S^2-PepAnalyst: A Web Tool for Predicting Plant Small Signalling Peptides

Kelly L. Vomo-Donfack[1,2], Mariem Abaach[3], Ana M. Luna[4], Grégory Ginot[1], Verónica G. Doblas[4], Ian Morilla [1,2,4*]

[1] Université Sorbonne Paris Nord, LAGA, CNRS, UMR 7539, F-93430, Villetaneuse, France

[2] University of Malaga, Department of Applied Mathematics, MLiMO, 29010, Málaga, Spain

[3] MAP5 UMR CNRS 8145, Université de Paris, 45 rue des Saints-Pères, 75006 Paris, France

[4] Instituto de Hortofruticultura Subtropical y Mediterránea La Mayora, Universidad de Málaga-Consejo Superior de Investigaciones Científicas, 29010, Málaga, Spain

Small signalling peptides (SSPs) play crucial roles in plant growth, development, and stress responses. However, accurately identifying and characterising SSPs remains challenging due to their structural diversity and the limitations of current prediction tools. Here, we introduce S^2-PepAnalyst, a novel web tool designed to enhance the prediction of SSPs in plants. By integrating comprehensive plant-specific datasets into a machine learning model, S^2-PepAnalyst offers versatility, improved accuracy of 99.5% on average, and reliability with a low rate of false negatives compared to existing tools. S^2-PepAnalyst provides essential resources for plant biologists and facilitates new discoveries in plant peptide signalling.

Small Signalling Peptides (SSPs), defined as proteins consisting of fewer than 200 amino acids (AAs), are among the smallest biological molecules in the plant proteome, playing diverse roles in plant growth, development, reproduction, symbiotic interactions, and stress responses [1-3]. The precursor (named prepropeptide) possesses a signal sequence in the N-terminal part, directing it for vesicle-based transport into the apoplast. The secreted peptide is recognised by a transmembrane receptor, triggering signal transduction [4].

Despite the increasing number of studies on SSPs revealing the vast diversity of structures impacting on the enconding genes, their experimental identification remains challenging due to their low abundance and specific expression patterns in certain cell types or developmental stages, making proteomic approaches often unfeasible [5,6].

Current computational prediction tools for SSPs, such as SignalP 6.0 [7], face significant limitations in accurately identifying these peptides, leading to high rates of false positives and negatives. Other recent methods, as the approach developed in [8] to identify phytocytokines, offer a low versatility. Moreover, the lack of comprehensive plant-specific datasets further exacerbates these limitations, necessitating the development of more advanced and specialised prediction tools [9]. To address these challenges, we present S^2-PepAnalyst, a simple and versatile all-in-one web tool designed to enhance the prediction of SSPs in plants (Fig. 1).

S^2-PepAnalyst integrates plant-specific datasets with scalable machine learning models, including protein language models (LMs) [10] and geometric-topological analysis [11, 12]. Our tool leverages the bidirectional DNA-BERT transformer, utilizing vast datasets comprising millions of unannotated protein sequences across diverse biological domains to generate semantic representations of proteins, encoding their biological characteristics and structures [13-16]. These representations are refined using a specific reinforcement learning algorithm designed to achieve better generalisation to new, unseen protein sequences, thereby improving prediction accuracy (Fig. 1a,b).

The preprocessing steps involve extensive curation of plant-specific datasets, ensuring high-quality input data for the model (Methods). Data preprocessing includes signal sequence identification, sequence alignment, and noise reduction techniques to enhance model training. By transforming AA sequences into high-resolution images and employing pre-trained deep convolutional neural networks (CNNs) for cleavage site prediction [17], our model captures critical topological features essential for accurate SSP classification (Fig. 1c).

Our benchmarking tests reveal that S^2-PepAnalyst outperforms existing tools such as SignalP 6.0 in predicting small signalling peptides in plants (Methods). The model demonstrates higher

sensitivity and specificity across various plant proteomes, significantly reducing false negatives. The success rate achieved the 98.1% on average for all the proteomes, reflecting the model's precision and robustness (Supplementary Fig. 1). For example, the final validation accuracies for *Solanum lycopersicum* were 98.18%, 98.18%, and 98.01% for non-overfitted feature extraction methods. Similar results were observed across other proteomes (https://github.com/MorillaLab/s2PEPANALYST/).

We initially interrogated our model with a curated baseline dataset (supplementary Table 1), achieving a 100% success rate in predicting SSP regions (Fig. 2a) and subsequently classifying them into their corresponding families: post-translationally modified peptides (e.g., PSK, CLE, IDA, etc.) or cysteine-rich peptide (e.g., RALF, EPF, LURE, etc.). The accuracy, precision, and Matthews correlation coefficient (MCC), and recall rates were 0.99 and 1, respectively (Fig. 2b middle). In comparison, SignalP 6.0 achieved a 65% success rate without family classification output, as it was not designed for such tasks (Fig. 2a). This family classification is an additional advantage of our model, which includes double-checking based on literature fine-tuned alignments to initially non-SSP predicted sequences [18].

