A Symbolic Regression Screening Approach Within Peptide Optimisation

Aidan Murphy[1(✉)], Mark Kocherovsky[2], Nir Dayan[2], Ilya Miralavy[2], Assaf Gilad[2], and Wolfgang Banzhaf[2]

[1] University College Dublin, Dublin, Ireland
aidan.murphy@ucd.ie
[2] Michigan State University, East Lansing, MI, USA
{kocherov,dayannir,miralavy,gilad,banzhafw}@msu.edu

Abstract. The Protein Optimization Evolving Tool is a genetic programming based peptide generation tool which has successfully created novel peptides with improved performance for MRI imaging. However, like all supervised machine learning techniques, it may overfit to its library of training peptides and create peptides which do not improve functionality. To overcome this problem we create symbolic regression models to act as another predictor of peptide function. We create a set of 76 features of physicochemical, theoretical and composite properties for each peptide and evolve the models using Grammatical Evolution on two datasets, one containing 74 peptides and the other 100 peptides. Models trained using these 76 features can successfully predict peptide functionality with a median MSE of 0.427 on the first dataset and 0.179 on the larger dataset, achieving state of the art results on both. We next investigate if a reduced set of 8 real-world features, which could result in more interpretable models, can accurately predict protein functionality. The models created on this reduced set were outperformed by model with used the full set on features on the first dataset but were statistically equivalent on the second dataset. Finally, we down sample the data at 10%, 33% and 50% to evaluate the robustness of this approach. Our results show that models trained on as little as 7 peptides can be used as an additional measure of functionality within the Protein Optimization Evolving Tool.

Keywords: Peptide Generation · Symbolic Regression · Grammatical Evolution

1 Introduction

Protein design and discovery is the foundation of many medical advances, from drug and MRI contrast design, to creating proteins for cellular reprogramming. Recent advances in structural protein prediction have enabled rapid design of proteins with desired structural properties. However, how to design proteins with specific functional properties is not well-understood. This is critically important

P. García-Sánchez et al. (Eds.): EvoApplications 2025, LNCS 15613, pp. 492–506, 2025.
https://doi.org/10.1007/978-3-031-90065-5_30

for smaller proteins, peptides, where there is little to no secondary/tertiary structure to predict.

A Genetic Programming (GP) approach for novel peptide discovery, the Protein Optimization Evolving Tool (POET), allows for the rapid functional improvement of peptides and proteins sequences [13,20]. POET differs from transformer-based Protein Language Model (PLM) approaches as improved, novel peptides can be discovered with very few training examples [5]. This is crucially important because many useful peptides will be novel and it is likely very few exist in the search space. Therefore, building a large library - necessary to train a PLM - may not be feasible or prohibitively expensive. This means POET can discover novel peptides that a PLM may be unable to produce while also being faster and cheaper to run. Crucially, POET allows for explainability of the peptides found, impossible when using PLMs.

The problem of overfitting in PLMs, when PLMs generate incorrect or nonsensical content, is a well known and researched phenomena [21]. To date, however, there has been no research examining the problem of overfitting within POET. Due to the small peptides which POET creates (often between 10 and 12 amino acids long), using powerful protein structure prediction tools, such as AlphaFold, will not be insightful because processes such as folding will not carry important influence in such small proteins. Therefore, we propose to use a symbolic regression approach as a secondary prediction tool to screen generated peptides from POET and remove predictions which may be a result of overfitting in the POET process. We use Structural Grammatical Evolution to create symbolic regression models to predict protein function using as features 76 physicochemical, theoretical and composite properties of the generated peptides. We also build models using a reduced set of real world features in order to increase interpretability of the best models found. Finally, we randomly sample the original dataset to create datasets that are 10%, 33% and 50% of the size of the original. We train models using these datasets in order to gauge how effective and robust the proposed symbolic regression methods can be on screening POET predictions on different domains when there may be very limited data.

Section 2 reviews the background to this research, including explaining peptides, peptide creation using AI and POET and how overfit predictions using POET can be identified. It also briefly discusses Grammatical Evolution and Structured Grammatical Evolution, the technique used to create the models in this study. Section 3 details the domain in which the peptides are being created, describes the features used within our symbolic regression approach and the experimental parameters used. Section 4 presents the main results of the experiments described in Sect. 3. Finally, Sect. 5 summarizes the research and discusses future work suitable for investigation.

