
peptides

Release 0.3.2

Martin Larralde

Apr 12, 2023

CONTENTS

1	Overview	3
2	Setup	5
3	Library	7
3.1	Installation	7
3.2	Contributing to <code>peptides.py</code>	8
3.3	API Reference	8
3.4	Changelog	36
4	License	39
	Index	41

Physicochemical properties, indices and descriptors for amino-acid sequences.

OVERVIEW

`peptides.py` is a pure-Python package to compute common descriptors for protein sequences. It started as a port of `Peptides`, the R package written by [Daniel Osorio](#) for the same purpose, but now also provides some more features from `EMBOSS`, `ExPASy Protein Identification and Analysis Tools`, and `Rcpi`. This library has no external dependency and is available for all modern Python versions (3.6+).

A non-exhaustive list of available features:

- Amino-acid Statistics:
 - Number of occurrences in the peptide sequence
 - Frequency in the peptide sequence
- QSAR descriptors:
 - BLOSUM indices
 - Cruciani properties
 - FASGAI vectors
 - Kidera factors
 - MS-WHIM scores
 - PCP descriptors
 - ProtFP descriptors
 - Sneath vectors
 - ST-scales
 - T-scales
 - VHSE-scales
 - Z-scales
- Sequence profiles:
 - Hydrophobicity profile using one of 39 proposed scales.
 - Hydrophobic moment profile based on [Eisenberg, Weiss and Terwilliger \(1984\)](#).
 - Membrane position based on [Eisenberg \(1984\)](#).
- Physical-chemical properties:
 - Aliphatic index proposed in [Ikai \(1980\)](#).
 - Instability index proposed in [Boman \(2003\)](#).

- Theoretical net charge based on the [Henderson-Hasselbach equation](#).
 - Isoelectric point using one of 8 pKa scales.
 - Molecular weight, taking into account isotope labelling, using one of 3 average weight tables.
- Biological properties:
 - Structural class using methods and reference data from either [Nakashima, Nishikawa & Ooi \(1985\)](#), [Chou \(1989\)](#), [Chou & Zhang \(1992\)](#), or [Chou, Liu, Maggiora & Zhang \(1998\)](#).

SETUP

Run `pip install peptides` in a shell to download the latest release, or have a look at the [Installation page](#) to find other ways to install `peptides.py`.

3.1 Installation

3.1.1 PyPi

`peptides.py` is hosted on GitHub, but the easiest way to install it is to download the latest release from its [PyPi repository](#).

```
$ pip install --user pyrodigal
```

3.1.2 GitHub + pip

If, for any reason, you prefer to download the library from GitHub, you can clone the repository and install the repository by running (with the admin rights):

```
$ git clone --recursive https://github.com/althonos/peptides.py
$ pip install --user ./peptides.py
```

Caution: Keep in mind this will install always try to install the latest commit, which may not even build, so consider using a versioned release instead.

3.1.3 GitHub + setuptools

If you do not want to use `pip`, you can still clone the repository and run the `setup.py` file manually, although you will need to install the build dependencies (mainly [astor](#)):

```
$ git clone --recursive https://github.com/althonos/peptides.py
$ cd peptides.py
# python setup.py build_py install
```

Danger: Installing packages without `pip` is strongly discouraged, as they can only be uninstalled manually, and may damage your system.

3.2 Contributing to peptides.py

For bug fixes or new features, please file an issue before submitting a pull request. If the change isn't trivial, it may be best to wait for feedback.

3.2.1 Running tests

Tests are written as usual Python unit tests with the `unittest` module of the standard library. Running them requires the data files to be built locally:

```
$ python setup.py codegen --inplace
$ python -m unittest discover -vv
```

3.2.2 Coding guidelines

This project targets Python 3.6 or later.

Docstrings

The docstring lines should not be longer than 76 characters (which allows rendering without soft-wrap of the entire module in a 80x24 terminal window). Docstrings should be written in Google format.

3.3 API Reference

3.3.1 Peptide

class `peptides.Peptide`(*typing.Sequence*[*str*])

A sequence of amino acids.

sequence

The peptide primary sequence, encoded in the IUPAC one-letter code.

Type

`str`

__init__(*sequence: str*) → `None`

Create a new peptide object with the given sequence.

Parameters

sequence (`str`) – A sequence of amino acids encoded with the IUPAC one-letter code. Non-standard (*O*, *U*), ambiguous (*B*, *Z*, *J*) and unknown (*X*) residues are supported in some methods, but not all of them.

aliphatic_index() → `float`

Compute the aliphatic index of the peptide.

The aliphatic index of a protein was proposed in Ikai (1980). It is defined as the relative volume occupied by aliphatic side chains (Alanine, Valine, Isoleucine, and Leucine). It may be regarded as a positive factor for the increase of thermostability of globular proteins.

Returns

`float` – The computed aliphatic index for the peptide sequence, between 0 and 100.

Example

```
>>> peptide = Peptide("SDKEVDEVDAALSDLEITLE")
>>> peptide.aliphatic_index()
117.0
```

References

- Ikai, A. *Thermostability and Aliphatic Index of Globular Proteins*. Journal of Biochemistry. Dec 1980;88(6):1895–98. PMID:7462208.

auto_correlation(table: *Dict[str, float]*, lag: *int* = 1, center: *bool* = True) → *float*

Compute the auto-correlation index of a peptide sequence.

Example

```
>>> peptide = Peptide("SDKEVDEVDAALSDLEITLE")
>>> table = peptides.tables.HYDROPHOBICITY["KyteDoolittle"]
>>> peptide.auto_correlation(table=table)
-0.3519908...
>>> peptide.auto_correlation(table=table, lag=5)
0.00113355...
```

auto_covariance(table: *Dict[str, float]*, lag: *int* = 1, center: *bool* = True) → *float*

Compute the auto-covariance index of a peptide sequence.

Example

```
>>> peptide = Peptide("SDKEVDEVDAALSDLEITLE")
>>> table = peptides.tables.HYDROPHOBICITY["KyteDoolittle"]
>>> peptide.auto_covariance(table)
-0.414005...
>>> peptide.auto_covariance(table, lag=5)
0.0010003...
```

blosum_indices() → *BLOSUMIndices*

Compute the BLOSUM62-derived indices of the peptide.

See *BLOSUMIndices* for more information.

Returns

peptides.BLOSUMIndices – The computed average BLOSUM indices for all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("KLKLLLLLKLK")
>>> for i, b in enumerate(peptide.blosum_indices()):
...     print(f"BLOSUM{i+1:<3} {b: .4f}")
BLOSUM1    -0.4827
BLOSUM2    -0.5618
BLOSUM3    -0.8509
BLOSUM4    -0.4173
BLOSUM5     0.3173
BLOSUM6     0.2527
BLOSUM7     0.1464
BLOSUM8     0.1427
BLOSUM9    -0.2145
BLOSUM10   -0.3218
```

boman() → float

Compute the Boman (potential peptide interaction) index.

The potential interaction index proposed by Boman (2003) is an index computed by averaging the solubility values for all residues in a sequence. It can be used to give an overall estimate of the potential of a peptide to bind to membranes or other proteins.

Returns

float – The Boman index for the peptide. A value greater than 2.48 indicates that a protein has high binding potential.

Example

```
>>> peptide = Peptide("FLPVLAGLTPSIVPKLVCLLTKKC")
>>> peptide.boman()
-1.2358...
```

Note: The potential protein interaction index was originally proposed as an easy way to differentiate between the action mechanism of hormones (protein/protein) and antimicrobial peptides (protein/membrane).

References

- Boman, H. G. *Antibacterial Peptides: Basic Facts and Emerging Concepts*. Journal of Internal Medicine. 2003 Sep;254(3):197–215. doi:10.1046/j.1365-2796.2003.01228.x. PMID:12930229.
-

charge(pH: float = 7, pKscale: str = 'Lehninger') → float

Compute the theoretical net charge of a peptide sequence.

This function computes the theoretical net charge of a peptide sequence, based on the Henderson-Hasselbach equation described by Dexter S. Moore (1985). The net charge can be computed at a given pH using one of the 9 pKa scales available.

Parameters

- pH (float) – The pH value for which to compute the charge.

- **pKscale** (*str*) – The name of the pKa scale to be used. A list of all the allowed values can be retrieved from the keys of the `peptides.tables.PK` dictionary.

Returns

float – The net charge of the peptide.

Example

