Data-Driven Design of Spontaneously-Organized Super-Peptides on Atomic Single Layer Solids

Swapil Paliwal

A thesis submitted in partial fulfillment of

the requirements for the degree of

Master of Science

(Materials Science and Engineering)

University of Washington

2017

Committee:

Mehmet Sarikaya

Hanson Fong

Sami Dogan

Program Authorized to Offer Degree:

Department of Materials Science and Engineering

University of Washington

Abstract

Data-Driven Design of Spontaneously-Organized Super-Peptides on Atomic Single Layer Solids

Swapil Paliwal

Chair of the Supervisory Committee:

Professor Mehmet Sarikaya

Department of Materials Science and Engineering

Rational design and analysis of protein databanks *via* data-driven algorithms have significantly accelerated drug discovery, in particular, and a wide range of biological research topics, in general, during last decades. A similar approach is gaining momentum in materials research but has garnered limited attention in areas such as the design of soft interfaces formed by solid-binding peptides at solid materials interfaces. The GEMSEC Laboratory (Genetically-Engineered Materials Science and Engineering Center) has been working towards expanding this strategy in materials research *via* the development of peptide-based bioelectronic interfaces incorporating solid-binding peptides and single layer materials and, thereby, bridge biology to solid-state devices such as graphene field-effect transistors. We are presented with a challenge in peptide-based materials design as, in general, a vast store of relevant data is not available in materials science that is similar to protein databanks that are available in fields such as molecular biology. Thus, there is need for a knowledge-base, but that requires decades of research to draw on. In the present research, this was accounted by utilizing an innovative integration of combinatorial selection of solid-binding peptides, their rational design and bioinformatics based approach to model specific peptide-material interactions.

From a data-base of 10s if not hundreds of peptides selected by this approach, the basis of the present method is to generate libraries of materials specific super-peptides that can attach, assemble and perform specific functions on atomically-flat material surfaces. As solid-state systems, single atomic layer materials, such as graphene and those that provide flat surfaces, such as quartz, have been chosen. Using these libraries, peptides that are capable of binding to their counterpart solid material of interest can be identified by performing combinatorial selection based on phage display approach. Typically, 50+ individual peptides are selected from of an original pool of ~10^{15} variants, which are then classified based on their binding strength using, e.g., fluorescent microscopy.

Needleman-Wunsch based similarity analysis and machine learning algorithms are then used to create a scoring matrix capable of identifying robust and weak binders for the particular material amongst millions of random permutations of amino acid sequences in the peptides. The most powerful of these binders are fed into a decision-tree based rational design consisting of selection rules on hydropathicity, iconicity, aromaticity, and polarity of peptides identified to be capable of self-assembly from the previously conduted experiments. This process filters peptides and identifies those that are capable of strongly binding to as well as readily assembling on the atomically flat solid crystals. These model-based designed peptide sequences are then chemically synthesized and subsequently evaluated experimentally in terms of their binding and assembly characteristics using, e.g., atomic force microscopy to validate the success of the predictive model. As the experimental data become available in the assembly of the peptides under specific experimental parameters that are related to the particular chemistry of the sequences, the approach progressively creates a better outcome. Consequently, the model upon each experimental validation is further improvised and provides further knowledge and supply related sequences to the library to advance peptide-guided functional solid-state materials for practical nanotechnology and nanomedicine applications.

Acknowledgements

I would like to express sincere thanks to Prof. Mehmet Sarikaya for his excellent supervision, guidance, and support during this master's project. His calm attitude and faith in my abilities have been a great confidence booster. He has been great a source of inspiration, and thought-provoking ideas and this work would have been an impossible task without his help and support.

Also, I would also like to thank Dr. Jevin West, Dr. Joshua Blumenstock, Professor Greg Hay and Professor Joel Smith from the Information school for introducing me to the exciting world of data science. They did a wonderful job very amicably and appealingly, and taught me some interesting tools and techniques in machine learning and data science that have made me feel confident of taking this research forward. I am also very thankful to Dr. David Beck from the e-Science Institute for being a great teacher and mentor, and especially for taking out time for my projects during his data science office hours.

I am also thankful to all my awesome lab mates and advisors in Sarikaya Labs, especially Dr. Sefa Dag, Deniz Yucesoy, Dr. Hanson Fong, Carolyn Gresswell, Sanaz Sadt, Richard Lee and David Starkebaum for their input and many great scientific conversations. They have provided a ton of help during my initial learning phase, and for assisting me understand experimental aspects of peptide design. I would also like to thank Tanmay Modak, Elton Dias, Blake Hough, Christopher Fu, I-Hsuan Huang and Melissa Gaughan for being amazing groupmates in data science projects and helping me learn some critical analytical and teamwork skills.

Special thanks also to Ms. Karen Wetterhahn, for being a very helpful academic advisor in the Materials Science and Engineering department and helping me stay on schedule with my degree program. Finally, I would like to thank all my friends, colleagues and well-wishers for all the suggestions and support. Last, but not least, I would like to thank my parents for their support and patience, and believing in me.

List of Figures

- **Figure 15:** (a) amino acid to number conversion key for faster processing of data in algorithms. (b) Splitting the dataset into training and test datasets containing 50 peptides in training set and 10 in the test set.

- **Figure 16:** Peptides that were randomly chosen to be a part of test data set

- **Figure 17:** 3D representation of the effect of the number of strong binders on average $TSS_{SS\text{-}SW}$ (hidden axis is Gop/Gep).

- **Figure 18:** 2D representation of Effect of the number of strong binders on average $TSS_{SS\text{-}SW}$.

- **Figure 19:** 3D representation of the effect of gap opening penalty (gop) on $TSS_{SS\text{-}SW}$ (hidden axis is strong binders).

- **Figure 20:** Effect of ratio of strong binders to weak binders on $TSS_{SS\text{-}SW}$ scores.(color and size represent Weak binders).

- **Figure 21:** Similarity Scores for training set based on optimized parameters for the BLOSUM62 scoring matrix.

- **Figure 22:** Results of BLOSUM62 based scoring model's classification on the test dataset.

