nature machine intelligence

Article

https://doi.org/10.1038/s42256-024-00928-1

Reshaping the discovery of self-assembling peptides with generative AI guided by hybrid deep learning

Received: 25 March 2024

Accepted: 18 October 2024

Published online: 19 November 2024

Check for updates

A list of authors and their affiliations appears at the end of the paper

Supramolecular peptide-based materials have great potential for revolutionizing fields like nanotechnology and medicine. However, deciphering the intricate sequence-to-assembly pathway, essential for their real-life applications, remains a challenging endeavour. Their discovery relies primarily on empirical approaches that require substantial financial resources, impeding their disruptive potential. Consequently, despite the multitude of characterized self-assembling peptides and their demonstrated advantages, only a few peptide materials have found their way to the market. Machine learning trained on experimentally verified data presents a promising tool for quickly identifying sequences with a high propensity to self-assemble, thereby focusing resource expenditures on the most promising candidates. Here we introduce a framework that implements an accurate classifier in a metaheuristic-based generative model to navigate the search through the peptide sequence space of challenging size. For this purpose, we trained five recurrent neural networks among which the hybrid model that uses sequential information on aggregation propensity and specific physicochemical properties achieved a superior performance with 81.9% accuracy and 0.865 F1 score. Molecular dynamics simulations and experimental validation have confirmed the generative model to be 80–95% accurate in the discovery of self-assembling peptides, outperforming the current state-of-the-art models. The proposed modular framework efficiently complements human intuition in the exploration of self-assembling peptides and presents an important step in the development of intelligent laboratories for accelerated material discovery.

Molecular self-assembly (SA) driven by weak, non-covalent interactions represents one of the fundamental chemical processes observed in living organisms[1,2]. Peptides, composed of 20 gene-encoded amino acids, serve as a versatile toolbox for obtaining supramolecular materials with rich chemical and structural properties[3]. It is no surprise that the organization of peptidic building blocks into three-dimensional structures gives rise to materials that exhibit remarkable complexities and emerging functionalities, including catalysis and molecular recognition[4–6]. As a result, the rational and computational design of peptide-based materials, accompanied by extensive experimental validations, has led to the establishment of relevant and applicable design principles, enabling substantial progress in the development of supramolecular materials for a wide range of applications[3,7–9]. However, understanding of the sequence-to-structure and function relationships of peptides still remains beyond our comprehension.

Although expensive and time-consuming, the experimental discovery of new self-assembling peptides remains the prevailing approach[10]. However, this nondeterministic polynomial time (NP)-hard

e-mail: daniela.kalafatovic@uniri.hr; goran.mausa@uniri.hr

https://doi.org/10.1038/s42256-024-00928-1

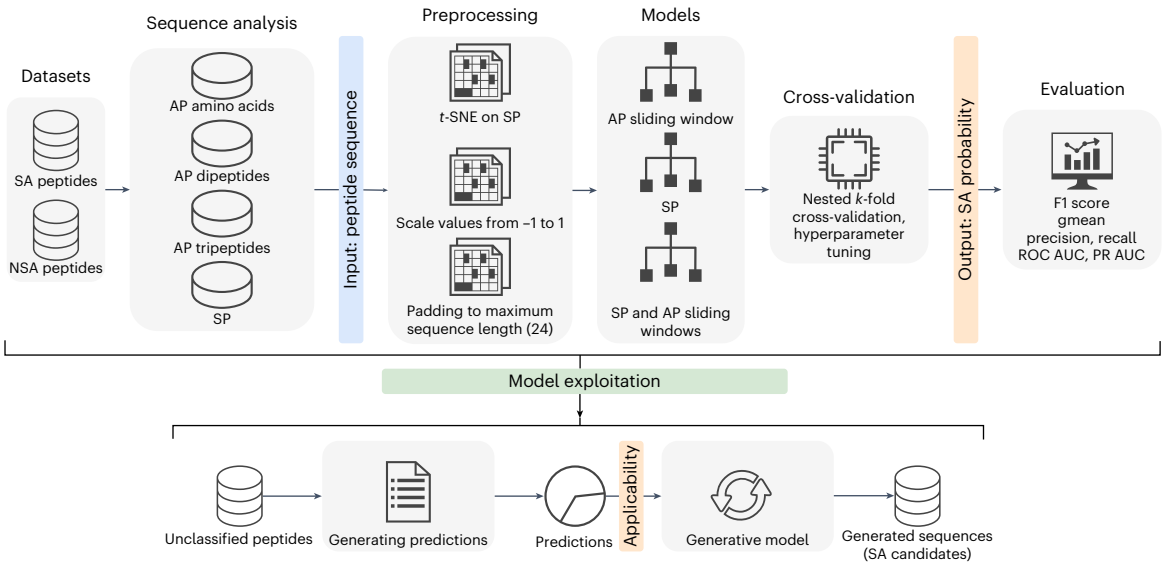


Fig. 1 | Overview of the proposed research pipeline. The models are based on heterogeneous data obtained by applying AP sliding windows of varying lengths (amino acids, dipeptides and tripeptides) in combination with SP. Data preprocessing included (1) *t*-SNE for dimensionality reduction, (2) scaling of AP and SP values to a range of [−1, 1] to facilitate gradient flow and (3) padding shorter sequences to a maximum length of 24 residues to expedite training. The models were optimized and evaluated using the nested fivefold cross-validation sampling technique before yielding the final model ready for exploitation. With the aim of discovering sequences with high SA propensity, the hybrid AP–SP model was used as a guideline in a genetic-algorithm-based generative model.

discovery process advocates alternative methods for the efficient navigation of the chemical space that grows exponentially with each amino acid added to the sequence[11]. Furthermore, laboratory investigations of supramolecular peptide nanostructures, including synthesis, purification and characterization, can last for weeks and require highly skilled experts and sophisticated instrumentation[12–17]. On the contrary, the intractability of an exhaustive examination of the entire search space when considering peptides longer than three amino acids[10] and the sparseness of useful molecules in such spaces[18] led to design rules for peptide SA, namely, patterning strategies manipulating hydrophobic–hydrophilic balance[19] and molecular templating[20]. Such procedures tend to constrain peptide design and reduce the number of peptides that reside in the available search space to a more manageable level. Nevertheless, they introduce an unwanted bias towards specific regions of the search space, thereby disregarding potentially promising areas and limiting the discovery of novel sequences.

Molecular dynamics (MD) simulations contributed to considerably reducing the time required to characterize individual compounds and increased the available screening throughput[21–25]. However, they tend to introduce errors into the results, the extent of which partially depends on the simulation setup details (for example, coarse-grained (CG) MD[26], simulation time and system size[27]). Nevertheless, MD has been successfully applied to the estimation of aggregation propensity (AP) as a precursor stage of peptide SA for dipeptides[27] and tripeptides[3]; however, extending this approach to tetrapeptides and beyond remains challenging due to high computational costs[10,28,29].

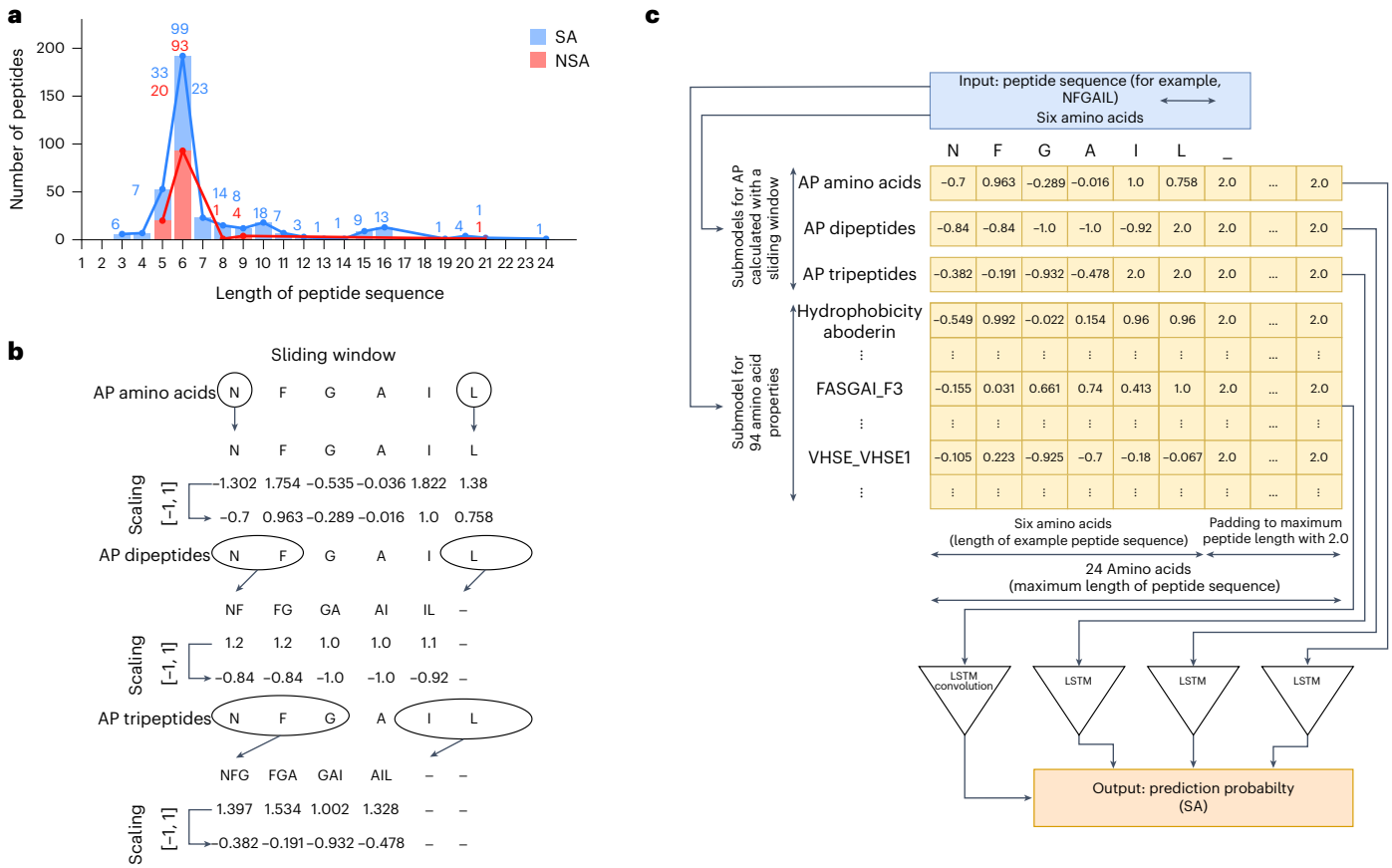
Machine learning (ML) has emerged as an efficient alternative to MD for the in silico screening and optimization of therapeutic peptides[30–41]. ML algorithms run faster than MD simulations and can achieve acceptable accuracy with large datasets and the right choice of architecture[10,42,43]. However, they are mostly unexplored for the prediction of peptide SA propensity due to the scarcity and imbalance of data[44]. Recently, ML-based sequence preselections based on support vector machine and random forest (RF) models were applied to guide the search for self-assembling and hydrogel-forming peptides, demonstrating advantages over human experts[10,45]. This approach produced higher average AP scores compared with an exhaustive search, and its scalability made it applicable to search spaces comprising longer sequences, including octapeptides and proteins[46]. Although a shift towards using artificial intelligence (AI) models for rapid screening is evident[47,48], many studies still rely on MD for the final AP score calculations. As we progress towards exploring peptide spaces of increasing size, MD simulations ought to be complemented with ML to facilitate fast and unbiased exploration of the peptide space[49,50]. In a recent study, a least absolute shrinkage and selection operator regression model trained with 163 sequences was used to search for de novo self-assembling peptides for viral gene delivery[51]. A subset of 16 promising hexapeptides showed a low accuracy of 25% when tested experimentally, which was improved to 50% after filtering the subset according to AP. Therefore, better models and more sophisticated representation schemes are needed to overcome the issue of complex multiparameter and multiscale dependencies therein.

Recurrent neural networks (RNNs) are particularly effective in processing data sequences due to their ability to capture sequential dependencies, distinguishing them from other ML models and network architectures[52–55]. Consequently, they have found widespread application in sequence-to-function inference, such as sentiment analysis[56,57]—a task highly similar to the prediction of peptide activity. Therefore, RNNs can exhibit performance superior to those of ML techniques such as RF, support vector machine and non-RNNs to assess SA propensity by modelling context-aware sequential relationships between peptide constituents[58,59]. In this paper, we introduce an RNN-based approach to assess the SA potential of unclassified peptides using irregularly sampled features of unequal length, based on AP scores of amino acids, dipeptides and tripeptides[3,27,60] as predictor variables for any given peptide of interest (Fig. 1). Furthermore, the RNN classifier is used as a fitness function in a search-based genetic algorithm to form a generative model for the discovery of sequences with a high propensity towards SA. This model complements human intuition in an attempt to identify new peptides with high SA propensity, based on unbiased sequence-space exploration aided by ML.

