GEMSEC Neuropeptide Docs

January 28, 2019

Chris Pecunies, Aaron, Savvy Gupta

Led by: Jacob Rodriguez

PI: Mehmet Sarikaya

Abstract

This documentation covers the phase of the GEMSEC Neuropeptide project involved in the downselection of three candidate peptides through the implementation of several similarity measures in the creation of two similarity matrices, one for each of two known graphene-binding 12-length peptides, principally utilizing Irena Cosic's Resonant Recognition Model.

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 have been experimentally determined by GEMSEC to bind to graphene. To achieve this goal, we will be using several different measures of peptide similarity to ultimately select three peptides from a set of several thousand neuropeptides to experimentally determine the graphene binding of affinity of, ultimately in the overarching goal of determining the relative efficacy of these techniques highly specified peptide synthesis. To determine candidate peptides to be experimented on, we have utilized three principal methods: cross-sequence entropy, PAM30 sequence similarity, and RRM relative to two known binders, *GrBP5* and *M6*

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.

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 is a theoretical approach in determining protein functionality, proposed by Irena Cosic. The Resonant Recognition Model (hereafter, RRM) posits that the primary determinant of biological or physiochemical functionlity for a protein or peptide lies in the stochastic delocalized electron potential along its carbon backbone. This premise is motivated by the observed high correlation between the spectra of an amino acid sequence represented numerically according to some unique categorical physical functionality and its biological activity.

The electron-ion interaction potential is determined by the pseudopotential of delocalized electrons along the backbone of the carbon backbone. In this manner, each of the 20 essential amino acids which form the set of neuropeptides to be analyzed are represented by a scalar EIIP value ranging from 0 (Leucine, Ile) to 0.1263 (Asp). These EIIP values are then used to calculate a spectrum unique to each sequence through a Discrete Fourier Transform (DFT) which is calculated, as proposed by Cosic:

$$X(n) = \sum x(m)e^{-j\frac{2nm\pi}{N}} \subset n = 1, 2 \ldots N/2 \tag{1}$$

These DFT coefficients are ultimately calculated for each neuropeptide sequence. Cosic proposes cross-spectral function between EIIP spectra calculated in this manner, which allows for the determination of shared ”characteristic peaks” which, depending on the magnitude and signal-to-noise ratio of the peak, allows for one to theoretically determine how similar one peptide's biochemical and physiochemical functionality is to another. This cross spectrum is calculated as follows:

$$S(n) = X(n)Y^{*}(n) \subset n = 1, 2 \ldots N/2 \tag{2}$$

This is implemented in our peptide similarity matrix with a manual DFT implementation corresponding to results which cohere to the DFT spectra in the introductory paper and book on the Resonant Recognition Model by Cosic. We initially attempted to implement the DFT using the NumPy and SciPy FFT packages, but struggled to acquire initial results which corresponded to the DFT and cross spectra in the aforementioned paper and book, so the manual implementation was added in with the help of Sid and Aaron.

In the cross spectrum generated by our manual implementation of the DFT outlined by Cosic, the signal-to-noise ratio, calculated by dividing the peak value (whose frequency, should the peak value be distinct enough from its background, corresponds to its *characteristic frequency*) by the mean of all values in the cross spectrum, can be deemed a measure of similarity between the two peptides whose DFT spectra are used to create the cross spectrum. In this manner, we used both the signal-to-noise ratio of the cross spectrum of each neuropeptide to the two known-binding peptides, as well as the correlation coefficient of each neuropeptide's DFT spectrum to the DFT spectrum of the two known-binding peptides as similarity values in the output similarity matrix for each binding peptide.

Sequence-Function Domain Matching

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). We then implemented a function which returned all possible subsequences of each known-binding peptide, and then for each peptide, returned a list of the longest matching subsequences and their indices for each neuropeptide. This was done for explicit amino acid subsequence matches, as well as for subsequence matches indicated by the seven aforementioned bio-functional categories, which, given the limited domain, corresponded to a less discriminatory pattern-matching algorithm for each neuropeptide. Thus, we implemented two columns, one containing a list of amino acid subsequence matches and one for functional encoding matches (as tuples, paired with their index of coincidence), in our output similarity matrix.

We also implemented a primitive scoring algorithm, which assigned a cumulative score of 1 multiplied by each independent subsequence match added to 1.5 multiplied by the length of each subsequence match, which allowed us to assign a scalar magnitude to the subsequence matches found. It is our intention to optimize the weight tuple (instead of using the arbitrary 1 and 1.5 aforementioned weights) used in this scoring method further in the project, and implement more complex scoring based on the weight of each match's functional mapping, some which may carry more information and explanatory power than others.