- **Figure 23:** Similarity score calculation by using BLOSUM62 Matrix with all combinatorially selected peptides as input

- **Figure 24:** BLOSUM62 and 1st perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1, max($TSS_{SS\text{-}SW}$) = 0.675.

- **Figure 25:** BLOSUM62 and 158th perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 158, max($TSS_{SS\text{-}SW}$) = 4.54.

- **Figure 26:** BLOSUM62 and 534th perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1000, max($TSS_{SS\text{-}SW}$) = 14.23.

- **Figure 27:** PAM250 and 450th perturbation to PAM250 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1000, max($TSS_{SS\text{-}SW}$) = 10.809.

- **Figure 28:** Similarity Scores for training set based on optimized parameters for 534th perturbation on BlOSUM62 derived scoring matrix with Gep/Gop of 0.8/8 units.

- **Figure 29:** Results of Newly generated matrix from 534th perturbation of BLOSUM62 based scoring model's classification on the test dataset.

- **Figure 30:** Similarity scores of all combinatorially selected peptides based on Newly generated BLOSUM62's 534th perturbation matrix with gop and gep values of 8 and 0.8 and ten strong, 26 moderate and 24 weak binders.

- **Figure 31:** Predictions on 100,000 randomly generated peptides binding properties based on the newly generated scoring matrix. (a) top 20 strongest binders (b) top 20 weakest binders and (c) score distribution of 100,000 random peptides.

- **Figure 32:** Alternative methods that can supplement the scoring matrix based classification of peptides.

- **Figure 33:** list of available peptides that have been studied for self-assembly behavior on graphite.

- **Figure 34:** Domains of a GrBP5-WT peptide identified by So. et al. which impact self-assembly pattern of GrBP5.

- **Figure 35:** Effects of substitutions changing the amphiphilic domains (regions I and II) in GrBP5 and mutants.

- **Figure 36:** Decision tree and selection rule based rational design to classify self-assembling capabilities of peptides on graphite surface

- **Figure 37:** New peptides predicted to be capable of self-assembly out of 100000 random peptides

- **Figure 38:** Schematic diagram indicating a potential DNA Reader Construct using graphene-based super-peptide

- **Figure 39:** Schematic diagram for periodic table type demonstration of solid-binding peptides.

- **Figure 40:** Experiments based on graphene, peptides & neurons to make a biocompatible bio-electronic interface. A) Schematic showing the peptide self-organization with a series of the surface processes: binding, diffusion, and self-organization. B) AFM image of the self-assembling peptide on single-layer graphene. The bright lines are self-assembled peptides forming organized nanostructures. C) An optical image of neurons cultured on the border of graphene (left) and TCPS (right). D) scanning electron microscopy image of neurons on graphene, E) MTT-measured viability of neurons cultured on TCPS and graphene after seven days, F) LDH activity of neurons after seven days incubation on TCPS and graphene

- **Figure 41:** Proposed components of the GEMSEC GEPI and Bio-mimetic intelligence suite for Super peptide design and analysis of material peptide interactions.

Contents

Chapter 1 – Introduction: Data Driven Design of Peptides

1.1 Introduction to Solid-Biding Peptide Design

Peptides are a class of naturally occurring macromolecules which of short chains of amino acids linked by amide bonds. These are chemically and functionally very similar to their larger counterparts called proteins. Both peptides and proteins are essential components of cells and responsible for carrying out vital biological functions in organisms.[i] The primary distinguishing factors separating the two is their size and conformational structure. Peptides typically consist of less than 50 amino acids, whereas proteins are made up of 50 or more amino acids. Also, peptides typically have less defined structure than proteins, which can adopt pretty complex secondary, tertiary, and quaternary structures. Peptides, which have few amino acids (e.g., 2 to 20) are called oligopeptides. in the current study, our primary focus is on these oligopeptides typically consisting of 10-13 Amino acids.

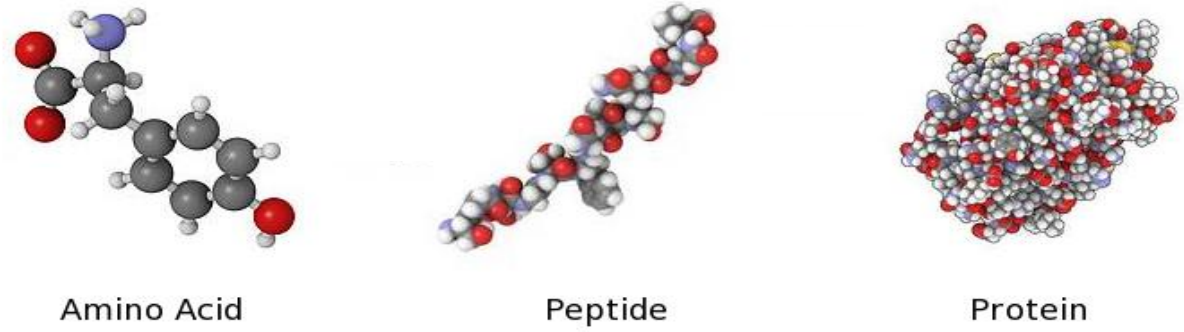


Figure 1: Ball and stick model are showcasing differences between amino acids, peptides, and protein. (Adapted from www.peptidesciences.com/glossary/peptides-vs-proteins[ii])

There are typically three main methods in peptide engineering namely, combinatorial selection, rational design and data-driven design.[iii] All three have their unique advantages and challenges as mentioned below. In the present study, we employ some aspects of all three methods to engineer novel single layer material binding super-peptides.

Combinatorial mutagenesis is reiterative, in vitro method used to find the preferred material–binding peptides from a variety of randomized sequences[iv]. Phage display and Cell surface display are two commonly employed methods in the combinatorial selection of peptides. Phage display uses randomized sequences from a bacteriophage genome and allows for the expression of a variety of viral coat fusion proteins[iv]. These proteins are used in the selection of peptides that bind to a material. In cell surface display random peptides are expressed on the surface of a bacterial cell or its flagellum and are directly accessible for the substrate or binding partner in binding studies. Figure 2 below shows a typical combinatorial selection cycle for both kinds of systems.