Results and discussion

The task of predicting the SA propensity in peptides of arbitrary length was approached by a supervised learning classifier with sequence-based input and RNN architecture (Fig. 1). RNNs are deep learning algorithms

https://doi.org/10.1038/s42256-024-00928-1



e

Value	AP	SP	AP-SP	*t*-SNE SP	*t*-SNE AP–SP
Number of cells	–	64 (13/25)	32 (12/25)	48 (10/25)	64 (10/25)
Kernel size	–	4 (10/25)	4 (13/25)	6 (12/25)	6 (11/25)
Dense	64(12/25)	–	64 (12/25)	–	128 (10/25)

f

Value	AP	SP	AP-SP	*t*-SNE SP	*t*-SNE AP-SP
Max acc	77.11%	93.21%	93.88%	73.32%	84.01%
Min loss	0.623	0.258	0.273	0.644	0.494
Avg acc	71.71% (4.33%)	85.01% (7.63%)	85.21% (7.57%)	67.66% (5.54%)	76.74% (5.90%)
Avg loss	0.713 (0.073)	0.451 (0.155)	0.465 (0.149)	0.723 (0.068)	0.609 (0.104)

Fig. 2 | Neural network setup from dataset to sliding window mechanism and hyperparameter optimization. a, Distribution of peptide lengths within the dataset with the indicated numbers of SA (blue) and NSA (red) instances. **b**, Schematic of the sliding window preprocessing procedure identifying individual amino acids, dipeptides and tripeptides within the sequence. **c**, Structure of the input data for an example sequence NFGAIL and a hybrid AP–SP RNN model that combines 94 SP and 3 AP values. **d**, Model construction workflow diagram. **e**, Most commonly selected hyperparameter values, along with the number of occurrences: 'Num cells' in a bidirectional LSTM layer, 'kernel size' used in the convolutional layer and 'dense' that presents the number of units in the final densely connected layer of the model. **f**, Maximum accuracy (Max Acc), minimal loss (Min Loss), average accuracy (Avg Acc) and average loss (Avg Loss) during training, along with standard deviations.

often used for sequential or temporal problems, such as language translation and processing, in Apple Siri[61] and Google Translate[62]. Their architecture facilitates memorizing previously received inputs, which is an important advantage, meaning that the output values they produce are influenced by information acquired in previous time steps[63]. To date, ML-based approaches have faced limitations due to the insufficient size and quality of available datasets[28]. To overcome this challenge, we manually curated a dataset comprising 368 peptides with experimentally validated assembly status (Supplementary Data 1), labelled as self-assembling (SA; 249 peptides) and non-assembling (NSA; 119 peptides).

Dipeptides and tripeptides, such as FF, YY, WW and FFF, have been widely used as basic building blocks in peptide nanotechnology to generate highly organized supramolecular materials with diverse architectures[3,64,65]. Among them, FF is the most studied one[2] and, alongside VFF, constitutes the key motif found in β-amyloid proteins associated with Alzheimer's disease[66]. The SA process and the resulting morphology of supramolecular assemblies are influenced by amino acid composition, physicochemical properties, position of specific amino acids within the sequence and their neighbouring residues[2,67,68]. Although distinct sequence patterns exist, accurately predicting the SA propensity solely based on an amino acid sequence remains challenging[51]. For this reason, the AP scores of amino acids, dipeptides and tripeptides were incorporated as crucial input values in our models, which aim to predict the SA propensity of longer sequences based on the existing knowledge of minimalistic ones. To enhance the predictive

power of the models, we combined the AP scores with the sequential properties (SP) representation scheme[69], allowing the computation of heterogeneous data and therefore enriching the peptide feature space.

The RNN models were trained by leveraging three peptide feature categories based on (1) AP scores, (2) SP values and (3) hybrid AP–SP. Previous studies have demonstrated the value of using AP scores as predictors of SA[10,46,66]. However, they faced limitations in efficiently predicting the SA propensity of longer sequences based solely on AP scores. To overcome this, we propose a sliding window approach to extract the AP scores of contiguous amino acids, dipeptides and tripeptides within the original sequence and use their combination to boost information gain and improve the predictive performance for longer sequences. In addition, the SP representation scheme, which proved effective for therapeutic peptides[69], was used to capture the sequential and physicochemical characteristics of peptides.

Building and fine-tuning the neural network models

Data preprocessing to obtain structured datasets preceded model training and included the scaling of AP and SP values to a range [−1, 1] to facilitate gradient flow and improve the prediction performance[70,71]. Furthermore, to accelerate training, all the peptides were padded to the maximum peptide length (within the dataset) of 24 residues, allowing the sequences to be processed in batches. An arbitrary value of 2, outside the [−1, 1] scaling range, was used as the padding value for peptide descriptors. The identification and property calculation of minimal sequence-building motifs was carried out using sliding windows of variable lengths (Fig. 2b). The way the input data is processed within the SP (Supplementary Fig. 1), AP (Supplementary Fig. 2) and hybrid AP–SP (Fig. 2c) models demonstrates their increasing complexity in terms of the required sliding windows and long short-term memory (LSTM) layers. LSTM layers were chosen due to the profound effectiveness of LSTM cells in addressing the vanishing gradient problem, ensuring efficient training, and improving performance when dealing with extensive sequentially dependent data[72].

As the number of SP descriptors is much larger than that of AP (94 compared with 3), we hypothesized that such a ratio contributes to a diminished influence of AP values on SA prediction in the hybrid AP–SP model. Therefore, to bring the number of AP and SP features to a common scale, we applied the *t*-distributed stochastic neighbour embedding (*t*-SNE)[73] dimensionality reduction technique and extracted 3 meta-features from the 94 physicochemical properties for each amino acid. *t*-SNE was used due to its established use in peptide research to generate meta-features and facilitate data visualization[74,75]. In total, five models with three main architectures (Fig. 3) were developed: (1) AP model; (2) SP model; (3) hybrid AP–SP model; and two models with *t*-SNE preprocessing of SP properties, namely, (4) *t*-SNE SP and (5) *t*-SNE AP–SP.

During training, stratified fivefold cross-validation was used to avoid overfitting and yield an unbiased performance estimate of our models (Methods), whereas the original SA–NSA class ratio of approximately 2:1 in the dataset (Fig. 2a) was maintained when generating folds. A grid search procedure was used as part of nested fivefold cross-validation to optimize the hyperparameters (Fig. 2d and Supplementary Table 1) by selecting the values that resulted in the best performance (Fig. 2e). The 368 peptides within the dataset were allocated to different subsets, leaving 236–237 peptides for training the model, 58–59 for validation and 73–74 for testing in each run. Training and validation loss along with the respective accuracy scores were monitored during hyperparameter optimization (Fig. 2f and Supplementary Fig. 3). A similar level of loss between training and validation, together with a clear trend of improvement, indicates that overfitting is successfully avoided. The hybrid AP–SP RNN model achieved the highest average accuracy of 85.21% and the second-lowest average loss of 0.465 during training. It is closely followed by the SP model with the second-highest average accuracy of 85.01% and the lowest average loss of 0.451.

Testing and benchmarking the models

The performance of the proposed models in terms of their prediction probability distribution in all the test folds (Fig. 3d–h) shows a clear overlap of the SA and NSA groups (Fig. 3d), indicating the inability of the AP model to effectively discriminate between positive and negative classes. However, the introduction of SP features into the models led to a greater number of true-positive and true-negative cases, visually manifested as a widened gap between the SA and NSA classes (Fig. 3e,f). This resulted in increased accuracy when the models were applied to sequences not seen during training.

For a comprehensive assessment of the neural networks' performance, the following set of evaluation metrics was used: precision–recall (PR) and receiver operating characteristic (ROC) curves, the corresponding area under the curves (PR AUC and ROC AUC), classification accuracy, F1 score, and geometric mean (gmean) of true-negative rates and true-positive rates. The sigmoid output of the models was assigned to a binary class using the optimal classification thresholds determined during the hyperparameter optimization (Supplementary Table 2). Along with the standard classification threshold of 0.5, thresholds were also estimated from the ROC and PR curves using the shortest distance to the ideal performance point (100% true-positive rate and 0% false-positive rate for ROC, and 100% precision and recall for PR) (Supplementary Figs. 4–8). Although certain improvements in gmean can be achieved using ROC or 0.5 thresholds, PR thresholds produce the highest F1 values for all the models (Table 1). Considering that F1 successfully addresses the class imbalance and provides an unbiased estimate of the predictive power of the models, we applied the PR thresholds to the remaining analyses.

The AP–SP model achieved the best performance according to every evaluation setup, reaching an accuracy of 81.9% and an F1 score of 0.865 (Table 1). This indicated that the model successfully differentiates between the SA and NSA classes. It is closely followed by the SP model with an accuracy of 80.4% and an F1 score of 0.856, whereas the AP model had a diminished performance compared with both SP and AP–SP (Table 1), which is consistent with the predictive power of the models (Fig. 3d–f). Reducing the number of SP features by *t*-SNE to match the number of AP descriptors failed to improve performance. However, the hybrid *t*-SNE AP–SP still attained a better result than the AP model, leading to the conclusion that the prediction of SA benefits from the synergistic effect of heterogeneous features. This is further supported by the statistically significant difference found between the classification output of the SP and hybrid AP–SP models ($P = 0.043$, $n = 1{,}840$ peptides; McNemar's two-sided test). These results are closely aligned with the performance achieved during hyperparameter optimization and training (Fig. 2f).

The proposed models were benchmarked with RF, as a simpler representative of ML models that achieved excellent performance in related studies[10,76,77], along with more complex neural networks of related architectures such as RNN[78], LSTM[79], Bi-LSTM[80], multilayer perceptron (MLP)[81] and Transformer[82] taken from another work[83]. All neural network models sequentially process the amino acids as a series of tokens. Although benchmarking models used the phrase embedding strategy for obtaining a numerical representation of tokens, our models used AP and SP values for embedding the amino acids, and AP values for dipeptides and tripeptides. Finally, RF used the SP-equivalent encoding strategy, but the properties were calculated for the whole peptide due to its inability to process sequential type of data.

When tested on the aggregated set of peptides, the Transformer achieved the best performance in terms of gmean (0.819) and accuracy (83.7%), and the same level of F1 score (0.878) as the LSTM architecture (Extended Data Table 1). Although their performance is greater than of the AP–SP, McNemar's two-sided statistical tests ($\alpha = 0.05$, $n = 1{,}840$) showed that their classification outputs are not significantly different. A significant difference was only observed for RNN and MLP ($P < 0.05$), which were outperformed by AP–SP (Extended Data Table 1).