2 Background

2.1 Peptides

Peptides are molecules composed of amino acids (AA) joined by peptide bonds. Peptides are short sequences and usually between 5 and 40 AA in length and have a diverse range of applications and many advantages, including their ability to be produced at a large scale. Peptides play an extremely important role in the life cycle of organisms and perform many natural functions, including forming muscle tissues, creating enzymes and are the building blocks of food. Combining amino acids in different orders and lengths produces a unique molecular structure and will result in a peptide with a unique biological function with differing levels of toxicity, stability, digestibility and other properties. Identifying peptides which can perform a particular application is therefore a difficult task as most of the possible proteins are not even being used or even explored by nature [2].

Significant advancements in synthetic and recombination technologies have been a driving force in bringing bio-active peptides back to center stage as therapeutic and diagnostic tools. The peptide global market is rapidly expanding with its value estimated at $14.4 billion, accounting for 1.5% of the total worldwide pharmaceutical market [1]. However, similar advances in artificial intelligence (AI) and data analytics methods for peptides have lagged behind this innovation. Recent advances in structural protein prediction have enabled rapid design of proteins with desired structural properties. However, understanding how to design proteins with specific functional properties is still challenging. This is critically important for peptides as, being very small proteins, there is often less secondary or tertiary structure to predict and the functionality has a great dependence on extrinsically bound factors.

A major stumbling block is the lack of high quality data because peptide datasets are generally much smaller than protein training datasets. This makes it much harder to build relevant AI representations, especially for non-canonical amino acids.

2.2 AI Peptide Design and POET

There are a total of 20 natural amino acids that can code for proteins, therefore finding a peptide comprising of 12 amino acids means the search space has 20^{12} amino acid sequences. This makes designing new peptides, even very small ones, an incredibly complex task due to the vastness of the search space. Indeed, millions of years of natural evolution have only created a fraction of the potential peptides possible.

Evolutionary algorithms are well suited for exploring such spaces and have already been used to design novel peptides. The Protein Optimization Engineering Tool (POET)[1] is a GP tool for predicting protein functionality to generate candidate peptides [13]. In contrast to transformer based Large Protein Language models it uses directed computational evolution to discover potentially

[1] https://github.com/elemenohpi/POET.

useful protein structures and substructures. It has been demonstrated previously to successfully create peptides with improved performance on magnetic resonance imaging using chemical exchange saturation transfer (CEST). POET has been shown it can produce peptides, using sparse data, with superior features that have not been - and may not be - developed by deep learning techniques.

POET uses GP to learn valuable protein subsequences. These sequences can either be represented as motifs, a collection of AAs, or as regular expressions [20]. For both approaches, POET assigns a weight to these sequences and assembles sequences and weights into a model. That is to say, a POET model, or individual, is a collection of sequences (either motifs or regular expressions) and an associated weight for each sequence. During evaluation of an individual, POET will check each peptide of the training set for the existence of the sequence in the individual and update the predicted score of the model for that peptide according to the weight of that sequence. The final score of a model is then compared to the known value of CEST contrast for the peptide and the model error is calculated for each peptide in the training set. The evolutionary process allows for both the changing of the sequences and the weight values in the model. Once enough generations of mutation and selection have been done, POET chooses the proteins that are fittest in predicted function.

2.3 Grammatical Evolution

Grammatical Evolution (GE) is a popular evolutionary computation technique which creates structures in any arbitrary language using a grammar [19], usually a context-free grammar written in Backus Naur Form (BNF) [10]. The grammar defines the possible structure of final programs or expressions and therefore establishes the search space of the problem. Grammars have shown many benefits, including allowing for domain knowledge to be easily encapsulated and incorporated into the creation of individuals, among many others [15].

To create our symbolic regression models Structured Grammatical Evolution (SGE) is used [12]. SGE is a variant of GE proposed to overcome some of the perceived weaknesses of GE while retaining GE's strengths. Namely, SGE aims to reduce the many invalid (unmapped) solutions GE can create during a run and the poor locality it exhibits when performing crossover and mutation. It has shown it can outperform GE on most tasks and continues to grow in popularity. In the context of creating a peptide regression model, SGE has shown that it suffers from less disruption through crossover and mutation than traditional GE, therefore creating solutions which tend to exhibit limited bloat. Having smaller solutions with as little bloat as possible is crucial for model interpretability.

