GEMSEC Neuropeptide Docs

January 28, 2019

Chris Pecunies, Aaron, Savvy Gupta

Led by: Jacob Rodriguez

PI: Mehmet Sarikaya

Abstract

The purpose of this documentation is to serve as a reference for the motivation, background, methods, and results of the GEMSEC project to discover the method of signal transduction in human neuropeptides, to ultimately discover candidate peptides with binding affinity to graphene, as well as to illuminate their mechanism of binding.

Introduction

The overarching aim of this project is to uncover the mechanism of signal transduction in neuropeptides which are experimentally determined to bind to graphene. To determine candidate peptides to be experimented on, we have utilized three principal methods: cross-sequence entropy, PAM30 sequence similarity, and RRM[^1] relative to two known binders, *GrBP5*[^2] and *M6*[^3]

Background

While there is extensive literature surrounding the subject of peptide-solid binding and its most important determinants, we have focused on two primary methods: sequence domain functionality clustering and the *Resonant Recognition Model* proposed by Irena Cosic [^4].

The *Resonant Recognition Model*, or RRM, is a method proposed by Irena Cosic for the analysis of peptides and proteins. The model assumes electron-ion interaction potential (EIIP, hereafter) along the backbone of amino acid chains to be the most significant deteriminant in a peptide or protein's biochemical phenomenological features. For each amino acid, a unique scalar EIIP value can be determined, which allows a amino acid sequence to be converted to a scalar sequence of EIIP values in its most simple form.

Methods

RRM

The Resonant Recognition Model in particular is a physical signal processing which interprets a protein or peptide's biological function as primarily determined by the corresponding translation numerical sequence of EIIP values, justified by the finding that there is significant correlation between the spectra of

an amino acid sequence represented numericaly and its biological activity [ˆ3]. In our computational studies, we have extended the principle underlying Cosic's Resonant Recognition Model to another numerical amino acid mapping, itself a representation of the amino acid's most significant biological characteristic.

From (where?) we used a mapping of the 20 amino acids to seven different categories, specifically: aromatic structure, hydrophobic function, polar structure, proline (itself), glycine (itself), negative charge, positive charge, and Cystine which is excluded (assigned to a separate category from all others, like proline and glycine. With the EIIP and biological function mapping keys, we were able to derive two unique numerical sequences per neuropeptide. From each of these translated sequences, following Irena Cosic's RRM construction, we could then perform a discrete Fourier transform to determine the spectrum of the numerical representation, which generated a characteristic peak in most cases which can be considered to be representative of an amino acid sequence's biological function.

Because our experiment is principally concerned with the relative similarity of sequences to two known graphene-binding peptide sequences (GrBP5 and M6), our initial goal was not to determine the biological function responsible for graphene binding, but to downselect three neuropeptides for experimentation, after which we could apply statistical methods to determine the most likely method of binding in the case where the candidates are successfully selected binding peptides. As such, we followed Cosic's method to determine the cross spectrum of each neuropeptide's EIIP and biological mapping spectra, and generated a cross spectra with the EIIP and biological mapping spectra of GrBP5 and M6. In this cross spectrum, the signal to noise ratio may be considered a primitive measure of "similarity" in biological function between the two peptides, should the RRM be valid in and of itself for determining such a function through characteristic peaks, because the S/N measures the degree to which constructive interference occurs between the two spectra, indicating similar biological function (which, in our case, is the potential to bind to graphene).

String distance

Although not the primary aim of our project, in the aim of reasonable comprehensiveness we included several algorithm implementations of sequence similarity in the hopes of providing possible insight into the validity of other higher-order derived similarity metrics. The specfic similarity algorithm outputs which we included are as follows: *Jaro-Winkler*, *Needleman-Wunsch*, *Smith-Waterman*, and *Levenshtein*. Each of the aforementioned sequence alignment metrics is unique enough from one another to justify inclusion without potential redundancy in the similarity matrix, and each implements a scoring method that is principally aimed at producing similarity measures for bioinformatic sequences, especially amino acid sequences. Although these are the four measures we included in our final data output for the neuropeptide sequences, the final SequenceSimilarity class provides for many more to be calculated and included

in the output .csv file, all included and implemented through the usage of the *textdistance* Python package [**omnium_textdistance_2019**]

Substitution Matrix Similarity

The PAM (point accepted mutation) matrices, initially developed by Margaret Dayhoff[^5] are commonly used in measuring the similarity of two peptide or protein sequences. The PAM30 matrix, specifically, is a sequence alignment matrix that allows 30 point accepted per 100 amino acids. The PAM30 matrix is a "shallow" sequence alignment matrix, in that it is more appropriate in determining alignment for more "similar" sequences.

Results

Fundamentally, our results were calculated on the basis of two well-established similarity matrices (the PAM30 substitution matrix and the BLOSUM substitution matrix) as well as two key-value lookups associating an amino acid with a scalar quantity representing its notable biological function (in the case of the statistical clustering key:value pairing) or its propensity for electron-ion interaction (in the case of EIIP). Since these latter two methods represent our most novel foray into characterizing peptide similarity, we first determined their correlation to one another. Since the biological function mapping is categorical, and the EIIP mapping is continuous, we utilizd the Kendall Tau correlation parameter (commonly used to compare categorical and continuous data) and found these two mappings to have a correlation (tau) of 0.1129 and corresponding p_value of 0.5083, indicating very slight agreement.