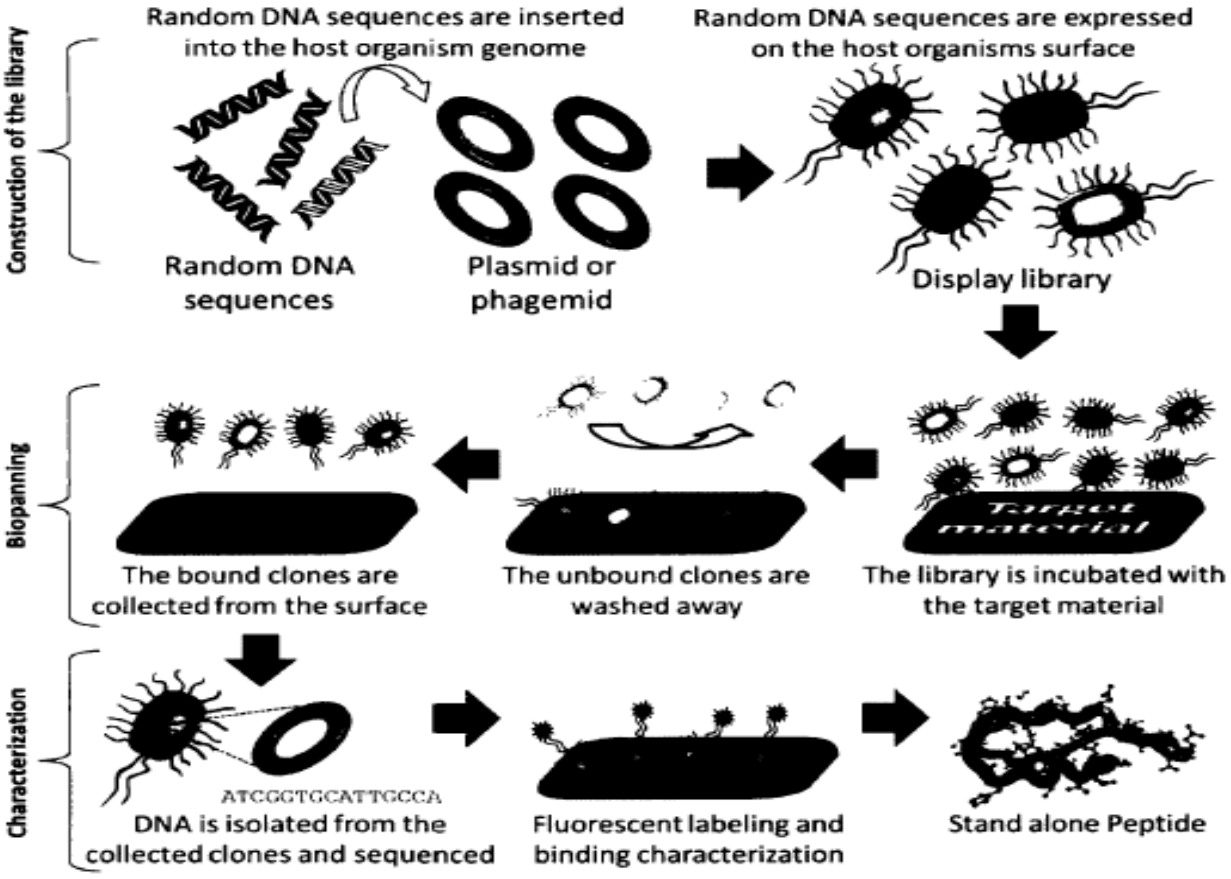
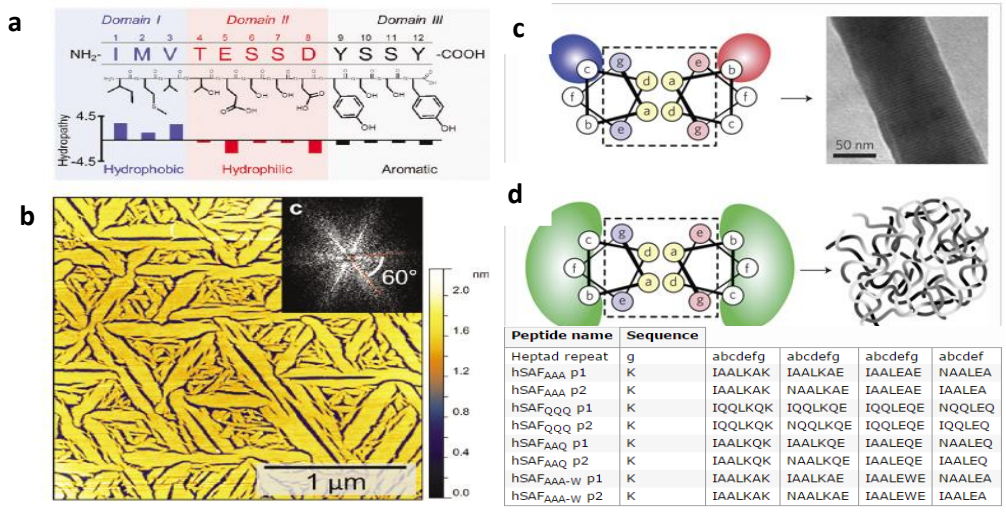


Figure 2: Schematic description of Combinatorial selection using cell surface display technologies (Adapted from the work of Gungormus, M. (2012)[v])

Rational Design is the strategy to generate new molecules with a certain function in mind. The design is based on the ability to predict how the molecule's structure and chemical properties will affect its behavior. The evidence is gathered through physical or molecular models and thorough experimentation. The design rules are either derived by copying nature (like mimicking amino acids in an α-helix or a β-sheet) or via exploration of new properties obtained through careful mutations of amino acids based on peptide derivatives (peptide amphiphiles, π-stacking systems). Figure 3a and 3b showcase how a rational design containing peptide amphiphiles with three separate domains of sequential hydrophobic, hydrophilic and aromatic amino acids can be a choice to make peptides capable of assembling onto a graphite surface as demonstrated by So *et. al.* in 2016 [vi]. Similarly figure 3c and 3d is an example of how design rules can be used for the creation of hydrogelating self-assembling fibers (hSAF's) based on a two-component system of coiled-coil heptads, of type abcdefg[vii]. Here, a, d, e and g positions are responsible for directing the dimer interface while the b, c and f positions, are exposed on the surfaces of the coiled-coil assemblies. Strong specific charged interactions between some b and c positions can lead to peptide alignment and fiber thickening. Whereas, weaker and more general interactions at all b, c and f sites, can result in smaller, more flexible, bundles of thinner fibers.