In a generalised dataset of 779 sequences from 46 different proteomes [19], our model achieved a 99% success rate compared to 84% for SignalP 6.0, demonstrating its superior predictive capabilities (Fig. 2a). Notably, SignalP 6.0 reached its peak accuracy of 88% with 1,177 sequences (Methods), whereas our model maintained a high success rate of minimum 98% regardless of the number of sequences (Fig. 2a). This indicates that while SignalP 6.0 requires a substantially larger number of sequences to achieve good accuracy, S^2-PepAnalyst performs well even with fewer sequences. This characteristic is particularly advantageous, as signalling peptide research typically involves datasets with a limited number of sequences. In terms of false negatives, our model exhibited a significantly lower rate (2 out of 80 for the 1,177-sequence dataset and 0 out of 8 for the 18-sequence dataset) compared to SignalP 6.0 (Fig. 2b). Furthermore, S^2-PepAnalyst consistently outperformed SignalP 6.0 in accuracy, MCC (0.985 on average), and F1-Score, achieving a precision rate of 97%, closely matching SignalP 6.0's peak precision of 99% (Fig. 2c).

S^2-PepAnalyst offers a significant improvement over existing prediction tools. Unlike SignalP 6.0, has a low false negative rate, while providing a simpler, all-in-one experience, than other approaches such the one described in Rhodes et Zipfel, 2024. Our tool's superior performance in predicting SSPs, underscores its robustness and versatility. However, it is essential to acknowledge its limitations. The model's performance is inherently dependent on the quality and diversity of the input data. Future work should focus on expanding the datasets to include a broader range of plant species and environmental conditions to further enhance the tool's accuracy and applicability.

This study presents S^2-PepAnalyst, a machine learning model designed to enhance the prediction of SSPs in plants, covering a wide range of peptide families. By leveraging plant-specific datasets and employing geometric-topological analysis, S^2-PepAnalyst accurately predicts sequences of unknown origin and evolutionarily distant proteins, thus positioning our model favourably for future research. This advancement enables a deeper understanding of plant peptide signalling mechanisms, with significant implications for plant biology research.

References

1. Olsson, V. et al. Look closely, the beautiful may be small: precursor-derived peptides in plants. *Annu. Rev. Plant Biol*. 70, 153-186 (2019).
2. Czyzewicz, N. et al. Plant peptide signaling: a new level of complexity. *Annu. Rev. Cell Dev. Biol*. 29, 21–51 (2013).
3. Takahashi et al., Hormone-like and small coding genes in plant stress signaling and development. *Current Opinion in Plant Biol*. 51, 88-95 (2019).
4. Bender, K.W., Zipfel, C. Paradigms of receptor kinase signaling in plants. *Biochem. J.*, 480, 835-854 (2023).
5. Han, Z., Sun, Y., Chai, J. Structural insight into the activation of plant receptor kinases. *Current Opinion in Plant Biology* 20, 55-63 (2014).
6. Moussu, S., Santiago, J. Structural biology of cell surface receptor-ligand interactions. *Curr Opin Plant Biol*. 52, 38-45 (2019).
7. Taufel, F. et al. SignalP 6.0 predicts all five types of signal peptides using protein language models. *Nat. Biotechnol*. 40, 1023-1025 (2022).

8. Rhodes, J. and Zipfel, C. Identification of Bioactive Phytocytokines Using Transcriptomic Data and Plant Bioassays. Edited in Plant Peptide Hormones and Growth Factors by Schaller. *Methods Mol Biol.* 2731, 23-25 (2024).
9. Hellmann, E. MtSSPdb: A New Database for the Small Secreted Peptide Research Community. *Plant Physiol*. 183, 31–32 (2020).
10. Rao, R. et al. Transformer protein language models are unsupervised structure learners. BioRxiv (2020).
11. Abaach, M., Morilla, I. GeoTop: Advancing Image Classification with Geometric-Topological Analysis. arXiv:2311.16157.
12. Gouiaa, F., Vomo-Donfack, K.L., Tran-Dinh, A., Morilla, I. Novel dimensionality reduction method, Taelcore, enhances lung transplantation risk prediction. *Computers in Biology and Medicine* 169, 107969 (2024).
13. Ji, Y., Zhou, Z., Liu, H. & Davuluri, R. V. DNABERT: pre-trained bidirectional encoder representations from transformers model for DNA-language in genome. *Bioinformatics* **37**, 2112–2120 (2021).
14. Sanabria, M., Hirsch, J., Joubert, P.M. *et al.* DNA language model GROVER learns sequence context in the human genome. *Nat Mach Intell* (2024).
15. Dalla-Torre, H. et al. The Nucleotide Transformer: building and evaluating robust foundation models for human genomics. BioRxiv (2023).
16. Yang, M. et al. Integrating convolution and self-attention improves language model of human genome for interpreting non-coding regions at base-resolution. *Nucleic Acids Res.* **50**, e81 (2022).
17. Sara, T.S., Hasan, M., Ahmad, A., Shatabda, S. Convolutional neural networks with image representation of amino acid sequences for protein function prediction. *Computational Biology and Chemistry* 92. 107494 (2021).
18. Altschul,S.F., Gish, W., Miller, W., Myers, E.W., Lipman, D.J. “Basic local alignment search tool”. *Journal of Molecular Biology* 215 (3): 403-410 (1990).
19. Campbell, L., Turner SR. A Comprehensive Analysis of RALF Proteins in Green Plants Suggests There Are Two Distinct Functional Groups. *Front. Plant Sci* 8, 37 (2017).

