[1]Section for Bioinformatics, Department of Health Technology, Technical University of Denmark, Kgs Lyngby, Denmark

[2]Department of Applied Mathematics and Computer Science, Technical University of Denmark, Kgs Lyngby, Denmark

Correspondence
Henrik Nielsen, Section for Bioinformatics, Department of Health Technology, Technical University of Denmark, 2800 Kgs Lyngby, Denmark
Email: henni@dtu.dk

Funding information

GPI-anchors constitute a very important post-translational modification, linking many proteins to the outer face of the plasma membrane in eukaryotic cells. Since experimental validation of GPI-anchors is slow and costly, computational approaches for predicting them from amino acid sequences are needed. However, the most recent GPI predictor is more than a decade old, and considerable progress has been made in machine learning since then. We present a new dataset and a novel method, NetGPI, for GPI prediction. The predictor is based on recurrent neural networks, incorporating an attention mechanism that simultaneously detects GPI-anchors and points out the location of their ω-sites. The performance of NetGPI is superior to existing methods with regards to discrimination between GPI-anchors and other proteins and approximate (± 1 position) placement of the ω-site. NetGPI is available at: https://services.healthtech.dtu.dk/service.php?NetGPI-1.0.

KEYWORDS
glycosylphosphatidylinositol, lipid anchored proteins, post-translational modification, protein sorting, prediction, neural networks

ORIGINAL ARTICLE

Prediction of GPI-Anchored proteins with pointer neural networks

Magnús Halldór Gíslason[1,2] | Henrik Nielsen[1] | José Juan Almagro Armenteros[1]* | Alexander Rosenberg Johansen[2]*

Abbreviations: GPI, glycosylphosphatidylinositol; HMM, hidden Markov model; LIME, Local Interpretable Model-agnostic Explanations; LSTM, long short-term memory; MCC, Matthews correlation coefficient; RNN, recurrent neural network; SVM, support vector machine.

* Equally contributing authors.

1 | INTRODUCTION

Some of the proteins that follow the secretory pathway are attached to the membrane of eukaryotic cells by specific mechanisms. One of these mechanisms is a post-translational modification where a glycosylphosphatidylinositol (GPI) anchor is attached to the protein. The identification of proteins that undergo this modification is of high interest due to the diversity of functions that they perform. GPI-anchored proteins are essential in the development of fungi and animal cells [1, 2]. They are also involved in certain diseases such as paroxysmal nocturnal haemoglobinuria, an acquired haematopoietic stem-cell disorder [3], and in the defense mechanisms of various protozoan parasites such as *Leishmania* and *Trypanosoma* [4] . Consequently, the development of computational tools that are able to detect proteins with this modification is of high impact on the research of eukaryote cell biology [5].

GPI-anchored proteins have two signals in their primary sequence: an N-terminal sequence for endoplasmic reticulum targeting (signal peptide) and a C-terminal signal sequence directing the attachment of the GPI-anchor. This attachment is carried out by a GPI transamidase which recognizes the C-terminal signal sequence and cleaves the peptide bond at the GPI-anchor attachment site, known as the ω-site. This cleavage creates a link between the GPI and the C-terminus of the cleaved protein, allowing the protein to remain tethered to the membrane. C-terminal signal sequences are generally composed by five regions, which are determined by the amino acids before the ω site (ω-minus) and after (ω-plus). The five regions are: a stretch of polar amino acids that form a flexible linker region ($\omega - 10$ to $\omega - 1$); the ω site amino acid; the $\omega + 2$ amino acid, a restrictive position with mostly G, A or S; a spacer region of moderately charged amino acids ($\omega + 3$ to $\omega + 9$ or more), and a stretch of hydrophobic amino acids starting approximately at $\omega + 10$ [6].

In order to detect proteins that carry this signal, experimental assays are required. Such experiments are generally low throughput and costly, which has resulted in a low amount of experimentally annotated GPI-anchored proteins. To overcome this limitation, fast computational methods that can approximate the experimentally validated process are needed. For this purpose, current machine learning methods exist for predicting GPI-anchors [7, 8, 9]. Nonetheless, these methods were developed more than a decade ago and do not utilize recent progress in machine learning methods nor access to new data sources. Deep learning methods, such as the Recurrent Neural Network (RNN) [10], have recently proven effective at protein prediction tasks [11]. However, Deep Learning requires large amounts of annotated examples to generalize well [12].

In this paper we present a new tool for detecting GPI-anchored proteins and determining the position of the ω-site using recurrent neural networks. To overcome the low amounts of experimentally validated data we build a new training set utilizing manually annotated predicted GPI anchored proteins, mostly reserving the experimentally verified data for a held-out test set. Regardless, our method achieves state-of-the-art performance on the GPI-anchor prediction task. Moreover, we show that the model learns biologically meaningful characteristics.