2.4 Peptide Screening

A unique characteristic of POET predictions for CEST contrast is that they have been experimentally synthesized and validated under wet-lab laboratory conditions for both POET approaches, using motifs and regular expressions. 10

peptides were generated by each technique for each of two cycles, or epochs, giving a total of 40 peptides.

Given the time and cost associated with these experiments, it is crucial that POET produces high quality predictions and does not suffer from hallucinations. This can easily happen when very short sequences (as low as 1 or 2 AAs) are given particularly large weights, which will result in POET potentially overusing small sequences to make predictions repeating the same AA continually. From an expert's point of view it can be difficult to distinguish if POET is being *creative* or nonsensical.

There are some potential avenues which can provide context to POET generated peptides:

Structural Information: Protein structure prediction algorithms, such as the revolutionary AlphaFold [8], have transformed protein modeling and have come close to solving the protein folding question. However, POET usually creates peptides as small as 10-12 AAs in length which do not exhibit these folding behaviors. Therefore, as no complex folding or other behavior will occur, AlphaFold or any other system of this type will not generate many useful insights into the predicted peptide.

Protein Language Model: Foundational PLMs have the ability to generate or predict the fitness of a protein sequence. However as above, CEST contrast peptides are often too small to be used. For example, ProGen2 [17], the state-of-the-art suite of PLMs, specifies a minimum protein size of 50 AA for prediction.

Fine-tuned PLM: Creating a custom PLM, specializing on the particular task at hand is another potential solution. However, as well as having to deal with hallucinations in this model, the data required for fine-tuning is often orders of magnitude larger that what is available from experiments. As well as the expense of collecting and curating this vast amount of data, there is also the expense of training and running the model.

Peptide Feature Regression Model: A final potential strategy to give a generated POET peptide context could be to build a regression model based on the features of the peptides in the training set, and assign the peptide a secondary score [11]. These features would be a mix of properties relating to the AA make up of the peptide and would ensure that POET predictions share similar characteristics (positive or negative charge, molecular weight, etc.) to those in the training set. It is this approach we shall investigate in this paper.

3 Experimental Setup

3.1 CEST Contrast Dataset

CEST is a magnetic resonance imaging (MRI) contrast approach in which peptides with exchangeable protons or molecules are saturated and detected indirectly through enhanced water signals after transfer [6]. Two datasets are used for Epoch 1 and Epoch 2. Epoch 1's training set consists of 74 peptides, all of length 12, with known normalized CEST contrast scores, its test set has 20 peptides (10 predictions each from POET using motifs and regular expressions). Epoch 2 has a training size of 100, made up of the original 74 peptides and the 20 predictions from Epoch 1 plus an additional 6 peptides. The test set of Epoch 2 is 20 further predictions from POET, again 10 from each POET variant.

3.2 Peptide Features

To build the peptide symbolic regression model we collected 76 peptide features related to several physicochemical, theoretical and composite properties of the 12 amino acids constituting a sequence [18]. We use these features to build a regression model to act as a secondary predictor for CEST contrast which can be used to screen the predicted sequences from POET. A description of each of the features used is shown in Table 3.2.

While helpful in prediction, many of these features are composite or constructed features and do not have real world significance for domain experts. In order to increase the interpretability of final models, we also use a reduced set of real world properties which can be given to a domain expert and should allow them to apply their expertise and potentially aid POET during the evolutionary search or in the optimisation step of generating new peptides.

The features included in the reduced set during experimentation were:

1. Charge
2. Polarity
3. Hydrophobicity
4. H-bonding
5. Molecular Weight
6. Mass over Charge
7. Isoelectric Point
8. Lipophilicity

Table 1. List of all features used to train the symbolic regression models, including the number of features, their names, and descriptions.