Methods

Sequence data. S^2-PepAnalyst leverages an extensive dataset of experimentally validated plant signalling peptides, augmenting the training data used by SignalP 6.0. This dataset includes a diverse range of plant species such as *Arabidopsis thaliana*, tomato, avocado (Hass and Gwen varieties), and mango [20-23]. These species were chosen due to their varied significance in global food security and agricultural sustainability. For broader analysis, s^2-PepAnalyst incorporates the Cleavage Site score (CS) from SignalP 6.0 as a pre-training step. Starting with one or multiple amino acid sequences of interest, derived, for instance, from some RNAseq studies, we pretrained the selected proteome to have only two classes: signalling or non-signalling peptides. From this resulting dataset, we chose short sequence amino acids ranging from 2 to 200 for our analysis in accordance with the observed literature of signal peptide recognition.

Modelling. S^2-pepAnalyst uses two different pretrained protein language models to encode amino acid sequences. In this study, we utilised Tasks Assessing Protein Embeddings (TAPE) [24], a semi-supervised learning task in protein biology, which generates a shaped tensor that is subsequently averaged to produce a vector representation of length 768. Additionally, we used Evolutionary Scale Modelling (ESM-2) [25], which provides an averaged vector of length 1280. Each vector was padded and reshaped to create images of sizes 28x28 and 36x36, respectively. These images were then used to capture topological information via GeoTop, an application of Topological Data Analysis (TDA) and Lipschitz-Killing Curvatures (LKCs) for feature extraction, which we embedded into the images for accuracy assessment [26-29].

In brief, for any two values s and $t \in \mathbb{R}$, where $s \leq t$, the relationship $X_t(x) \subseteq X_s(x)$ holds. This indicates an expanding family of binary images parameterised by t. This family is referred to as the "superlevel sets filtration" of X. The subset inherits its topological structure, allowing us to compute the homology groups (H_k) of $X_t(x)$.

Through a series of trial-and-error experiments, we established a set of thresholds denoted as T, consisting of 200 equidistant points spanning from the minimum to the maximum values of the images and their corresponding excursion sets. From these, we compute three scaled

geometrical features: area ($\check{\mathcal{A}}$), perimeter ($\check{\mathcal{P}}$), and Euler characteristic ($\check{\mathcal{E}}$), with $\widetilde{LKC} = \frac{LKC}{m^2}$, where m is the image of size.

Deep learning integration. LeNet-5 [30], a convolutional neural network (CNN) architecture, was then employed. This architecture was embedded in a reinforcement learning algorithm to capture the diverse information stored in the various LeNet models (Fig. 1d). The first model has an input dimension of 32x32, while the second model has an input dimension of 41x41. For the reinforcement learning section, the input dimension is 41x41, with the strategy aiming to capture all the information provided by each embedding model and enhance the learning process.

Given s_t as the state at time step s_t (e.g., the current semantic representation of the protein r=TAPE(q)), a_t as the action (a includes the decision to classify a sequence as a signal peptide or not, and further classification into specific families) taken at time step t (modifications to the representation), r_t as the reward received after taking action a_t in state s_t., and $\pi(a|s)$ as the policy, which represents the probability of taking action a given state s, the goal is to maximise the expected cumulative reward $\mathcal{R}$:

$$\mathcal{R}(\pi) = \mathbb{E}\left[\sum_{t=0}^{T} \gamma^t r^t\right],$$

where T is the total number of time steps and $0 \leq \gamma \leq 1$ is the count factor. The expectation $\mathbb{E}_{\pi}$ indicates that the rewards are averaged over the probability distribution of states and actions as dictated by the policy π.

For optimising semantic representations of proteins using DNA-BERT, the reward function r_t should capture the desired attributes of the representation. The tailored reward function r_t is:

$$r_t = \alpha \cdot \text{Accuracy}(s_t) + \beta \cdot Plausibility(s_t) + \delta \cdot Quality(s_t),$$

where $Accuracy(s_t)$ measures performance on protein function prediction, $Plausibility(s_t)$ evaluates how well the representation aligns with known biological facts, on structural motifs (specific arrangements of AAs forming distinct three-dimensional structures), and $Quality(s_t)$ assesses the overall quality of the representation (e.g., coherence and robustness). $\alpha, \beta, and\ \delta$ are weights that balance the contributions of these components.

The agent's policy $\pi(a|s)$, which maps states to actions, is optimised using reinforcement Policy Gradients [31]. The objective is to find the policy that maximises the expected cumulative reward $\mathcal{R}(\pi)$:

$$\pi^* = arg\ max_\pi\ \mathcal{R}(\pi)$$

Family classification. For smoother classification for the families in *Arabidopsis*, we calculated the persistence diagram of dimension 1 for the *Arabidopsis* dataset (H_1), followed by computing the Wasserstein distance among all elements in the dataset [32, 33]. This process resulted in the classification of SSP candidates, enabling the clustering of different families and observation of their connections.