1.1 | Related works

Initial work on predicting the presence of GPI-anchors and the ω-site was published by Eisenhaber et al. [13]. This work, known as the Big-Π Predictor, details a method that evaluates amino acid type preferences at positions near a supposed ω-site as well as the concordance with general physical properties encoded in multi-residue correlation within the motif sequence [13]. Big-Π provides kingdom-specific predictions as it was trained on metazoan, protozoan, fungi [14] and plant [15] proteins separately.

Fankhauser and Mäser [8] presented a neural network based prediction tool called KohGPI/GPI-SOM. GPI-SOM utilizes a Kohonen Self Organizing Map structure which takes as input the average position of a given amino acid

relative to its proximity to the C-terminal, the hydrophobicity of the amino acid at 22 C-terminal positions and 2 units representing the quality of the presumed ω-site and its position. Both GPI-SOM and Big-Π utilize an external signal peptide predictor known as SignalP [16] to preselect proteins. A genome-wide study by the authors indicated that the percentage of GPI-anchored proteins in a given proteome was in the same order of magnitude as their reported error rate, not accounting for the error rate of the version of SignalP used.

In 2008 Pierleoni, Martelli & Casadio published Pred-GPI, a GPI-anchor predictor utilizing a Hidden Markov Model (HMM) for the prediction of the position of the ω-site and a Support Vector Machine (SVM) for the presence of a GPI-anchor [9]. The HMM has 46 states, with varying probabilities for amino acids and the potential ω-site assigned the 26th state. The SVM takes as input the negative log-likelihood computed by the HMM as well as 82 features intended to describe the overall composition of the sequence, the features of the N-terminal regions comprising the signal peptide, and the features of the C-terminal regions containing the cleaved GPI-anchor signal. Pred-GPI supplies two different variants: One model where the potential ω-site is restricted to be one of Cysteine, Aspartic acid, Glycine, Asparagine, and Serine, this approach they refer to as the conservative model; and another model having no such restriction. Unlike the two other methods, Pred-GPI does not rely on an external signal peptide predictor, such as SignalP.

2 | MATERIALS AND METHODS

2.1 | Dataset

All data used in this project are extracted from the UniProt database [17]. The dataset construction follows two main steps: data gathering and homology partitioning. First, we select all the eukaryotic proteins with experimental evidence (ECO:0000269) of a signal peptide. This dataset is divided into proteins with and without a GPI-anchor signal in the subcellular location field, defining the positive and negative set respectively. The positive set is composed of proteins with three different levels of annotation: Experimental evidence of a GPI-anchor signal, experimental evidence of the ω-site position in a "LIPID" feature table entry, and non-experimental evidence of a GPI-anchor signal. All the proteins are truncated to the 100 last (C-terminal) positions, since this is where we expect to find the GPI-anchor signal. The truncated sequences are then homology clustered using CD-HIT [18] with a similarity threshold of 20% resulting in 1866 clusters.

For homology partitioning we assign the clusters of proteins into either one test partition or one of five training/-validation partitions. Clusters composed exclusively of proteins with experimental evidence are assigned to the test partition, whereas clusters with both experimental evidence and predicted GPI-anchors are assigned to one of the training/validation partitions. The partitions are constructed such that the distribution of positive and negative classes and the kingdom composition (animal, fungi and plant) is the same across all partitions.

These steps result in a training/validation set containing 2823 samples, of which 658 are GPI-anchored. A total of 25 proteins with experimental evidence of the GPI-anchor signal were placed in the training/validation partitions. The test set contains 594 samples, of which 111 are experimentally verified GPI-anchored, but without a verified ω-site, and 50 have an experimentally verified ω-site for a total of 161 GPI-anchored samples.

2.2 | Objective

The objective of GPI prediction is to decide whether a GPI signal is present and, if present, to determine the position of the ω-site in a protein sequence. We combine these two tasks by reducing them to the single task of maximizing the probability of a position in a sequence. To achieve this, we add a placeholder to the end of the protein sequence which

TABLE 1 Dataset composition for the training/validation set and held out test set.

Data-set	Samples	Not GPI-Anchored	%	GPI-Anchored	%
Trainining/Validation	2823	2165	76,7%	658	23,3%
Held-out test	594	433	72,9%	161	27,1%

serves to indicate that it is non GPI-anchored. Thus we formally define the objective as maximizing the probability of a position in $\hat{D}$, which is known as pointing [19].

$$\max_{\theta} P_\theta(C_i|\hat{D}) \tag{1}$$

$$\hat{D} = [D, \mathrm{z}] \tag{2}$$

Where $D \in \Sigma^T$ is an amino acid sequence and Σ is a dictionary of the twenty common amino acids as well as the token X, which represents any encountered amino acid not in Σ. We only consider the last 100 amino acids in the protein sequence, such that the length $T \leq 100$. C_i corresponds to a position in $\hat{D}$.