https://doi.org/10.1038/s42256-024-00928-1

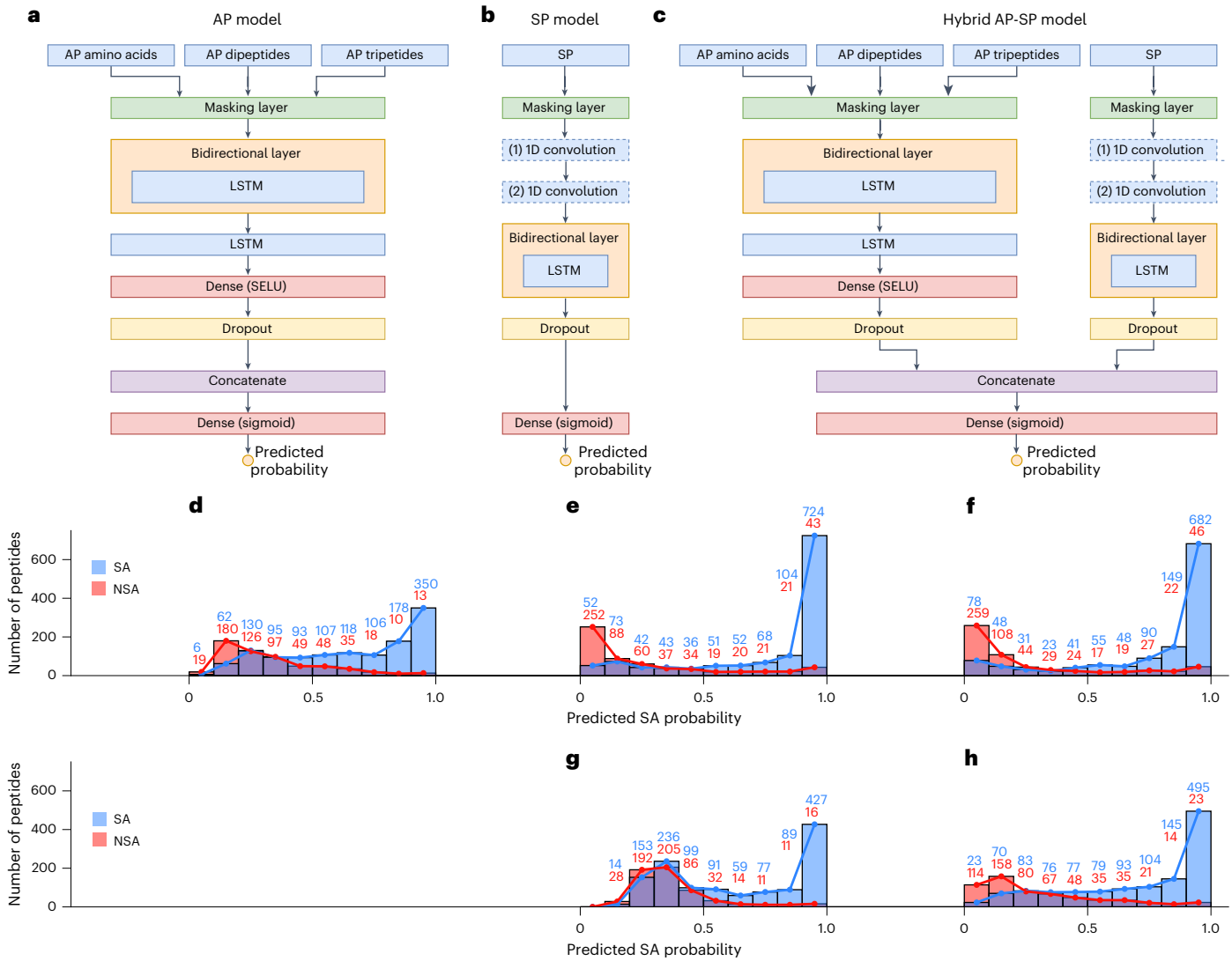


Fig. 3 | RNN architectures and the respective performance assessments. **a**–**c**, Schematic of the architectures, depicting differences in input layers and configuration of LSTM or convolutional layers, for AP (**a**), SP (**b**) and hybrid AP–SP (**c**) models. SELU, scaled exponential linear unit; 1D, one-dimensional. **d**–**h**, Histograms of the prediction probability distribution on the aggregated test folds for the AP (**d**), SP (**e**), hybrid AP–SP (**f**), *t*-SNE SP (**g**) and *t*-SNE AP–SP (**h**) models demonstrate the ability of each model to discriminate between the two classes of peptides: SA (blue) and NSA (red).

Benchmarking of the models was further investigated in a real-world use-case scenario of 11 pentapeptides proposed by human experts and 9 predicted by AI expert based on RF[10], with an experimentally verified SA status. Considering that human and AI experts predicted only the positive class, their true-negative rate and gmean values were equal to 0 (Extended Data Table 1). Furthermore, the 20 pentapeptides exhibited a 60%:40% ratio skewed towards the assembling sequences (Supplementary Fig. 13a), making the F1 score the best option for ranking the models. All models trained on the curated dataset, except *t*-SNE SP, performed better than human experts (Supplementary Table 3). Among the benchmarked models (Extended Data Table 1), AP–SP is the best performing one (gmean, 0.928; F1 score, 0.930; accuracy, 92%), followed by LSTM (gmean, 0.841; F1 score, 0.861; accuracy, 84%). Furthermore, the AP–SP and SP models (F1 score, 0.942; accuracy, 93%) performed similarly (Supplementary Table 3), both outperforming the AI expert (F1 score, 0.800; accuracy, 67%). With standard deviation values below 2% for the full dataset and below 4% for 20 pentapeptides across all the evaluation metrics, this suggests that the AP–SP model is highly stable and robust, as well as capable of effectively generalizing information. Stable performance was observed for pentapeptides and hexapeptides, which constitute the majority of the dataset (Fig. 2a), as well as for sequences composed of 8, 9, 3 or 16 residues, which had fewer samples for training (Supplementary Table 4). An additional analysis of model exploitation for over 6,000 pentapeptides processed by MD[10] showed that reducing the number of features and retaining the most informative ones can help models in generalizing their knowledge more successfully on unseen data (Supplementary Section 2).

On the basis of this, we envision that our models will complement rather than compete with human intuition in the discovery of new knowledge about peptide SA by facilitating a guided search of the sequence space, providing a platform specialized in offering educated guesses to the peptide SA problem.

Model applicability in unexplored sequence space

The effectiveness of the AP–SP model becomes more apparent when used as a knowledgeable tool aimed at discovering sequences with a high SA propensity. In this context, the model served as a guideline in a genetic-algorithm-based generative approach[84]. The algorithm was conditioned to generate hexapeptides (Fig. 4a) as well as 5–10 residue

https://doi.org/10.1038/s42256-024-00928-1

Table 1 | Performance of the proposed RNN models

Metric	AP	SP	AP–SP	*t*-SNE SP	*t*-SNE AP–SP
ROC AUC	0.817 (0.004)	**0.864 (0.006)**	**0.862 (0.017)**	0.772 (0.010)	0.839 (0.014)
gmean (ROC threshold)	0.742 (0.005)	**0.796 (0.009)**	**0.798 (0.018)**	0.702 (0.008)	0.759 (0.013)
F1 (ROC threshold)	0.772 (0.006)	0.833 (0.009)	**0.841 (0.008)**	0.688 (0.010)	0.784 (0.011)
Acc (ROC threshold)	72.5% (0.6%)	79.0% (1.0%)	**79.7% (1.1%)**	66.2% (0.9%)	73.9% (1.2%)
PR AUC	0.896 (0.005)	**0.919 (0.003)**	**0.920** (0.009)	0.890 (0.004)	0.911 (0.010)
gmean (PR threshold)	0.639 (0.013)	0.768 (0.010)	**0.794 (0.016)**	0.582 (0.021)	0.728 (0.013)
F1 (PR threshold)	0.824 (0.006)	0.856 (0.002)	**0.865 (0.008)**	0.796 (0.005)	0.830 (0.008)
Acc (PR threshold)	74.6% (0.8%)	80.4% (0.4%)	**81.9% (1.1%)**	70.4% (0.8%)	77.0% (1.0%)
gmean (0.5)	0.739 (0.005)	**0.797 (0.012)**	**0.801 (0.017)**	0.716 (0.007)	0.760 (0.017)
F1 (0.5)	0.771 (0.006)	0.844 (0.010)	**0.853 (0.006)**	0.717 (0.008)	0.800 (0.014)
Acc (0.5)	72.3% (0.6%)	79.9% (1.2%)	**80.9% (0.9%)**	68.2% (0.8%)	75.2% (1.6%)

The average values and the standard deviation (in brackets) are calculated over an aggregated set of test folds with five different seeds in terms of AUC; the geometric mean of true-positive rate and true-negative rate; F1 score; and classification accuracy for the ROC, PR and standard 0.5 thresholds. The best scores per metric, when rounded to two decimal places, are marked in bold. McNemar's two-sided statistical test showed a statistically significant difference between the classification probability of SP and AP–SP models (P=0.043, n=1,840). The test was conducted using predictions from the five-times-repeated testing phase, which resulted in five associated predictions for each of the 368 peptides in the dataset.

peptides (Fig. 4b) because 85% of the sequences in our dataset fall within this range, with the most prevalent length being six (Fig. 2a). When conditioned to generate sequences 5–10 residues long, the generative model showed a strong tendency towards decapeptides (Fig. 4b and Supplementary Table 5).

To stress the importance of searching through the unexplored regions of the peptide chemical space, we visualized the resulting sequences as plot points in two-dimensional space based on their mutual similarities calculated by Needleman–Wunsch global sequence alignment[85]. The distances among the generated hexapeptides (Fig. 4a, blue points) and among the peptides 5–10 residues long (Fig. 4b, green points) were optimized by stochastic gradient descent to reflect their similarity to each other (Sim_{gen}). Having all the sequences contained in the training set represented as a single point (Fig. 4a,b, red), the average similarity between the generated peptides and the training data (Sim_{train}) was used as a more important indicator of similarity and intentionally maintained at a lower visualization error (below 6%) in this plot.

The top five generated peptides with the highest SA probability (Fig. 4a,b) show low Sim_{train} but 40% or higher Sim_{gen}, which derives from the tendency of the generative model to converge to a single best peptide. This is a general characteristic of the single-criterion optimization procedures, supported by a selection pressure that ensures a rise in SA propensity as the algorithm progresses. Although a single experiment yields a homogeneous population, by studying the motifs and sequences across distinct experiments (Supplementary Table 7), we can conclude that the stochastic component of the algorithm ensures convergence to different peptides each time, offering diverse solutions for researchers to choose from. Furthermore, algorithm convergence resulted in the emergence of specific motifs during evolution (Supplementary Section 3).

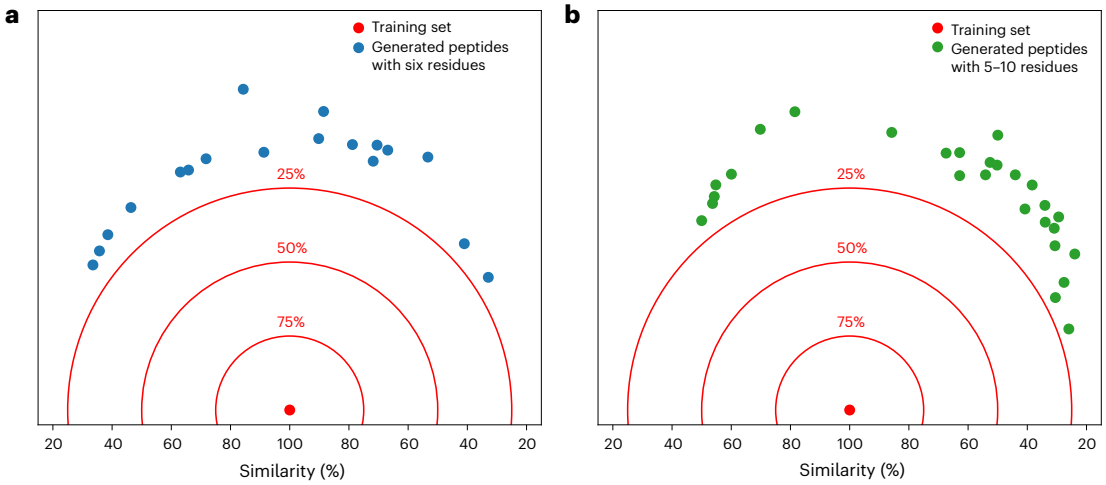
The generated peptides (Fig. 4a,b) were validated using CG-MD simulations. Their AP was assessed through changes in solvent-accessible surface area (SASA) by calculating the AP_{SASA} score following 200 ns simulations. AP scores, as well as visual inspection of initial (t = 0 ns) and final (t = 200 ns) simulation frames, confirmed that all the generated hexapeptides and decapeptides show a strong tendency towards aggregation (Fig. 4c,d and Extended Data Fig. 1). Furthermore, interpeptide contacts ($AP_{contact}$)[86] were inspected for additional confirmation of the aggregation behaviour. The previously established AP_{SASA} cut-off value (1.75; Supplementary Section 2) and the $AP_{contact}$ threshold reported in the literature (0.5 (ref. 86)) were used as the aggregation criteria. Together, these two metrics can indicate structural shapes or aggregate numbers, giving an additional insight into aggregation dynamics.

All the decapeptides aggregated according to the set criteria, which can be visually confirmed by the differences between the initial and final frames (Fig. 4d, Supplementary Fig. 9 and Extended Data Fig. 1). Furthermore, all the hexapeptides scored above the set AP_{SASA} cut-off, and the 200 ns frames confirmed the formation of aggregates (Fig. 4c, Supplementary Fig. 10 and Extended Data Fig. 1). IMGIIA with $AP_{contact}$ marginally below the threshold showed aggregation similar to the other hexapeptides, but with a more flat morphology (Extended Data Fig. 1). IMCIEW failed to meet the AP_{SASA} criterion after 100 ns; although it passed the AP_{SASA} threshold after 200 ns, it showed less pronounced differences between the initial and final frames, possibly indicating a lower propensity towards aggregation or a less-ordered supramolecular structure compared with other generated peptides.

