decoy scores more than the training or testing scores, indicated by the position of the red cluster with respect to the line y=x.

Default HMMs do not encounter these issues as they are able to develop any alignment
for a given decoy sequence. It can then be deduced that for methods where only alignment consensus columns are viewed, entropy weighting may not be as helpful for effectively testing for homology.

4.3 Examination of Selected Data

PF00512 was one of very few families where the RBM was able to reach a lower total of errors at the optimal score threshold. Albeit minimally, as the RBM had seven errors at its optimal threshold and the inputted-alignment HMM without entropy weighting had nine. Although we do not currently know what the RBM was able to detect in PF000512 that the HMM was not, further investigation into families like PF00512 through other modes of analysis could aid in determining what the RBM is seeing. PF00512 is also unique in the sheer size of its training and testing set, with 17,111 and 269 sequences, respectively. As Table (5) presents, family size is something that we recognized could have an effect on the way in which the RBM profile is constructed and something that could be adequately tested in further experiments to discover more about.

PF00549 also contained an interesting visual result indicative of a larger trend. While all seven methods struggled to concretely determine homology, the non-entropy weighted global HMM line on the ROC plot features a sharp point of inflection around -2 on the x-axis (representing the log_{10}(FPR)). This inflection in the non-entropy weighted global HMM line appeared with decent frequency among families where the primary methods of focus struggled. Akin to the atypical pattern in PF00512, further analysis into what families with this inflection share biologically and structurally could be useful in designing future tools for remote homology search.

5 Conclusion and Next Steps

We sought to answer the question of whether a restricted Boltzmann machine, which Tubiana et al. showed to be able to learn constitutive motifs from MSAs, could apply its profile of an MSA to detect remote homologs in a benchmark homology search. In order to evaluate that, we prepared benchmark tests over all available families in the Pfam database [27]. We see that the benchmark experiments taught us the following lessons. Primarily, we were able to verify that while not perfectly, the RBM was able to apply the information
learned from training about an MSA to the context of detecting homology in a benchmark remote homology search. For 78% of the families presented, the RBM was able to perfectly detect remote homologs from decoy sequences and only exceptionally failed on families where the other key comparison methods also struggled. Further experimentation is required to concretely determine whether or not the RBM would be able to become a viable means for remote homology searching. As discussed earlier, the benchmarking process highlights the importance of ensuring that the comparison model in a benchmark setting be as similar as possible to the target model, else results can become skewed as a product of the test design.

Given the inconclusive answer on whether or not the RBM can be more effective at learning protein motifs and applying those to remote homology search, there are a number of important next steps anyone looking into applying RBMs to remote homology should evaluate. As previously alluded to, experimentation into what the RBM sees on an MSA could prove helpful to future applications of the model. Using methods that can reduce a large MSA to a sequence space-type representation, such as Principal Component Analysis (PCA) or Multi-Dimensional Scaling (MDS) could help indicate exactly what structures or subfamilies within a larger MSA are being favored by the RBM. Ideally, we could use one of these dimensional reduction analyses to determine which groups of proteins within a family are particularly high or low scoring with the RBM and assess what about their particular biology and sequence the RBM finds particularly favorable or unfavorable.

Another critical next step for an RBM implementation is to view how the form of the hidden layer potentials affect the training and ability of the RBM. Tubiana et al. adjusted hidden local potential forms in their experimentation to produce favorable results for their questions, which do not necessarily align with remote homology search. Therefore, it is possible that adjusting the format of the hidden potentials from double rectified linear units, which they showed to be critical to the RBM’s ability to learn higher order information on an MSA, to another distribution could better capture information necessary to detect remote homologs.

Additionally, the number of weights and hidden units each RBM has is a hyperparameter Tubiana et al. experimentally determined for the RBM to perform effectively. Therefore, adjusting the number of weights generated for a single MSA could affect how well the RBM can search; perhaps there exists a linear relationship between the scope of information an RBM can learn from an MSA and the number of hidden units and weights exist in the model.
Specifically, setting the number of weights and hidden units to zero could be an interesting experiment, as that would provide a precise apples to apples comparison between the RBM to the inputted-alignment HMM without entropy weighting as two site-independent models against each other. Such a control test would allow us to examine if there are any quirks in Tubiana et al.’s RBM implementation.

Looking into methods for the RBM to, given a model and set of sequences, conduct E-value estimation would also help enable the RBM to become a more powerful tool for homology too. An E-value, effectively, is the number of false positives based on the expected number of hits at a given score threshold from a database of i.i.d. random sequences [5]. In effect, equipping the RBM to estimate E-values allows future experiments to adequately assess how much random background noise exists for a given model. With the RBM as currently implemented, an algorithm to estimate E-values would start by scoring a very large number of randomly generated sequences in order to estimate the distribution of non-homolog scores.

Perhaps the most valuable investigation for the RBM is determining how to develop a probabilistic alignment process such that the RBM can be trained on and score unaligned sequences. The actual functionality of the RBM is limited by the lack of alignment algorithm— in effect, the benchmark was only able to be run because HMMER was able to align sequences for the RBM to then be fed to score. Therefore, developing an alignment algorithm for the RBM would enable it to actually function in detecting remote homologs not just in a fabricated benchmark search.

The restricted Boltzmann machine showed promising results for its ability to apply learned family motifs from an MSA and apply them to remote homology search. Future investigations should continue to see whether the model can be improved to be a stronger tool for homology search.
6 References


http://pfam.xfam.org/family/PF00703#tabview=tab0 (accessed Mar. 01, 2021).

[31]“Pfam: Family: HisKA (PF00512),” pfam.xfam.org.

http://pfam.xfam.org/family/PF01127#tabview=tab0 (accessed Mar. 01, 2021).

https://pfam.xfam.org/family/PF00672#tabview=tab0 (accessed Mar. 01, 2021).

[34]“Pfam: Family: EF-hand_1 (PF00036),” pfam.xfam.org.
http://pfam.xfam.org/family/pf00036#tabview=tab0 (accessed Mar. 01, 2021).

http://pfam.xfam.org/family/PF00347#tabview=tab0 (accessed Mar. 01, 2021).

https://pfam.xfam.org/family/PF00549#tabview=tab0 (accessed Mar. 01, 2021).