If the sequence does not contain an ω-site we maximize the probability of the protein being non GPI-anchored. Inspired by work in natural language processing [20, 21], we represent the lack of an ω-site by maximizing the placeholder position known as the sentinel, z, at the end of the amino acid sequence. This results in $\hat{D} \in \hat{\Sigma}^{T+1}$ where $\hat{\Sigma} = \Sigma \cup \{z\}$.

To parameterize the conditional probability distribution P_θ we use a neural network architecture known as the Long-Short Term Memory (LSTM) Cell [22] and distributed representations of the amino acids [23] as shown in equation 3.

$$\begin{aligned} z_i &= \texttt{embedding}(\hat{D}_i) \\ h &= \texttt{LSTM}(z) \\ g_i &= \texttt{tanh}(h_i W) \\ P(C_i|\hat{D})_\theta = \texttt{softmax}(gV)_i &= \frac{\exp(g_i V)}{\sum_{j=0}^{T} \exp(g_j V)} \end{aligned} \tag{3}$$

Where $\texttt{embedding} : \hat{\Sigma} \rightarrow \mathbb{R}^d$ turns each amino acid into a distributed representation of real numbers using a linear trainable weight of size d and $i, j \in \mathbb{N} \leq T$ are indexes of the protein sequence including the sentinel position. The `LSTM` is a non-linear transformation of a sequence of real values. It uses trainable recurrent units to distribute sequential information across the protein sequence, $\texttt{LSTM} : \mathbb{R}^{T \times d} \rightarrow \mathbb{R}^{T \times d'}$, where d' is the output size of the `LSTM`. As we use a bidirectional `LSTM` [24] we end up with two hidden representations of size d'. To get the probability over the sequence we project the output of every position to a logit, $g_i V \in \mathbb{R}$, followed by a $\texttt{softmax} : \mathbb{R}^T \rightarrow [0, 1]^T$ that normalizes the logits into a probability distribution over the sequence. To create the logits we use a two layer feed forward neural network on top of the `LSTM` hidden states, $h \in \mathbb{R}^{T \times 2d'}$, with a `tanh` activation function, $W \in \mathbb{R}^{2d' \times d''}$, and $V \in \mathbb{R}^{d''}$. This usage of `softmax` over a sequence length is a modification of attention where the interaction size d'' of gV is the attention hidden representation size, which is known as a pointer network [19].

The `embedding`, `LSTM`, W, and V are all trainable with stochastic gradient descent using back-propagation through time [25]. We have visualized our model in Figure 1.

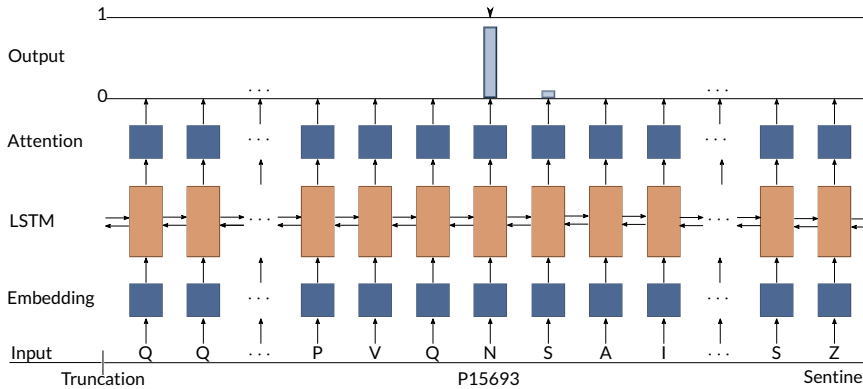


FIGURE 1 Diagram of the model, illustrating how the model points to a position in a sequence, in this case, the entry with UniProt accession number P15693. The sequence is truncated to the last 100 amino acids and the sentinel, z, is appended. The predicted ω-site is an Asparagine (N). If the position with highest likelihood had been the sentinel position, then the protein would have been predicted as non GPI-anchored.

2.2.1 | Quantitative evaluation criteria

To evaluate the discrimination between GPI-anchored and non GPI-anchored proteins we use the Matthews Correlation Coefficient (MCC) and for ω-site prediction evaluation we use the F1 score [26]. Due to the dual nature of the problem and as well as the lack of experimental $\omega-$site evidence in the training set, a simple heuristic is devised. The heuristic is a composition of the two evaluation methods. The F1 score is calculated with a tolerance of two positions from the annotated ω-site. We allow for this flexibility when calculating the F1 score as the training set contains only non-experimentally verified ω-site samples, which are not as reliable as the experimentally verified. The MCC is weighed twice as important as the F1 score. We weigh the MCC more as we want to emphasize the GPI-anchoring discrimination over the ω-site prediction performance. The model with the combination of hyperparameters that gives the best heuristics, on the validation partition, is chosen for each fold. This heuristic also controls when the model's parameters are stored as an early stopping approach. The self evaluation during training is the Cross Entropy Loss.