As a control, hexapeptides and decapeptides with low propensity to assemble were generated; when evaluated by MD, only VNGYSPKWPG had AP_{SASA} above the threshold (Supplementary Figs. 9 and 10). A clear distinction between the distribution of positive and negative classes in terms of AP_{SASA} can be confirmed by the box-plot diagrams (Supplementary Fig. 11). Although this suggests that the AP–SP classifier successfully discriminates between classes, its intended purpose in the generative model is not to produce sequences that lack a target property. Moreover, the number of NSA peptides in the training set is lower than the number of SA peptides, which might be an additional challenge for the ML model.

We experimentally validated five sequences selected based on their highest SA probability given by the AP–SP classifier or the highest AP_{SASA} scores after 100 ns simulations (Extended Data Fig. 1a), as follows: FMGIIF (FF6) with AP_{SASA} = 2.20; IMGIIA (IA6) with SA probability = 99.4%; IMCIEW (IW6) with SA probability = 99.0%; FATAAGGNMF (FF10) with AP_{SASA} = 2.27 and FGDAAGGNTT (FT10) with SA probability = 99.9%. The AP and formation of β-sheet-like assemblies was assessed in MilliQ water, at pH 7, across a range of peptide concentrations (5 mM to 0.039 mM). The sample's opacity was monitored by optical density (OD) measurements at 600 nm (OD600). FF10, IW6, IA6 and FF6 exhibited an OD600 value greater than 0.1 (for water, OD600 = 0.036), indicating aggregation[10]. Specifically, FF6 and IW6 formed cloudy suspensions. IA6 and FF10 resulted in clearer, viscous suspensions. FT10 remained a clear solution throughout the concentration range (Fig. 5a). The formation of ordered supramolecular β-sheet-like assemblies was confirmed by an increase in Thioflavin T (ThT) fluorescence at 480 nm that was more pronounced for FF10,

https://doi.org/10.1038/s42256-024-00928-1



Peptide	SA probability	$\overline{\text{Sim}}_{\text{train}}$ *	$\overline{\text{Sim}}_{\text{gen}}$ **	$\text{AP}_{\text{contact}}$	AP_{SASA}
IMGIIA	99.4%	9.2%	62.5%	0.47	1.86
IMCIEW	99.0%	9.1%	41.7%	0.54	1.77
VMGIMF	98.9%	7.2%	50.0%	0.52	2.13
FMGIIF	98.8%	9.7%	58.3%	0.61	2.29
IMGIIN	95.2%	9.0%	62.5%	0.70	2.00

* Average sequence similarity to training data
** Average sequence similarity to other generated peptides

Peptide	SA probability	$\overline{\text{Sim}}_{\text{train}}$ *	$\overline{\text{Sim}}_{\text{gen}}$ **	$\text{AP}_{\text{contact}}$	AP_{SASA}
FGDAAGGNTT	99.9%	7.3%	74.3%	0.85	1.85
FATAAGGNNF	99.7%	7.2%	76.8%	0.92	2.38
FGDAAGGNNF	99.7%	6.6%	81.6%	0.82	1.88
FGDAAGGNTF	99.7%	7.1%	81.8%	0.85	1.90
FATAAGGNMF	98.3%	8.1%	74.5%	0.85	2.19

* Average sequence similarity to training data
** Average sequence similarity to other generated peptides

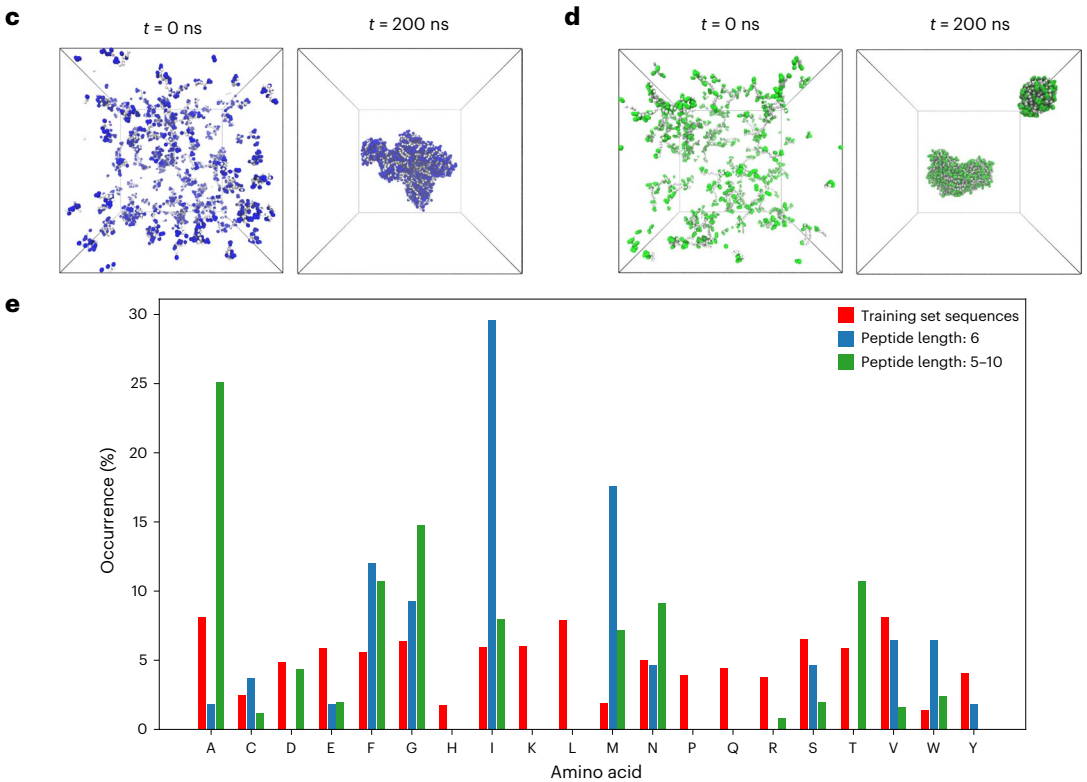


Fig. 4 | Peptides generated by the AP–SP-guided generative model. a,**b**, A plot depicting the similarities between the generated peptides and the sequences used for model training, along with the accompanying table containing the top five generated sequences with the highest probability of SA for hexapeptides (visualization errors: 5.82% for $\text{Sim}_{\text{train}}$ and 23.75% for Sim_{gen}) (**a**) and peptides with lengths between 5 and 10 residues (visualization errors: 5.97% for $\text{Sim}_{\text{train}}$ and 18.89% for Sim_{gen}) (**b**). $\text{Sim}_{\text{train}}$ and Sim_{gen} present exact calculations in terms of the average similarities of the generated peptides with the training data and other generated sequences in the same table, respectively. The full list of generated peptides is given in Supplementary Table 6. **c**,**d**, AP of the generated peptides was assessed using CG-MD simulations where the initial (0 ns) and final (200 ns) frames are shown for an example hexapeptide (FMGIIF) (**c**) and decapeptide (FATAAGGNNF) (**d**). **e**, Comparison of amino acid distributions within the training dataset (n = 9,539 amino acids), generated sequences with six residues (n = 108 amino acids) and generated sequences with 5–10 residues (n = 251 amino acids).

https://doi.org/10.1038/s42256-024-00928-1

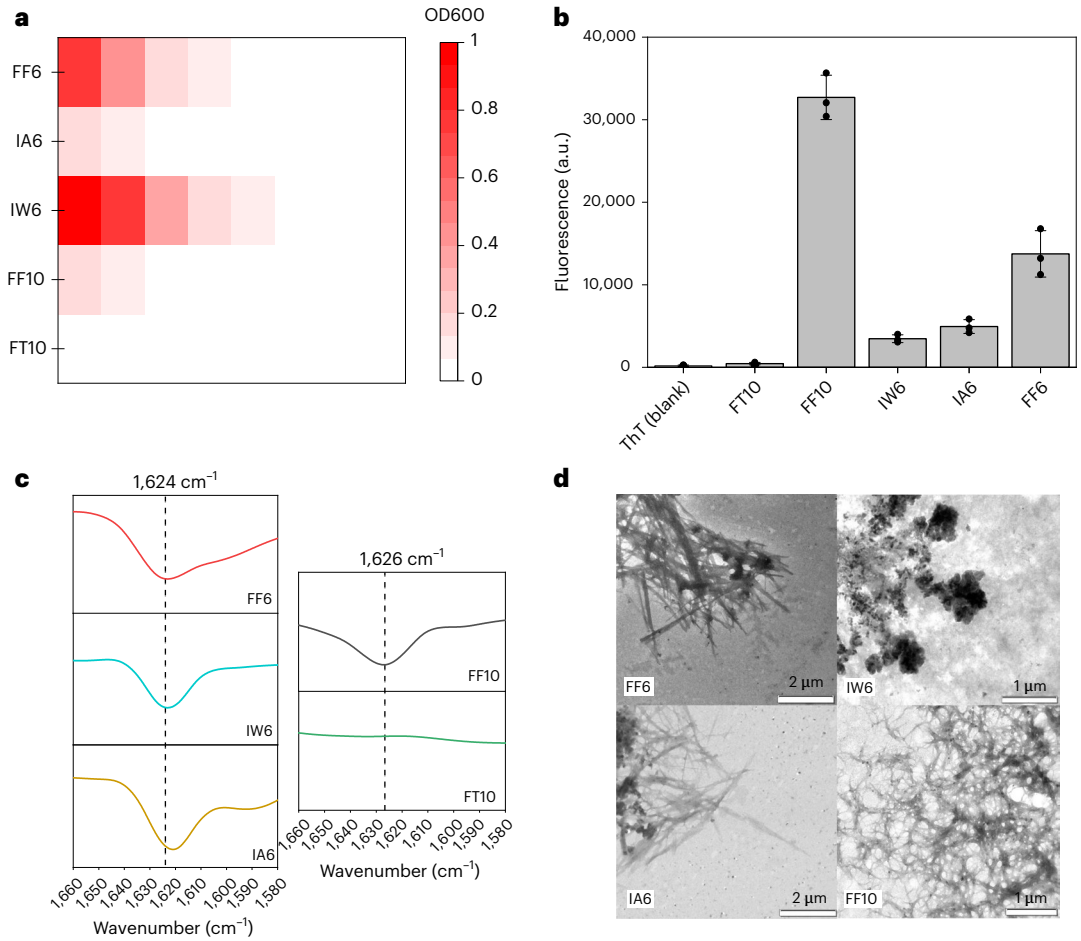


Fig. 5 | Experimental validation of selected generated peptides. a, Heat map representing the sample opacity, indicating aggregation or assembly. Each sample (FF6, IA6, IW6, FF10 and FT10) was measured across a concentration range from 5 mM to 0.039 mM. The opacity, measured as OD600, increases with concentration for all the peptides, with the exception of FT10. **b**, ThT fluorescence intensity indicating the formation of supramolecular β-sheets. Data are presented as mean values ($n = 3$ independent experiments) ± standard deviation in the bar charts, overlaid by individual measurements in the dot plots. **c**, ATR-FTIR spectra displaying the characteristic peak for the hydrogen-bonding pattern in β-sheets at 1,624 cm^{-1} for FF6, IA6 and IW6 and 1,626 cm^{-1} for FF10. **d**, TEM images showing fibrillar morphologies for FF6 (scale bar, 2 µm), IA6 (scale bar, 2 µm), FF10 (scale bar, 1 µm) and amorphous aggregates for IW6 (scale bar, 1 µm) at a concentration of 5 mM.

followed by FF6, IA6 and IW6 (Fig. 5b), as well as a characteristic peak in the Fourier transform infrared (FTIR) spectra at 1,624 cm^{-1} for hexapeptides and 1,626 cm^{-1} for FF10, indicating β-sheet hydrogen-bonding patterns (Fig. 5c). The different fluorescence emission intensities (Fig. 5b) could be due to the varying binding affinities of ThT for specific structural and chemical features within amyloid-like fibrils[87], suggesting different supramolecular morphology for each peptide. Fluorescence microscopy provided a visual confirmation of ThT binding to peptide aggregates, from bundles of fibres for IA6 to larger aggregates of undefined morphology for FF6, IW6 and FF10 (Supplementary Fig. 17), confirming amyloid-like aggregation. Transmission electron microscopy (TEM) showed entangled nanofibres for FF6 and IA6, fibrillar networks for FF10 and amorphous aggregates for IW6 (Fig. 5d). On the contrary, FT10 did not show a tendency to form β-sheet-like structures, as evidenced by OD600, FTIR and fluorescence (Fig. 5), indicating a lack of aggregation or SA. However, to rule out its assembly potential, further studies are required under different solvent and pH conditions.