Peptide name	Sequence				
Heptad repeat	g	abcdefg	abcdefg	abcdefg	abcdef
$hSAF_{AAA}$ p1	K	IAALKAK	IAALKAE	IAALEAE	NAALEA
$hSAF_{AAA}$ p2	K	IAALKAK	NAALKAE	IAALEAE	IAALEA
$hSAF_{QQQ}$ p1	K	IQQLKQK	IQQLKQE	IQQLEQE	NQQLEQ
$hSAF_{QQQ}$ p2	K	IQQLKQK	NQQLKQE	IQQLEQE	IQQLEQ
$hSAF_{AAQ}$ p1	K	IAALKQK	IAALKQE	IAALEQE	NAALEQ
$hSAF_{AAQ}$ p2	K	IAALKQK	NAALKQE	IAALEQE	IAALEQ
$hSAF_{AAA\text{-}W}$ p1	K	IAALKAK	IAALKAE	IAALEWE	NAALEA
$hSAF_{AAA\text{-}W}$ p2	K	IAALKAK	NAALKAE	IAALEWE	IAALEA

Figure 3: 3a and 3b showcase self-assembly of a graphite binding peptide following an order of hydrophobic, hydrophilic and aromatic type amino acids in the sequence. 3c and 3d showcase effects of specific vs. non-specific interactions amongst b, c and f units in a peptide dimerization unit (a, d, e & g) containing heptapeptides. 3a,3b are Adapted from work by So *et. al.*[vi] and 3c, 3d are Adapted from work by Banwell *et.al.*[vii].

Data Driven Design of peptides approach uses computational modeling, bioinformatics algorithms and other available knowledge on specific peptide and materials interactions to limit library size. This approach allows the identification of variable substitutions that may lead to large effects but which might get had been discounted in combinatorial selection or rational design. A specific limitation of this approach is that it requires a large and increasing databank of experiments on functional variants, to be able to identify specific amino acid residues to vary to increase the chances of obtaining a variant molecule with the desired properties. An example of such a study is demonstrated by Oren *et. al.* in their experiments to design new quartz-binding peptides.[viii] They obtained around 60 quartz-binding peptides via a combinatorial selection approach and then created bioinformatics based data models called similarity matrices to characterize the similarities amongst better binders and their differences from weak binders. They trained the models on different scoring matrices like BLOSUM62 and PAM250 and used these models to predict some newly designed reliable quartz-binding peptides from amongst millions of variants using similarity scores to powerful binders. Figure 4a showcases the similarity scores, binding strength, and prediction of the binding strength of various peptides. Figure 4b showcases how models scored different peptides classified as strong(S), Moderate(M) and weak(W) based on binding strength data. Figure 4c shows some predictions on newly classified peptides from a million random variants.

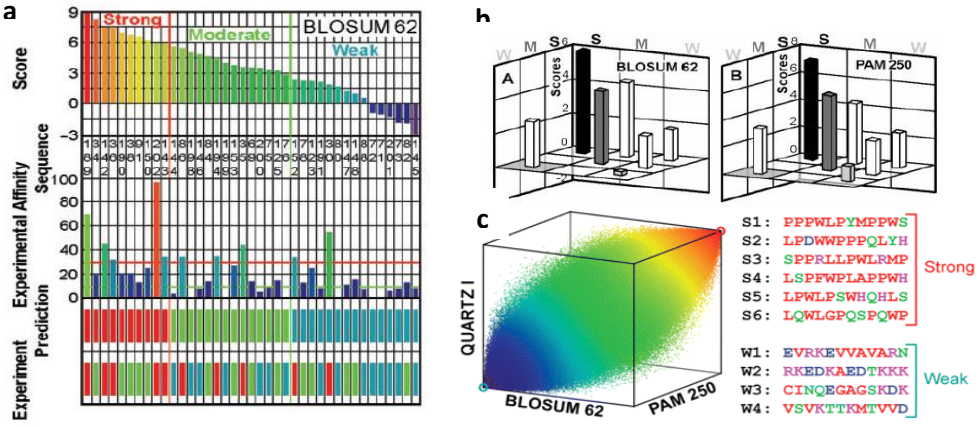


Figure 4: A bioinformatics based data-driven model to design quartz-binding peptides demonstrated by Oren *et. al.* (Adapted from work by Oren *et. al.* [viii].)

1.2 Motivation and Justification

The motivation of this research to develop a library of solid state material binding functional peptides stems from an article titled “Molecular Biomimetics: nanotechnology through biology” published in Nature journal in 2003 by Sarikaya et.al.[ix] In this paper, the authors described how we could learn from a diverse phenomenon involving proteins in mother nature and apply that to develop modular materials for applications in materials technology. The authors talk about how in biology, Proteins containing very long amino acid sequences are specialized to guide solid-state materials formation making up hard and soft tissues such as ones found in bones, teeth, magnetotactic bacteria or sea shells. Proteins can do this guided mineralization because of their unique and specific interactions with other macromolecules and inorganics. This feat is usually achieved by a combination of multiple subdomains of proteins which work together in tandem. A part of protein sticks to the site where mineralization occurs, second part attracts catalysts and reactant ions, and another part acts as action site for mineralization. Similarly, in materials technology solid-binding peptides, which can be one of a modular subdomain in protein or which are artificially designable, can guide the formation &assembly of solid-state materials & devices. Bioinformatics, a field involving studies of biomolecules like proteins has greatly benefited through a data savvy and rational approaches in protein engineering in last three or four decades and is a decently mature field now. We wish to expand a similar approach now to design novel materials like ones that can potentially integrate biology with electronics. The integration of biology and electronics is achievable through studies of material biomolecule interfaces like self-assembling peptides on solid materials.