The *p*-Wasserstein distance between two persistence diagrams D_1 and D_2 is the infimum over all bijections $\gamma: D_1 \cup \Delta \rightarrow D_2 \cup \Delta$ of

$$\left(\sum_{x \in D_1 \cup \Delta} \|x - \gamma(x)\|_\infty^p \right)^{1/p}$$

Where $\| - \|_\infty$ is defined for $(x, y) \in \mathbb{R}^2$ by $\max\{|x|, |y|\}$.

Thus, if such a distance is null, this is akin to using BLAST but invariant to changes of scale [19]. Since protein sequences from different species may have undergone evolutionary changes in length, this facilitates the study of homologous proteins (i.e., *Arabidopsis*) and evolutionary relationships. Additionally, proteins often contain functional domains that are conserved in structure but may vary in size. Scale invariance helps in detecting these domains accurately, regardless of their length, leading to better functional annotation of proteins.

Training. We employed TAPE, ESM, and Geotop for embedding and feature extraction. Initially, diagrams from Geotop embeddings were selected and transformed [34], then concatenated with the original embeddings to form a 1024-dimensional feature set.

The dataset was divided into training (80%) and testing (20%) subsets. Each set was standardised separately, and the data was reshaped to suit a convolutional neural network (CNN), including an additional dimension for compatibility. This helps to evaluate the model on unseen data and identify and mitigate issues such as over-fitting, where the model becomes overly specialised to the training data and fails to generalise well.

We assessed training label distribution for class balance and verified preprocessing accuracy through sample visualisation. The CNN model, featuring several convolutional layers, average pooling layers, and dense layers, was trained with 18,433 to 116,737 parameters for *Solanum lycopersicum,* as an example among the chosen proteomes. Training involved 50 epochs, with loss and accuracy tracked for both training and validation sets using Stochastic Gradient descent (with momentum) SGD and Root Mean Squared Propagation RMSprop optimisers from Keras [35].

Evaluation and benchmarking. As a baseline for our evaluation, we curated a dataset comprising 18 sequences, focusing on those that are new or difficult to predict, with the assistance of domain experts. Using this dataset (i.e., 4 non SSPs and 14 SSPs) of 8 different proteomes and various SSPs families, we conducted testing with SignalP 6.0 and S^2-pepAnalyst (Fig. 2a).

Every step was meticulously taken to ensure that the training and validation datasets accurately represented the broader spectrum of plant proteomes, thus enhancing the generalisability of our model.

Following this initial evaluation, we further tested our model on two additional datasets comprising 779 and 1,177 sequences from 46 different species. The first dataset comprised 779 known SSPs, while the second contained 779 SSPs and 398 known non-SSPs [19]. These

tests further validated the superior performance of S^2-pepAnalyst in accurately predicting small signal peptides across a diverse range of species (Fig. 2b).

To evaluate the performance of the models, we primarily focused on classification accuracy. A probability threshold of 0.5 from the CNN's final output layer was used to distinguish SSPs from non-SSPs. In addition to accuracy, we utilised precision, recall, MCC-binary [36], and the F1 score as supplementary metrics (see Fig. 2c). We employed validation loss to implement early stopping, improve model generalisation, and optimise hyperparameters. Validation loss is calculated by feeding a subset of the total dataset (not used for training) through the trained model and computing the loss on this dataset. The binary cross-entropy loss function used is as follows:

$$L = -\frac{1}{N}\sum(y \times log(\hat{y}) + (1-y) \times log(1-\hat{y})),$$

where N represents the total number of samples in the validation dataset, y denotes the true label (i.e., $\wp(y|r)$), and $\hat{y}$ (i.e., $\wp(y|r_t)$) signifies the predicted probability of the positive class.

Hence, the total validation loss is the average of all sample losses:

$$Val_{loss} = \frac{1}{N}\sum(L).$$

Model complexity, scalability, and computational time cost. The computational complexity of our S^2-PepAnalyst model can be analysed by considering the number of parameters and operations in the CNN layers. The total number of parameters in a convolutional neural network is a significant factor influencing its computational requirements.

In each convolutional layer, the number of parameters is determined by the size of the filters and the number of filters. For a convolutional layer with k filters, each of size $f \times f$, applied to an input of size $m \times m$, the number of parameters is $k \times f^2$. If there are n such layers, the total number of parameters is given by $\text{Total parameters} = n \times k \times f^2$.

The overall complexity for the convolutional operations in one layer is:

$$\mathcal{O}(n \times m \times k \times f^2),$$

where n represents the number of sequences, m the average sequence length, k the number of convolutional filters, f filter size, and ℓ number of layers. Considering ℓ layers, the total complexity becomes:

$$\mathcal{O}(n \times m \times k \times f^2 \times \ell).$$

For example, considering the architecture of LeNet-5, which has a total of 22 filters (6 in the first convolutional layer and 16 in the second convolutional layer), and $f = 32, \text{and } \ell = 2$, the complexity for processing 10 sequences, each with a typically average length of 80, is:

$$\mathcal{O}(10 \times 80 \times 22 \times 32^2 \times 2) = \mathcal{O}(36{,}044{,}800).$$

During inference, the model's efficiency is maintained through optimised convolution operations, enabling rapid predictions even for larger datasets. Although the computational time cost is higher during the initial training phase due to extensive parameter tuning and hyperparameter optimisation, this is alleviated using GPU acceleration, which brings the time to practical levels for real-world applications. For instance, the inference time for a single sample, defined as the time taken to train a single batch to convergence and perform inference, took 0.63 seconds. Predicting and determining the family of peptide sequences predicted as SSP took 15.7 seconds (Supplementary Fig. 2).