The propensity of peptides to form supramolecular structures is influenced by their sequence, which determines the nature of interactions that guide their assembly and by the experimental conditions[5,88]. Even small changes can drastically affect the aggregation behaviour and morphology of supramolecular assemblies[89]. Therefore, exploring a wider range of peptide concentrations, buffer compositions, pH, temperatures and ionic strengths will complement this preliminary screening of aggregation behaviour and reveal the full potential these sequences might have as peptide materials.

These findings indicate that our model can be effectively used in generative AI approaches to create sequences characterized by diverse compositions and a strong propensity for SA, as well as maintaining minimal resemblance to previously reported sequences in the literature and the current dataset. This is further supported by the comparison of the amino acid distribution between the training and generated data (Fig. 4e). In this context, our model offers a means to explore novel regions of the chemical space rather than simply reproducing sequences encountered during the training process.

Conclusion

The challenge of identifying new peptide-based supramolecular materials on the basis of their sequence still persists. A thorough examination of compounds residing in a specific part of the chemical space constitutes an NP-hard combinatorial challenge, which remains infeasible for sequences longer than three residues. Although experimental discovery complemented by MD simulations remains the prevailing approach for uncovering new self-assembling peptides, such methods are burdened by high computational costs, require substantial time and need highly skilled experts.

The premise of this research is that longer peptides can be represented by their simpler building blocks (amino acids, dipeptides and tripeptides) whose properties may be used to predict their SA status. Five sequence-to-assembly RNN-based prediction models were developed by varying the architecture, input data and training parameters. Using precomputed AP scores obtained through the use of sliding windows that were 1, 2 or 3 residues in length, along with specific physicochemical properties, the models were trained on experimental data curated from the literature. This enabled the models to analyse sequences of arbitrary length without the need for extensive AP score calculations using MD.

The hybrid AP–SP model discriminated between SA and NSA peptides with a high F1 score of 0.865 and its ability to generalize knowledge to regions of the chemical space that are unexplored by the existing datasets was put to test in a generative model. The validation of the generated peptides (ten SA and ten NSA) using MD simulations confirmed the model precision of 90–100%. The ground-truth experimental validation was conducted for three hexapeptides and two decapeptides. OD, attenuated total reflectance (ATR)-FTIR, ThT assay and TEM (Fig. 5) measurements confirmed that four out of five peptides self-assemble, which is consistent with the accuracy of the AP–SP classifier (81.9%) used in the ML-guided generative model.

Consequently, the generative model outperformed human and AI experts with 25% to 35% greater accuracy[10]. Given the resource-intensive nature of the existing methods for SA inference, the ML models can pinpoint sequences with a high propensity towards SA while requiring substantially less time and fewer resources. We believe that the accuracy of the generative model demonstrates that the developed ML models successfully captured the underlying rules stored in the experimentally validated data and, as such, present a way to complement human intuition in the discovery of peptides with a high probability to self-assemble and offer opportunities for the development of intelligent and self-driving laboratories in the future that will allow for a faster and sustainable discovery of new materials.

Methods

The source code for this research was written in the Python programming language (version 3.10.13).

Dataset

A total of 368 peptides with validated SA behaviour were extracted from published articles, including 249 SA and 119 NSA sequences. The dataset and details on the specific methods used for the experimental validation of peptide SA are provided in Supplementary Data 1, as well as in the cited literature. Peptide SA is studied mainly in sequences shorter than 24 residues[90] and, consequently, this is the maximal sequence length in the dataset (Fig. 2a).

Sliding window approach to feature extraction

The AP values for the input to the models (amino acids, dipeptides and tripeptides as subsets of the peptide sequence) were obtained using a sliding window of the corresponding size. When amino acids were used, the array of extracted AP values was of the same length as the peptide under consideration. When dipeptides or tripeptides were examined as building units of the original sequence, the arrays were shorter by one or two entries, respectively. The sliding window of size 1—analysing physicochemical descriptors of individual amino acids—was applied to obtain SP feature vectors for the SP models, as described elsewhere[69].

Scaling of input values

The AP values for amino acids, dipeptides and tripeptides were obtained from the literature[3,27,60], and are not directly comparable in terms of scale and the underlying interpretation. The AP scores of dipeptides and tripeptides represent the percentage of the surface of a peptide exposed to water before and after aggregation, whereas the AP values of amino acids represent the energy released during the formation of supramolecular structures and can assume negative values. In addition to AP, 94 physicochemical properties were used as amino acid descriptors[69]. To boost performance and mitigate problems that may arise due to the varying magnitudes among different parts of the input data[70,71], we used min–max scaling and mapped the values of each feature into the range [−1, 1].

Sequence padding for accelerated training

Training the model in batches accelerates the process and enables faster convergence[91]. However, this requires that all the sequences in a batch are padded to an equal length. The padding value was arbitrarily set to 2, meeting the requirement of being outside the feasible [−1, 1] value range of the input features. The padded values were masked for processing in ML models to ensure that the padding is ignored and does not affect the inference.

Initial settings and callbacks during training

The training was limited to 70 epochs. The batch size was set at 600 to ensure that all the peptides were processed in a single batch, thereby obtaining the fastest speed of operation and smoother gradients. The initial learning rate was set to 0.01 and was reduced by approximately 10% per epoch (multiplied by $e^{-0.1}$), starting from the tenth epoch onwards. Only the model with the lowest validation loss was saved.

Hyperparameter optimization

A nested cross-validation consisting of five folds in the inner and outer loops was used to determine the best hyperparameters and prevent information leakage, which could result in overestimated capabilities of the models. This indicates a detailed optimization procedure that yields five repeated measurements (Fig. 2d). The outer loop of the nested fivefold cross-validation split the dataset into (1) an outer training and validation fold and (2) an outer test fold, whereas the inner loop additionally divided the outer training and validation fold into (3) an inner training fold and (4) an inner validation fold. The models were trained using each inner training fold and all the possible combinations of hyperparameters chosen for the grid search. The hyperparameters that yielded a model with the lowest loss averaged over all the inner validation folds were applied for training and testing of the model with the outer folds ((1) and (2), respectively).

Architecture tuning

The submodels that used the AP values applied two LSTM layers with five units to the input data, with the first layer being bidirectional. These submodels also encompassed a dense layer with 64 or 128 units, followed by the scaled exponential linear unit activation. On the contrary, the submodels that processed the SP values used two one-dimensional convolutional layers, each having five filters and a kernel size of 4 or 6. This presents a notable reduction in the number of filters compared with previous research on therapeutic peptides[69], which we attribute to the smaller dataset. The kernels were convolved over the peptides' spatial dimension to produce output tensors. The subsequent application of a bidirectional LSTM layer with 32, 48 or 64 units enabled capturing intricate dependencies within the peptide sequences. The models that used lower-dimensionality descriptors in the feature space compensated for the lack of information available to characterize the data with a larger output-space size.

Overfitting control and final predictions

A 50% dropout regularization was applied to the final layer of all submodels (Fig. 3a–c) as a mechanism to prevent overfitting[92]. Because relatively complex models were applied to a small dataset, a large dropout percentage was necessary to prevent the models from overly adapting to the distribution of the training data. Prevalent dropout values of 10%, 20% and 30% from the literature[92] were also examined; however, they were not

part of the hyperparameter optimization procedure. The sigmoid activation function was applied to the final layer of each model to yield a value between 0 and 1, representing the probability of an input sequence exhibiting SA. The small variations in true-positive, true-negative, false-positive and false-negative predictions (Supplementary Fig. 12) proved that overfitting was successfully avoided.

Configuration of models used for benchmarking

All the deep-learning models taken from ref. 83 were trained with an initial learning rate of 0.2, a vocabulary size of 21 (representing the number of amino acids), a maximum peptide sequence length of 24 and a batch size of 1,024. Only the MLP model was trained for 50 epochs, whereas others were trained for 100. A grid search with nested fivefold cross-validation was used to optimize the RF hyperparameters for each split separately. It explored the number of trees in the forest with values set to 100, 200, 300, 400 and 500, and tried the Gini, entropy and log-loss splitting criteria. The maximum depth of the trees was evaluated with values of 3, 6, 9, 12 and unlimited depth, whereas the minimum number of instances required to split an internal node was set to 2, 5 and 10. Additionally, the minimum number of instances required to be at a leaf node was tested with values of 1, 2 and 4.

Generative model

The generative approach utilized a genetic algorithm from another work[84] with the proposed hybrid AP–SP SA prediction model serving as a fitness function. The initial population of 50 sequences contained peptides of varying lengths (from 3 to 24 residues) and was constructed by randomly sampling a set of 20 proteinogenic amino acids. In each of the 30 algorithm iterations, 30 new sequences were created by performing tournament selection with three individuals and applying a single-point crossover. Each sequence had a 5% chance of being mutated using four equiprobable mutations: amino acid insertion, deletion, swap and change. The algorithm was conditioned to keep the population diverse and favour specific peptide lengths (6 in the first case and 5–10 in the second case) by introducing two penalty factors that measured (1) the average similarity of a peptide's amino acid distribution to the other sequences in the population and (2) the difference between the length of the peptide and the preferred length range.

MD validation of the generated peptides

The validation of the AP of the generated peptides was performed using MD, as described in previous studies[3,10,27,46]. Briefly, Protein Data Bank coordinate files for MD studies were prepared in PyMOL v1.2. Transformation of all-atom coordinate files into CG representations was achieved using the martinize.py script[93]. The input for this script was defined using Martini force field v. 2.2P with the secondary-structure input set to the extended β-sheet (the 'E' symbol in the Dictionary of Secondary Structure of Proteins)[3,27,46]. Simulations were carried out in polarizable water by converting CG water with the triple-w.py program[94]. To maintain a consistent total amino acid count (approximately 1,200 amino acids) relative to the box size, each simulation was performed by the random placement of 200 hexapeptides or 120 decapeptides in a $20 \times 20 \times 20$ nm^3 cubic system, resulting in final concentrations of 0.042 M and 0.025 M for hexapeptides and decapeptides, respectively. Each simulation underwent a three-step energy minimization, a two-step equilibration and two separate dynamics runs lasting 100 ns and 200 ns each. The initial peptide setup was minimized before the addition of water. After adding polarizable water, a 'soft-core' minimization lasting 20,000 steps of 20 fs was performed. The system was then minimized again using standard steepest descent algorithms for 50,000 steps with shorter 10 fs time steps. Equilibration was conducted in two phases: an initial short V-rescale thermostat and Berendsen barostat isotropic equilibration, followed by a more extended Nose–Hoover thermostat and Parrinello–Rahman barostat in a semi-isotropic system. This approach was adopted to combine the stability of the former method with the precision of the latter method, providing an accurate thermodynamic ensemble[95]. The V-rescale/Berendsen phase had a time step of 6 fs and 15,000 steps, whereas the Nose–Hoover/Parrinello–Rahman phase had 25 fs time steps and 500,000 steps. The dynamics runs lasted for 100 ns or 200 ns each, using 20 fs time steps. The total wall time for each simulation—from system preparation to the conclusion of the dynamics run—was approximately 5 h for 100 ns and 7.5 h for 200 ns simulations per peptide system on 10 Intel Xeon E5-2690v3 processors. The GROMACS SASA tool[96] was used to calculate the AP_{SASA} scores. $AP_{contact}$ score calculations were performed using interpeptide distances to score aggregation[86]. On constructing a matrix of paths that visit each peptide in the system exactly once, the weighted-average distance between two peptides for each path was calculated, after which the maximum value was taken as $AP_{contact}$. Distances were weighted using equation (1), where x represents the closest Euclidean distance between two beads in distinct peptides of interest. The simulations were carried out using GROMACS v. 2023.2-IMPI2021.5 gcc13.1 p3.10.5.

$$w(x) = \begin{cases} 1 & x < 4\,\text{Å} \\ e^{[-(x-4)]} & 4\,\text{Å} \le x \le 12\,\text{Å} \\ 0 & x > 12\,\text{Å} \end{cases} \quad (1)$$

Pearson's and Spearman's correlation coefficients

The sigmoid outputs from the models were assigned to a binary class using different classification thresholds determined during the hyperparameter optimization on the validation folds, as well as a fixed threshold of 0.5. These binary classes were compared with actual peptide labels that signify SA status. The corrcoef function from the numpy library (v. 1.23.5) was used to obtain the Pearson product–moment correlation coefficients, whereas the Spearman's correlation coefficient was assessed using the spearmanr function from the scipy.stats package (v. 1.9.3).