2.3 | Model Details

We train the neural network with a batch size of 64 and up to 30 epochs. We set the embedding size $d = 22$. To find the optimal values for: the size of the bidirectional `LSTM` cell hidden representation d', the attention hidden representation d'', the number of `LSTM` layers, the dropout between `LSTM` layers, the optimizer's weight decay, and learning rate we use a validation set. Dropout between `LSTM` layers forces each hidden unit in subsequent layers to work with a randomly chosen set of hidden units from the previous layer [27].

To better utilize data we do a five-fold split of the training set and optimize the neural network hyperparameters individually for each split. The best performing model from each split is used in an ensemble for the test set. Each model of the ensemble is transformed with a logarithm before being averaged. This is done to emphasize confident model predictions.

We evaluate 192 different hyperparameter settings on the validation set for each fold. The hyperparameters we

TABLE 2 The combination of hyperparameters with the best validation performance for each partition.

Ensemble model	LSTM layers	LSTM hidden units (d')	Weight decay
0	3	110	0.0010
1	2	44	0.0006
2	3	110	0.0006
3	2	44	0.0006
4	2	88	0.0006

Dimensions shared by all models: Attention hidden units(d'') = 110, LSTM dropout = 0.4, embedding size(d) = 22, learning rate = 0.006

found optimal are shown in table 2.

The neural network is trained with stochastic gradient descent using the Adam optimizer [28]. Our models are implemented with the PyTorch deep learning framework [29].

2.3.1 | Qualitative evaluation methods

To better understand the decisions the model makes we performed a feature importance analysis using the Local Interpretable Model-agnostic Explanations (LIME) package [30]. This analysis is performed on the held-out test set. In the LIME analysis, amino acids contributing to a GPI-anchored prediction will have a positive importance, while amino acids contributing to the non GPI-anchored prediction will have a negative importance. The larger the weight the larger the contribution to the prediction.

Furthermore, we investigate the sequence composition around the ω-site to uncover possible model biases.

3 | RESULTS AND DISCUSSION

3.1 | Quantitative results

To benchmark the performance of the current tools the held-out test set was submitted to the three tools currently available; Big-Π, GPI-SOM, and PredGPI. In the case of Big-Π we separated the held-out test set according to kingdom and submitted to the corresponding versions of the tool. Big-Π annotates its predictions according to likelihood. Predictions with high likelihood are labeled as P, twilight zone predictions are labeled as S, and non-potentially GPI-anchored proteins are labeled as N. We regarded any protein predicted as potentially GPI-anchored (P or S) as a GPI-anchored prediction.

PredGPI ranks and classifies predictions according to specificity. Predictions are regarded as highly probable, probable, weakly probable, and not GPI-anchored. We measure the performance for two settings of PredGPI; designating weakly probable either as GPI-anchored or non GPI-anchored. Assuming weakly probable as negative predictions gives the best performance according to MCC.

For predicting the presence of GPI-anchors, NetGPI achieves the highest MCC of **0.962**. If we regard PredGPI's weakly probable as negative, the second highest MCC is PredGPI, otherwise the second highest is Big-Π. NetGPI also attains the highest true positive rate (TPR), **0.975**, the second highest being GPI-SOM. NetGPI achieves the second

TABLE 3 Comparison of the GPI-anchor presence prediction performance of NetGPI and benchmarked methods.

All (594)	TP	FP	FN	TN	TPR	Prec.	FPR	MCC
NetGPI	**157**	5	**4**	428	**0.975**	0.969	0.012	**0.962**
PredGPI*	148	13	13	420	0.919	0.919	0.030	0.889
PredGPI**	151	25	10	408	0.938	0.858	0.058	0.857
GPI-SOM	153	45	8	388	0.950	0.773	0.104	0.798
Big-Π	132	**2**	29	**431**	0.820	**0.985**	**0.005**	0.867
Filtered*** (295)	**TP**	**FP**	**FN**	**TN**	**TPR**	**Prec.**	**FPR**	**MCC**
NetGPI	**98**	**1**	**4**	**192**	**0.961**	**0.990**	**0.005**	**0.970**
Big-Π	73	**1**	29	**192**	0.716	0.986	**0.005**	0.754

Abbreviation: TP = True positive, FP = False Positive, FN = False Negative, TN = True Negative, TPR = True Positive Rate, Prec. = Precision, FPR = False Positive Rate, MCC = Matthews Correlation Coefficient.
* No difference in the conservative or non-conservative options for PredGPI was observed, this is the results when weakly probable predictions are regarded as negative.
** This is the result for PredGPI when weakly probable predictions are regarded as positive.
*** Here we have limited the test set to samples not in Big-Π's reported training set and made available on UniProt after 2004-03-19

highest precision, 0.969, and false positive rate (FPR), 0.012, the highest being Big-Π with a precision of 0.985 and FPR of 0.005. For a detailed comparison see table 3. Noticeably, Big-Π uses kingdom information in its predictor. We tried a similar approach, but found no improvement in our performance using kingdom features during hyperparameter optimization, which is why we did not include it in our predictor.