```
>>> peptide = Peptide("FLPVLAGLTPSIVPKLVCLLTKKC")
>>> peptide.charge(pKscale="Bjellqvist")
2.7373...
>>> peptide.charge(pKscale="EMBOSS")
2.9141...
>>> peptide.charge(pKscale="Murray")
2.9075...
>>> peptide.charge(pKscale="Sillero")
2.9198...
>>> peptide.charge(pKscale="Solomon")
2.8444...
>>> peptide.charge(pKscale="Stryer")
2.8765...
>>> peptide.charge(pKscale="Lehninger")
2.8731...
>>> peptide.charge(pKscale="Dawson")
2.8444...
>>> peptide.charge(pKscale="Rodwell")
2.8197...
```

References

- Bjellqvist, B., G. J. Hughes, C. Pasquali, N. Paquet, F. Ravier, J. C. Sanchez, S. Frutiger, and D. Hochstrasser. *The Focusing Positions of Polypeptides in Immobilized pH Gradients Can Be Predicted from Their Amino Acid Sequences*. Electrophoresis. 1993 Oct;14(10):1023–31. doi:10.1002/elps.11501401163. PMID:8125050.
- Dawson, R. M. C. and D. C. Elliott. *Data for Biochemical Research*. Oxford: Clarendon Press. 2002;3:592. ISBN:978-0-19-855299-4.
- Kiraga, J. *Analysis and computer simulations of variability of isoelectric point of proteins in the proteomes*. PhD thesis, University of Wroclaw, Poland. 2008.
- Lehninger, A. L., D. L. Nelson, and M. M. Cox. *Lehninger Principles of Biochemistry*. 4th ed. New York: W.H. Freeman. 2005;4:1100. ISBN:978-0-7167-4339-2.
- Murray, R. K. *Harper's Illustrated Biochemistry*. New York: Lange Medical Books/McGraw-Hill. 2006;27. ISBN:978-0-07-146197-9.
- Rodwell, J.D. *Heterogeneity of Component Bands in Isoelectric Focusing Patterns*. Analytical Biochemistry. 1982 Jan;119(2):440-49. doi:10.1016/0003-2697(82)90611-x. PMID:7072964.
- Sillero, A., and A. Maldonado. *Isoelectric Point Determination of Proteins and Other Macromolecules: Oscillating Method*. Computers in Biology and Medicine. 2006 Feb;36(2): 157–66. doi:10.1016/j.compbiomed.2004.09.006. PMID:16389075.
- Solomons, T. W. G. *Fundamentals of Organic Chemistry*. New York: Wiley. 1997;5. ISBN:978-0-471-28298-3.

- Stryer, L., J. Augustyniak, and J. Michejda. *Biochemia*. Warszawa: Wydawnictwo Naukowe PWN. 2000. ISBN:978-83-01-12044-3.
-

counts() → Dict[str, int]

Return a table of amino-acid counts in the peptide.

Returns

dict – A dictionary mapping each amino-acid code to the number of times it occurs in the peptide sequence.

Example

```
>>> p = Peptide("SDKEVDEVDAAALS")
>>> {k:v for k,v in p.counts().items() if v != 0}
{'A': 2, 'D': 3, 'E': 2, 'L': 1, 'K': 1, 'S': 2, 'V': 2}
```

cross_covariance(table1: Dict[str, float], table2: Dict[str, float], lag: int = 1, center: bool = True) → float

Compute the cross-covariance index of a peptide sequence.

Example

```
>>> peptide = Peptide("SDKEVDEVDAAALSDLEITLE")
>>> table1 = peptides.tables.HYDROPHOBICITY["KyteDoolittle"]
>>> table2 = peptides.tables.HYDROPHOBICITY["Eisenberg"]
>>> peptide.cross_covariance(table1, table2)
-0.3026609...
>>> peptide.cross_covariance(table1, table2, lag=5)
0.0259803...
```

cruciani_properties() → CrucianiProperties

Compute the Cruciani properties of the peptide.

See [CrucianiProperties](#) for more information.

Returns

peptides.CrucianiProperties – The computed average Cruciani properties of all the amino acids in the corresponding peptide sequence.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> for i, b in enumerate(peptide.cruciani_properties()):
...     print(f"PP{i+1:<3} {b: .4f}")
PP1    -0.1130
PP2    -0.0220
PP3     0.2735
```

descriptors() → Dict[str, float]

Create a dictionary containing every protein descriptor available.

Example

```
>>> peptide = Peptide("SDKEVDEVDAALSDLEITLE")
>>> sorted(peptide.descriptors().keys())
['BLOSUM1', ..., 'F1', ..., 'KF1', ..., 'MSWHIM1', ..., 'PP1', ...]
```

Hint: Use this method to create a DataFrame containing the descriptors for several sequences.

fasgai_vectors() → *FasgaiVectors*

Compute the FASGAI vectors of the peptide.

See *FasgaiVectors* for more information.

Returns

peptides.FasgaiVectors – The computed average FASGAI vectors for all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, b in enumerate(peptide.fasgai_vectors()):
...     print(f"F{i+1:<3} {b: .5f}")
F1   -0.13675
F2   -0.45485
F3   -0.11695
F4   -0.45800
F5   -0.38015
F6    0.52740
```

frequencies() → *Dict[str, float]*

Return a table of amino-acid frequencies in the peptide.

Returns

dict – A dictionary mapping each amino-acid code to its frequency in the peptide sequence.

Example

```
>>> p = Peptide("AALS")
>>> {k:v for k,v in p.frequencies().items() if v != 0}
{'A': 0.5, 'L': 0.25, 'S': 0.25}
```

hydrophobic_moment (*window: int = 11, angle: int = 100*) → *float*

Compute the maximal hydrophobic moment of a protein sequence.

This function computes the hydrophobic moment based on Eisenberg *et al* (1984). Hydrophobic moment is a quantitative measure of the amphiphilicity perpendicular to the axis of any periodic peptide structure, such as the -helix or -sheet.

Parameters

- **angle** (*int*) – A protein rotational angle, in **degrees**. Usual values are *100* for -helix, and *160* for -sheet.
- **window** (*int*) – The size of the sliding window for which to compute the local hydrophobic moment.

Returns

float – The maximal hydrophobic moment of the peptide.

Example

```
>>> peptide = Peptide("FLPVLAGLTPSIVPKLVCLLTKKC")
>>> peptide.hydrophobic_moment(angle=100)
0.519922...
>>> peptide.hydrophobic_moment(angle=160)
0.270590...
```

See also:

The *hydrophobic_moment_profile* method, which builds a profile for each amino acid position instead of simply extracting the global maximum.

References

- Eisenberg, D., R. M. Weiss, and T. C. Terwilliger. *The Hydrophobic Moment Detects Periodicity in Protein Hydrophobicity*. Proceedings of the National Academy of Sciences of the United States of America. 1984 Jan;81(1):140–44. doi:10.1073/pnas.81.1.140. PMID:6582470.

hydrophobic_moment_profile(*window: int = 11, angle: int = 100*) → *Sequence[float]*

Build a hydrophobic moment profile of a sliding window.

This function builds a profile computing the hydrophobic moment of a section of the peptide based on the primary sequence.

Parameters

- **window** (*int*) – The size of the sliding window for which to compute the local hydrophobic moment.
- **angle** (*int*) – A protein rotational angle, in **degrees**. Usual values are *100* for -helix, and *160* for -sheet.

Example

```
>>> peptide = Peptide("ARQQNLFINFCLILIFLLLI")
>>> uH = peptide.hydrophobic_moment_profile(window=12, angle=100)
>>> [round(x, 3) for x in uH]
[0.353, 0.317, 0.274, 0.274, 0.253, 0.113, 0.113, 0.108, 0.132]
```

See also:

The *hydrophobic_moment* method, which computes the maximal hydrophobic moment instead of building a profile.

hydrophobicity(*scale: str = 'KyteDoolittle'*) → float

Compute the hydrophobicity index of a protein sequence.

This function calculates the hydrophobicity index of an amino acid sequence by averaging the hydrophobicity values of each residue using one of the 39 scales from different sources.

Parameters

scale(*str*) – The name of the hydrophobicity scale to be used. A list of all the allowed values can be retrieved from the keys of the `peptides.tables.HYDROPHOBICITY` dictionary.

Returns

float – The hydrophobicity index of the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> peptide.hydrophobicity(scale="Aboderin")
3.84...
>>> peptide.hydrophobicity(scale="AbrahamLeo")
0.092...
```

Note: The hydrophobicity is an important stabilization force in protein folding; this force changes depending on the solvent in which the protein is found.

References

- Aboderin, A. A. *An Empirical Hydrophobicity Scale for -Amino-Acids and Some of Its Applications*. International Journal of Biochemistry. 1971 Oct;2(11):537–44. doi:10.1016/0020-711X(71)90023-1.
- Abraham, D.J., and A. J. Leo. *Extension of the Fragment Method to Calculate Amino Acid Zwitterion and Side Chain Partition Coefficients*. Proteins: Structure, Function, and Genetics. 1987;2(2):130–52. doi:10.1002/prot.340020207.
- Argos, P., J. K. Rao, and P. A. Hargrave. *Structural Prediction of Membrane-Bound Proteins*. European Journal of Biochemistry. Nov 1982;128(2–3):565–75. doi:10.1111/j.1432-1033.1982.tb07002.x. PMID:7151796.
- Barley, M. H., N. J. Turner, and R. Goodacre. *Improved Descriptors for the Quantitative Structure–Activity Relationship Modeling of Peptides and Proteins*. Journal of Chemical Information and Modeling. Feb 2018;58(2):234–43. doi:10.1021/acs.jcim.7b00488. PMID:29338232.
- Black, S. D., and D. R. Mould. *Development of Hydrophobicity Parameters to Analyze Proteins Which Bear Post- or Cotranslational Modifications*. Analytical Biochemistry. Feb 1991;193(1):72–82. doi:10.1016/0003-2697(91)90045-u. PMID:2042744.
- Bull, H. B., and K. Breese. *Surface Tension of Amino Acid Solutions: A Hydrophobicity Scale of the Amino Acid Residues*. Archives of Biochemistry and Biophysics. Apr 1974;161(2):665–70. doi:10.1016/0003-9861(74)90352-x. PMID:4839053.
- Casari, G., and M. J. Sippl. *Structure-Derived Hydrophobic Potential. Hydrophobic Potential Derived from X-Ray Structures of Globular Proteins Is Able to Identify Native Folds*. Journal of Molecular Biology. Apr 1992;224(3):725–32. doi:10.1016/0022-2836(92)90556-y. PMID:1569551.
- Chothia, C. *The Nature of the Accessible and Buried Surfaces in Proteins*. Journal of Molecular Biology. Jul 1971;105(1):1–12. doi:10.1016/0022-2836(76)90191-1. PMID:994183.