Num of Features	Name	Description	Num of Features	Name	Description
1	index	Aliphatic index of the peptide sequence.	10	blosum	BLOSUM62 derived indices of the peptide sequence.
1	boman	Potential Protein Interaction of the peptide (boman index).	1	charge	Theoretical Charge of the peptide sequence.
3	cruciani	Cruciani properties of the peptide sequence.	6	fasgai	Factor Analysis Scales of Generalized Amino Acid Information vectors of the peptide sequence.
2	hydrophobic	Hydrophobic moment the peptide sequence at two angles (100 and 160).	1	hydrophobicity	Hydrophobicity index of the peptide sequence using the *KyteDoolittle* scale.
1	instaIndex	Instability index of the peptide sequence.	10	kidera	Kidera factors of the peptide sequence.
3	mswhim	The 3 principal components from PCA applied to MS-WHIM 3D description matrix.	1	mw	Molecular Weight of the peptide.
1	mz	Mass over charge (m/z) of the peptide sequence.	1	pI	The isoelectric point of the peptide sequence.
8	protFP	The ProtFP descriptors of the peptide sequence.	8	stScale	The structural topology scales of the peptide sequence.
5	tScale	T-scale of the peptide sequence.	8	vhseScale	Vectors of Hydrophobic, Steric, and Electronic properties of the peptide sequence.
5	zScales	Z-scales of the peptide sequence.	-	-	-.

3.3 Experimental Parameters

The full experimental setup and all associated parameters are shown in Table 2. The initial population was created using Local Optimised Probabilistic Tree Creation 2 (LO-PTC2) [14]. We compare the results of the SGE models with 3 other machine learning methods; Random Forest [3], XGBoost [4] and LightGBM [9]. The hyper-parameters for each of these methods underwent a simple grid-search optimisation prior to execution. The grammar used is shown in Fig. 1. There are four protected operators used: protected exponential (which returns 1 if a value error occurs), protected log (which returns 0 if a negative number or 0 is passed) and protected square root (which returns the absolute value of the argument passed). Protect division was not used due to its negative impact on generalisation performance [16].

Table 2. List of the main parameters used to run SGE

Parameter	Value
Runs	50
Total Generations	500
Population	300
Elitism	1%
Selection	Tournament (5)
Fitness Function	r^2
Crossover	0.9
Mutation	0.1
For LO-PTC2:	
Minimum Expansions:	4
Maximum Expansions:	30
Maximum Initial Evaluations:	50

$$
\begin{aligned}
&< exp >::= < expr >< op >< expr > \mid < pre\_op > (< expr >) \mid < var > \\
&< op >::= + \mid - \mid * \mid / \\
&< pre\_op >::= sin \mid cos \mid exp \mid log \mid sqrt \\
&< var >::= x[0] \mid ... \mid x[75]
\end{aligned}
$$

Fig. 1. Grammar used for the SGE symbolic regression experiments.

4 Results

4.1 Best Regression Method

A comparison of the regression approaches for each Epoch dataset is shown in Table 3. SGE was seen to perform the best of all models on both epochs.

In Epoch 1 it attained a mean squared error score (MSE) of 0.104, followed by LightGBM and Random Forest at 0.212 and 0.213, respectively. The worst performing method was XGBoost at 0.266. This is perhaps unsurprising because GP has been shown to be particularly effective in creating accurate symbolic regression models on very few data points. On the larger dataset in Epoch 2, SGE remains the best performing method, halving its MSE and achieving a best test score of 0.051. It again outperforms LightBGM, 0.078, Random Forest, 0.086 and XGBoost, 0.129. These results show that SGE finds the state-of-the-art results and, crucially, SGE will yield an interpretable model which allows inspection by a domain expert.

Table 3. Experimental Test Results for the Best Model found for each method on both Epoch 1 and Epoch 2 Test sets. The best result is shown in bold.