We find that Big-Π has at least 59 overlapping samples with our positive test set and an unknown overlap with our negative test set. This might cause the performance of Big-Π to be overestimated. We filter the dataset of test samples to GPI-positive samples not found in Big-Π's reported training set and non GPI-anchored samples made available on UniProt after 2004-03-19 (the publishing date of Eisenhaber et al. [14]). In the filtered comparison NetGPI has comparable performance (+0.008 MCC), while Big-Π's performance decreases (-0.113 MCC).

For the prediction of the position of the ω-site we only consider the 50 proteins with an experimentally verified ω-site. NetGPI predicts 36 out of 50 correctly, achieving an F1-score of 0.692, while Big-Π correctly predicts 39 out of 50, resulting in an F1 score of 0.812. However, when we allow for one-off errors NetGPI positions 45 out of 50 and attains an F1 score of 0.865, while Big-Π positions 41 out of 50 with an F1 score of 0.854. For a detailed overview see table 4.

As pointed out there is an overlap with the Big-Π training set and our test set. This is overly prevalent for GPI anchors with experimentally verified ω-sites. Out of the 50 ω-sites, 33 are used for training the Big-Π model. Both NetGPI and Big-Π correctly position 9 of the 17 which are not in the Big-Π training set. Allowing for one-off errors, NetGPI correctly locates 14 out of 17 whereas Big-Π locates 10 out of 17.

TABLE 4 Comparison of the ω-site position prediction performance of NetGPI and the benchmarked methods.

Known*** (50)	±0	F1	Sens.	Prec.	±1	F1	Sens.	Prec.	±2	F1	Sens.	Prec.
NetGPI	36	0.692	0.720	0.667	**45**	**0.865**	**0.900**	0.833	**46**	**0.885**	**0.920**	0.852
PredGPI*	29	0.580	0.509	0.542	35	0.700	0.614	0.654	36	0.720	0.632	0.673
PredGPI	28	0.560	0.491	0.523	36	0.720	0.632	0.673	37	0.740	0.649	0.692
PredGPI*, **	29	0.487	0.580	0.507	35	0.588	0.700	0.507	36	0.605	0.720	0.522
PredGPI**	28	0.471	0.506	0.406	36	0.605	0.720	0.522	37	0.622	0.536	0.536
GPI-SOM	30	0.423	0.600	0.326	33	0.465	0.660	0.359	33	0.465	0.660	0.359
Big-Π	**39**	**0.812**	**0.780**	**0.848**	41	0.854	0.820	**0.891**	41	0.854	0.820	**0.891**

Abbreviation: ±0 = The number of correctly predicted ω-sites, ±1 = The number of ω-site predictions within one position away from the correct position, ±2 = The number of ω-site predictions within two positions away from the correct position, F1 = f1-score, Sens. = Sensitivity, Prec. = Precision.

* PredGPI provides two options, this is their conservative option.

** This is the result for PredGPI when weakly probable predictions are regarded as positive.

*** For the position prediction we use the experimentally tested sequences with known ω-sites. The precision is calculated w.r.t. the experimentally tested sequences with known ω-sites as well as all negative samples in the test set.

3.2 | Qualitative results

In the qualitative analysis we investigate the importance of biological features when NetGPI predicts GPI-anchor presence and the ω-site. In addition, we do a statistical analysis of the ω-site composition to understand the neighborhood of true and predicted ω-site positions. Lastly, we investigate model likelihood of the predictions, and how it relates to model correctness, on the held-out test set.

3.2.1 | Feature Importance Analysis

Figure 2 illustrates the results of the LIME analysis for both positive (see Figure 2a) and negative (see Figure 2b) samples. We observe that the presence of a hydrophobic tail contributes the most towards a positive prediction. This is consistent with the literature [6], which defines the presence of a hydrophobic region from the position $\omega + 10$. From that position the feature importance is much higher than for the rest of the sequence, which means that the main feature driving the positive prediction of NetGPI is the presence of the hydrophobic region. Regarding the negative predictions, we observe that the amino acids contributing the most towards a negative prediction are charged and polar amino acids. This indicates that the model is attributing higher importance to non-hydrophobic amino acids, indicating a lack of hydrophobic tail, when making a negative prediction.

3.2.2 | ω-site composition

Of the 50 experimentally verified ω-sites 54% are Serine, while the other amino acids observed are Asparagine, Glycine, Aspartic acid, Cysteine and Alanine, in decreasing order of frequency. All Glycine and Cysteine ω-sites are correctly predicted, one of the Asparagine ω-sites is off by 2 positions and both of the Alanine sites are off by one, where the preceding Serine is predicted instead, see table 5. Positioning errors made by NetGPI are mostly specific to Aspartic acid. Of the experimentally verified ω-sites, 8% are Aspartic acid, however we predict it in 14% of the 50 experimentally

TABLE 5 NetGPI's and Big-Π's ω-site position prediction performance for the 50 true ω-site amino acid in the test set. We see that both models only predict 1 out of 4 Aspartic acid ω-sites correctly. NetGPI has 9 one-off errors, 7 of which are actually Serine ω-sites.