- Cid, H., M. Bunster, M. Canales, and F. Gazitúa. *Hydrophobicity and Structural Classes in Proteins*. Protein Engineering. Jul 1992;5(5):373–75. doi:10.1093/protein/5.5.373. PMID:1518784.
- Cowan, R., and R. G. Whittaker. *Hydrophobicity Indices for Amino Acid Residues as Determined by High-Performance Liquid Chromatography*. Peptide Research. Apr 1990;3(2):75–80. PMID:2134053.
- Eisenberg, D., E. Schwarz, M. Komaromy, and R. Wall. *Analysis of Membrane and Surface Protein Sequences with the Hydrophobic Moment Plot*. Journal of Molecular Biology. Oct 1984;179(1):125–42. doi:10.1016/0022-2836(84)90309-7. PMID:6502707.
- Engelman, D. M., T. A. Steitz, and A. Goldman. *Identifying Nonpolar Transbilayer Helices in Amino Acid Sequences of Membrane Proteins*. Annual Review of Biophysics and Biophysical Chemistry. 1986;15:321–53. doi:10.1146/annurev.bb.15.060186.001541. PMID:3521657.
- Fasman, G. D. *Prediction of Protein Structure and the Principles of Protein Conformation*. Springer US. 1989. doi:10.1007/978-1-4613-1571-1. ISBN:978-0-306-43131-9.
- Fauchère, J-L., and Pliska V. *Hydrophobic Parameters of Amino-Acid Side Chains from the Partitioning of N-Acetyl-Amino-Acid Amides*. European Journal of Medicinal Chemistry. 1983;18(4):369–75.
- Goldsack, D. E., and R. C. Chalifoux. *Contribution of the Free Energy of Mixing of Hydrophobic Side Chains to the Stability of the Tertiary Structure of Proteins*. Journal of Theoretical Biology. Jun 1973;39(3):645–51. doi:10.1016/0022-5193(73)90075-1. PMID:4354159.
- Guy, H. R. *Amino Acid Side-Chain Partition Energies and Distribution of Residues in Soluble Proteins*. Biophysical Journal. Jan 1985;47(1):61–70. doi:10.1016/S0006-3495(85)83877-7. PMID:3978191.
- Hopp, T. P., and K. R. Woods. *Prediction of Protein Antigenic Determinants from Amino Acid Sequences*. Proceedings of the National Academy of Sciences of the United States of America. Jun 1981;78(6):3824–28. doi:10.1073/pnas.78.6.3824. PMID:6167991.
- Janin, J. *Surface and inside Volumes in Globular Proteins*. Nature. Feb 1979;277(5696):491–92. doi:10.1038/277491a0. PMID:763335.
- Jones, D. D. *Amino Acid Properties and Side-Chain Orientation in Proteins: A Cross Correlation Approach*. Journal of Theoretical Biology. Mar 1975;50(1):167–83. doi:10.1016/0022-5193(75)90031-4. PMID:1127956.
- Juretić, D., D. Zucić, B. Lucić, and N. Trinajstić. *Preference Functions for Prediction of Membrane-Buried Helices in Integral Membrane Proteins*. Computers & Chemistry. Jun 1998;22(4):279–94. doi:10.1016/s0097-8485(97)00070-3. PMID:9680689.
- Kawashima, S., H. Ogata, and M. Kanehisa. *AAindex: Amino Acid Index Database*. Nucleic Acids Research. Jan 1999;27(1):368–69. doi:10.1093/nar/27.1.368. PMID:9847231.
- Kawashima, S., and M. Kanehisa. *AAindex: Amino Acid Index Database*. Nucleic Acids Research. Jan 2000;28(1):374. doi:10.1093/nar/28.1.374. PMID:10592278.
- Kawashima, S., P. Pokarowski, M. Pokarowska, A. Kolinski, T. Katayama, and M. Kanehisa. *AAindex: Amino Acid Index Database, Progress Report 2008*. Nucleic Acids Research. Jan 2008;36:D202–205. doi:10.1093/nar/gkm998. PMID:17998252.
- Kidera, A., Y. Konishi, M. Oka, T. Ooi, and H. A. Scheraga. *Statistical Analysis of the Physical Properties of the 20 Naturally Occurring Amino Acids*. Journal of Protein Chemistry. Feb 1985;4(1):23–55. doi:10.1007/BF01025492.
- Kuhn, L. A., C. A. Swanson, M. E. Pique, J. A. Tainer, and E. D. Getzoff. *Atomic and Residue Hydrophilicity in the Context of Folded Protein Structures*. Proteins. Dec 1995;23(4):536–47. doi:10.1002/prot.340230408. PMID:8749849.

- Kyte, J., and R. F. Doolittle. *A Simple Method for Displaying the Hydropathic Character of a Protein*. Journal of Molecular Biology. May 1982;157(1):105–32. doi:10.1016/0022-2836(82)90515-0. PMID:7108955.
- Levitt, M. *A Simplified Representation of Protein Conformations for Rapid Simulation of Protein Folding*. Journal of Molecular Biology. Jun 1976;104(1):59–107. doi:10.1016/0022-2836(76)90004-8. PMID:957439.
- Manavalan, P., and P. K. Ponnuswamy. *Hydrophobic Character of Amino Acid Residues in Globular Proteins*. Nature. Oct 1978;275(5681):673–74. doi:10.1038/275673a0. PMID:703834.
- Miyazawa, S., and R. L. Jernigan. *Estimation of Effective Interresidue Contact Energies from Protein Crystal Structures: Quasi-Chemical Approximation*. Macromolecules. Mar 1985;18(3):534–52. doi:10.1021/ma00145a039.
- Nakai, K., A. Kidera, and M. Kanehisa. *Cluster Analysis of Amino Acid Indices for Prediction of Protein Structure and Function*. Protein Engineering. Jul 1988;2(2):93–100. doi:10.1093/protein/2.2.93. PMID:3244698.
- Nozaki, Y., and C. Tanford. *The Solubility of Amino Acids and Two Glycine Peptides in Aqueous Ethanol and Dioxane Solutions. Establishment of a Hydrophobicity Scale*. The Journal of Biological Chemistry. Apr 1971;246(7):2211–17. PMID:5555568.
- Parker, J. M., D. Guo, and R. S. Hodges. *New Hydrophilicity Scale Derived from High-Performance Liquid Chromatography Peptide Retention Data: Correlation of Predicted Surface Residues with Antigenicity and X-Ray-Derived Accessible Sites*. Biochemistry. 1986;25(19):5425–32. doi:10.1021/bi00367a013. PMID:2430611.
- Ponnuswamy, P. K. *Hydrophobic Characteristics of Folded Proteins*. Progress in Biophysics and Molecular Biology. 1993;59(1):57–103. doi:10.1016/0079-6107(93)90007-7. PMID:8419986.
- Prabhakaran, M. *The Distribution of Physical, Chemical and Conformational Properties in Signal and Nascent Peptides*. The Biochemical Journal. Aug 1990;269(3):691–96. doi:10.1042/bj2690691. PMID:2390062.
- Rao, J. K. M., and P. Argos. *A Conformational Preference Parameter to Predict Helices in Integral Membrane Proteins*. Biochimica Et Biophysica Acta. Jan 1986;869(2):197–214. doi:10.1016/0167-4838(86)90295-5. PMID:2935194.
- Rose, G. D., A. R. Geselowitz, G. J. Lesser, R. H. Lee, and M. H. Zehfus. *Hydrophobicity of Amino Acid Residues in Globular Proteins*. Science (New York, N.Y.). Aug 1985;229(4716):834–38. doi:10.1126/science.4023714. PMID:4023714.
- Roseman, M. A. *Hydrophilicity of Polar Amino Acid Side-Chains Is Markedly Reduced by Flanking Peptide Bonds*. Journal of Molecular Biology. Apr 1988;200(3):513–22. doi:10.1016/0022-2836(88)90540-2. PMID:3398047.
- Sweet, R. M., and D. Eisenberg. *Correlation of Sequence Hydrophobicities Measures Similarity in Three-Dimensional Protein Structure*. Journal of Molecular Biology. Dec 1983;171(4):479–88. doi:10.1016/0022-2836(83)90041-4. PMID:6663622.
- Tomii, K., and M. Kanehisa. *Analysis of Amino Acid Indices and Mutation Matrices for Sequence Comparison and Structure Prediction of Proteins*. Protein Engineering. Jan 1996;9(1):27–36. doi:10.1093/protein/9.1.27. PMID:9053899.
- Welling, G. W., W. J. Weijer, R. van der Zee R, and S. Welling-Wester. *Prediction of Sequential Antigenic Regions in Proteins*. FEBS Letters. Feb 1985;188(2):215–8. doi:10.1016/0014-5793(85)80374-4. PMID:2411595.
- White, S. H., and W. C. Wimley. *Membrane Protein Folding and Stability: Physical Principles*. Annual Review of Biophysics and Biomolecular Structure. 1999;28:319–65.