Epoch	Method	Best MSE
1	**SGE**	**0.104**
	Random Forest	0.213
	XGBoost	0.266
	LightGBM	0.212
2	**SGE**	**0.051**
	Random Forest	0.086
	XGBoost	0.129
	LightGBM	0.078

4.2 POET Screening

We next investigate if the performance of SGE is accurate enough to act as a screening method for POET. As well as reporting the overall MSE, results were split by POET type (motif and regular expression) to investigate if there was any difference in the performance of predicting peptides generated by the two approaches. The first approach, *Full*, uses all 76 features to build models while the second approach, *Reduced*, uses the reduced set of 8 real-world features. Both approaches were performed on both sets of data, Epoch 1 and Epoch 2. The results from these experiments are seen in Table 4. We report the median test performance of the best of run model across all 50 runs. The performance of the models on each POET approach, using motif's or regular expressions, is shown (third and fourth columns) as well as the overall MSE on all peptides predicted using POET (fifth column). The last column show's the MSE of the best model found from all 50 runs. Using Epoch 1 data, models performed better predicting peptides produced from POET motif than POET with regular expressions, regardless of using the full or reduced data. This was not the case for Epoch 2, with both performing equally.

Table 4. Experimental Test Results using both the full and reduced datasets. Each setup was run 50 times. Underlined results denote that the setup performed significantly better than the other according to Wilcoxon tests.

Epoch	Dataset	Median MSE Motif	Median MSE Regex	Median MSE	Best MSE
1	Full	0.222	<u>0.427</u>	<u>0.337</u>	0.104
	Reduced	0.184	0.688	0.469	0.205
2	Full	0.185	0.179	0.188	0.051
	Reduced	0.231	0.253	0.257	0.031

It can be seen that using the full data in Epoch 1 leads to more accurate symbolic regression models, column four in Table 4, with models trained with all 76 features, Full, statistically significantly outperforming models trained on the reduced dataset, Reduced. Full models attained a median MSE of 0.337 compared to a median MSE of 0.469 for models trained using on 8 features. The best model found on the Epoch 1 dataset used the Full dataset, finding an MSE of 0.104.

Using the larger Epoch 2 dataset both the Full and Reduced feature sets were seen to find comparable results, with their performance difference not statistically significantly different. Indeed, the best model found throughout all experimentation was created using the reduced dataset, attaining an MSE of 0.031, compared to the full dataset with found a best test error of 0.051

The plots of the best performing models for Epoch 1 and Epoch 2 can be seen in Fig. 2 and Fig. 3, respectively. Each plot shows the predicted score from the SGE symbolic regression model on the x-axis and actual wet-lab measurement of the peptides which were predicted using POET on the y-axis. Each shape represents a different POET approach, circles are motif and triangles are regular expression, with filled shapes designating predictions from the model where all 76 features were used while hollow shapes are those predictions from the model which used the reduced set. It should be noted that each of these peptides was predicted by POET to have scores ≥ 2.0. In a real world context, all peptides which would fail to score 1.5 would not be synthesized and measured in the wet lab due to the cost and their relative lack of improvement over the current best peptide. For Epoch 1, it can be seen that both SGE symbolic regression models using the full data (filled circles and triangles) and the reduced data (hollow circles and triangles) can successfully identify five of the worst performing peptides, correctly predicting scores of ≤ 1.5 rendering them not desirable for synthesis. These five are shown in the green shaded area and were from POET Motif, which the models were better at modeling. The models were not unerring, however, incorrectly missing two predictions which should have been removed (bottom right red area) and the reduced feature models incorrectly recommend to remove up to four peptides which were in fact promising (top left red area). Epoch 2 produced far better POET and symbolic regression models, easily observed in Fig. 3. There was only one poor POET prediction from POET