Consensus motifs of generated peptides

Multiple-sequence alignment was performed using Clustal Omega[97] through the EMBL-EBI's online service[98]. Data were analysed through Jalview[99], where the BLOSUM62 colouring was used to show sequence similarity. The motifs were derived from the phylogenetic tree. The sequence logo graphs were generated by entering the FASTA notation into the WebLogo online tool[100].

Experimental validation

The selected peptides were custom synthesized by GeneCust and are as follows: FMGIIF (FF6; M_w, 727.0 Da; purity, 96.29%), IMGIIA (IA6; M_w, 616.88 Da; purity, 95.51%), IMCIEW (IW6; M_w, 794.06 Da; purity, 96.04%), FATAAGGNMF (FF10; M_w, 986.22 Da; purity, 95.39%), FGDAAGGNTT (FT10; M_w, 910.02 Da; purity, 98.79%). For all the peptides, salt exchange was performed using 10 mM HCl to remove any traces of trifluoroacetic acid. Thioflavin T (ThT) and deuterium oxide were purchased from Sigma-Aldrich.

Sample preparation. The lyophilized powder of each peptide was weighed and dissolved in 10 mM NaOH prepared in MilliQ water or deuterium oxide (Sigma-Aldrich). The pH was adjusted to 7 with HCl prepared in MilliQ water or deuterium oxide. The samples were either used as prepared or diluted for further characterization.

Optical density. Sample opacity was used as an indicator of aggregation or assembly. Here 100 µl of each sample was added to a 96-well plate with concentrations ranging from 5 mM to 0.039 mM following twofold serial dilutions and the absorbance was recorded at 600 nm (OD600, Hidex Sense microplate reader).

ThT binding assay. ThT binding assay was performed using a peptide concentration of 5 mM for all the samples. The samples were incubated for 15 min with a ThT stock solution prepared in methanol to a final concentration of 25 μM of ThT. Then, 100 μl of each sample was added to a 96-well plate and excited at 450 nm, and the emission spectra were recorded at 480 nm (Tecan Infinite M200 PRO microplate reader).

FTIR. Hydrogen-bonding patterns characteristic of peptide SA were investigated using ATR-FTIR. The ATR-FTIR spectra of peptides were recorded in deuterium oxide (Sigma-Aldrich) using an Agilent Technologies Cary 630 FTIR instrument (Sigma-Aldrich) in the range of 650–4,000 cm^{-1} with a resolution of 16 cm^{-1}.

Fluorescent microscopy. The peptides (5 mM) were stained with 25 μM ThT. Then, 1 μl of the peptide solution was dropped on a glass slide and covered with a coverslip. Microscopy was performed using an Olympus IX73 inverted fluorescent microscope equipped with differential interference contrast and fluorescence optics (mirror units; U-FUNA (blue): EX360-370, DM410, EM420-460; U-FBWA (green): EX460-495, DM505, EM510-550; and U-FGW (red): EX530-550, DM570, EM575IF (Olympus)). The images were acquired with an Olympus XM10 monochrome camera, U-FBWA (green) filter, ×60 magnification with 1.42-numerical-aperture oil-immersion objective and CellSens Standard 2.3 software. The images were analysed using ImageJ software (v. 1.54).

TEM. TEM images were obtained using a JEOL JEM-1400Flash microscope equipped with a 20 MP complementary metal–oxide–semiconductor XAROSA camera (EMSIS). The 5 mM samples were deposited on copper grids covered with Formvar (Structure Probe) and dried before imaging.

Reporting summary

Further information on research design is available in the Nature Portfolio Reporting Summary linked to this article.

Data availability

The datasets used in this research, along with pretrained models in the H5 format, are available via GitHub at https://github.com/mnjirjak/ml_peptide_self_assembly (ref. 101). The MD coordinates for the initial and final frames of the simulations are available via figshare at https://figshare.com/s/463150e29f478cc5e25e. We also provide a workbook (Supplementary Data 1) detailing the self-assembling and non-assembling sequences taken from the literature along with the DOI and characterization methods, and raw fluorescence microscopy and TEM data (Supplementary Data 2). Source data are provided with this paper.

Code availability

The source code for this research is available via GitHub at https://github.com/mnjirjak/ml_peptide_self_assembly (ref. 101).

References

1. Lampel, A. Biology-inspired supramolecular peptide systems. *Chem* **6**, 1222–1236 (2020).
2. Janković, P., Šantek, I., Pina, A. S. & Kalafatovic, D. Exploiting peptide self-assembly for the development of minimalistic viral mimetics. *Front. Chem.* **9**, 723473 (2021).
3. Frederix, P. W. et al. Exploring the sequence space for (tri-) peptide self-assembly to design and discover new hydrogels. *Nat. Chem.* **7**, 30–37 (2015).
4. Lampel, A., Ulijn, R. & Tuttle, T. Guiding principles for peptide nanotechnology through directed discovery. *Chem. Soc. Rev.* **47**, 3737–3758 (2018).
5. Levin, A. et al. Biomimetic peptide self-assembly for functional materials. *Nat. Rev. Chem.* **4**, 615–634 (2020).
6. Chatterjee, A., Reja, A., Pal, S. & Das, D. Systems chemistry of peptide-assemblies for biochemical transformations. *Chem. Soc. Rev.* **51**, 3047–3070 (2022).
7. Ramakrishnan, M., van Teijlingen, A., Tuttle, T. & Ulijn, R. V. Integrating computation, experiment, and machine learning in the design of peptide-based supramolecular materials and systems. *Angew. Chem. Int. Ed.* **62**, e202218067 (2023).
8. Lampel, A. et al. Polymeric peptide pigments with sequence-encoded properties. *Science* **356**, 1064–1068 (2017).
9. Smith, D. J. et al. A multiphase transitioning peptide hydrogel for suturing ultrasmall vessels. *Nat. Nanotechnol.* **11**, 95–102 (2016).
10. Batra, R. et al. Machine learning overcomes human bias in the discovery of self-assembling peptides. *Nat. Chem.* **14**, 1427–1435 (2022).
11. Pierce, N. A. & Winfree, E. Protein design is NP-hard. *Protein Eng.* **15**, 779–782 (2002).
12. Hu, K. et al. Self-assembly of constrained cyclic peptides controlled by ring size. *CCS Chem.* **2**, 42–51 (2020).
13. Hu, K. et al. Tuning peptide self-assembly by an in-tether chiral center. *Sci. Adv.* **4**, 5907 (2018).
14. Chan, K. H., Lee, W. H., Ni, M., Loo, Y. & Hauser, C. A. C-terminal residue of ultrashort peptides impacts on molecular self-assembly, hydrogelation, and interaction with small-molecule drugs. *Sci. Rep.* **8**, 17127 (2018).
15. Kim, J. et al. Role of water in directing diphenylalanine assembly into nanotubes and nanowires. *Adv. Mater.* **22**, 583–587 (2010).
16. Nguyen, P. K. et al. Self-assembly of a dentinogenic peptide hydrogel. *ACS Omega* **3**, 5980–5987 (2018).
17. Yan, X. et al. Reversible transitions between peptide nanotubes and vesicle-like structures including theoretical modeling studies. *Chem. A Eur. J.* **14**, 5974–5980 (2008).
18. Yang, K. K., Wu, Z. & Arnold, F. H. Machine-learning-guided directed evolution for protein engineering. *Nat. Methods* **16**, 687–694 (2019).
19. Mandal, D., Shirazi, A. N. & Parang, K. Self-assembly of peptides to nanostructures. *Org. Biomol. Chem.* **12**, 3544–3561 (2014).
20. Shmilovich, K. et al. Discovery of self-assembling π-conjugated peptides by active learning-directed coarse-grained molecular simulation. *J. Phys. Chem. B* **124**, 3873–3891 (2020).
21. Gocheva, G., Peneva, K. & Ivanova, A. Self-assembly of doxorubicin and a drug-binding peptide studied by molecular dynamics. *Chem. Phys.* **525**, 110380 (2019).
22. Guo, C., Luo, Y., Zhou, R. & Wei, G. Triphenylalanine peptides self-assemble into nanospheres and nanorods that are different from the nanovesicles and nanotubes formed by diphenylalanine peptides. *Nanoscale* **6**, 2800–2811 (2014).
23. Lee, O.-S., Cho, V. & Schatz, G. C. Modeling the self-assembly of peptide amphiphiles into fibers using coarse-grained molecular dynamics. *Nano Lett.* **12**, 4907–4913 (2012).
24. Hauser, C. A. et al. Natural tri-to hexapeptides self-assemble in water to amyloid β-type fiber aggregates by unexpected α-helical intermediate structures. *Proc. Natl Acad. Sci. USA* **108**, 1361–1366 (2011).
25. Frederix, P. W., Patmanidis, I. & Marrink, S. J. Molecular simulations of self-assembling bio-inspired supramolecular systems and their connection to experiments. *Chem. Soc. Rev.* **47**, 3470–3489 (2018).
26. Takahashi, K., Oda, T. & Naruse, K. Coarse-grained molecular dynamics simulations of biomolecules. *AIMS Biophys.* **1**, 1–15 (2014).
27. Frederix, P. W., Ulijn, R. V., Hunt, N. T. & Tuttle, T. Virtual screening for dipeptide aggregation: toward predictive tools for peptide self-assembly. *J. Phys. Chem. Lett.* **2**, 2380–2384 (2011).
28. Zhou, P., Yuan, C. & Yan, X. Computational approaches for understanding and predicting the self-assembled peptide hydrogels. *Curr. Opin. Colloid Interface Sci.* **62**, 101645 (2022).