Overall, S^2-PepAnalyst offers a balanced trade-off between accuracy and computational efficiency, making it suitable for high-throughput small signalling peptide prediction tasks.

Data availability

The sequence data used for training and testing S^2-pepAnalyst can be downloaded from https://github.com/MorillaLab/s2-PEPANALYST/data. The investigated reference proteomes are available from TAIR, Phytozome, avobase, and mangodb at https://www.arabidopsis.org/,

https://phytozome-next.jgi.doe.gov, https://www.avocado.uma.es/, and https://www.mangobase.org/.

Code availability

S^2-pepAnalyst is freely available as a web tool at http://www.s2-pepanalyst.uma.es. Users can upload peptide sequences for prediction and access detailed results through the user-friendly interface. The model source code in Keras 2.15, PyTorch 2.2.1, and Transformer 4.38.2 is available at https://github.com/MorillaLab/s2-PEPANALYST/.

References

20. Rhee, S.Y., Beavis, W., Berardini, T.Z., Chen, G., Dixon, D., Doyle, A., Garcia-Hernandez, M., Huala, E., Lander, G., Montoya, M., Miller, N., Mueller, L.A,, Mundodi, S., Reiser, L., Tacklind, J. Weems, D.C., Wu, Y., Xu, I., Yoo, D., Yoon, J., Zhang, P., The *Arabidopsis* Information Resource (TAIR): a model organism database providing a centralized, curated gateway to *Arabidopsis* biology, research materials and community , *Nucleic Acids Research* 31, 224–228 (2003).
21. Zhou, Y., Zhang, Z., Bao, Z., Li, H., Lyu, Y., Zan, Y., Wu, Y., Cheng, L., Fang, Y., Wu, K., Zhang, J., Lyu, H., Lin, T., Gao, Q., Saha, S., Mueller, L., Fei, Z., Städler, T., Xu, S., … Huang, S. Graph pangenome captures missing heritability and empowers tomato breeding. *Nature* 606, 527–534 (2022).
22. Solares, E., Morales-Cruz, A., Balderas, R.F., Focht, E., Ashworth, V.E.T.M., Wyant, S., Minio, A., Cantu, D., Arpaia, M.L., Gaut, B.S. Insights into the domestication of avocado and potential genetic contributors to heterodichogamy. *G3* (Bethesda). 13, jkac323 (2023).
23. Mango Genome Consortium, Bally, I.S.E., Bombarely, A., Chambers, A.H., Cohen, Y., Dillon, N.L., Innes, D.J., Islas-Osuna, M.A., Kuhn, D.N., Mueller, L.A., Ophir, R., Rambani, A., Sherman, A., Yan, H. The 'Tommy Atkins' mango genome reveals candidate genes for fruit quality. *BMC Plant Biology* 21, (2021).
24. Rao, R., Bhattacharya, N., Thomas, N., Duan, Y., Chen, X., Canny, J., Abbeel, P., Song, Y.S. Evaluating Protein Transfer Learning with TAPE. *Adv Neural Inf Process Syst*. 32, 9689-9701 (2019).

25. Rives, A. et al. Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences. bioRxiv (2020).
26. Martin, C. B., Simon, C., Clouchoux, C. & Histace, A. TDA-Clustering Strategies for the Characterization of Brain Organoids.. In J. S. H. Baxter, I. Rekik, R. Eagleson, L. Zhou, T. F. Syeda-Mahmood, H. Wang & M. Hajij (eds.), *EPIMI/ML-CDS@MICCAI*, Springer, 113-122, ISBN: 978-3-031-23223-7 (2022).
27. Munch, E. A User's Guide to Topological Data Analysis. *The Journal of Learning Analytics*. https://doi.org/10.18608/jla.2017.42.6 (2017).
28. Vidotto, A. "Random Lipschitz–Killing curvatures: Reduction Principles, Integration by Parts and Wiener chaos." *Theory of Probability and Mathematical Statistics* 0, 157-1175 (2021).
29. Cammarota, V., Marinucci, D., Rossi, M. Lipschitz-Killing Curvatures for Arithmetic Random Waves. arXiv (2020), arXiv:2010.14165v1.
30. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE*, 86, 2278-2324 (1998).
31. Ciosek, K., and Whiteson, S. Expected Policy Gradients for Reinforcement Learning. Journal of Machine Learning Research 21, 1-51 (2020).
32. Li, J. & Lin, L. Canonical Variates in Wasserstein Metric Space. CoRR, abs/2405.15768 (2024).
33. Taejong, J., Donggu, K., and Byunghoon, K. Regularizing activations in neural networks via distribution matching with the Wasserstein metric. International Conference on Learning Representations (2020).
34. Li, A., Pearl, J. Unit Selection with Causal Diagram. *AAAI*, 5765-5772. *AAAI Press.* ISBN: 978-1-57735-876-3 (2022).
35. https://keras.io/about/
36. Chicco, D., and Jurman, G. The Matthews correlation coefficient (MCC) should replace the ROC AUC as the standard metric for assessing binary classification. *BioData Min*. 16, 1 (2023).
37. https://pytorch.org/
38. https://huggingface.co/docs/transformers/en/index
39. https://liambai.com/protein-representation/