1.3 Brief History

Proteins are a highly diverse and biologically very significant category of molecules which are an absolute necessity for the existence of life. They are known to have unique and specific interactions with other macromolecules and inorganics, and thus they can control structures and functions of biological hard and soft tissues in organisms[x]. Based on the observation that proteins can control materials behavior a sub-field in molecular biomimetics started to emerge wherein people started looking at material specific peptide interactions and the impacts of peptides on material properties. An important first milestone towards gaining more information about this new sub-field to was to identify ways in which peptide material interaction can be studied exclusively and in a scalable manner. The introduction of combinatorial selection method via phage display technique in 1985 by Smith *et. al.* paved the way for the creation of peptide libraries based on binding affinities for specific materials.[xi] A few decades later in 2003, Sarikaya *et. al.* highlighted the importance of combinatorial methods for selection of peptides and how genetically engineered proteins for inorganics (GEPIs) can be used in the assembly of functional nanostructures [ix]. They also raised important questions about how data-driven methods or more intelligent designs can further exploit the potential of peptide driven materials design which forms a premise of this current study. A few years later in 2007, Oren *et. al.* showcased that by applying data-driven classification model's like bioinformatics based similarity scoring techniques to experimentally characterized quartz-binding peptides they could design new sequences with specific affinities towards quartz computationally[viii]. Few more years later in 2012, first practical application of such techniques emerged when Gungormus *et. al.* used similar bioinformatics based scoring matrix models for hydroxyapatite in addition to molecular modeling of amelogenin protein [v]. They were able to identify regions within amelogenin that were shared with a set of hydroxyapatite-binding peptides (HABPs) previously selected by phage display. They also, found that a 22-amino acid long peptide region within

amelogenin could facilitate the formation of a hydroxyapatite layer on demineralized human dentin even without the presence of cells or actual protein.

Another study in 2012, by So *et. al.* demonstrated that short peptides selected by combinatorial selection could not only bind on graphite surface but also self-assemble by forming long-range-ordered biomolecular nanostructures [vi]. They identified three amino acid domains that could steer ordering behavior in self-assembled peptides on graphite. They further applied some simple mutations and could fine tune interfacial processes like initial binding, surface aggregation, growth kinetics, and intermolecular interactions. Their study thus provided a proof of concept on how rational design can be a compelling technique to design new self-assembled peptide interfaces on materials like graphite.

Lastly, In 2016, Hayamizu *et. al.* demonstrated that some combinatorial-selection derived peptides and their synthetic variants can self-assemble into peptide nanowires on two-dimensional nanosheets, single-layer graphene, and MoS_2 [xii]. Their observations and experiments have now paved a path for novel device applications, especially for bioelectronic interfaces.

All these studies and articles provided great insights on how peptide design methods such as combinatorial selection can be combined with either rational design or data-driven design to device different material specific peptide interfaces with useful applications in healthcare and devices.

In the present study, we will use these learnings and combine all three techniques namely combinatorial selection, rational design, and data-driven design to generate a peptide material library for single layer materials for bioelectronic or other applications.

1.4 Potential Applications

Peptides have a strong potential to be the building blocks of novel materials with tailorable properties as can be seen via numerous ways proteins impact biology of living organisms. Peptides by their various

sequences, mutations, structures, and specific interaction sites can enrich materials by assisting in the production of complex shapes, scaffolds, and tunable chemical properties. Through data-driven modeling and rational design, we can generate a library of peptides that interact favorably with certain materials as shown in a periodic table of GEPI's in figure 5 below and create a knowledge base on numerous unexplored peptide material interactions while gaining a fundamental understanding of how the peptides properties can impact material properties. Data-driven design and rational design can also help in the creation of multifunctional materials using chimeric peptides as a molecular glue thus allowing us to make self-assembled molecular circuits. These material peptide interactions can be exploited to create novel sensors since peptides will have specific interactions with materials. Device integration, bioelectronics, and antibacterial coatings can be some other use cases besides healthcare once we start application of these newly designed solid-state materials.

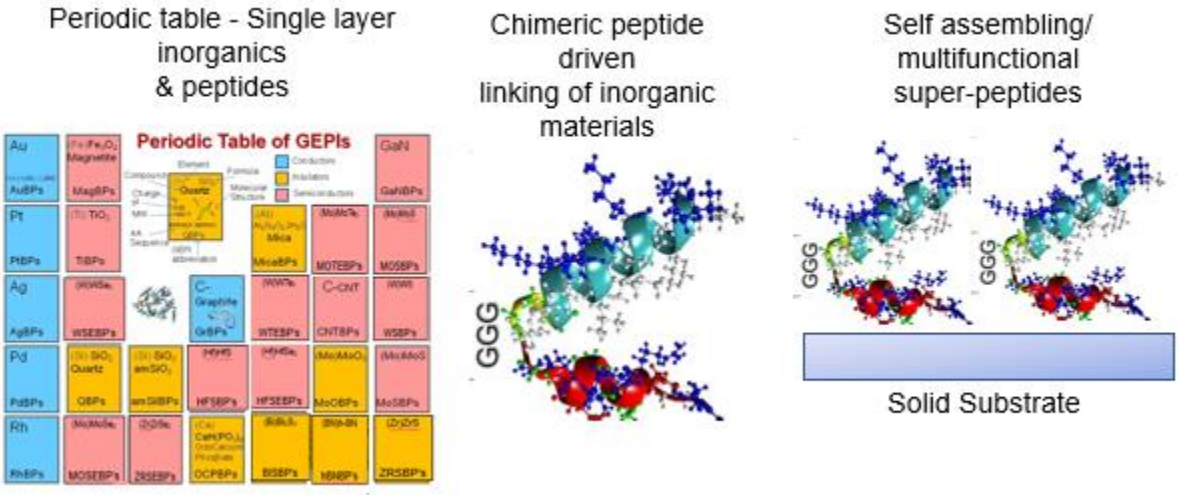


Figure 5: Potential applications of data-driven and rational design includes (a) database of materials specific peptides and their properties, (b) chimeric peptides and (c) antibacterial coatings.
. (Figures: Courtesy of M. Sarikaya, U. Washington, 2017)

1.5 Overview of Current Research

The goal of current research is to apply combinatorial selection, data-driven design, and self- assembly based rational design methods to design a library of single layer solid state binding super-peptides that can assemble on materials of choice and perform specialized functions.