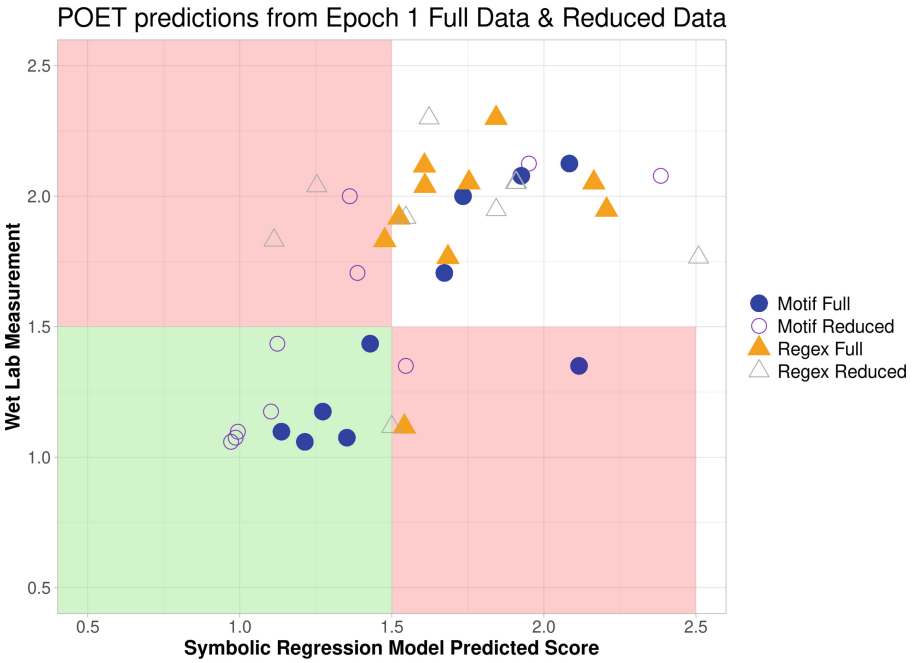

Fig. 2. Plot of the predicted scores from the symbolic regression model vs the actual measured CEST contrast in the wet lab for the POET predicted peptides in Epoch 1.

Regex and this was identified by models using the full and reduced features. It is notable too, that the symbolic regression models did not recommend to screen peptides which are promising as none fall in the red regions.

4.3 Explainability

We next look at the actual expressions which are found using SGE. By investigating the expressions themselves the domain expert may be able to understand the logic of the model and can aid the search by, for example, augmenting the grammar to bias certain features or encapsulate certain functionality to protect it from destruction. The best of run models from each setup from Table 4 are shown in Table 5. As highlighted earlier, the models using the full dataset make use of many composite features, such as the structural topology scales, which have no real world interpretation but are instead meta-features or components from PCA analysis. This is seen in the first and second expressions, which use $stScale3$, $ProtFP4$, $stScale8$ and $stScale7$. ST-scales were created by performing PCA on structural and topological variables of AAs. Likewise, the ProtFP descriptors are found using PCA and are useful for analysis but both sets of features inhibit interpretability. The models using only real world features do not contain these composite features but are more verbose and contain many non-linear functions making their interpretability equally challenging. While the

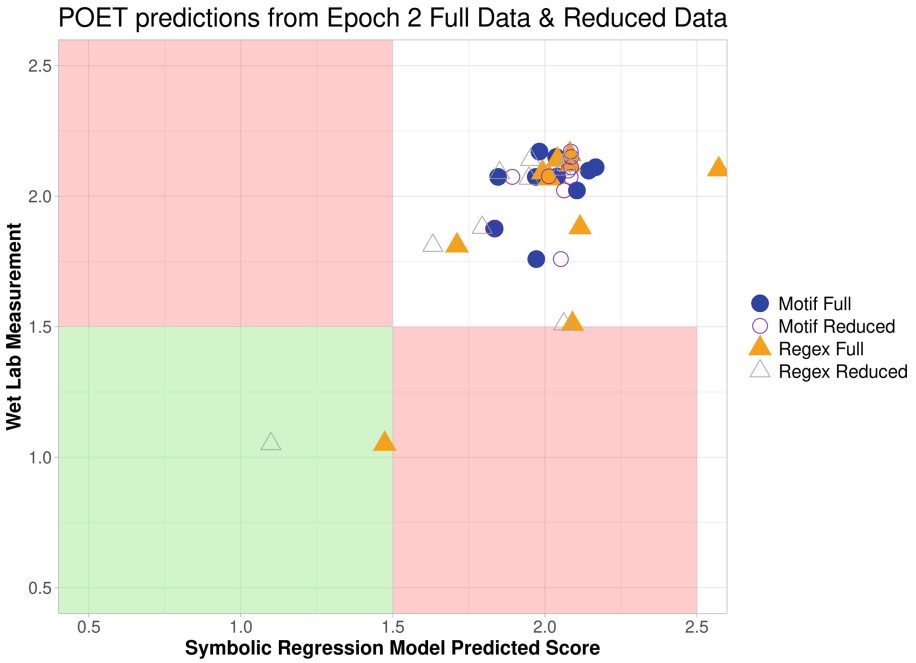

Fig. 3. Plot of the predicted scores from the symbolic regression model vs the actual measured CEST contrast in the wet lab for the POET predicted peptides in Epoch 2.

presence or absence of certain features may allow some insight to be gained into what logic the model is using, the models in their current state cannot be said to allow much explainability. Indeed, an initial consultation with a domain expert in the field confirmed that, while the predictions were indeed very useful, the models themselves in their current state are not.