Acknowledgements

We gratefully acknowledge funding from the "Proyecto de Jóvenes Investigadores" (UMA B1-2022-29 to I Morilla), Consejería de Universidades, Ciencias y Desarrollo, fondos FEDER de la Junta de Andalucía (ProyExec_0499 to I Morilla), MINISTERIO DE CIENCIA E INNOVACIÓN (PID2020-113378RA-I00 to VG Doblas).

Author contributions

K.V-D. collected the datasets, implemented the S^2-pepAnalyst model and performed the experiments with help from V.G.D. and I.M. K.V-D. developed the reinforcement learning optimisation task. M.A. provided the GeoTop code and guided its application with help from I.M. A.M.L. and G.G. provided suggestions during the design of S^2-pepAnalyst. G.G. supervised the application of TDA analysis with help from I.M. V.G.D. and I.M. supervised and guided the project. All authors edited and approved the manuscript.

Competing interests

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Figure legends

Figure 1. Architecture of S^2-PepAnalyst learning proteins as images and refining LM semantic by RL. (a) Schematic of the encoding process for generating optimal semantic protein representations [34] using Reinforcement Learning (RL). Each peptide sequence is encoded, averaged, and transformed into a high-dimensional vector [39]. **(b)** Scatter plot comparing encoded SSPs with the Arabidopsis database, highlighting distinct clustering patterns that indicate similarities and differences in the protein representations. Non-SSPs are highlighted in blue and SSP predictions in yellow. **(c)** Workflow diagram illustrating the progression from peptide sequence input to machine learning-based prediction. The process involves encoding the sequence with a prior pre-training using its cleavage site, generating an optimal semantic representation, image learning, and applying convolutional neural networks (CNNs) and computer vision GeoTop algorithm for final prediction and family classification: the upper figure depicts post-translationally modified peptides, showing possible hydroxylation and

glycosylation of Proline and sulphation of Tyrosine. The lower figure shows cysteine-rich peptides with di-sulphide bridges forming between cysteine residues.

Figure 2. S^2-PepAnalyst strongly predicts and classifies SSPs in plants. (a) Baseline and generalised comparative analysis of S^2-PepAnalyst and SignalP 6.0 for SSP prediction accuracy. S^2-PepAnalyst demonstrates higher sensitivity and specificity across various plant proteomes, **(b)** The top classification matrix corresponds to SignalP 6.0, while the middle and bottom matrices represent S^2-PepAnalyst's performance on datasets comprising 18 and 1,177 sequences, respectively. S^2-PepAnalyst exhibits a lower false-negative rate, indicating enhanced predictive accuracy. **(c)** Comparative analysis of MCC, accuracy, precision, and F1 score between S^2-PepAnalyst and SignalP 6.0. S^2-PepAnalyst consistently achieves higher metrics, with precision approaching 99%, underscoring its robustness and reliability.

Figure S1. Training and validation accuracy on average across epochs and proteomes. This plot illustrates the accuracy of the S^2-PepAnalyst model during training and validation phases over 30 epochs and the 5 analysed proteomes. The blue line represents the accuracy on the training dataset, while the orange line indicates the accuracy on the validation dataset. The model shows a steady increase in accuracy, achieving over 98.1% accuracy for both training and validation data, demonstrating robust performance and effective learning.

Figure S2. Memory usage during model execution. This graph displays the memory consumption of the S^2-PepAnalyst model over time, measured in MiB. The analysis reveals a stable memory usage pattern with a peak around 9 MiB, indicating the model's efficiency and scalability. The consistent memory usage throughout the process highlights the model's suitability for real-world applications with limited computational resources.