Figure 6 below shows a typical schematic on how all things can be combined sequentially to arrive at a library of super peptides visualized as a periodic table of solid-binding peptides or generate multifunctional moieties via physical peptide material interfaces such as antibacterial coatings on a solid surface.

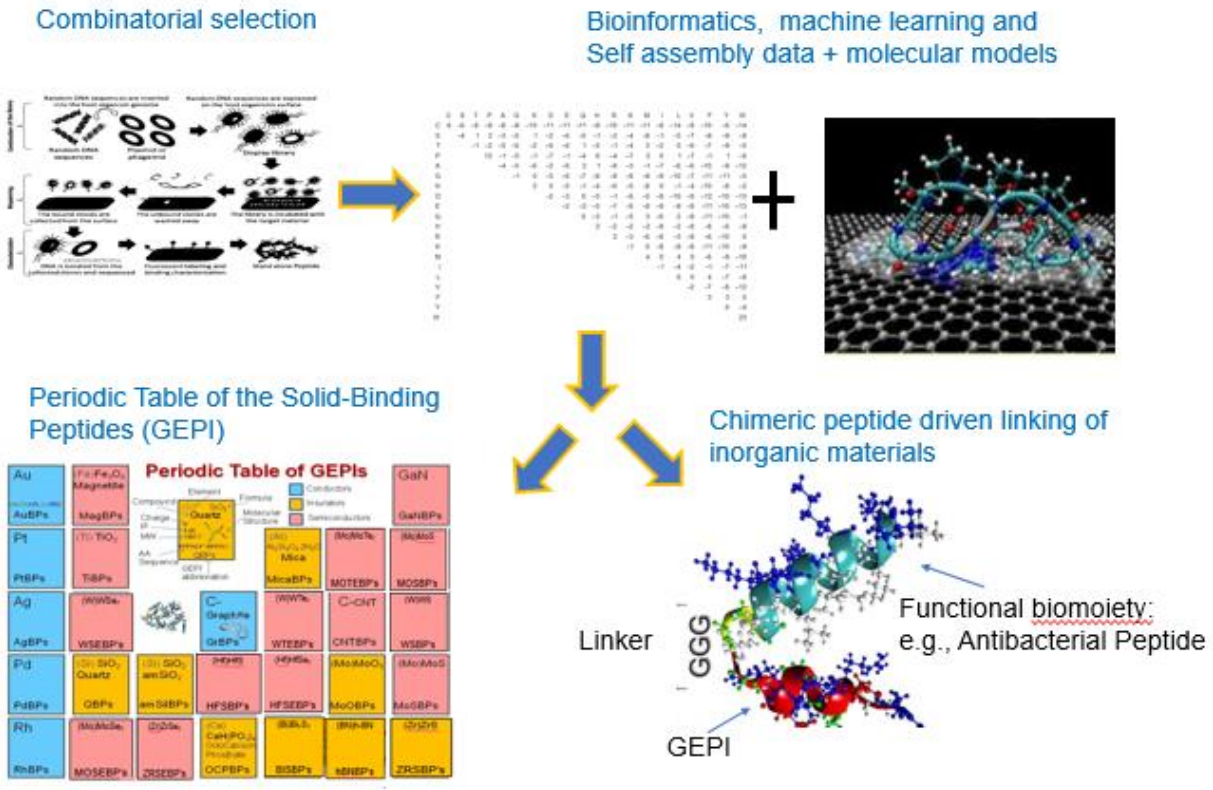


Figure 6: Schematic diagram showcasing usage of combinatorial selection, data-driven design and rational design to generate functional solid peptide interfaces and library of material specific super-peptides (Courtesy of M. Sarikaya Lab, Univ of Washington, 2017)

Figure 7 below shows a more detailed schematic diagram on how data-driven approach and rational design can be combined. As seen in figure 6 we start with a combinatorial selection of peptides through phage display experiments and generate a classification model using quantified binding strength data on some 50-100 peptides. We then apply bioinformatics and machine learning algorithms to model the characteristics of strong binding peptides. Thus, we can at arrive at optimal features of the classification model and generate a scoring matrix. The scoring matrix-based model is depicted on the top left of the dashboard style image in figure 6. We then use this optimized model from binding characteristics data to predict new peptides that are capable of binding strongly with the material. We then create a rational design based on studies of self-assembling peptides. We typically identify features such as amphiphilicity, subunit lengths, hydropathy, acidity, and aromaticity and create selection rules for peptides capable of self-assembly. We pass the binding capable peptides into this rational model and select prospective super peptides on which actual self-assembly experiments are done to validate the model.