doi:10.1146/annurev.biophys.28.1.319. PMID:10410805

- White, S. H., and W. C. Wimley. *Hydrophobic Interactions of Peptides with Membrane Interfaces*. Biochimica Et Biophysica Acta. Nov 1998;1376(3):339–52. doi:10.1016/s0304-4157(98)00021-5. PMID:9804985.
 - Wilson, K. J., A. Honegger, R. P. Stötzl, and G. J. Hughes. *The Behaviour of Peptides on Reverse-Phase Supports during High-Pressure Liquid Chromatography*. The Biochemical Journal. Oct 1981;199(1):31–41. doi:10.1042/bj1990031. PMID:7337711.
 - Wimley, W. C., and S. H. White. *Experimentally Determined Hydrophobicity Scale for Proteins at Membrane Interfaces*. Nature Structural Biology. Oct 1996;3(10):842–48. doi:10.1038/nsb1096-842. PMID:8836100.
 - Wimley, W. C., T. P. Creamer, and S. H. White. *Solvation Energies of Amino Acid Side Chains and Backbone in a Family of Host-Guest Pentapeptides*. Biochemistry. Apr 1996;35(16):5109–24. doi:10.1021/bi9600153. PMID:8611495.
 - Wolfenden, R., L. Andersson, P. M. Cullis, and C. C. Southgate. *Affinities of Amino Acid Side Chains for Solvent Water*. Biochemistry. Feb 1981;20(4):849–55. doi:10.1021/bi00507a030. PMID:7213619.
 - Zimmerman, J. M., N. Eliezer, and R. Simha. *The Characterization of Amino Acid Sequences in Proteins by Statistical Methods*. Journal of Theoretical Biology. Nov 1968;21(2):170–201. doi:10.1016/0022-5193(68)90069-6. PMID:5700434.
-

hydrophobicity_profile(window: *int* = 11, scale: *str* = 'KyteDoolittle') → Sequence[float]

Build a hydrophobicity profile of a sliding window.

Example

```
>>> peptide = Peptide("ARQQNLFINFCLILIFLLLI")
>>> h = peptide.hydrophobicity_profile(window=12, scale="Eisenberg")
>>> [round(x, 3) for x in h]
[0.083, 0.147, 0.446, 0.632, 0.802, 0.955, 0.955, 0.944, 0.944]
```

instability_index() → float

Compute the instability index of a protein sequence.

This function calculates the instability index proposed by Guruprasad *et al* (1990). This index predicts the stability of a protein based on its dipeptide composition.

Returns

float – The instability index of the peptide. A protein whose instability index is smaller than 40 is predicted as stable, a value above 40 predicts that the protein may be unstable.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> round(peptide.instability_index(), 2)
83.68
```

References

- Guruprasad, K., B.V. Bhasker Reddy, and M. W. Pandit. *Correlation between Stability of a Protein and Its Dipeptide Composition: A Novel Approach for Predicting in Vivo Stability of a Protein from Its Primary Sequence*. Protein Engineering, Design and Selection. 1990 Dec;4(2):155–61. doi:10.1093/protein/4.2.155. PMID:2075190.

isoelectric_point(pKscale: *str* = 'EMBOSS') → float

Compute the isoelectric point of a protein sequence.

The isoelectric point (*pI*), is the *pH* at which a particular molecule or surface carries no net electrical charge.

Parameters

pKscale (*str*) – The name of the pKa scale to be used. A list of all the allowed values can be retrieved from the keys of the `peptides.tables.PK` dictionary.

Returns

float – The pH at which the peptide has a neutral net charge.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> peptide.isoelectric_point(pKscale="EMBOSS")
9.71...
>>> peptide.isoelectric_point(pKscale="Murray")
9.81...
>>> peptide.isoelectric_point(pKscale="Sillero")
9.89...
>>> peptide.isoelectric_point(pKscale="Solomon")
9.58...
>>> peptide.isoelectric_point(pKscale="Stryer")
9.62...
>>> peptide.isoelectric_point(pKscale="Lehninger")
9.93...
>>> peptide.isoelectric_point(pKscale="Dawson")
9.56...
>>> peptide.isoelectric_point(pKscale="Rodwell")
9.71...
```

Note: The *pI* is a variable that affects the solubility of the peptides under certain conditions of *pH*. When the *pH* of the solvent is equal to the *pI* of the protein, it tends to precipitate and lose its biological function.

References

- Rice, P., I. Longden, and A. Bleasby. *EMBOSS: The European Molecular Biology Open Software Suite*. Trends in Genetics. June 2000;16(6):276–77. doi:10.1016/s0168-9525(00)02024-2. PMID:10827456

kidera_factors() → *KideraFactors*

Compute the Kidera factors of the peptide.

See *KideraFactors* for more information.

Returns

peptides.KideraFactors – The computed average Kidera factors for all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("KLKLLLLLKLK")
>>> for i, kf in enumerate(peptide.kidera_factors()):
...     print(f"KF{i+1:<3} {kf: .4f}")
KF1   -0.7855
KF2    0.2982
KF3   -0.2364
KF4   -0.0818
KF5    0.2100
KF6   -1.8936
KF7    1.0291
KF8   -0.5127
KF9    0.1118
KF10   0.8100
```

linker_preference_profile(window: *int* = 15) → *Sequence*[float]

Compute the linker preference profile of a protein sequence.

The linker preference profile is a measure used as a basis for the DomCut method in Suyama & Ohara (2002). The resulting profile can then be used to identify putative domain boundaries in the input protein, either:

- Using prior knowledge of the estimated domain count D , in which case the $D - 1$ global minimums in the sequence can be used as cutting points
- Without prior knowledge of the domain count, using a fixed threshold to estimate the number of domains and linkers. A cutoff value of -0.09 was selected by the authors optimizing on the specificity / selectivity tradeoff.

References

- Suyama, M., O. Ohara O. *DomCut: prediction of inter-domain linker regions in amino acid sequences*. Bioinformatics. Mar 2003;19(5):673-4. doi:10.1093/bioinformatics/btg031. PMID:12651735.

New in version 0.3.0.

mass_shift(aa_shift: *Optional*[*Union*[*str*, *Dict*[*str*, float]]] = 'silac_13c', monoisotopic: *bool* = True) → float

Compute the mass difference of modified peptides.

This function calculates the mass difference of peptides introduced by chemical modifications or heavy isotope labelling.

Parameters

- **aa_shift** (*str* or *dict*) – Either the key to a pre-defined isotope label (see *peptides.tables.MASS_SHIFT*), or a dictionary mapping each amino acid to its mass difference in Dalton (use *nTer* and *cTer* keys for N-terminal and C-terminal modifications).
- **monoisotopic** (*bool*) – Flag whether monoisotopic weights of amino-acids should be used.

Returns

`float` – The mass difference of the modified peptide.

Example

```
>>> peptide = Peptide("EGVNDNECEGFFSAR")
>>> peptide.mass_shift(aa_shift="silac_13c")
6.020129...
>>> peptide.mass_shift(aa_shift=dict(R=10.00827))
10.00827...
```

References

- Ong, S-E., I. Kratchmarova, and M. Mann. *Properties of ^{13}C -Substituted Arginine in Stable Isotope Labeling by Amino Acids in Cell Culture (SILAC)*. Journal of Proteome Research. Apr 2003;2(2):173–81. doi:10.1021/pr0255708. PMID:12716131.
- Picotti, P., B. Bodenmiller, L. N. Mueller, B. Domon, and R. Aebersold. *Full Dynamic Range Proteome Analysis of *S. Cerevisiae* by Targeted Proteomics*. Cell. Aug 2009;138(4):795–806. doi:10.1016/j.cell.2009.05.051. PMID:19664813.

membrane_position_profile(window: `int = 11`, angle: `int = 100`) → `List[str]`

Compute the theoretical class of a protein sequence.

This function builds a profile predicting the theoretical class of a section of the peptide based on the relationship between the hydrophobic moment and hydrophobicity scale as proposed by Eisenberg (1984).

Parameters

- **window** (`int`) – The window size to consider when building the profile.
- **angle** (`int`) – The protein rotational angle, in **degrees**, for which to compute the hydrophobic moment profile. Usual values are *100* for -helix, and *160* for -sheet.

Returns

`list` of `str` – A list containing a one-character code for each window starting position: either 'G' for globular, 'T' for transmembrane, or 'S' for surface.

Example