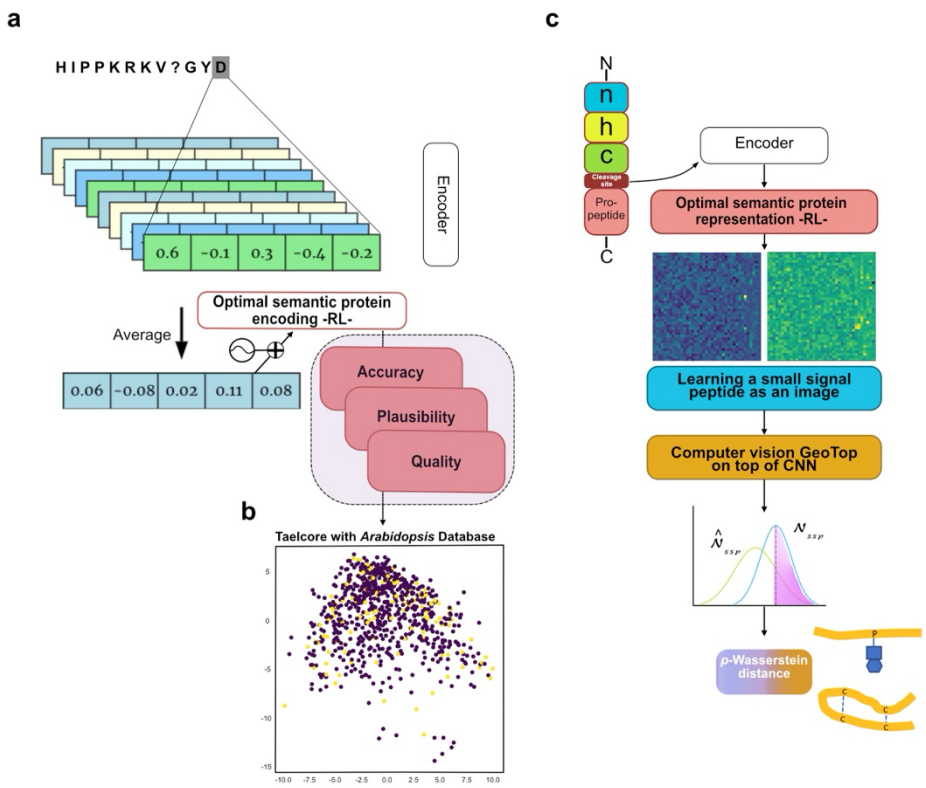

Figure 1. Architecture of S^2-PepAnalyst learning proteins as images and refining LM semantic by RL. (a) Schematic of the encoding process for generating optimal semantic protein representations [34] using Reinforcement Learning (RL). Each peptide sequence is encoded, averaged, and transformed into a high-dimensional vector [39]. **(b)** Scatter plot comparing encoded SSPs with the Arabidopsis database, highlighting distinct clustering patterns that indicate similarities and differences in the protein representations. Non-SSPs are highlighted in blue and SSP predictions in yellow. **(c)** Workflow diagram illustrating the progression from peptide sequence input to machine learning-based prediction. The process involves encoding the sequence with a prior pre-training using its cleavage site, generating an optimal semantic representation, image learning, and applying convolutional neural networks (CNNs) and computer vision GeoTop algorithm for final prediction and family classification: the upper figure depicts post-translationally modified peptides, showing possible hydroxylation and glycosylation of Proline and sulphation of Tyrosine. The lower figure shows cysteine-rich peptides with di-sulphide bridges forming between cysteine residues.