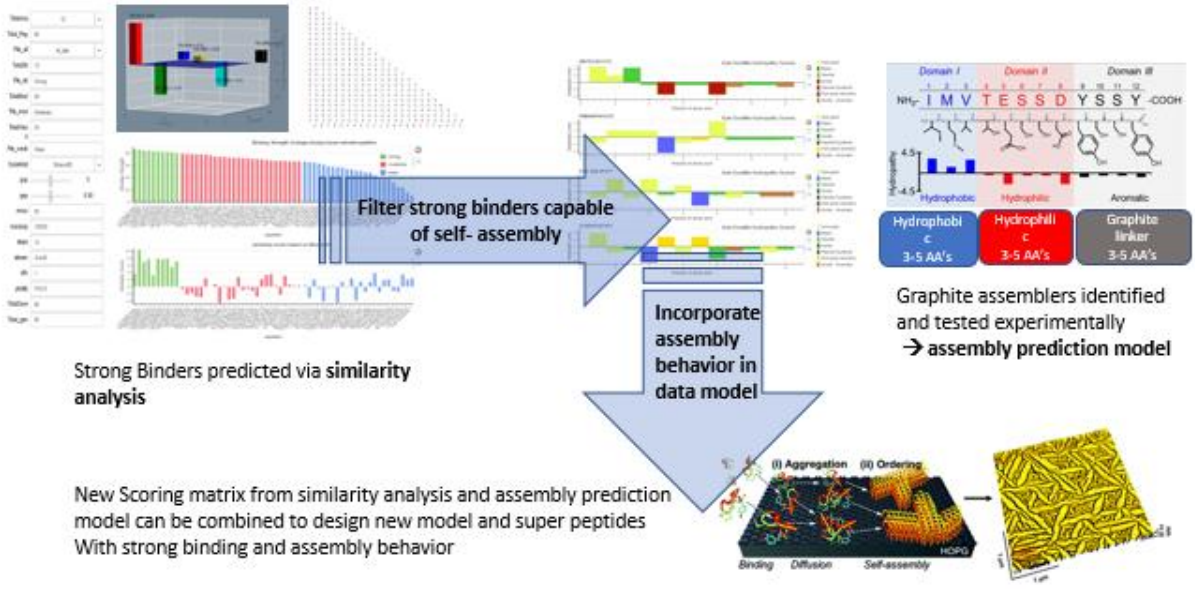


Figure 7: Detailed schematic diagram showing a combination of data-driven model and rational design to generate super-peptides.

Chapter 2 - Overview of Software and Analytical Tools

2.1 Software Design Introduction

The overall aim of the project involving data-driven and rational design of single layer material based super-peptides is to create a scalable system that enables researchers to apply bioinformatics models, machine learning algorithms and molecular modeling approaches to generate knowledge libraries about useful peptides and their interactions with a variety of materials. This system should be able to store and process information about millions and billions of potential peptide molecules of interest and deliver substantial analytics and reports about their behavior within hours or minutes of processing time. An end user shall be able to change just few model parameters preferably in an interactive manner via a dashboard and get statistical and graphical results about useful models and peptides of interest based on desired materials interface properties. We are thus exploring a combination of tools involving C++, Python and R programming languages, SQL Server database, Knime pipeline development tools and Tableau visualization software in current research. The combination of these tools offers unique features to achieve the purpose, and most necessary features are either supported by a large open source community or the tools are provided a strong professional support for data science and machine learning applications. Also, they can all be easily integrated into more scalable and complex environments like cheminformatics toolkits and molecular modeling parallel software with ease.

2.2 Software Details

The following paragraphs describe why we intend to use of various tools and how they can be made to work in sync to achieve the aims of the current research.

Figure below showcases a typical workflow on these tools are being used in conjunction with each other to create a system for data-driven peptide design.

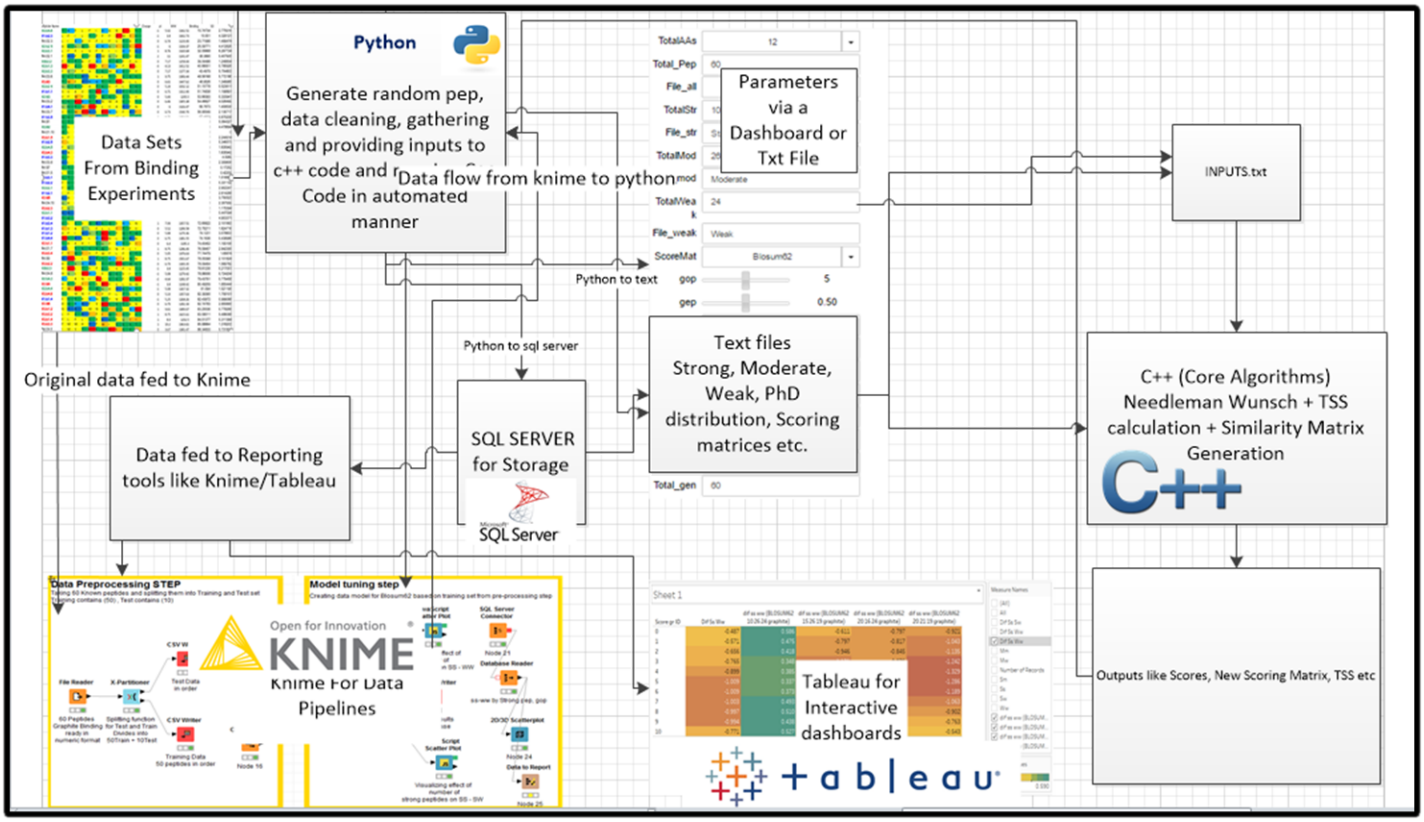


Figure 8: Data workflow is showing interactions and key processes taking place at various components of the system involving Python, C++, Knime, SQL Server, and Tableau.