```
>>> peptide = Peptide("ARQQNLFINFCLILIFLLLI")
>>> peptide.membrane_position_profile(window=12, angle=100)
['G', 'G', 'G', 'T', 'S', 'T', 'T', 'T', 'T']
>>> peptide.membrane_position_profile(window=12, angle=160)
['G', 'G', 'G', 'S', 'S', 'S', 'S', 'S', 'S']
```

References

- Eisenberg, D. *Three-Dimensional Structure of Membrane and Surface Proteins*. Annual Review of Biochemistry. July 1984;53:595–623. doi:10.1146/annurev.bi.53.070184.003115. PMID:6383201.

- Eisenberg, D., E. Schwarz, M. Komaromy, and R. Wall. *Analysis of Membrane and Surface Protein Sequences with the Hydrophobic Moment Plot*. Journal of Molecular Biology. Oct 1984;179(1):125–42. doi:10.1016/0022-2836(84)90309-7. PMID:6502707.
 - Eisenberg, D., R. M. Weiss, and T. C. Terwilliger. *The Helical Hydrophobic Moment: A Measure of the Amphiphilicity of a Helix*. Nature. Sep 1982;299(5881):371–74. doi:10.1038/299371a0. PMID:7110359
-

molecular_weight (*average*: *str* = 'expasy', *aa_shift*: *Optional[Union[*str*, Dict[*str*, float]]]* = None) → float

Compute the molecular weight of a protein sequence.

This function calculates the molecular weight of a protein sequence. It is calculated as the sum of the mass of each amino acid using one of the 3 available scales. It also supports mass calculation of proteins with predefined or custom stable isotope mass labels.

Parameters

- **average** (*str*) – The name of the average amino acid average weight scale. See `peptides.tables.MOLECULAR_WEIGHT` for a list of appropriate values.
- **aa_shift** (*str*, *dict* or *None*) – Either an appropriate shift value to pass to *Peptide.mass_shift*, or *None* to get the unmodified weight.

Returns

float – The molecular weight of the peptide, in Dalton.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> peptide.molecular_weight()
2485.91...
>>> peptide.molecular_weight(average="mascot")
2485.89...
>>> peptide.molecular_weight(average="monoisotopic")
2484.11...
```

References

- Wilkins, M. R., E. Gasteiger, A. Bairoch, J. C. Sanchez, K. L. Williams, R. D. Appel, and D. F. Hochstrasser. *Protein Identification and Analysis Tools in the ExPASy Server*. Methods in Molecular Biology. 1992;112: 531–52. doi:10.1385/1-59259-584-7:531. PMID:10027275
-

ms_whim_scores() → *MSWHIMScores*

Compute the MS-WHIM scores of the peptide.

See *MSWHIMScores* for more information.

Returns

peptides.MSWHIMScores – The compute average of MS-WHIM scores of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("KLKLLLLLKLK")
>>> for i, mw in enumerate(peptide.ms_whim_scores()):
...     print(f"MSWHIM{i+1:<3} {mw: .4f}")
MSWHIM1    -0.6564
MSWHIM2     0.4873
MSWHIM3     0.1164
```

mz(charge: *int* = 2, aa_shift: *Optional[Union[str, Dict[str, float]]]* = None, cysteins: *float* = 57.021464) → *float*

Compute the m/z (mass over charge) ratio for a peptide.

This function calculates the (monoisotopic) mass over charge ratio (m/z) for peptides, as measured in mass spectrometry.

Parameters

- **charge** (*int*) – The net charge for which the m/z should be computed.
- **aa_shift** (*str*, *dict* or *None*) – Either an appropriate shift value to pass to *Peptide.mass_shift*, or *None* to get the unmodified weight.
- **cysteins** (*float*) – The mass shift (in Dalton) of blocked cysteins. Default corresponds to cysteins blocked by iodoacetamide.

Returns

float – The m/z ratio of the peptide.

Example

```
>>> peptide = Peptide("EGVNDNECEGFFSAR")
>>> peptide.mz()
865.857...
>>> peptide.mz(aa_shift=dict(K=6.020129, R=6.020129))
868.867...
>>> peptide.mz(aa_shift="silac_13c", cysteins=58.005479)
869.359...
```

pcp_descriptors() → *PCPDescriptors*

Compute the Physical-Chemical Properties descriptors of the peptide.

See *PCPDescriptors* for more information.

Returns

peptides.PCPDescriptors – The computed average of PCP descriptors of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, pcp in enumerate(peptide.pcp_descriptors()):
...     print(f"E{i+1:<3} {pcp: .5f}")
E1    0.01090
E2    0.03810
```

(continues on next page)

(continued from previous page)

E3	0.12505
E4	0.04095
E5	-0.10595

physical_descriptors() → *PhysicalDescriptors*

Compute the Physical Descriptors of the peptide.

See *PhysicalDescriptors* for more information.**Returns**

`peptides.PhysicalDescriptors` – The computed average of Physical Descriptors of all the amino acids in the peptide. *PD1* is related to volume while *PD2* is related to hydrophilicity.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, pd in enumerate(peptide.physical_descriptors()):
...     print(f"PD{i+1:<3} {pd: .4f}")
PD1    0.1190
PD2    0.2825
```

profile(*table: Dict[str, float]*, *window: int = 1*, *default: float = 0.0*) → *Sequence[float]*

Compute a generic per-residue profile from per-residue indices.

Parameters

- **table** (*dict*) – The values per residue to apply to the whole protein sequence.
- **window** (*int*) – The window size for computing the profile. Leave as *1* to return per-residue values.
- **default** (*float*) – The default value to use for amino-acids that are not present in the given table.

Returns

`collections.abc.Sequence` of *float* – The per-residue profile values, averaged in the given window size. When window is larger than the available number of residues, an empty sequence is returned.

Example

```
>>> peptide = Peptide("PKLVCLKKC")
>>> peptide.profile(peptides.tables.CHARGE['sign'])
[0.0, 1.0, 0.0, 0.0, -1.0, 0.0, 1.0, 1.0, -1.0]
>>> peptide.profile(peptides.tables.MOLECULAR_WEIGHT['expasy'], 5)
[108..., 111..., 111..., 114..., 115...]
```

New in version 0.3.0.

protfp_descriptors() → *ProtFPDescriptors*

Compute the ProtFP descriptors of the peptide.

See *ProtFPDescriptors* for more information.

Returns

peptides.ProtFPDescriptors – The computed average of ProtFP descriptors of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCCGWGPGRRCVVRWC")
>>> for i, fp in enumerate(peptide.protfp_descriptors()):
...     print(f"ProtFP{i+1:<3} {fp: .4f}")
ProtFP1    0.2065
ProtFP2   -0.0565
ProtFP3    1.9930
ProtFP4   -0.2845
ProtFP5    0.7315
ProtFP6    0.7000
ProtFP7    0.1715
ProtFP8    0.1135
```

classmethod `sample`(length: int, frequencies: str = 'SwissProt2021') → *Peptide*

Generate a peptide with the given amino-acid frequencies.

This method is useful for testing, but using amino-acid frequencies to generate a peptide is not a biologically accurate method, instead consider sampling based on dipeptide frequencies in a particular organism, or using k-mer shuffling.

Parameters

- **length** (int) – The desired length for the generated peptide.
- **frequencies** (str) – The name of the amino-acid frequency table to use: either *KingJukes* to use the amino-acid frequencies for vertebrate organisms reported in King & Jukes (1969), or *SwissProt2021* to use the amino-acid frequencies in all the proteins from the January 2021 release of SwissProt.

Returns

Peptide – A new peptide. The first amino-acid will always be a Methionine for biological accuracy.

References

- King, J. L., and T. H. Jukes. *Non-Darwinian Evolution*. Science. May 1969;164(3881):788–98. doi:10.1126/science.164.3881.788. PMID:5767777.
- The UniProt Consortium. *UniProt: The Universal Protein Knowledgebase in 2021*. Nucleic Acids Research. Jan 2021;49(D1):D480–89. doi:10.1093/nar/gkaa1100. PMID:33237286.

sneath_vectors() → *SneathVectors*

Compute the Sneath vectors for the peptide.

See *SneathVectors* for more information.

Returns

peptides.SneathVectors – The computed average of Sneath vectors of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, fp in enumerate(peptide.sneath_vectors()):
...     print(f"SV{i+1:<3} {fp: .5f}")
SV1    0.19620
SV2    0.04655
SV3    0.04050
SV4    0.02775
```

st_scales() → *STscales*

Compute the ST-scales of the peptide.

See *STscales* for more information.

Returns

peptides.STscales – The computed average of ST-scales of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, st in enumerate(peptide.st_scales()):
...     print(f"ST{i+1:<3} {st: .5f}")
ST1    -0.63760
ST2     0.07965
ST3     0.05150
ST4     0.07135
ST5    -0.27905
ST6    -0.80995
ST7     0.58020
ST8     0.54400
```

structural_class(frequencies: *str* = 'Nakashima', distance: *str* = 'mahalanobis') → *str*

Predict the structural class of the peptide from its sequence.