NetGPI	S (27)	N (9)	G (6)	D (4)	C (2)	A (2)
±0	19	8	6	1	2	0
±1	26	8	6	1	2	2
±2	26	9	6	1	2	2
Big-Π	**S (27)**	**N (9)**	**G (6)**	**D (4)**	**C (2)**	**A (2)**
±0	22	9	4	1	2	1
±1	23	9	5	1	2	1
±2	23	9	5	1	2	1

Abbreviation: ±0 = The number of correctly predicted ω-sites, ±1 = The number of ω-site predictions within one position away from the correct position, ±2 = The number of ω-site predictions within two positions away from the correct position.

verified ω-sites. NetGPI has 9 off by one errors, 7 of which are actually Serine ω-sites, all of whom belong to the species *Arabidopsis thaliana*. Of those, 6 are predicted as an Aspartic acid where the actual ω-site is the preceding Serine, and together they belong to the 4-mer PTSD – an ω-site motif that does not occur in our training set. Both Big-Π and NetGPI are unable to position 3 out of 4 Aspartic acid ω-sites, see table 5. This may be related to the $\omega + 2$ position, as these 3 samples have a non-standard amino acid (i.e. something other than G, A, or S).

It is worth mentioning that out of the 50 experimentally verified, 13 belong to the species *Arabidopsis thaliana* and 14 to *Homo sapiens*. Only one of the 50 proteins with an experimentally verified ω-site is predicted non GPI-anchored by NetGPI. This example has a very unusual $\omega + 2$ amino acid, namely Lysine (K).

3.2.3 | Likelihood and correctness

In addition to the classification of the sequence and the most likely position of the ω-site, NetGPI reports the likelihood of the chosen position. For positive predictions this is the predicted ω-site, while for negative predictions it is the sentinel.

As our model is trained with cross entropy, it is penalized with a logarithm of the correct prediction. If we predict incorrectly, with a very low likelihood for the correct position, the loss can be immense. We should thus expect that answers with a high likelihood are more credible.

In Figure 3 we display the likelihood distribution of the predictions on the held-out test set. We observe differences in the likelihood of correct and incorrect predictions implying a correlation between likelihood and correctness. Furthermore, we observe higher likelihood in negative predictions than positive. This is expected as this is the probability distribution over the last 100 amino acids as well as the added sentinel, where only the sentinel position denotes a negative prediction, while a positive prediction is spread across the 100 amino acid positions. This means that positive prediction likelihood has to cover all potential ω-site positions, while the negative prediction likelihood is limited to one position. Therefore, using the likelihood as ranking should be done separately for negative and positive results.

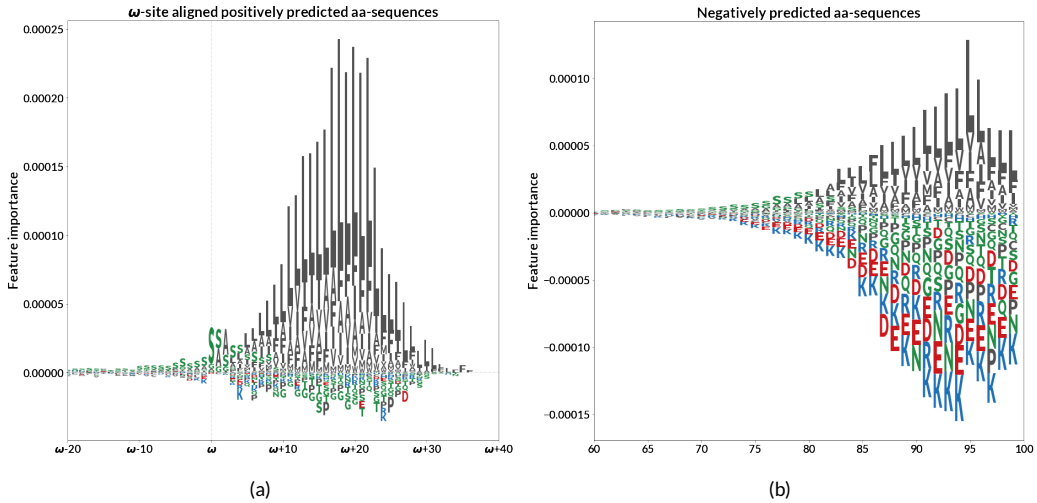
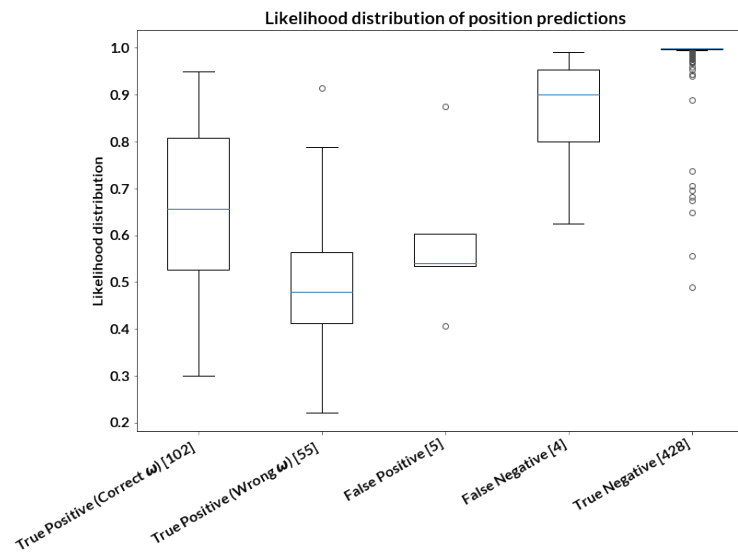


