peptidy: A light-weight Python library for peptide representation in machine learning

Rıza Özçelik[1,2*], Laura van Weesep[1], Sarah de Ruiter[1], and Francesca Grisoni[1,2*]

[1]Eindhoven University of Technology, Institute for Complex Molecular Systems and Dept. Biomedical Engineering, Eindhoven, Netherlands.
[2]Centre for Living Technologies, Alliance TU/e, WUR, UU, UMC Utrecht, Netherlands.
[*]Corresponding authors: r.ozcelik@tue.nl, f.grisoni@tue.nl

Abstract

In this work, we introduce peptidy – a lightweight Python library that facilitates converting peptides (expressed as aminoacid sequences) to numerical representations suited to machine learning. peptidy is free from external dependencies, integrates seamlessly into modern Python environments, and supports a range of encoding strategies suitable for both predictive and generative machine learning approaches. Additionally, peptidy supports peptides with post-translational modifications, such as phosphorylation, acetylation, and methylation, thereby extending the functionality of existing Python packages for peptides and proteins. peptidy is freely available with a permissive license on GitHub at the following URL: https://github.com/molML/peptidy.

1 Introduction

Peptides are relevant molecular entities in chemistry and biology, with applications ranging from drug discovery [1, 2] to food technology [3, 4]. Machine learning has accelerated peptide discovery, *e.g.*, for de novo design, sequence optimization, and property/bioactivity prediction [5–8].

A key step for machine learning is peptide representation [9, 10], whereby relevant structural information is converted into numerical formats for model training. Several strategies can be adopted to encode peptide information, *e.g.*, via description of physicochemical features [10], one-hot encoding [11], and/or evolutionary information [12]. Each of these approaches captures different structural information, might be suited for different machine learning approaches [13], and might uniquely contribute to model performance [11, 14]. While public implementations are available for specific encoding methods (*e.g.*, [15, 16]), bringing them together into a single project often requires dependency alignment and possible incompatibility hurdles.

Here, we introduce peptidy – a light-weight Python library that implements various peptide representations for machine learning. Key features of peptidy are the following:

- it has no external dependency and integrates smoothly into all modern Python environments;
- it encompasses a range of strategies for peptide encoding, useful for predictive and generative machine learning applications;
- it supports several post-translational modifications to amino acids *e.g.,* phosphorylation, acetylations, and methylation, extending the capabilities of existing Python packages.

Thanks to its light-weight character, peptidy is expected to accelerate the analysis of different pep-

https://doi.org/10.26434/chemrxiv-2024-bm3lv **ORCID:** https://orcid.org/0000-0002-4052-9721 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

Table 1: Description of encoding methods in `peptidy`. For each approach, the chemical information captured and the output dimension are reported (L: number of amino acids in the sequence.)

Method	**Information captured**	**Dimension**
Peptide descriptors[a]	Captures 48 physicochemical properties of peptides as numeric values (descriptors). A full description of the available properties can be found in the technical documentation. The selection of a subset is possible.	48×1
Amino acid descriptors[a,b]	Encodes 18 physicochemical properties at the amino acid level (selection of a subset possible). A full description of the available properties can be found in the technical documentation.	$L \times 18$
BLOSUM62 encoding[a,b]	Represents amino acids in terms of their evolutionary similarity to each other and represents the peptide as the sequence of such vectors.	$L \times 21$
One-hot encoding[a,b]	Creates fixed vectors per amino acid type (where 1 indicates its presence in a specific position in the sequence, and 0 indicates its absence).	$L \times 28$
Label encoding[a,b]	Maps each amino acid to an integer (label) and represents the peptide as a sequence of labels.	$L \times 1$

[a] Supports post-translational modifications.

[b] Supports generative deep learning.

tide representation strategies for machine learning, and to be easy to adopt and expand upon. Comprehensive online documentation and user guides were developed to facilitate the adoption of `peptidy` by the scientific community. We expect `peptidy` to further contribute to the application of machine learning for peptide discovery and optimization.