29. Palmer, N., Maasch, J. R., Torres, M. D. & de la Fuente-Nunez, C. Molecular dynamics for antimicrobial peptide discovery. *Infect. Immun.* **89**, 00703-20 (2021).
30. Wan, F., Wong, F., Collins, J. J. & de la Fuente-Nunez, C. Machine learning for antimicrobial peptide identification and design. *Nat. Rev. Bioeng.* **2**, 392–407 (2024).
31. Das, P. et al. Accelerated antimicrobial discovery via deep generative models and molecular dynamics simulations. *Nat. Biomed. Eng.* **5**, 613–623 (2021).
32. Yoshida, M. et al. Using evolutionary algorithms and machine learning to explore sequence space for the discovery of antimicrobial peptides. *Chem* **4**, 533–543 (2018).
33. Zeng, W.-F. et al. AlphaPeptDeep: a modular deep learning framework to predict peptide properties for proteomics. *Nat. Commun.* **13**, 7238 (2022).
34. Bukhari, S. N. H., Webber, J. & Mehbodniya, A. Decision tree based ensemble machine learning model for the prediction of Zika virus T-cell epitopes as potential vaccine candidates. *Sci. Rep.* **12**, 7810 (2022).
35. Melo, M. C., Maasch, J. R. & de la Fuente-Nunez, C. Accelerating antibiotic discovery through artificial intelligence. *Commun. Biol.* **4**, 1050 (2021).
36. Chen, J., Cheong, H. H. & Siu, S. W. XDeep-AcPEP: deep learning method for anticancer peptide activity prediction based on convolutional neural network and multitask learning. *J. Chem. Inf. Model.* **61**, 3789–3803 (2021).
37. Akbar, S. et al. iAtbP-Hyb-EnC: prediction of antitubercular peptides via heterogeneous feature representation and genetic algorithm based ensemble learning model. *Comput. Biol. Med.* **137**, 104778 (2021).
38. Aronica, P. G. et al. Computational methods and tools in antimicrobial peptide research. *J. Chem. Inf. Model.* **61**, 3172–3196 (2021).
39. Hasan, M. M. et al. HLPpred-Fuse: improved and robust prediction of hemolytic peptide and its activity by fusing multiple feature representation. *Bioinformatics* **36**, 3350–3356 (2020).
40. Manavalan, B., Shin, T. H., Kim, M. O. & Lee, G. AIPpred: sequence-based prediction of anti-inflammatory peptides using random forest. *Front. Pharmacol.* **9**, 276 (2018).
41. Oeller, M. et al. Sequence-based prediction of the intrinsic solubility of peptides containing non-natural amino acids. *Nat. Commun.* **14**, 7475 (2023).
42. Liu, Y. et al. A survey on evolutionary neural architecture search. *IEEE Trans. Neural Netw. Learn. Syst.* **34**, 550–570, (2023).
43. Elsken, T., Metzen, J. H. & Hutter, F. Neural architecture search: a survey. *J. Mach. Learn. Res.* **20**, 1997–2017 (2019).
44. Li, F. et al. Design of self-assembly dipeptide hydrogels and machine learning via their chemical features. *Proc. Natl Acad. Sci. USA* **116**, 11259–11264 (2019).
45. Xu, T. et al. Accelerating the prediction and discovery of peptide hydrogels with human-in-the-loop. *Nat. Commun.* **14**, 3880 (2023).
46. van Teijlingen, A. & Tuttle, T. Beyond tripeptides two-step active machine learning for very large data sets. *J. Chem. Theory Comput.* **17**, 3221–3232 (2021).
47. Gromski, P. S., Henson, A. B., Granda, J. M. & Cronin, L. How to explore chemical space using algorithms and automation. *Nat. Rev. Chem.* **3**, 119–128 (2019).
48. Attique, M., Farooq, M. S., Khelifi, A. & Abid, A. Prediction of therapeutic peptides using machine learning: Computational models, datasets, and feature encodings. *IEEE Access* **8**, 148570–148594 (2020).
49. Scott, G. G., Börner, T., Leser, M. E., Wooster, T. J. & Tuttle, T. Directed discovery of tetrapeptide emulsifiers. *Front. Chem.* **10**, 822868 (2022).
50. Heydari, S., Raniolo, S., Livi, L. & Limongelli, V. Transferring chemical and energetic knowledge between molecular systems with machine learning. *Commun. Chem.* **6**, 13 (2023).
51. Kaygisiz, K. et al. Inverse design of viral infectivity-enhancing peptide fibrils from continuous protein-vector embeddings. *Biomater. Sci.* **11**, 5251–5261 (2023).
52. Deo, D. R. et al. Brain control of bimanual movement enabled by recurrent neural networks. *Sci. Rep.* **14**, 1598 (2024).
53. Singh, S. H., van Breugel, F., Rao, R. P. & Brunton, B. W. Emergent behaviour and neural dynamics in artificial agents tracking odour plumes. *Nat. Mach. Intell.* **5**, 58–70 (2023).
54. Hong, T. & Stauffer, W. R. Computational complexity drives sustained deliberation. *Nat. Neurosci.* **26**, 850–857 (2023).
55. Bengio, Y., Simard, P. & Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **5**, 157–166 (1994).
56. Yang, G., Jiayu, Y., Dongdong, X., Zelin, G. & Hai, H. Feature-enhanced text-inception model for Chinese long text classification. *Sci. Rep.* **13**, 2087 (2023).
57. Zhang, L., Wang, S. & Liu, B. Deep learning for sentiment analysis: a survey. *WIREs Data Mining Knowl. Discov.* **8**, e1253 (2018).
58. Zhang, X. et al. Deeptap: an RNN-based method of TAP-binding peptide prediction in the selection of tumor neoantigens. *Comput. Biol. Med.* **164**, 107247 (2023).
59. Zhou, Z., Qiu, C. & Zhang, Y. A comparative analysis of linear regression, neural networks and random forest regression for predicting air ozone employing soft sensor models. *Sci. Rep.* **13**, 22420 (2023).
60. De Groot, N., Pallarès, I., Avilés, F., Vendrell, J. & Ventura, S. Prediction of 'hot spots' of aggregation in disease-linked polypeptides. *BMC Struct. Biol.* **5**, 18 (2005).
61. Siri Team. Hey Siri: An on-device DNN-powered voice trigger for Apple's personal assistant. *Machine Learning Research at Apple* https://machinelearning.apple.com/research/hey-siri (2017).
62. Le, Q. V. & Schuster, M. A neural network for machine translation, at production scale. *Google AI Blog* **27** (2016).
63. Su, T., Sun, L., Wang, Q.-F. & Wang, D.-H. in *Deep Learning: Fundamentals, Theory and Applications* 31–55 (Springer, 2019).
64. Guo, C. et al. Expanding the nanoarchitectural diversity through aromatic di- and tri-peptide coassembly: nanostructures and molecular mechanisms. *ACS Nano* **10**, 8316–8324 (2016).
65. Reches, M. & Gazit, E. Formation of closed-cage nanostructures by self-assembly of aromatic dipeptides. *Nano Lett.* **4**, 581–585 (2004).
66. Conchillo-Solé, O. et al. AGGRESCAN: a server for the prediction of 'hot spots' of aggregation in polypeptides. *BMC Bioinform.* **8**, 65 (2007).
67. Lee, S. et al. Self-assembling peptides and their application in the treatment of diseases. *Int. J. Mol. Sci.* **20**, 5850 (2019).
68. Lopez-Silva, T. L. & Schneider, J. P. From structure to application: progress and opportunities in peptide materials development. *Curr. Opin. Chem. Biol.* **64**, 131–144 (2021).
69. Otović, E., Njirjak, M., Kalafatovic, D. & Mauša, G. Sequential properties representation scheme for recurrent neural network-based prediction of therapeutic peptides. *J. Chem. Inf. Model.* **62**, 2961–2972 (2022).
70. Singh, D. & Singh, B. Investigating the impact of data normalization on classification performance. *Appl. Soft Comput.* **97**, 105524 (2020).
71. Nawi, N. M., Atomi, W. H. & Rehman, M. Z. The effect of data pre-processing on optimized training of artificial neural networks. *Proc. Technol.* **11**, 32–39 (2013).
72. Pascanu, R., Mikolov, T. & Bengio, Y. On the difficulty of training recurrent neural networks. In *Proc. 30th International Conference on Machine Learning* 1310–1318 (PMLR, 2013).

73. Van der Maaten, L. & Hinton, G. Visualizing data using *t*-SNE. *J. Mach. Learn. Res.* **9**, 2579–2605 (2008).
74. Wei, L., Ye, X., Sakurai, T., Mu, Z. & Wei, L. ToxIBTL: prediction of peptide toxicity based on information bottleneck and transfer learning. *Bioinformatics* **38**, 1514–1524 (2022).
75. Dean, S. N., Alvarez, J. A. E., Zabetakis, D., Walper, S. A. & Malanoski, A. P. PepVAE: variational autoencoder framework for antimicrobial peptide generation and activity prediction. *Front. Microbiol.* **12**, 725727 (2021).
76. Negovetić, M., Otović, E., Kalafatovic, D. & Mauša, G. Efficiently solving the curse of feature-space dimensionality for improved peptide classification. *Digital Discov.* **3**, 1182–1193 (2024).
77. Capecchi, A. et al. Machine learning designs non-hemolytic antimicrobial peptides. *Chem. Sci.* **12**, 9221–9232 (2021).
78. Elman, J. L. Finding structure in time. *Cogn. Sci.* **14**, 179–211 (1990).
79. Hochreiter, S. & Schmidhuber, J. Long short-term memory. *Neural Comput.* **9**, 1735–1780 (1997).
80. Schuster, M. & Paliwal, K. K. Bidirectional recurrent neural networks. *IEEE Trans. Signal Process.* **45**, 2673–2681 (1997).
81. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).
82. Vaswani, A. et al. Attention is all you need. *Adv. Neural Inf. Process. Syst.* **30**, 5999–6010 (2017).
83. Liu, Z. et al. Efficient prediction of peptide self-assembly through sequential and graphical encoding. *Brief. Bioinfor.* **24**, 409 (2023).
84. Mauša, G., Njirjak, M., Otović, E. & Kalafatovic, D. Configurable soft computing-based generative model: the search for catalytic peptides. *MRS Adv.* **8**, 1068–1074 (2023).
85. Needleman, S. B. & Wunsch, C. D. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.* **48**, 443–453 (1970).
86. Thapa, S., Clark, F., Schneebeli, S. & Li, J. Multiscale simulations to discover self-assembled oligopeptides: a benchmarking study. *J. Chem. Theory Comput.* **20**, 375–384 (2023).
87. Biancalana, M., Makabe, K., Koide, A. & Koide, S. Molecular mechanism of Thioflavin-T binding to the surface of β-rich peptide self-assemblies. *J. Mol. Biol.* **385**, 1052–1063 (2009).
88. Li, T., Lu, X.-M., Zhang, M.-R., Hu, K. & Li, Z. Peptide-based nanomaterials: self-assembly, properties and applications. *Bioact. Mater.* **11**, 268–282 (2022).
89. Ghosh, G. et al. Control over multiple nano-and secondary structures in peptide self-assembly. *Angew. Chem. Int. Ed.* **61**, 202113403 (2022).
90. Hu, X. et al. Recent advances in short peptide self-assembly: from rational design to novel applications. *Curr. Opin. Colloid Interface Sci.* **45**, 1–13 (2020).
91. Ioffe, S. & Szegedy, C. Batch normalization: accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning* 448–456 (PMLR, 2015).
92. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **15**, 1929–1958 (2014).
93. de Jong, D. H. et al. Improved parameters for the martini coarse-grained protein force field. *J. Chem. Theory Comput.* **9**, 687–697 (2013).
94. Yesylevskyy, S. O., Schäfer, L. V., Sengupta, D. & Marrink, S. J. Polarizable water model for the coarse-grained MARTINI force field. *PLoS Comput. Biol.* **6**, e1000810 (2010).
95. Hünenberger, P. H. in *Thermostat Algorithms for Molecular Dynamics Simulations* (eds Holm, C. & Kremer, K.) 105–149 (Springer, 2005).
96. Eisenhaber, F., Lijnzaad, P., Argos, P., Sander, C. & Scharf, M. The double cubic lattice method: efficient approaches to numerical integration of surface area and volume and to dot surface contouring of molecular assemblies. *J. Comput. Chem.* **16**, 273–284 (1995).
97. Sievers, F. & Higgins, D. G. Clustal omega. *Curr. Protoc. Bioinform.* **48**, 1.25.1–1.25.33 (2014).
98. Madeira, F. et al. Search and sequence analysis tools services from EMBL-EBI in 2022. *Nucleic Acids Res.* **50**, 276–279 (2022).
99. Waterhouse, A. M., Procter, J. B., Martin, D. M. A., Clamp, M. & Barton, G. J. Jalview Version 2—a multiple sequence alignment editor and analysis workbench. *Bioinformatics* **25**, 1189–1191 (2009).
100. Crooks, G. E., Hon, G., Chandonia, J.-M. & Brenner, S. E. WebLogo: a sequence logo generator. *Genome Res.* **14**, 1188–1190 (2004).
101. Njirjak, M. et al. ML peptide self-assembly. *Zenodo* https://doi.org/10.5281/zenodo.13847868 (2024).

Acknowledgements

This work was supported by the Croatian Science Foundation (grant nos. UIP-2019-04-7999 (D.K.), DOK-2021-02-3496 (D.K.) and DOK-2020-01-4659 (G.M.)); the University of Rijeka (grant nos. UNIRI-23-78 (G.M.), UNIRI-INOVA-3-23-1 (G.M.), UNIRI-23-16 (D.K.) and UNIRI-INOVA-3-23-2 (D.K.)). This work utilized resources of the Bura supercomputer facility at the University of Rijeka, Center for Advanced Computing and Modeling. We would like to thank L. Grbčić (Lawrence Berkeley National Laboratory), R. Ulijn (Advanced Science Research Center, City University New York), C. Gruber (Medical University of Vienna) and S. Marchesan (University of Trieste) for their support, feedback and invaluable discussions. We also thank J. Ban (University of Rijeka), H. Fulgosi (Institute Ruder Bošković) and R. Frkanec (University of Zagreb) for their support with fluorescent and electron microscopy measurements.

Author contributions

M.N. and L.Ž. contributed to the methodology, model creation, validation and benchmarking, writing and visualization. M.B. contributed to the methodology and simulations. P.J. contributed to the dataset collection, methodology and laboratory experiments. E.O. contributed to the methodology, model creation and benchmarking. D.K. and G.M. contributed to the conceptualization, methodology, assessment, visualization, writing, funding and project supervision. All authors reviewed the paper.

Competing interests

The authors declare no competing interests.

Additional information

Extended data is available for this paper at https://doi.org/10.1038/s42256-024-00928-1.

Supplementary information The online version contains supplementary material available at https://doi.org/10.1038/s42256-024-00928-1.