The structural class of a protein, as defined in Levitt and Chothia (1976), can be either , , +, or /, with being later defined for irregular proteins. It depends on the secondary structure of the protein. Several methods have been proposed to elucidate the structural class from the amino acid sequence, all based on similarity with proteins which structures have been elucidated.

Parameters

- **frequencies** (*str*) – The frequencies of the amino acids in proteins of different structural classes to use as reference centroids. Use "Chou" to load the frequencies of the 64 proteins analyzed in Chou (1989), "Nakashima" to use the normalized frequencies of the 135 proteins analyzed in Nakashima *et al.* (1986) and Zhang & Chou (1995), or "ChouZhang" to load the frequencies of 120 proteins used in Chou & Zhang (1995).
- **distance** (*str*) – The distance metric to use in the 20-D space formed by the 20 usual amino acid to find the nearest structural class for the peptide. Use "cityblock" to use the Manhattan distance like in Chou (1989), "euclidean" to use the Euclidean distance like in Nakashima *et al* (1986), "correlation" to use the correlation distance like in Chou

& Zhang (1992), "mahalanobis" to use the Mahalanobis distance like in Chou & Zhang (1995), or "discriminant" to use the Bayes discriminant like in Chou *et al.* (1998).

Returns

str – The structural class the protein most likely belongs to. Note that some classes may not be predictable, depending on the reference frequencies being used (at the moment, the class can only be predicted from the *Nakashima* frequencies with *euclidean* or *manhattan* distances).

Example

Predict the structural class of the skipjack tuna Cytochrome C, (P0025), an protein

```
>>> p = Peptide(
...     "MGDVAKGKKTFFVQKCAQCHTVENGKGKHKVGPNLWGLFGRKTGQAEGYSYT"
...     "DANKSKGIVWNENTLMEYLENPKKIYIPGTMIFAGIKKKGERQDLVAYLK"
...     "SATS"
... )
>>> p.structural_class("Nakashima", distance="mahalanobis")
'alpha'
>>> p.structural_class("ChouZhang", distance="mahalanobis")
'beta'
>>> p.structural_class("Chou", distance="correlation")
'alpha'
>>> p.structural_class("Nakashima", distance="euclidean")
'alpha'
>>> p.structural_class("Chou", distance="cityblock")
'alpha+beta'
```

Predict the structural class of the sea krait Erabutoxin B (Q90VW1), a protein:

```
>>> p = Peptide(
...     "MKTLLTLVVVTIVCLDLGYTRICFNHQSSQPQTTKTCSPGESSCYHKQW"
...     "SDFRGTIIERGCGCPTVKPGIKLSCESEVCNN"
... )
>>> p.structural_class("Nakashima", distance="mahalanobis")
'beta'
>>> p.structural_class("ChouZhang", distance="mahalanobis")
'alpha+beta'
>>> p.structural_class("Chou", distance="correlation")
'beta'
>>> p.structural_class("Nakashima", distance="euclidean")
'zeta'
>>> p.structural_class("Chou", distance="cityblock")
'beta'
```

Predict the structural class of the *Arthrosira platensis* Ferredoxin (P00246), a protein:

```
>>> p = Peptide(
...     "MATYKVTLINAEGINETIDCDDDTYILDAAEEAGLDLPYSCRAGACSTC"
...     "AGTITSGTIDQSDQSFLDDQIEAGYVLTCAVYPTSDCTIKTHQEEGLY"
... )
>>> p.structural_class("Nakashima", distance="euclidean")
'zeta'
```

References

- Chou, K-C., W-M. Liu, G. M. Maggiora, and C-T. Zhang. *Prediction and Classification of Domain Structural Classes*. Proteins: Structure, Function, and Genetics. Apr 1998;31(1):97–103. PMID:9552161.
 - Chou, K-C., and C-T. Zhang. *Prediction of Protein Structural Classes*. Critical Reviews in Biochemistry and Molecular Biology. Feb 1995;30:275–349. doi:10.3109/10409239509083488. PMID:7587280.
 - Chou, K-C., and C-T. Zhang. *A Correlation-Coefficient Method to Predicting Protein-Structural Classes from Amino Acid Compositions*. European Journal of Biochemistry. 1992;207(2):429–33. doi:10.1111/j.1432-1033.1992.tb17067.x. PMID:1633801.
 - Chou, P. Y. *Prediction of Protein Structural Classes from Amino Acid Compositions*. In Prediction of Protein Structure and the Principles of Protein Conformation, edited by G. D. Fasman. Springer US. 1989:549–86. doi:10.1007/978-1-4613-1571-1. ISBN:978-0-306-43131-9.
 - Nakashima, H., K. Nishikawa, and T. Ooi. *The Folding Type of a Protein Is Relevant to the Amino Acid Composition*. Journal of Biochemistry. Jan 1986;99(1):153–62. doi:10.1093/oxfordjournals.jbchem.a135454. PMID:3957893.
 - Zhang, Chun-Ting, and Kuo-Chen Chou. *An Eigenvalue-Eigenvector Approach to Predicting Protein Folding Types*. Journal of Protein Chemistry. Jul 1995;14(5):309–26. doi:10.1007/BF01886788. PMID:8590599.
 - Zhou, G.P., and N. Assa-Munt. *Some Insights into Protein Structural Class Prediction*. Proteins: Structure, Function, and Bioinformatics. 2001;44(1):57–59. doi:10.1002/prot.1071. PMID:11354006.
-

svger_descriptors() → *SVGERDescriptors*

Compute the SVGER descriptors of the peptide.

See *SVGERDescriptors* for more information.

Returns

peptides.SVGERDescriptors – The computed average of SVGER descriptors of all the amino acids in the peptide.

New in version 0.3.2.

t_scales() → *TScales*

Compute the T-scales of the peptide.

See *TScales* for more information.

Returns

peptides.TScales – The computed average of T-scales of all the amino acids in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGGWGPGRRYCVRWC")
>>> for i, t in enumerate(peptide.t_scales()):
...     print(f"T{i+1:<3} {t: .4f}")
T1   -3.2700
T2   -0.0035
T3   -0.3855
```

(continues on next page)

(continued from previous page)

T4	-0.1475
T5	0.7585

vhse_scales() → *VHSEscales*

Compute the VHSE-scales of the peptide.

See *VHSEscales* for more information.

Returns

peptides.VHSEscales – The computed average of VHSE-scales of the amino acids in the peptide. *VHSE1* and *VHSE2* represent hydrophobic properties, *VHSE3* and *VHSE4* represent steric properties, while *VHSE5*, *VHSE6*, *VHSE7* and *VHSE8* represent electronic properties.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, vhse in enumerate(peptide.vhse_scales()):
...     print(f"VHSE{i+1:<3} {vhse: .4f}")
VHSE1  -0.1150
VHSE2   0.0630
VHSE3  -0.0055
VHSE4   0.7955
VHSE5   0.4355
VHSE6   0.2485
VHSE7   0.1740
VHSE8  -0.0960
```

z_scales() → *Zscales*

Compute the Z-scales of the peptide.

See *Zscales* for more information.

Returns

peptides.Zscales – The computed average of Z-scales of all the amino acid in the peptide.

Example

```
>>> peptide = Peptide("QWGRRCGWPGRRYCVRWC")
>>> for i, z in enumerate(peptide.z_scales()):
...     print(f"Z{i+1:<3} {0.0+round(z, 5): .4f}")
Z1     0.5520
Z2     0.0985
Z3     0.0000
Z4     0.8130
Z5    -0.8285
```

3.3.2 Descriptors

class peptides.**BLOSUMIndices**(*typing.NamedTuple*)

The BLOSUM62-derived indices of a peptide.

BLOSUM indices were derived of physicochemical properties that have been subjected to a VARIMAX analysis and an alignment matrix of the 20 natural AAs using the BLOSUM62 matrix.

References

- Georgiev, A. G. *Interpretable Numerical Descriptors of Amino Acid Space*. Journal of Computational Biology. May 2009;16(5):703–23. doi:10.1089/cmb.2008.0173. PMID:19432540.
-

class peptides.**CrucianiProperties**(*typing.NamedTuple*)

The Cruciani properties of a peptide.

The Cruciani properties are a collection of scaled principal component scores that summarize a broad set of descriptors calculated based on the interaction of each amino acid residue with several chemical groups (or “probes”), such as charged ions, methyl, hydroxyl groups, and so forth.

References

- Cruciani, G., M. Baroni, E. Carosati, M. Clementi, R. Valigi, and S. Clementi. *Peptide Studies by Means of Principal Properties of Amino Acids Derived from MIF Descriptors*. Journal of Chemometrics. 2004;18(3-4):146–55. doi:10.1002/cem.856.
-

class peptides.**FasgaiVectors**(*typing.NamedTuple*)

FASGAI vectors of a peptide.

The FASGAI vectors (Factor Analysis Scales of Generalized Amino Acid Information) are a set of amino acid descriptors, that reflect hydrophobicity, alpha and turn propensities, bulky properties, compositional characteristics, local flexibility, and electronic properties, that can be utilized to represent the sequence structural features of peptides or protein motifs.