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(2019)]

Substitution Matrix Similarity

The PAM (point accepted mutation) matrices, initially developed by Margaret Dayhoff 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

Although we implemented several distinct metrics of similarity in our final two output similarity matrices, the initial and primary motivation for the project laid in exploring Irena Cosic's Resonant Recognition Model as a means for determining peptide behavior, especially with regards to self-assembly, and as such, we weighted the signal-to-noise ratio and correlation coefficient metrics garnered from RRM results the highest, and all other similarity metrics are used at the behest of validating and interpreting the RRM results. The sequence-function-domain pattern matching algorithm represents a relatively novel similarity metric, but its full validity cannot be verified without more experimental effort (or a project on its own).

In implementing the Resonant Recognition Model, we first needed to convert the two known-binding peptides GrBP5 and M6 to its bio-functional encoding (using the mapping listed in the Appendix) as well as its EIIP encoding for DFT spectra calculations, used thereafter for cross-spectral calculations alongside each neuropeptide.

Table 1: *Two known-binding peptides under consideration, as well as their bio-functional encoding and scaled EIIP DFT spectrum*

seq	numseq	scaled dft
IMVTESSDYSSY	111252250220	[51.1, 0, 31.2, 21.8, 29.2, 100]
IMVTASSAYDDY	111212210550	[27.3, 84.4, 93.5, 0, 59.1, 100]

Our ultimate goal in the initial phase of this project is to downselect from the full list of neuropeptides to three candidate peptides, as determined principally through the calculation of cross-spectral signal-to-noise as an alias for peptide similarity. To this end, we used our manual RRM implementation to generate the aforementioned metric (column titled RRM SN in output matrices), and sorted all peptides by their signal-noise ratio with the known-binding peptide of interest, using other metrics as secondary benchmarks to judge candidacy. Below is the similarity matrix for the top five sequences, sorted by signal-to-noise ratio (minus distance values and amino acid encodings – PAM30 and BLOSUM45 scores scaled from 0 to 1):

Table 2a: *Top 5 sequences by RRM S/N ratio as similarity to GrBP5*

Sequences	PAM30	BLOSUM45	RRM_SN	RRM_Corr
INNDCQNFIGNR	0.359	0.370	5.46	0.69
VPLRPEEDELID	0.485	0.522	5.42	0.65
LNSLDGAGFGFE	0.612	0.543	5.42	0.89
RSFGCRFGTCTV	0.359	0.196	5.39	0.64
SSGGGDGSGMWF	0.427	0.283	5.27	0.78

Table 2b: *Top 5 sequences by RRM S/N ratio as similarity to M6*

Sequences	PAM30	BLOSUM45	RRM_SN	RRM_Corr
PQNWNKLNSLWG	0.233	0.174	4.76	0.25
SGLMSEGSSLEA	0.398	0.348	4.56	0.52
SKYVSKQKFYSW	0.612	0.630	4.44	0.93
NSLLGIPKVMND	0.427	0.283	4.33	0.71
NSILGLPKVMND	0.456	0.326	4.33	0.71

It was generally observed that RRM cross-spectral signal-to-noise demonstrated low correlation towards most other metrics of similarity (see table 3). Furthermore, in Cosic's introduction to the RRM, a signal-to-noise ratio above 20 was deemed to be notable – however, given that the signal in her examples (human alpha, beta hemoglobin) were presumed to be many magnitudes higher in length (dealing with proteins, rather than short-length peptides), it is expected that the signal-to-noise ratio would be less meaningful in our signals which are relatively protracted in length (on the order of 10s versus 100s).

Table 3: *Correlation matrix of similarity measures for GrBP5, minus string distance*

	PAM30	BLOSUM45	RRM_SN	RRM_Corr	sseq_score	num_score
PAM30	1.0	0.90	0.05	0.01	0.59	0.63
BLOSUM45	0.90	1.0	0.06	0.01	0.64	0.70
RRM_SN	0.05	0.06	1.0	0.52	0.06	0.07
RRM_Corr	0.01	0.01	0.52	1.0	0.05	0.05
sseq_score	0.59	0.64	0.06	0.05	1.0	0.43
num_score	0.63	0.70	0.07	0.05	0.43	1.0

However, given that we were most interested in using RRM metrics alongside sequence-domain pattern matching metrics, we decided to observe common sequence motifs among those peptides who had higher cross-spectral signal-to-noise relative to their counterparts, and found that, while some common patterns existed among those neuropeptides with the highest cross-spectral signal-to-noise with regards to a known-binding peptide, there was not enough information to fully conclude that any particular subsequence motif or functional encoding motif would provide any significant insight.