2 `peptidy`

`peptidy` is a light-weight and easy-to-use Python package (v3.6 or greater) to convert amino acid sequences into machine-learning-ready representations. Five popular peptide encoding methods are available in `peptidy` v0.0.1, with support for post-translational modifications in the sequences. The initial release of `peptidy` supports twenty standard amino acids and eight post-translational modifications, extending the capabilities of existing peptide processing tools. All the available representations (except for global descriptors) are also suited for generative deep learning, which is achieved by adding special elements to indicate the beginning and end of a given peptide sequence (*see* the technical documentation accompanying `peptidy` for more information). The implemented peptide representations are briefly described below and summarized in Table 1.

Peptide descriptors. `peptidy` implements a total of 48 global descriptors that capture physicochemical properties (*e.g.*, charge density, isoelectric point). Selecting a subset of descriptors is possible.

Amino acid descriptors. This encoding approach brings the physicochemical knowledge down to the amino acid level by representing a peptide as a sequence of amino acids, where each amino acid is encoded as predefined physicochemical properties. By default, this approach returns an $L \times 18$ dimensional list, where L is the number of amino acids in the peptide and a sub-selection of properties is possible.

https://doi.org/10.26434/chemrxiv-2024-bm3lv **ORCID:** https://orcid.org/0000-0002-4052-9721 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

BLOSUM62 encoding. BLOSUM62 is a matrix that contains amino acid similarities based on the reserved (sub)sequences on the phylogenetic trees [12]. Similar to descriptors, BLOSUM62 also introduces domain knowledge to the model, but encodes evolutionary information rather than physicochemical properties. BLOSUM62 encoding produces amino acid vectors such that each element in the vector encodes BLOSUM62 similarity score of one aminoacid to another particular amino acid. In other words, BLOSUM62 vectors represent amino acids in terms of their similarity to other amino acids. `peptidy` integrates post-translational modifications to BLOSUM62 representation by adding a new binary dimension (optional). This function returns $L \times 21$ matrices by default, where 20 dimensions encode the standard amino acids and the added dimension encodes post-translations.

One-hot encoding. One-hot-encoding represents the peptide sequence in an n-dimensional vocabulary such that each dimension encodes the presence of a particular amino acid in a particular position (denoted with a 1 if present). `peptidy` assigns a pre-defined position to each amino acid and post-translational modification in the vocabulary, and represents the peptide sequences accordingly. This function returns a $L \times 28$ dimensional matrix by default, where each column encodes the existence of an amino acid or a post-translation, for a total of 28 elements in the dictionary.

Label encoding. Label encoding assigns a unique index ('label') to each sequence element and represents sequences as a (random) list of integers. When combined with deep learning this encoding allows to learn optimal representations during training, starting from randomly initialized labels. This differs from one-hot encoding, where the vectors are fixed and pre-defined. Our implementation of label encoding supports the post-translational amino acid modifications.

3 Discussion

Machine learning for peptide discovery has been a fruitful research endeavour in the last decade, and is expect to gain further traction. Despite the availability of tools to compute peptide descriptors and representations, those tools often introduce friction during integration. This creates entry barriers for such an interdisciplinary field. `peptidy` aims to bridge the gap between peptide sequences and machine learning libraries by offering accessible encoding solutions out-of-the-box. `peptidy` not only makes popular encoding approaches as accessible as possible, it also extends the capabilities of available tools by supporting post-translational modifications that are critical to peptide properties.

`peptidy` is accompanied by extensive documentation and tutorials to facilitate accessibility. It is also open-sourced to allow feedback from researchers and extend its capabilities. We expect `peptidy` to be a useful tool for new machine learning researchers in the field.

Data and code availability

The Python code is available on GitHub at the following URL: https://github.com/molML/peptidy.

Author Contribution

Conceptualization: RÖ and FG. *Data curation*: RÖ and LvW. *Software*: RÖ, LvW, and SdR. *Writing – original draft*: RÖ. *Writing – review and editing*: all authors.