Table 5. Best of run expressions found.

Epoch	Dataset	Expression
1	Full	$stScale3/(exp((exp((sin(ProtFP4) + exp(stScale8))))))$
	Reduced	$stScale3 * (exp((stScale7 - (exp((exp(ProtFP4))))))))$
2	Full	$(Polarity/(((exp(pI))/((Lipophilicity - (Polarity - ((sin((mw + mz)))$ $/(exp((cos(Hydrophobicity)))))))) * (sqrt(charge)))) - Hydrophobicity))$
	Reduced	$(cos(((Lipophilicity/(log(pI))) - (sin(((H - bonding * Lipophilicity)$ $/(sqrt(H - bonding)))))))$

4.4 Robustness on Much Smaller Datasets

We finally examined the robustness of the SGE symbolic regression approach by randomly sampling Epoch 1 data at 10%, 33% and 50% at the beginning of each run and training models on this severely downsampled dataset. The models were trained using all 76 features and the experimental parameters were identical to those used in Sect. 4.2. The results can be seen in Table 6. As with Table 4, we show the median test performance of the best of run model across all 50 runs, split by POET approach and overall, and report the results of the single best model found across all 50 runs. SGE showed surprisingly strong performance across all sample sizes and there no statistically significant difference was observed. A 10% random sample, yielding a dataset of just 7 or 8 peptides, allowed SGE to generate models with a median MSE of 0.643, worse than 0.337 found using the full dataset, but the best model attained an impressive MSE of 0.118 which is only slightly worse than the best model found in experimentation which achieved an MSE of 0.104. These results highlight the robustness of the proposed method and suggests it's applicability alongside POET in a wide variety of domains regardless of the size of training dataset.

Table 6. Experimental Test Results using the randomly sampled datasets using all features. Each setup was run 50 times. The results were not significantly different from each other, indicating that extreme down sampling still leads to robust models.

Epoch	Sample Used	Median MSE Motif	Median MSE Regex	Median MSE	Best MSE
1	10%	0.402	0.895	0.643	0.118
	33%	0.351	0.715	0.539	0.113
	50%	0.316	0.696	0.535	0.116

5 Conclusions and Future Work

We successfully created symbolic regression models using Structured Grammatical Evolution to identify overfit peptides produced using the Protein Optimization Evolving Tool. Symbolic Regression models were built using 76 features which consisted of physicochemical properties, theoretical properties and composite properties of the peptide sequences. Structured Grammatical Evolution found the state of the art models when compared to Random Forest, XGBoost and LighGBM on two datasets which aim to predict chemical exchange saturation transfer contrast, one consisting of 74 peptides and the other 100 peptides.

In order to increase the interpretability of the final models found, we conducted experiments on a reduced dataset which only contains real world peptide properties, such as charge and molecular weight. On the first test dataset the full feature models outperformed the reduced feature models, however no significant difference was observed on the second, larger test set. Both methods were able to identify overfit predictions and can be used a secondary measure of

peptide functional performance. The best expressions found were next examined for their interpretability. Reducing the number of composite features was not seen to greatly increase the insight into the models due to their size and use of highly non-linear expressions.

Finally, we randomly down sampled the first training dataset and conducted experiments with training sets containing as few a 7 peptides. Our results showed that these heavily reduced datasets, while worse than models trained on the full training data, could attain competitive performance and shows that symbolic regression approaches using peptide features can be used in with paltry amounts of peptide data allowing labs with limited resources to maximise the effectiveness of the Protein Optimization Evolving Tool.

There are many avenues for future work. One is to increase interpretability by making interpretability an explicit objective of the search. This may also be aided by removing non-linear functions in the grammar and performing further grammar simplifications. The approach could also be improved by using active learning to select the most relevant peptides to use for training, which may allow training sets with handfuls of peptides to achieve the same performance as those with hundreds or thousands of peptides [7].