References

- Liang, G., G. Chen, W. Niu, and Z. Li. *Factor Analysis Scales of Generalized Amino Acid Information as Applied in Predicting Interactions between the Human Amphiphsin-1 SH3 Domains and Their Peptide Ligands*. Chemical Biology & Drug Design. Apr 2008;71(4):345–51. doi:10.1111/j.1747-0285.2008.00641.x. PMID:18318694.
-

class peptides.**KideraFactors**(*typing.NamedTuple*)

The Kidera factors of a peptide.

The Kidera Factors were originally derived by applying multivariate analysis to 188 physical properties of the 20 amino acids and using dimension reduction techniques.

kf1

A factor modeling the helix / bend preference.

Type

float

kf2

A factor modeling the side-chain size of each residue ([AAindex:KIDA850101](#)).

Type

float

kf3

A factor modeling the extended structured preference.

Type

float

kf4

A factor representing the hydrophobicity.

Type

float

kf5

A factor modeling the double-bend preference.

Type

float

kf6

A factor modeling the partial specific volume.

Type

float

kf7

A factor modeling the flat extended preference.

Type

float

kf8

A factor modeling the occurrence in alpha regions.

Type

float

kf9

A factor encoding the pK-C.

Type

float

kf10

A factor representing the surrounding hydrophobicity.

Type

float

References

- Kidera, A., Y. Konishi, M. Oka, T. Ooi, and H. A. Scheraga. *Statistical Analysis of the Physical Properties of the 20 Naturally Occurring Amino Acids*. Journal of Protein Chemistry. Feb 1985;4(1):23–55. doi:10.1007/BF01025492.
-

class peptides.**MSWHIMScores**(typing.NamedTuple)

The MS-WHIM scores of a peptide.

MS-WHIM scores were derived from 36 electrostatic potential properties derived from the three-dimensional structure of the 20 natural amino acids.

References

- Bravi, G., E. Gancia, P. Mascagni, M. Pegna, R. Todeschini, and A. Zaliani. *MS-WHIM, New 3D Theoretical Descriptors Derived from Molecular Surface Properties: A Comparative 3D QSAR Study in a Series of Steroids*. Journal of Computer-Aided Molecular Design. Jan 1997;11(1):79-92. doi:10.1023/a:1008079512289. PMID:9139115
 - Gancia, E., G. Bravi, P. Mascagni, and A. Zaliani. *Global 3D-QSAR Methods: MS-WHIM and Autocorrelation*. Journal of Computer-Aided Molecular Design. Mar 2000;14(3):293-306. doi:10.1023/a:1008142124682. PMID:10756483.
 - Zaliani, A., and E. Gancia. *MS-WHIM Scores for Amino Acids: A New 3D-Description for Peptide QSAR and QSPR Studies*. Journal of Chemical Information and Computer Sciences. May 1999;39(3):525-33. doi:10.1021/ci980211b.
-

class peptides.**PhysicalDescriptors**(typing.NamedTuple)

The Physical Descriptors of a peptide.

The PP descriptors were constructed by improving on existing PCA-derived descriptors (Z-scales, MS-WHIM and T-scales) after correcting for the hydrophilicity of Methionine, Asparagine and Tryptophan based on Feng *et al.*

pd1

A descriptor related to residue volume.

Type
float

pd2

A descriptor related to hydrophilicity.

Type
float

Note: Barley *et al* insisted on maintaining a minimal number of descriptors as a way to reduce the chances of finding spurious QSAM models that would be affected by mutation between interaction sites.

References

- Barley, M. H., N. J. Turner, and R. Goodacre. *Improved Descriptors for the Quantitative Structure-Activity Relationship Modeling of Peptides and Proteins*. Journal of Chemical Information and Modeling. Feb 2018;58(2):234-43. doi:10.1021/acs.jcim.7b00488. PMID:29338232.
 - Feng, X., J. Sanchis, M. T. Reetz, and H. Rabitz. *Enhancing the Efficiency of Directed Evolution in Focused Enzyme Libraries by the Adaptive Substituent Reordering Algorithm*. Chemistry. Apr 2012;18(18):5646-54. doi:10.1002/chem.201103811. PMID:22434591.
-

```
class peptides.PCPDescriptors(typing.NamedTuple)
```

The Physical-Chemical Properties descriptors of a peptide.

The PCP descriptors were constructed by performing multidimensional scaling of 237 physical-chemical properties.

References

- Mathura, V. S., and W. Braun. *New Quantitative Descriptors of Amino Acids Based on Multidimensional Scaling of a Large Number of Physical–Chemical Properties*. Molecular Modeling Annual. Dec 2001;7(12):445–53. doi:10.1007/s00894-001-0058-5.
 - Mathura, V. S., D. Paris, and M. J. Mullan. *A Novel Physico-Chemical Property Based Model for Studying the Effects of Mutation on the Aggregation of Peptides*. Protein and Peptide Letters. 2009;16(8):991–98. doi:10.2174/092986609788923220. PMID:19689427.
-

```
class peptides.ProtFPDescriptors(typing.NamedTuple)
```

The ProtFP descriptors of a peptide.

The ProtFP set was constructed from a large initial selection of indices obtained from the [AAindex](#) database for all 20 naturally occurring amino acids.

References

- van Westen, G. J., R. F. Swier, J. K. Wegner, A. P. Ijzerman, H. W. van Vlijmen, and A. Bender. *Benchmarking of Protein Descriptor Sets in Proteochemometric Modeling (Part 1): Comparative Study of 13 Amino Acid Descriptor Sets*. Journal of Cheminformatics. Sep 2013;5(1):41. doi:10.1186/1758-2946-5-41. PMID:24059694.
 - van Westen, G. J., R. F. Swier, I. Cortes-Ciriano, J. K. Wegner, J. P. Overington, A. P. Ijzerman, H. W. van Vlijmen, and A. Bender. *Benchmarking of Protein Descriptor Sets in Proteochemometric Modeling (Part 2): Modeling Performance of 13 Amino Acid Descriptor Sets*. Journal of Cheminformatics. Sep 2013;5(1):42. doi:10.1186/1758-2946-5-42. PMID:24059743.
-

```
class peptides.SneathVectors(typing.NamedTuple)
```

The Sneath vectors of a peptide.

These vectors were obtained in Sneath (1996) by running PCA on the [coefficient](#) to explain the dissimilarity between the 20 natural amino acids based on binary state encoding of 134 physical and chemical properties (such as presence/absence of a —CH₃ group, step-wise optical rotation, etc.).

sv1

A descriptor representing mainly aliphatic properties of each residue ([AAindex:SNEP660101](#)).

Type

[float](#)

sv2

A descriptor putatively modeling the number of reactive groups ([AAindex:SNEP660102](#)).

Type

[float](#)

sv3

A descriptor representing the aromatic properties of each residue ([AAindex:SNEP660103](#)).

Type
float

sv4

A descriptor with uncertain interpretation (AAindex:SNEP660104).

Type
float

References

- Sneath, P. H. A. *Relations between Chemical Structure and Biological Activity in Peptides*. Journal of Theoretical Biology. Nov 1996;12(2):157–95. doi:10.1016/0022-5193(66)90112-3. PMID:4291386.
-

class peptides.STScales(typing.NamedTuple)

The ST-scales of a peptide.

The ST-scales were proposed in Yang *et al* (2010), taking 827 properties into account which are mainly constitutional, topological, geometrical, hydrophobic, electronic, and steric properties of a total set of 167 amino acids.

References

- Yang, L., M. Shu, K. Ma, H. Mei, Y. Jiang, and Z. Li. *ST-Scale as a Novel Amino Acid Descriptor and Its Application in QSAM of Peptides and Analogues*. Amino Acids. Mar 2010;38(3):805–16. doi:10.1007/s00726-009-0287-y. PMID:19373543.
-

class peptides.SVGERDescriptors(typing.NamedTuple)

The SVGER descriptors of a peptide.

SVGER descriptors were constructed by Principal Component Analysis of 74 geometrical descriptors (svger1 to svger6), 44 eigenvalue descriptors (svger7, svger8 and svger9), and 41 Randić descriptors (svger10 and svger11) computed for the 20 proteinogenic amino acids.

References

- Tong, J., L. Li, M. Bai, and K. Li. *A New Descriptor of Amino Acids-SVGER and Its Applications in Peptide QSAR*. Molecular Informatics 36, no. 5–6 (2017): 1501023. doi:10.1002/minf.201501023.
 - Randic, M. *Molecular Shape Profiles*. Journal of Chemical Information and Computer Sciences 35, no. 3 (1 May 1995): 373–82. doi:10.1021/ci00025a005.
-

class peptides.TScales(typing.NamedTuple)

The T-scales of a peptide.

The T-scales are based on 67 common topological descriptors of 135 amino acids. These topological descriptors are based on the connectivity table of amino acids alone, and to not explicitly consider 3D properties of each structure.

References

- Tian, F., P. Zhou, and Z. Li. *T-Scale as a Novel Vector of Topological Descriptors for Amino Acids and Its Application in QSARs of Peptides*. Journal of Molecular Structure. Mar 2007;830(1):106–15. doi:10.1016/j.molstruc.2006.07.004.
-

```
class peptides.VHSEscales(typing.NamedTuple)
```

The VHSE-scales of a peptide.