FIGURE 2 This Figure shows the logo plots of the LIME analysis for both positive (2a) and negative (2b) samples from the test set. The logo plots are colored according to amino acid properties, where blue means positively charged, green means polar, red means negatively charged and gray means hydrophobic amino acids. The positive set (2a) is aligned to the predicted ω-site, while the negative set (2b) is aligned to the C-terminus. Positive feature importance contributes to a positive prediction whereas a negative feature importance contributes to a negative one. We see that the presence of a hydrophobic tail contributes the most towards a positive prediction, whereas charged and polar amino acids contribute the most towards a negative prediction.

FIGURE 3 The likelihood distribution for true positive, false positive, false negative and true negative predictions of the held-out test set. True positive are split into correctly positioned ω-sites and incorrectly positioned. The number of samples behind each are displayed in brackets.



4 | CONCLUSION

We have shown that GPI-anchor prediction can be improved using recurrent neural networks and up-to-date datasets, achieving state-of-the-art performance. Comparison with previous methods is challenging as there exists no standard

dataset for training and testing predictive methods. Given progress in protein annotation, we publish a new homology partitioned training and test set, using experimentally verified proteins for testing and manually annotated predicted proteins for training. However, due to the new dataset definition, the performance of current methods could be overestimated as their training sets overlap with our test set.

Our results show that proteins manually annotated by prediction methods or sequence similarity are useful for training a GPI-anchor predictor to perform well when evaluated on experimentally verified ω-sites. However, using these data comes with a caveat; ω-site predictions are sometimes off by one position. We believe that this limitation is necessary in order to obtain a larger training set and create a completely independent test set of experimentally verified GPI-anchors. If we were to use only the experimentally verified GPI-anchors to train and test the predictor, we would not have enough training samples to teach a deep neural network classifier, and the resulting test set would be too small to be representative.

A web server implementing NetGPI is available at https://services.healthtech.dtu.dk/service.php?NetGPI-1.0, and our training and testing data set can be downloaded from the same site.

ENDNOTES

REFERENCES

[1] Brul S, King A, Van der Vaart J, Chapman J, Klis F, Verrips C. The incorporation of mannoproteins in the cell wall of S. cerevisiae and filamentous Ascomycetes. Antonie van Leeuwenhoek 1997;72(3):229–237.

[2] Kawagoe K, Kitamura D, Okabe M, Taniuchi I, Ikawa M, Watanabe T, et al. Glycosylphosphatidylinositol-anchor-deficient mice: implications for clonal dominance of mutant cells in paroxysmal nocturnal hemoglobinuria. Blood 1996;87(9):3600–3606.

[3] Takeda J, Miyata T, Kawagoe K, Iida Y, Endo Y, Fujita T, et al. Deficiency of the GPI anchor caused by a somatic mutation of the PIG-A gene in paroxysmal nocturnal hemoglobinuria. Cell 1993;73(4):703–711.

[4] Masterson WJ, Raper J, Doering TL, Hart GW, Englund PT. Fatty acid remodeling: a novel reaction sequence in the biosynthesis of trypanosome glycosyl phosphatidylinositol membrane anchors. Cell 1990;62(1):73–80.

[5] Mayor S, Riezman H. Sorting GPI-anchored proteins. Nature Reviews Molecular Cell Biology 2004;5(2):110.

[6] Orlean P, Menon AK. GPI anchoring of protein in yeast and mammalian cells, or: how we learned to stop worrying and love glycophospholipids. J Lipid Res 2007;48(5):993–1011.

[7] Varki A, Cummings R, Esko J, Freeze H, Hart G, Marth J. Glycophospholipid Anchors. Cold Spring Harbor Laboratory Press; 1999. `https://www.ncbi.nlm.nih.gov/books/NBK20711/`.

[8] Fankhauser N, Mäser P. Identification of GPI anchor attachment signals by a Kohonen self-organizing map. Bioinformatics 2005;21(9):1846–1852.