Table 4a: *Top 5 sequences and subsequence matches to GrBP5 by RRM S/N ratio*

Sequences	RRM_SN	sseq_matches	num_matches
INNDCQNFIGNR	5.46	(’I’, 0)	(’1’, 0), (’22’, 5)
VPLRPEEDELID	5.42	(’D’, 7)	(’1’, 0), (’1’, 2), (’5’, 7)
LNSLDGAGFGFE	5.42		(’1’, 0), (’5’, 4), (’0’, 8)
RSFGCRFGTCTV	5.39		(’2’, 10)
SSGGGDGSGMWF	5.27		(’0’, 11)

Table 4b: *Top 5 sequences and subsequence matches to M6 by RRM S/N ratio*

Sequences	RRM_SN	sseq_matches	num_matches
PQNWNKLNSLWG	4.76		
SGLMSEGSSLEA	4.56		(’1’, 2), (’5’, 10)
SKYVSKQKFYSW	4.44		(’2’, 6), (’0’, 8), (’0’, 11)
NSLLGIPKVMND	4.33		(’1’, 2)
NSILGLPKVMND	4.33		(’1’, 2)

It is readily apparent that in producing DFT and cross spectra for peptides signals so protracted in length relative to the proteins typically used as input for RRM analysis, we would find many superficially similar patterns amongst those scoring highest in cross-spectral signal-to-noise, without any shared function necessarily being an explanatory factor (moreso than mere coincidence). Forgoing a significant analytical deep-dive into the RRM spectra characteristic frequencies as a means of determining specific functionality (as Cosic proposes), in the ultimate aim of selecting three peptides to forward for further experimental analysis for graphene-binding affinity, we simply joined the M6 and GrBP5 similarity matrices and filtered for high signal-to-noise for both binding peptides first, and for respective bio-functional encoding subsequence matches second, to produce a list of peptides which were deemed to be sufficiently similar in both bio-functional encoding pattern matching as well as cross-spectral similarity to both peptides to ultimately determine which three peptides to forward for further experimentation.

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 in a manner relatively more primitive, while still taking into account the cross-spectral signal-to-noise ratio for all neuropeptides with both GrBP5 and M6, as well as the subsequence pattern matches for both the aforementioned known-binding peptides. We considered briefly using the bio-functional encoding pattern matches as an additional criterion to the RRM signal-to-noise instead of subsequence matching, however, we felt that, in absence of any justifiable reason to apply less discrimination in determining only three candidate peptides, we would forgo the option (the resultant top 10 peptides using bio-functional encoding score are listed in the Appendex as Table 2a and Table 2b).

To do this, we simply merged the two similarity matrices, and created a new column which was the product of the average signal-to-noise ratio between each neuropeptide with GrBP5 and M6 (half of the summation of each respective RRM SNR) and the average subsequence match score between both GrBP5 and M6. The merged similarity matrix was then sorted by this new derived column to produce peptides which had high RRM signal-to-noise similarity to both M6 and GrBP5 while also having high numbers subsequence pattern matches for both M6 and GrBP5. From this metric, we found the top 10 neuropeptides to be as follows.

Table 5a: *Top 10 sequences by RRM SN and subsequence match score*

Sequences	RRM SN GrBP5	RRM SN M6	Score GrBP5	Score M6
IGVRKSARKWNN	3.9643732569500365	3.4386167986056866	6	6
LDLTPGSHVDSY	3.737484646877938	1.9401406448554996	7	8
LTLTPGSHVDSY	3.6612996255277976	1.7621098413241925	7	8
FDRISNSAFSDF	4.290469373065538	3.49429975994587	4	5
FDRISSSAFSDF	4.835410960463124	2.398641048540137	3	6
SGDTSSQAKGMW	3.6353709793420013	2.7674157609153496	4	6
QLSEDASKVITY	4.312190671490123	3.673904521769156	4	4
YDRISNSAFSDF	4.116755702909698	2.9028797282910577	4	5
YDRISGSAFSDF	4.125806287602751	2.8817836385812985	4	5
IFLPGSVILRAL	5.073675546565263	2.724680915699252	4	4

Table 5b: *Top 10 sequences by RRM SN and bio-functional encoding match score with specific encoding matches*