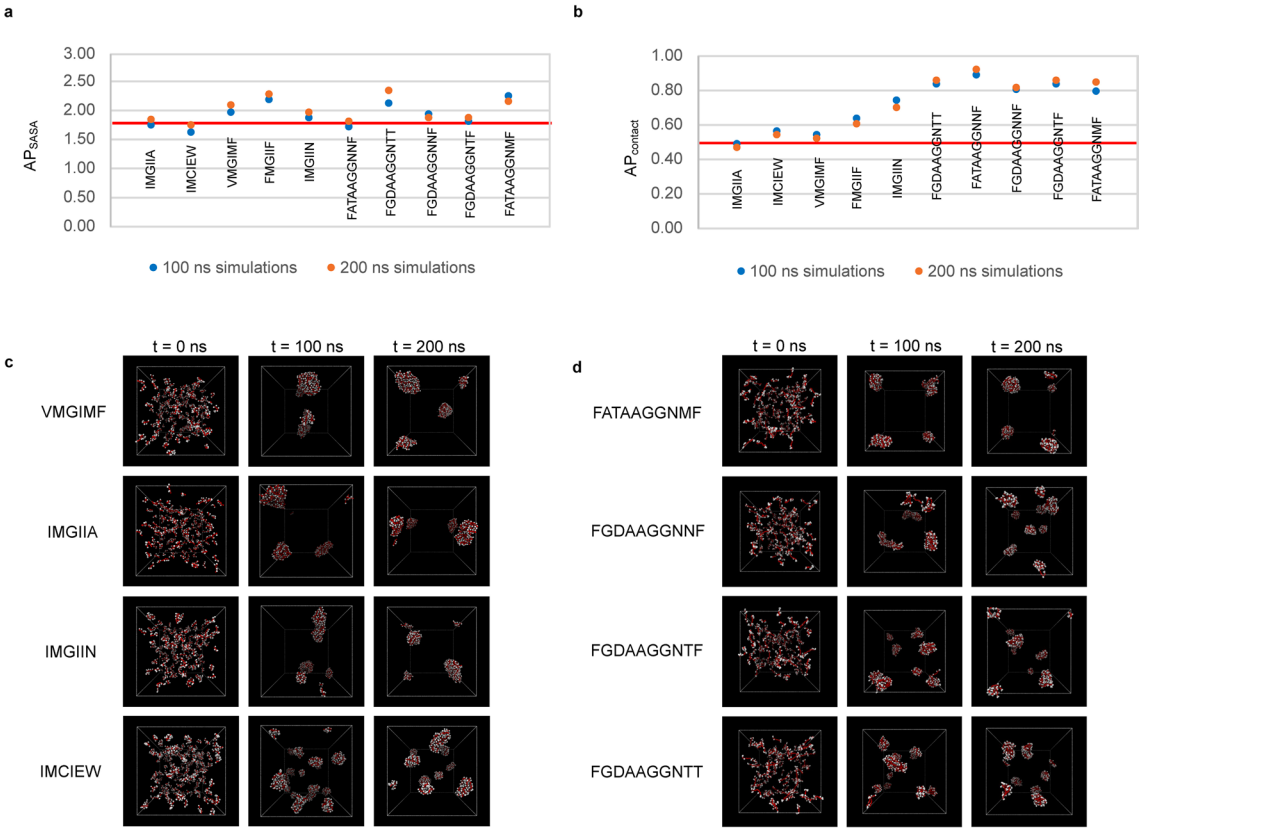
Correspondence and requests for materials should be addressed to Daniela Kalafatovic or Goran Mauša.

Peer review information *Nature Machine Intelligence* thanks Cesar de la Fuente and the other, anonymous, reviewer(s) for their contribution to the peer review of this work.

Marko Njirjak[1,4], Lucija Žužić[1,2,4], Marko Babić[3], Patrizia Janković[3], Erik Otović[1,2], Daniela Kalafatovic[2,3] ✉ & Goran Mauša[1,2] ✉

[1]University of Rijeka, Faculty of Engineering, Rijeka, Croatia. [2]University of Rijeka, Center for Artificial Intelligence and Cybersecurity, Rijeka, Croatia. [3]University of Rijeka, Faculty of Biotechnology and Drug Development, Rijeka, Croatia. [4]These authors contributed equally: Marko Njirjak, Lucija Žužić. ✉e-mail: daniela.kalafatovic@uniri.hr; goran.mausa@uniri.hr

https://doi.org/10.1038/s42256-024-00928-1



Extended Data Fig. 1 | Comparison of AP scores and the CG simulation images for the generated hexa- and decapeptides with high SA probability. a,**b** The AP_{SASA} (**a**) and $AP_{contact}$ (**b**) scores for CG simulations of 100 ns (in blue) and 200 ns (in orange). **c**,**d** The visual representation of the initial frame (t = 0 ns) and the final frames of two independent simulations at t=100 ns and t=200 ns for the hexapeptides (**c**) and decapeptides (**d**) obtained by the generative model. The red and white spheres represent the main and side chain beads, respectively. Horizontal red lines are drawn to show the thresholds for aggregation, 1.75 for AP_{SASA} and 0.5 for $AP_{contact}$.

https://doi.org/10.1038/s42256-024-00928-1

Extended Data Table 1 | Benchmarking the models: Comparison with the state-of-the-art

Metric	Transformer	RNN	LSTM	Bi-LSTM	MLP	RF	AP-SP	human*	AI**
gmean	**0.819 (0.011)**	0.746 (0.023)	0.805 (0.011)	0.797 (0.016)	0.748 (0.023)	0.772 (0.010)	0.794 (0.016)		
F1	**0.878 (0.005)**	0.852 (0.012)	**0.878 (0.005)**	0.873 (0.010)	0.848 (0.006)	0.875 (0.006)	0.865 (0.008)		
Acc	**83.7% (0.7%)**	79.5% (1.7%)	83.4% (0.7%)	82.8% (1.4%)	79.2% (1.0%)	82.6% (0.8%)	81.9% (1.1%)		
p-value	0.052	**0.025**	0.105	0.349	**0.011**	0.464	baseline		
gmean	0.780 (0.025)	0.775 (0.100)	0.841 (0.075)	0.803 (0.034)	0.712 (0.073)	0.702 (0.038)	**0.928 (0.041)**	0	0
F1	0.789 (0.025)	0.828 (0.066)	0.861 (0.065)	0.824 (0.028)	0.791 (0.043)	0.817 (0.032)	**0.930 (0.035)**	0.706	0.800
Acc	77% (2.4%)	79% (8.6%)	84% (7.3%)	80% (3.2%)	74% (5.8%)	76% (3.7%)	**92% (4.0%)**	55%	67%

* the human experts performance was estimated on 9 peptides they suggested (Batra et al., 2022.)
** the AI-expert performance was estimated on 11 peptides it predicted (Batra et al., 2022.)

The average and standard deviation (in brackets) is given for every model that was tested with 5 different seeds. The benchmarking models Transformer, RNN, LSTM, Bi-LSTM, MLP, and RF were compared against AP-SP (PR thr) on the aggregated set of peptides using the McNemar's two-sided test and the resulting p-values are marked in bold where a significant difference (< 0.05) exists. The comparison in the lower part of the table is given for 20 experimentally verified pentapeptides, where human and AI-experts participated with their predictions. The 20 verified peptides, although included in our dataset, were consistently allocated to the test fold, ensuring a rigorous setup that aims to fairly estimate and compare the performance of the models. The best scores per each metric, when rounded to two decimal places, are marked in bold.

nature portfolio

Corresponding author(s): DBPR - NATMACHINTELL-A24030878A

Last updated by author(s): Oct 10, 2024

Reporting Summary

Nature Portfolio wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Portfolio policies, see our Editorial Policies and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

n/a	Confirmed	
☐	☒	The exact sample size (*n*) for each experimental group/condition, given as a discrete number and unit of measurement
☐	☒	A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
☐	☒	The statistical test(s) used AND whether they are one- or two-sided *Only common tests should be described solely by name; describe more complex techniques in the Methods section.*
☒	☐	A description of all covariates tested
☒	☐	A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
☐	☒	A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
☐	☒	For null hypothesis testing, the test statistic (e.g. *F*, *t*, *r*) with confidence intervals, effect sizes, degrees of freedom and *P* value noted *Give P values as exact values whenever suitable.*
☒	☐	For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
☒	☐	For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
☐	☒	Estimates of effect sizes (e.g. Cohen's *d*, Pearson's *r*), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection	No software was used
Data analysis	MD simulations were run using GROMACS 2023.2-IMPI2021.5_gcc13.1_p3.10.5. McNemar's test statistic and P-values were determined using chi2.sf method from scipy.stats.distributions package version 1.11.4. A custom code was written for calculating interpeptide contacts (APcontact) according to the algorithm described in [1]. A set of 14 bash scripts, written in-house, execute the system preparation, simulation, simulation postprocessing, and calculation of SASA and AP scores. The scripts use atomistic PDB files generated by PyMOL v1.2 and the "fab" command to generate a Coarse-grained representation using martinize.py (v 2.6) software. We add polarizable water beads with triple-w.py. Both codes are run with Python-2.7.18. Visual Molecular Dynamics (VMD) v1.9.3 was used for visualizing aggregates. The rest of the experiment (ML models, generative AI approach, etc.) uses Python version 3.10.13. Matplotlib version 3.8.0 was used for plotting. The similarity between peptides was assessed using global_pairwise_align_protein from skbio.alignment (scikit-bio package version 0.5.9). Custom code implementing gradient descent was utilized to visualize plot points in Fig. 4a,b, as described in the paper. [1] Thapa, S., Clark, F., Schneebeli, S. T., & Li, J. (2023). Multiscale Simulations to Discover Self-Assembled Oligopeptides: A Benchmarking Study. Journal of Chemical Theory and Computation, 20(1), 375-384.

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors and reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Portfolio guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:

- Accession codes, unique identifiers, or web links for publicly available datasets
- A description of any restrictions on data availability
- For clinical datasets or third party data, please ensure that the statement adheres to our policy

The datasets used in this research, along with pre-trained models in H5 format, are available in a public GitHub repository: https://github.com/mnjirjak/ml_peptide_self_assembly. The molecular dynamics coordinates for the initial and final frames of the simulations are provided in a public Figshare repository: https://figshare.com/s/463150e29f478cc5e25e. We also provide a workbook (Supplementary Data 1) detailing self-assembling and non-assembling sequences taken from the literature along with DOI and characterisation methods, and raw fluorescence microscopy and TEM data (Supplementary Data 2).

Human research participants

Policy information about studies involving human research participants and Sex and Gender in Research.

Reporting on sex and gender	Not applicable
Population characteristics	Not applicable
Recruitment	Not applicable
Ethics oversight	Not applicable

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

☒ Life sciences ☐ Behavioural & social sciences ☐ Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

Sample size	A sample size for McNemar's test of statistical significance was 1840 peptides, which is sufficient given the general recommendations of >=25 samples. A dataset of 368 peptides with experimentally confirmed self-assembly status was divided into train, validation and test using nested 5-fold cross-validation procedure, which yielded 236-237 peptides for training the model, 58-59 for validation, and 73-74 for testing.
Data exclusions	No data were excluded.
Replication	Replication of the study is feasible by using the source code, pre-trained H5 models and a dataset which are provided in the GitHub repository. However, due to random seeds that were used for the stochastic processes, small deviations may occur. Moreover, we provide instructions on how random seeds can be set to a fixed value: - For the predictive models, the seed can be set by modifying the seed.txt file in the SA_ML_predictive/data folder. - For the generative model, the seed can be set in the header of the find_novel_peptides.py script in the SA_ML_generative folder. For molecular dynamics, standard methodology was applied, which is thoroughly described in the Methods section.
Randomization	For all stochastic processes, we utilized a random seed generated by the underlying Python libraries: - StratifiedKFold method from scikit-learn (version 1.1.3) for dividing the dataset into train, validation and test folds. The "shuffle" argument was set to "True". The original ratio of self-assembly to non self-assembly peptides was maintained in the folds. - For predictive models, we relied on the default parameters for random number generation provided by the underlying libraries (e.g. Tensorflow version 2.10.0). - Numpy version 1.26.3 was utilized for all operations that included randomness in the generative model.
Blinding	Blinding was not relevant to this study because splitting the data into train, validation and testing folds was done purely algorithmic. By using the k-fold cross-validation approach all peptide sequences were a part of the test folds exactly once.

Reporting for specific materials, systems and methods

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

Materials & experimental systems

n/a	Involved in the study
☒	☐ Antibodies
☒	☐ Eukaryotic cell lines
☒	☐ Palaeontology and archaeology
☒	☐ Animals and other organisms
☒	☐ Clinical data
☒	☐ Dual use research of concern

Methods

n/a	Involved in the study
☒	☐ ChIP-seq
☒	☐ Flow cytometry
☒	☐ MRI-based neuroimaging