Candidate peptides

The five similarity metrics generated for each neuropeptide served as a feature set which could be weighted and used to train a regression model, but without experimental data, the relative weights for each feature were chosen by [METHOD].

After forwarding the candidate peptides for experimentation, we utilized statistical clustering and signal processing methods to predict determinant sequence domains in graphene binding which could then be used with the experimental data once acquired.

Experimental results

After generating the list of [NUMBER] candidate peptides and receiving binding affinity metrics, we were then able to perform several statistical methods to ascertain the most influential factors in peptide graphene binding.

First, we trained a simple linear regression model, using the gathered experimental results as training data, and [...]

Signal transduction

From results gathered from model training, statistical clustering, and experimental cross-validation, we were able to identify several sequence patterns and corresponding functions which may prove significant. [...]

Discussion

Computational results

Our program, using all aforementioned computational methods, generated two "similarity tables" for 934,235 [@TODO: revise this] neuropeptides: one for GrBP5, and one for the wild-type peptide M6. This generated a table of the schema (example data – replace with top 10-20 candidate peptides):

Table 1: *Output schema of .csv similarity tables*

seq	numseq	rrmsn	rrmcorr	pam30	numentr	aaentr
IMVSTED	1132335	73	88	23	45	55
GTTYUEI	3224121	11	5	6	14	13

Where the columns correspond as follows:

- **seq**: Original amino acid sequence of the neuropeptide
- **numseq**: The original amino acid sequence converted to numbers corresponding to their biological function as determined by the biological function key-value table[^6]
- **rrmsn**: The signal-to-noise ratio of the cross-signal of the EIIP frequency spectum calculated for a given neuropeptide and the sequence of comparison, normalized in the domain [0, 100].
- **rrmcor**: The Pearson correlation coefficient between the EIIP frequency spectrum calculated for a given neuropeptide and the sequence of comparison, mapped to the domain [0, 100]
- **pam30**: The PAM30 similarity score (formula described in methods section); a measure of inverse distance using the PAM30 matrix, mapped to [0, 100]
- **numentr**: The cross-entropy of the numerical sequence represencted by `numseq`
- **aaentr**: The cross-entropy of the amino acid sequence

From this data, we determined that [...]

Experimental results

[to be completed after experimentaiton]

Clustering and signal analysis

After gathering the experimental results, we were then able to apply supervised statistical learning techniques to the full set of neuropeptides, as well as signal processing techniques. We first [...]

How to run

Prerequisites: A computer running Windows/Mac OSX/Linux with Python (3+) installed, and the Python libraries pandas, NumPy, and scikit-learn. For visualizations, the library matplotlib should be installed. If these libraries are not already installed, instructions for their installation will be listed below.

1. Download the .zip containing all .py files and sample data set **[!FIX]** and extract to preferred location.
2. Open a terminal and navigate to the directory containing the extracted files
    1. On Windows, press the Windows key and type "cmd", and the press enter. `dir` lists all files and folders in the working directory, while `cd dirName` changes the working directory to the specified folder (in this case, `dirname`). To move up the directory hierarchy, type `cd ../`
    2. On Mac OSX, enter spotlight search with Command + Spacebar and type "Terminal", then hit enter. Instructions are the same as for windows, but replace `dir` with `ls`.
    3. On Linux, a terminal is likely easily accessible. Commands are the same as for Mac OSX.
3. When in the directory containing the extracted files, type the following command to generate the "similarity tables" for the example dataset: `python main.py example_data.csv`
    - This can be run with any .csv sequence file, but it must follow the same schema as the example_data.csv table provided, and the sequences must all be of the same length relative to each other as well as to the known binder(s) (non length-matching input sequences will be ignored in output)
    - *(To implement)* The script accepts as a second argument a sequence (or list of sequences), each of which will generate its own similarity table for the input .csv.
        - If no argument is given (as above), it will generate two .csv similarity tables, one for GrBP5 and one for M6. To specify only one .csv output similarity table (for GrBP5), run `python main.py example_data.csv grbp5`.
        - This can be done for an arbitrary number of different sequences, for example: `python main.py example_data.csv IVTSSY UVGEASTT EEVTUSGMII` will output three .csv tables for peptides

in `example_data.csv` of lengths corresponding to each sequence specified by the user.

- Finally, the second argument can itself be a .csv of sequences, following the same schema as the first input .csv. In this way, for a .csv entered as a second argument with 10 rows of sequences will generate 10 separate .csv tables.

4. Similarity tables and visualizations will be generated in a folder titled "output" located in the same directory as the `main.py` file.

Appendix

Table 1: AA

Function	AA	Num	EIIP
Aromatic	F	0	0.0946
Aromatic	Y	0	0.0516
Aromatic	W	0	0.0548
Hydrophobic	A	1	0.0373
Hydrophobic	V	1	0.0057
Hydrophobic	I	1	O.0000
Hydrophobic	L	1	0.0000
Hydrophobic	M	1	0.0823
Polar	S	2	0.0829
Polar	T	2	0.0941
Polar	N	2	0.0036
Polar	Q	2	0.0761
Proline	P	3	0.0198
Glycine	G	4	0.0050
Charge (-)	D	5	0.1263
Charge (-)	E	5	0.0058
Charge (+)	K	6	0.0371
Charge (+)	H	6	0.0242
Charge (+)	R	6	0.0959
Excluded	C	7	0.0829

Bibliography