References

1. Apostolopoulos, V., et al.: A global review on short peptides: Frontiers and perspectives. Molecules **26**(2), 430 (2021)
2. Baker, D.: What has de novo protein design taught us about protein folding and biophysics? Protein Sci. **28**(4), 678–683 (2019)
3. Breiman, L.: Random forests. Mach. Learn. **45**, 5–32 (2001)
4. Chen, T., Guestrin, C.: XgBoost: A scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 785–794 (2016)
5. Ferruz, N., Höcker, B.: Controllable protein design with language models. Nature Machine Intelligence **4**(6), 521–532 (2022)
6. Gilad, A.A., Bar-Shir, A., Bricco, A.R., Mohanta, Z., McMahon, M.T.: Protein and peptide engineering for chemical exchange saturation transfer imaging in the age of synthetic biology. NMR Biomed. **36**(6), e4712 (2023)
7. Haut, N., Banzhaf, W., Punch, B.: Active learning in Genetic Programming: Guiding efficient data collection for symbolic regression. IEEE Trans. Evolutionary Comput. (2024)
8. Jumper, J., Evans, R., Pritzel, A., Green, T., Figurnov, M., Ronneberger, O., Tunyasuvunakool, K., Bates, R., Žídek, A., Potapenko, A., et al.: Highly accurate protein structure prediction with Alphafold. Nature **596**(7873), 583–589 (2021)
9. Ke, G., et al.: Lightgbm: a highly efficient gradient boosting decision tree. Advances in neural information processing systems **30** (2017)
10. Knuth, D.E.: Backus normal form vs. backus naur form. Communications of the ACM **7**(12), 735–736 (1964)
11. Li, K., et al.: Explainable machine learning identifies multi-omics signatures of muscle response to spaceflight in mice. npj Microgravity **9**(1), 90 (2023)

12. Lourenço, N., Pereira, F.B., Costa, E.: SGE: A structured representation for Grammatical Evolution. In: International Conference on Artificial Evolution (Evolution Artificielle), pp. 136–148. Springer (2015)
13. Miralavy, I., Bricco, A.R., Gilad, A.A., Banzhaf, W.: Using genetic programming to predict and optimize protein function. PeerJ Physical Chemistry **4**, e24 (2022)
14. Murphy, A., Mahdinejad, M., Ventresque, A., Lourenço, N.: An investigation into structured grammatical evolution initialisation. Genet. Program Evolvable Mach. **25**(2), 24 (2024). https://doi.org/10.1007/s10710-024-09498-y
15. Murphy, A., Murphy, G., Dias, D.M., Amaral, J., Naredo, E., Ryan, C.: Human in the loop fuzzy pattern tree evolution. SN Computer Science **3**(2), 1–14 (2022)
16. Nicolau, M., Agapitos, A.: Choosing function sets with better generalisation performance for symbolic regression models. Genet. Program Evolvable Mach. **22**(1), 73–100 (2021)
17. Nijkamp, E., Ruffolo, J.A., Weinstein, E.N., Naik, N., Madani, A.: ProGen2: Exploring the boundaries of protein language models. Cell Syst. **14**(11), 968–978 (2023)
18. Osorio, D., Rondón-Villarreal, P., Torres, R.: Peptides: a package for data mining of antimicrobial peptides. Small **12**, 44–444 (2015)
19. Ryan, C., Collins, J.J., Neill, M.O.: Grammatical Evolution: Evolving programs for an arbitrary language. In: Banzhaf, W., Poli, R., Schoenauer, M., Fogarty, T.C. (eds.) EuroGP 1998. LNCS, vol. 1391, pp. 83–96. Springer, Heidelberg (1998). https://doi.org/10.1007/BFb0055930
20. Scalzitti, N., Miralavy, I., Korenchan, D.E., Farrar, C.T., Gilad, A.A., Banzhaf, W.: Computational peptide discovery with a Genetic Programming approach. J. Comput. Aided Mol. Des. **38**(1), 17 (2024)
21. Schmirler, R., Heinzinger, M., Rost, B.: Fine-tuning protein language models boosts predictions across diverse tasks. Nat. Commun. **15**(1), 7407 (2024)