The VHSE-scales (principal components score Vectors of Hydrophobic, Steric, and Electronic properties), are derived from principal components analysis (PCA) on independent families of 18 hydrophobic properties, 17 steric properties, and 15 electronic properties, respectively, which are included in total 50 physicochemical variables of 20 coded amino acids.

Attribute:

vhse1 (**float**): A descriptor representing hydrophobic properties. vhse2 (**float**): Another descriptor representing hydrophobic

properties.

vhse3 (**float**): A descriptor representing steric properties. vhse4 (**float**): Another descriptor representing steric properties. vhse5 (**float**): A descriptor representing electronic properties. vhse6 (**float**): A second descriptor representing electronic

properties.

vhse7 (**float**): A third descriptor representing electronic properties.

vhse8 (**float**): A fourth descriptor representing electronic properties.

References

- Mei, H., Z. H. Liao, Y. Zhou, and S. Z. Li. *A New Set of Amino Acid Descriptors and Its Application in Peptide QSARs*. Biopolymers. 2005;80(6):775-86. doi:10.1002/bip.20296. PMID:15895431.
-

```
class peptides.ZScales(typing.NamedTuple)
```

The Z-scales of a peptide.

The Z-scales were proposed in Sandberg *et al* (1998) based on physicochemical properties of proteogenic and non-proteogenic amino acids, including NMR data and thin-layer chromatography (TLC) data.

z1

A descriptor quantifying lipophilicity.

Type

float

z2

A descriptor modeling steric properties like steric bulk and polarizability.

Type

float

z3

A descriptor quantifying electronic properties like polarity and charge.

Type

float

z4

A descriptor relating to electronegativity, heat of formation, electrophilicity and hardness.

Type
float

z5

Another descriptor relating to electronegativity, heat of formation, electrophilicity and hardness.

Type
float

References

- Sandberg, M., L. Eriksson, J. Jonsson, M. Sjöström, and S. Wold. *New Chemical Descriptors Relevant for the Design of Biologically Active Peptides. A Multivariate Characterization of 87 Amino Acids*. Journal of Medicinal Chemistry. Jul 1998;41(14):2481–91. doi:10.1021/jm9700575. PMID:9651153.
-

3.4 Changelog

All notable changes to this project will be documented in this file.

The format is based on [Keep a Changelog](#) and this project adheres to [Semantic Versioning](#).

3.4.1 Unreleased

3.4.2 v0.3.2 - 2023-04-01

Added

- `Peptide.svgger_descriptors` to compute SVGER descriptors from Tong *et al.* (2016).

Fixed

- `Peptide.physical_descriptors` not being computed by `Peptide.descriptors`.

Documentation

- Refactor documentation of individual descriptors into their own class.
- Add missing `PCPDescriptors` and `PhysicalDescriptors` to the API documentation.

3.4.3 v0.3.1 - 2022-09-01

Fixed

- `peptides.datasets` data files missing from the source distribution.

3.4.4 v0.3.0 - 2022-09-01

Added

- `Peptide.linker_preference_profile` to build a profile like used in the DomCut method from Suyama & Ohara (2002).
- `Peptide.profile` to build a generic per-residue profile from a data table (#3).

3.4.5 v0.2.1 - 2022-02-05

Fixed

- `Peptide.hydrophobic_moment` not working properly on sequences smaller than the provided window length (#1).

3.4.6 v0.2.0 - 2021-10-21

Added

- `Peptide.counts` method to get the number of occurrences of each amino acid in the peptide.
- `Peptide.frequencies` to get the frequencies of each amino acid in the peptide.
- `Peptide.pcp_descriptors` to compute the PCP descriptors from Mathura & Braun (2001).
- `Peptide.sneath_vectors` to compute the descriptors from Sneath (1966).
- Hydrophilicity descriptors from Barley (2018).
- `Peptide.structural_class` to predict the structural class of a protein using one of three reference datasets and one of four distance metrics.

Changed

- `Peptide.aliphatic_index` now supports unknown Leu/Ile residue (code *J*).
- Swap order of `Peptide.hydrophobic_moment` arguments for consistency with profile methods.
- Some `Peptide` functions now support vectorized code using `numpy` if available.

3.4.7 v0.1.0 - 2021-10-21

Initial release.

LICENSE

This library is provided under the [GNU General Public License v3.0](#). The original R [Peptides](#) package was written by [Daniel Osorio](#), [Paola Rondón-Villarreal](#) and [Rodrigo Torres](#), and is licensed under the terms of the [GNU General Public License v2.0](#).

This project is in no way not affiliated, sponsored, or otherwise endorsed by the original [Peptides](#) authors. It was developed by [Martin Larralde](#) during his PhD project at the [European Molecular Biology Laboratory](#) in the [Zeller](#) team.

Symbols

`__init__()` (*peptides.Peptide method*), 8

A

`aliphatic_index()` (*peptides.Peptide method*), 8
`auto_correlation()` (*peptides.Peptide method*), 9
`auto_covariance()` (*peptides.Peptide method*), 9

B

`blosum_indices()` (*peptides.Peptide method*), 9
`BLOSUMIndices` (*class in peptides*), 30
`boman()` (*peptides.Peptide method*), 10

C

`charge()` (*peptides.Peptide method*), 10
`counts()` (*peptides.Peptide method*), 12
`cross_covariance()` (*peptides.Peptide method*), 12
`cruciani_properties()` (*peptides.Peptide method*), 12
`CrucianiProperties` (*class in peptides*), 30

D

`descriptors()` (*peptides.Peptide method*), 12

F

`fasgai_vectors()` (*peptides.Peptide method*), 13
`FasgaiVectors` (*class in peptides*), 30
`frequencies()` (*peptides.Peptide method*), 13

H

`hydrophobic_moment()` (*peptides.Peptide method*), 13
`hydrophobic_moment_profile()` (*peptides.Peptide method*), 14
`hydrophobicity()` (*peptides.Peptide method*), 14
`hydrophobicity_profile()` (*peptides.Peptide method*), 18

I

`instability_index()` (*peptides.Peptide method*), 18
`isoelectric_point()` (*peptides.Peptide method*), 19

K

`kf1` (*peptides.KideraFactors attribute*), 30
`kf10` (*peptides.KideraFactors attribute*), 31
`kf2` (*peptides.KideraFactors attribute*), 30
`kf3` (*peptides.KideraFactors attribute*), 31
`kf4` (*peptides.KideraFactors attribute*), 31
`kf5` (*peptides.KideraFactors attribute*), 31
`kf6` (*peptides.KideraFactors attribute*), 31
`kf7` (*peptides.KideraFactors attribute*), 31
`kf8` (*peptides.KideraFactors attribute*), 31
`kf9` (*peptides.KideraFactors attribute*), 31
`kidera_factors()` (*peptides.Peptide method*), 19
`KideraFactors` (*class in peptides*), 30

L

`linker_preference_profile()` (*peptides.Peptide method*), 20

M

`mass_shift()` (*peptides.Peptide method*), 20
`membrane_position_profile()` (*peptides.Peptide method*), 21
`molecular_weight()` (*peptides.Peptide method*), 22
`ms_whim_scores()` (*peptides.Peptide method*), 22
`MSWHIMScores` (*class in peptides*), 31
`mz()` (*peptides.Peptide method*), 23

P

`pcp_descriptors()` (*peptides.Peptide method*), 23
`PCPDescriptors` (*class in peptides*), 32
`pd1` (*peptides.PhysicalDescriptors attribute*), 32
`pd2` (*peptides.PhysicalDescriptors attribute*), 32
`Peptide` (*class in peptides*), 8
`physical_descriptors()` (*peptides.Peptide method*), 24
`PhysicalDescriptors` (*class in peptides*), 32
`profile()` (*peptides.Peptide method*), 24
`protfp_descriptors()` (*peptides.Peptide method*), 24
`ProtFPDescriptors` (*class in peptides*), 33

S

`sample()` (*peptides.Peptide class method*), 25

sequence (*peptides.Peptide attribute*), 8
sneath_vectors() (*peptides.Peptide method*), 25
SneathVectors (*class in peptides*), 33
st_scales() (*peptides.Peptide method*), 26
structural_class() (*peptides.Peptide method*), 26
STScales (*class in peptides*), 34
sv1 (*peptides.SneathVectors attribute*), 33
sv2 (*peptides.SneathVectors attribute*), 33
sv3 (*peptides.SneathVectors attribute*), 33
sv4 (*peptides.SneathVectors attribute*), 34
svger_descriptors() (*peptides.Peptide method*), 28
SVGERDescriptors (*class in peptides*), 34

T

t_scales() (*peptides.Peptide method*), 28
TScales (*class in peptides*), 34

V

vhse_scales() (*peptides.Peptide method*), 29
VHSEScales (*class in peptides*), 35

Z

z1 (*peptides.ZScales attribute*), 35
z2 (*peptides.ZScales attribute*), 35
z3 (*peptides.ZScales attribute*), 35
z4 (*peptides.ZScales attribute*), 35
z5 (*peptides.ZScales attribute*), 36
z_scales() (*peptides.Peptide method*), 29
ZScales (*class in peptides*), 35