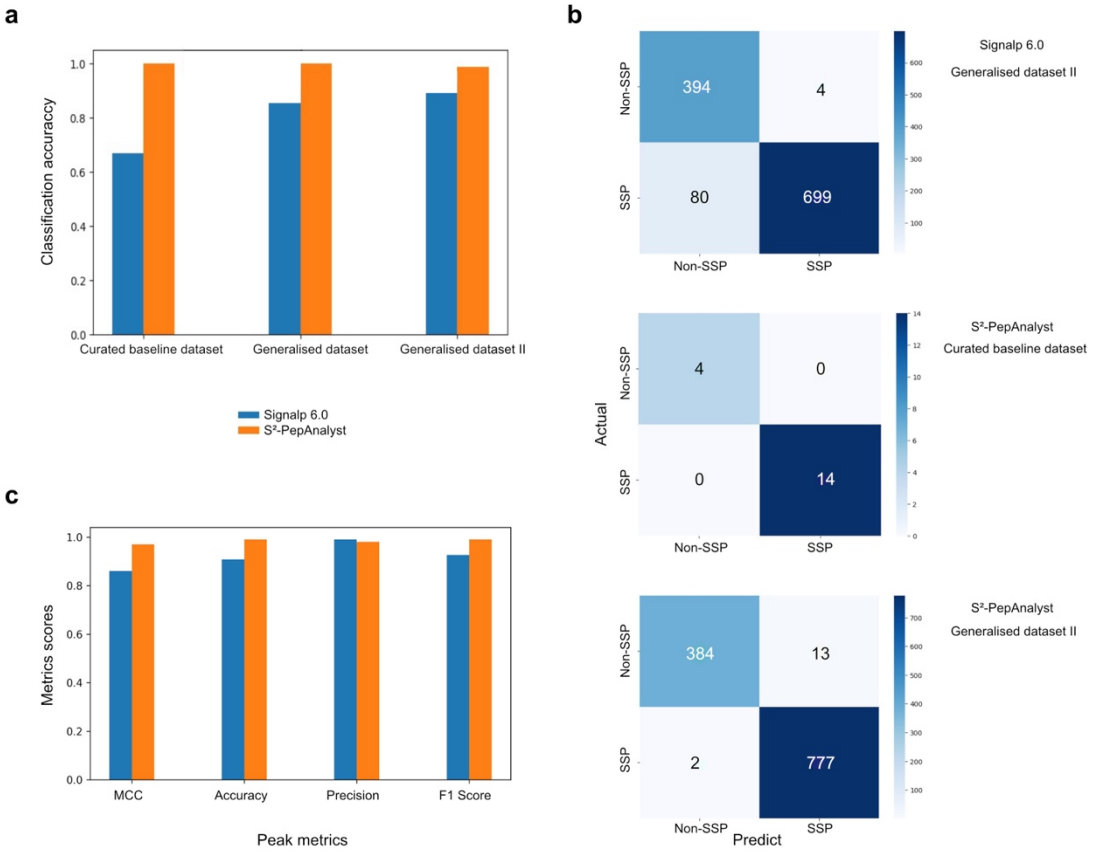

Figure 2. S^2-PepAnalyst strongly predicts and classifies SSPs in plants. (a) Baseline and generalised comparative analysis of S^2-PepAnalyst and SignalP 6.0 for SSP prediction accuracy. S^2-PepAnalyst demonstrates higher sensitivity and specificity across various plant proteomes, **(b)** The top classification matrix corresponds to SignalP 6.0, while the middle and bottom matrices represent S^2-PepAnalyst's performance on datasets comprising 18 and 1,177 sequences, respectively. S^2-PepAnalyst exhibits a lower false-negative rate, indicating enhanced predictive accuracy. **(c)** Comparative analysis of MCC, accuracy, precision, and F1 score between S^2-PepAnalyst and SignalP 6.0. S^2-PepAnalyst consistently achieves higher metrics, with precision approaching 99%, underscoring its robustness and reliability.

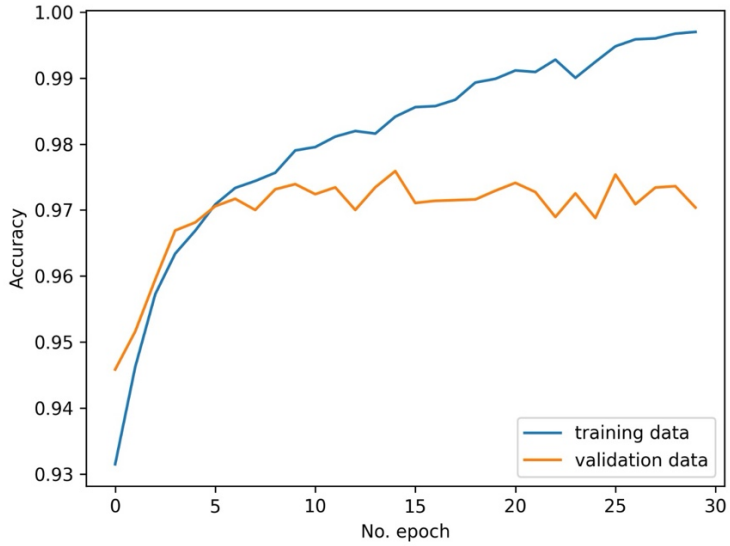

Figure S1. Training and validation accuracy on average across epochs and proteomes. This plot illustrates the accuracy of the S^2-PepAnalyst model during training and validation phases over 30 epochs and the 5 analysed proteomes. The blue line represents the accuracy on the training dataset, while the orange line indicates the accuracy on the validation dataset. The model shows a steady increase in accuracy, achieving over 98.1% accuracy for both training and validation data, demonstrating robust performance and effective learning.

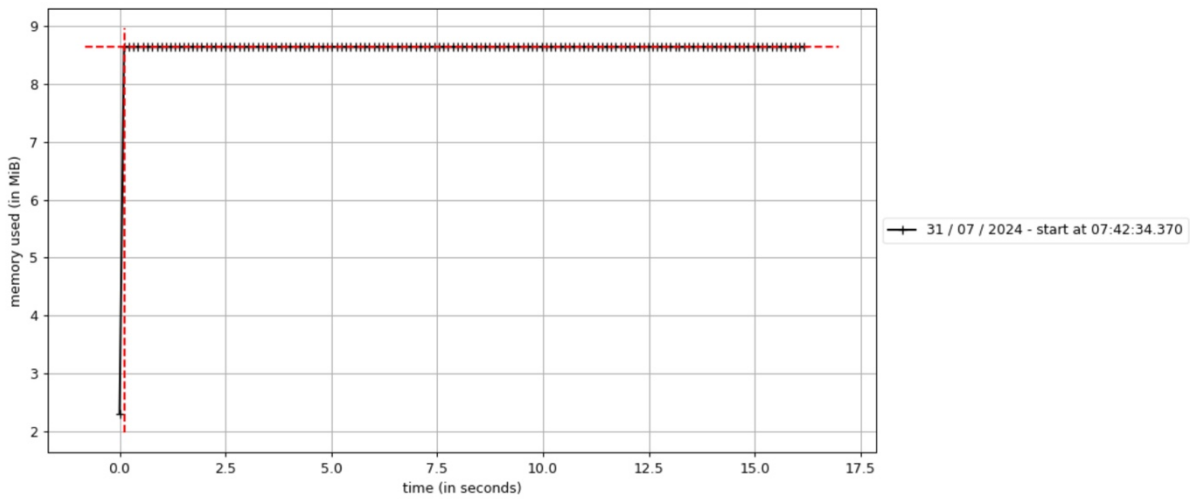

Figure S2. Memory usage during model execution. This graph displays the memory consumption of the S^2-PepAnalyst model over time, measured in MiB. The analysis reveals a stable memory usage pattern with a peak around 9 MiB, indicating the model's efficiency and scalability. The consistent memory usage throughout the process highlights the model's suitability for real-world applications with limited computational resources.