[9] Pierleoni A, Martelli PL, Casadio R. PredGPI: a GPI-anchor predictor. BMC Bioinformatics 2008;9:392. `https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2571997/`.

[10] Graves A. Supervised sequence labelling. In: Supervised sequence labelling with recurrent neural networks Springer; 2012.p. 5–13.

[11] Jurtz VI, Johansen AR, Nielsen M, Almagro Armenteros JJ, Nielsen H, Sønderby CK, et al. An introduction to deep learning on biological sequence data: examples and solutions. Bioinformatics 2017;33(22):3685–3690.

[12] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–444.

[13] Eisenhaber B, Bork P, Eisenhaber F. Prediction of Potential GPI-modification Sites in Proprotein Sequences. Journal of Molecular Biology 1999;292(3):741–758. http://www.sciencedirect.com/science/article/pii/S0022283699930693.

[14] Eisenhaber B, Schneider G, Wildpaner M, Eisenhaber F. A Sensitive Predictor for Potential GPI Lipid Modification Sites in Fungal Protein Sequences and its Application to Genome-wide Studies for Aspergillus nidulans, Candida albicans Neurospora crassa, Saccharomyces cerevisiae and Schizosaccharomyces pombe. Journal of Molecular Biology 2004;337(2):243–253. http://www.sciencedirect.com/science/article/pii/S002228360400083X.

[15] Eisenhaber B, Wildpaner M, Schultz CJ, Borner GHH, Dupree P, Eisenhaber F. Glycosylphosphatidylinositol Lipid Anchoring of Plant Proteins. Sensitive Prediction from Sequence- and Genome-Wide Studies for Arabidopsis and Rice. Plant Physiology 2003;133(4):1691–1701. https://www.ncbi.nlm.nih.gov/pmc/articles/PMC300724/.

[16] Almagro Armenteros JJ, Tsirigos KD, Sønderby CK, Petersen TN, Winther O, Brunak S, et al. SignalP 5.0 improves signal peptide predictions using deep neural networks. Nature Biotechnology 2019 Feb;37(4):420–423. https://www.nature.com/articles/s41587-019-0036-z.

[17] UniProt Consortium. UniProt: a hub for protein information. Nucleic acids research 2014;43(D1):D204–D212.

[18] Li W, Godzik A. Cd-hit: a fast program for clustering and comparing large sets of protein or nucleotide sequences. Bioinformatics 2006;22(13):1658–1659.

[19] Vinyals O, Fortunato M, Jaitly N. Pointer Networks. arXiv:150603134 [cs, stat] 2015;http://arxiv.org/abs/1506.03134.

[20] Merity S, Xiong C, Bradbury J, Socher R. Pointer Sentinel Mixture Models. arXiv:160907843 [cs] 2016;abs/1609.07843. http://arxiv.org/abs/1609.07843.

[21] McCann B, Keskar NS, Xiong C, Socher R. The Natural Language Decathlon: Multitask Learning as Question Answering. arXiv:180608730 [cs, stat] 2018;abs/1806.08730. http://arxiv.org/abs/1806.08730.

[22] Hochreiter S, Schmidhuber J. Long Short-Term Memory. Neural Comput 1997 Nov;9(8):1735–1780. http://dx.doi.org/10.1162/neco.1997.9.8.1735.

[23] Mikolov T, Sutskever I, Chen K, Corrado G, Dean J. Distributed Representations of Words and Phrases and Their Compositionality. In: Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2 NIPS'13, USA: Curran Associates Inc.; 2013. p. 3111–3119. http://dl.acm.org/citation.cfm?id=2999792.2999959.

[24] Schuster M, Paliwal KK. Bidirectional Recurrent Neural Networks. Trans Sig Proc 1997 Nov;45(11):2673–2681. http://dx.doi.org/10.1109/78.650093.

[25] Werbos PJ. Backpropagation through time: what it does and how to do it. Proceedings of the IEEE 1990;78(10):1550–1560.

[26] Baldi P, Brunak S, Chauvin Y, Andersen CA, Nielsen H. Assessing the accuracy of prediction algorithms for classification: an overview. Bioinformatics 2000;16(5):412–424.

[27] Zaremba W, Sutskever I, Vinyals O. Recurrent neural network regularization. arXiv preprint arXiv:14092329 2014;.

[28] Kingma DP, Ba J. Adam: A Method for Stochastic Optimization. arXiv:14126980 [cs] 2014;abs/1412.6980. http://arxiv.org/abs/1412.6980.

[29] Paszke A, Gross S, Chintala S, Chanan G, Yang E, DeVito Z, et al. Automatic Differentiation in PyTorch. In: NIPS Autodiff Workshop; 2017. https://openreview.net/forum?id=BJJsrmfCZ&noteId=BJJsrmfCZ.

[30] Ribeiro MT, Singh S, Guestrin C. "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv:160204938 [cs, stat] 2016;abs/1602.04938. http://arxiv.org/abs/1602.04938.