Sequences	num_matches_grbp5	num_matches_m6
IGVRKSARKWNN	[(’I’, 0), (’V’, 2), (’S’, 5)]	[(’I’, 0), (’V’, 2), (’S’, 5)]
LDLTPGSHVDSY	[(’T’, 3), (’S’, 6), (’SY’, 10)]	[(’T’, 3), (’S’, 6), (’D’, 9), (’Y’, 11)]
LTLTPGSHVDSY	[(’T’, 3), (’S’, 6), (’SY’, 10)]	[(’T’, 3), (’S’, 6), (’D’, 9), (’Y’, 11)]
FDRISNSAFSDF	[(’S’, 6), (’S’, 9)]	[(’SA’, 6), (’D’, 10)]
FDRISSSAFSDF	[(’SS’, 5)]	[(’SSA’, 5), (’D’, 10)]
SGDTSSQAKGMW	[(’T’, 3), (’S’, 5)]	[(’T’, 3), (’S’, 5), (’A’, 7)]
QLSEDASKVITY	[(’S’, 6), (’Y’, 11)]	[(’S’, 6), (’Y’, 11)]
YDRISNSAFSDF	[(’S’, 6), (’S’, 9)]	[(’SA’, 6), (’D’, 10)]
YDRISGSAFSDF	[(’S’, 6), (’S’, 9)]	[(’SA’, 6), (’D’, 10)]
IFLPGSVILRAL	[(’I’, 0), (’S’, 5)]	[(’I’, 0), (’S’, 5)]

Given more time to prepare the code for generating similarity based on subsequence matching as a supplement to RRM signal-to-noise calculations, it would perhaps have been beneficial to implement a more nuanced subsequence scoring metric, which takes into account locations of matches and scores each match uniquely depending on the index of matching, in the case of certain peptide locations influencing the mechanism of binding moreso than others.

After forwarding the candidate peptides for experimentation, we plan to utilize 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 matrices for 1,612 same-length neuropeptides: one matrix for similarity values relative to GrBP5, and one for the wild-type peptide M6.

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 (including those of different length to the input known-binding peptides), 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

Table 2a: *Top 10 sequences by RRM SN and bio-functional encoding match*

score

Sequences	RRM SN GrBP5	RRM SN M6	Score GrBP5	Score M6
LFADVSTIGDFF	3.90	2.12	9	13
LNSLDGQGFGFE	5.16	4.07	8	6
ALNSLDGAGFGF	4.74	3.72	7	8
VVLGKKQRFHSW	3.31	2.73	11	10
ALNSLDGNGFGF	4.69	3.72	7	8
ALNSLDGQGFGF	4.72	3.67	7	8
AGNSGANSGMWF	4.88	2.89	8	8
AGGTGANSAMWF	4.91	2.78	8	8
ALADEHNDNFLR	5.04	3.11	9	6
LADEHNDNFLRF	4.01	2.73	9	9

Table 2b: *Top 10 sequences by RRM SN and bio-functional encoding match score with specific encoding matches*

Sequences	num_matches_grbp5	num_matches_m6
LFADVSTIGDFF	(’1’, 0), (’1’, 2), (’22’, 5), (’0’, 11)	(’1’, 0), (’1’, 2), (’1221’, 4), (’5’, 9), (’0’, 11)
LNSLDGQGFGFE	(’1’, 0), (’5’, 4), (’2’, 6), (’0’, 8)	(’1’, 0), (’2’, 6), (’0’, 8)
ALNSLDGAGFGF	(’11’, 0), (’2’, 3), (’0’, 11)	(’11’, 0), (’21’, 3), (’0’, 11)
VVLGKKQRFHSW	(’111’, 0), (’2’, 6), (’0’, 8), (’20’, 10)	(’111’, 0), (’2’, 6), (’0’, 8), (’0’, 11)
ALNSLDGNGFGF	(’11’, 0), (’2’, 3), (’0’, 11)	(’11’, 0), (’21’, 3), (’0’, 11)
ALNSLDGQGFGF	(’11’, 0), (’2’, 3), (’0’, 11)	(’11’, 0), (’21’, 3), (’0’, 11)
AGNSGANSGMWF	(’1’, 0), (’2’, 3), (’2’, 6), (’0’, 11)	(’1’, 0), (’2’, 3), (’2’, 6), (’0’, 11)
AGGTGANSAMWF	(’1’, 0), (’2’, 3), (’2’, 6), (’0’, 11)	(’1’, 0), (’2’, 3), (’2’, 6), (’0’, 11)
ALADEHNDNFLR	(’111’, 0), (’5’, 4), (’25’, 6)	(’111’, 0), (’2’, 6)
LADEHNDNFLRF	(’11’, 0), (’2’, 5), (’0’, 8), (’0’, 11)	(’11’, 0), (’2’, 5), (’0’, 8), (’0’, 11)

Bibliography

References

[omnium(2019)] omnium. textdistance, October 2019. URL https://pypi.org/project/textdistance/.