Acknowledgements

The authors acknowledge the support from the Centre for Living Technologies (Utrecht, the Netherlands).

https://doi.org/10.26434/chemrxiv-2024-bm3lv **ORCID:** https://orcid.org/0000-0002-4052-9721 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0

References

[1] M. L. Mangoni, "Host-defense peptides: from biology to therapeutic strategies," *Cellular and Molecular Life Sciences*, vol. 68, pp. 2157–2159, 2011.

[2] K. Sharma, K. K. Sharma, A. Sharma, and R. Jain, "Peptide-based drug discovery: Current status and recent advances," *Drug Discovery Today*, vol. 28, no. 2, p. 103464, 2023.

[3] R. Hartmann and H. Meisel, "Food-derived peptides with biological activity: from research to food applications," *Current opinion in biotechnology*, vol. 18, no. 2, pp. 163–169, 2007.

[4] L. M. D. L. Rodriguez and Y. Hemar, "Prospecting the applications and discovery of peptide hydrogels in food," *Trends in Food Science & Technology*, vol. 104, pp. 37–48, 2020.

[5] E. Y. Lee, B. M. Fulan, G. C. Wong, and A. L. Ferguson, "Mapping membrane activity in undiscovered peptide sequence space using machine learning," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. 13588–13593, 2016.

[6] C. D. Fjell, H. Jenssen, K. Hilpert, W. A. Cheung, N. Panté, R. E. Hancock, and A. Cherkasov, "Identification of novel antibacterial peptides by chemoinformatics and machine learning," *Journal of medicinal chemistry*, vol. 52, no. 7, pp. 2006–2015, 2009.

[7] E. Y. Lee, G. C. Wong, and A. L. Ferguson, "Machine learning-enabled discovery and design of membrane-active peptides," *Bioorganic & medicinal chemistry*, vol. 26, no. 10, pp. 2708–2718, 2018.

[8] F. Grisoni, C. S. Neuhaus, G. Gabernet, A. T. Müller, J. A. Hiss, and G. Schneider, "Designing anticancer peptides by constructive machine learning," *ChemMedChem*, vol. 13, no. 13, pp. 1300–1302, 2018.

[9] Z.-X. Yue, T.-C. Yan, H.-Q. Xu, Y.-H. Liu, Y.-F. Hong, G.-X. Chen, T. Xie, and L. Tao, "A systematic review on the state-of-the-art strategies for protein representation," *Computers in Biology and Medicine*, vol. 152, p. 106440, 2023.

[10] H. Jenssen, "Descriptors for antimicrobial peptides," *Expert opinion on drug discovery*, vol. 6, no. 2, pp. 171–184, 2011.

[11] I. Erjavac, D. Kalafatovic, and G. Mauša, "Coupled encoding methods for antimicrobial peptide prediction: how sensitive is a highly accurate model?," *Artificial Intelligence in the Life Sciences*, vol. 2, p. 100034, 2022.

[12] S. R. Eddy, "Where did the BLOSUM62 alignment score matrix come from?," *Nature Biotechnology*, vol. 22, no. 8, pp. 1035–1036, 2004.

[13] H. ElAbd, Y. Bromberg, A. Hoarfrost, T. Lenz, A. Franke, and M. Wendorff, "Amino acid encoding for deep learning applications," *BMC bioinformatics*, vol. 21, pp. 1–14, 2020.

[14] F. Grisoni, C. S. Neuhaus, M. Hishinuma, G. Gabernet, J. A. Hiss, M. Kotera, and G. Schneider, "De novo design of anticancer peptides by ensemble artificial neural networks," *Journal of molecular modeling*, vol. 25, pp. 1–10, 2019.

[15] A. T. Müller, G. Gabernet, J. A. Hiss, and G. Schneider, "modlamp: Python for antimicrobial peptides," *Bioinformatics*, vol. 33, no. 17, pp. 2753–2755, 2017.

[16] D. Osorio, P. Rondón-Villarreal, and R. Torres, "Peptides: a package for data mining of antimicrobial peptides," *Small*, vol. 12, pp. 44–444, 2015.

https://doi.org/10.26434/chemrxiv-2024-bm3lv **ORCID:** https://orcid.org/0000-0002-4052-9721 Content not peer-reviewed by ChemRxiv. **License:** CC BY 4.0