In figure 8 datasets are being fed from excel file into Python and Knime as shown in the top left corner. Knime is used to split data into training, and test sets and these are fed into python for subsequent processing. In python, a dashboard for data inputs is created wherein people can manually update parameters to be fed into C++ code containing Needleman-Wunsch dynamic programming algorithm and Similarity Matrix generation code. The parameters are usually manipulated in python code blocks and sent into C++ where calculations happen, and the output is sent back to python for further processing. From python data is also forwarded to SQL server via a JDBC connection to store information from the

similarity analysis and this data is also sent to knime and tableau through either the SQL server interface for direct analytics or through Python for real time data manipulation and analysis.

2.2.1 C++ as Choice for Core Algorithms

C++ is often more complex regarding syntax and can pose to be very challenging as a prototyping language but it is often 10-100 times faster in code execution due to its efficient utilization of memory management, and it involves compiled form of code execution as opposed to interpreted manner in python[xiii, xiv]. Since we are dealing with a minimum of 100,000 to million peptides in our study and want the algorithms and code to scale well, C++ is a smart choice for core algorithms like Dynamic programming for similarity analysis and to generate scoring matrix based on similarity analysis.

Moreover, previously published results involving similar algorithms from Dr. Sarikaya's group on systems like quartz had been written in C++[viii]. Thus usage of C++ for this purpose allows for reuse of verified code. This reuse of code also allows for verification of modeling results on a system like quartz-binding peptides that has already been published.

2.2.2 Python and R for Basic Data Science, Visualization, and Automation

Python is an excellent general-purpose scripting language ideally suited for prototyping and automation of tasks. It is an interpreted language and has an emphasis on code readability. It follows a syntax that allows users to express complex concepts in very few lines of code in comparison to languages such as C++ or Java.[xiv, xv] The interpreted nature of this language while allowing for easy software development is also causing its slowness in term of performance in demanding applications. Some of this slowness can be overcome with the use of libraries like Numpy and pandas for scientific computing applications such as data driven In our case. Other advantages of using Python for this project is it contains a variety of libraries like Numpy, Pandas, SciKitlearn, stats models, bokeh, and matplotlib for general purpose data science applications[xvi]. There are also many useful libraries like biopython, pymol,

pychem, modeler and chimera that support bioinformatics, molecular modeling and data visualization which are highly suitable for peptides based materials research[xvii,xviii] .

Python has great interfaces and connectivity with databases like SQL Server and MongoDB that can be leveraged to write useful experimental data directly into the database and which can be used later for advanced analytics. Python can also be integrated with data pipelining tools such as Knime with ease, and this combination can be utilized for repeated data processing tasks via an interactive application much more easily. Libraries like Bokeh can help create a user dashboard, and interactive visualization interface that can be hosted on the internet and non-programmers can use them for submitting data and for analysis based on software generated through the data-driven design of peptides project.

In the current project, Python is being implemented as the main workhorse. It is used to write driver code that allows users to provide inputs like peptides to be used, the parameters to be passed into Needleman wunch algorithms and perturbations to be used in similarity matrix generation algorithm and then call in C++ based core algorithms. The C++ generates output in the form of text files which are again processed using pandas library of Python and important features are then moved to SQL Server for further analytics and storage. Some visualizations are also created using bokeh and matplotlib libraries.

R is also one of the leading programming languages for statistical analysis and consists of highly relevant tools for bioinformatics, statistics, data analysis, machine learning and has the best graphical visualization capabilities. Some packages that are very relevant to our studies are Bioconductor a great library for bioinformatics, Shiny which is a tool for interactive data visualization, data.table, ggplot2, plyr, dplyr, and caret which all are excellent tools for data manipulation and machine learning[xix]. R is not the best language for production level software in data science yet, but recently, Microsoft Corporation has been developing its version of R Tools in conjunction with Revolution Analytics which will allow scalable usage of R for data science software in very near future[xx]. It has however been used as an

excellent choice of language to work alongside SQL server, which is another product from Microsoft. R like Python allows easy integrating with other languages (C/C++, Java, Python) and pipelining tools like Knime and also enables interacting with many data sources and databases like SQL server without much hassle. So, R is a great tool to be used in conjunction with Python for data analytics, visualization, and bioinformatics and offers nice complementary features that might be lacking in Python.

Current project utilizes more of Python libraries like pandas, bokeh, matplotlib, pyodbc, and knime at the current stage but once the data gets bigger and messier a large scale usage of tools like R, shiny and tableau can be projected.

2.2.3 Microsoft SQL Server as a Choice of Database

While pursuing data-driven design project, in combination with combinatorial peptide selection, and rational design, we anticipate the generation of large volumes of useful information which we will need to store and serve on demand to get the most out of our modeling approaches. We will be studying millions and billions of peptides and need a place to store the information on which out of these can be useful and which are not used based on properties we desire out of certain materials. A relational database can serve as an excellent tool for organizing this vast information for us as they have been doing for large scale businesses for decades.

Microsoft SQL Server which is an enterprise-grade relational database management system developed by Microsoft can be of great service in data management aspect. It is a software product meant to store and retrieve data as requested by other software applications like Tableau, knime or Python and R based analysis programs. It allows users to access information in a secure and controlled manner which may either lie on the same computer or across an extensive and distant network (including the Internet). Figure 9 below showcases the entity relationship diagram of how different aspects of peptide materials interaction can be stored and mapped out in SQL Server.

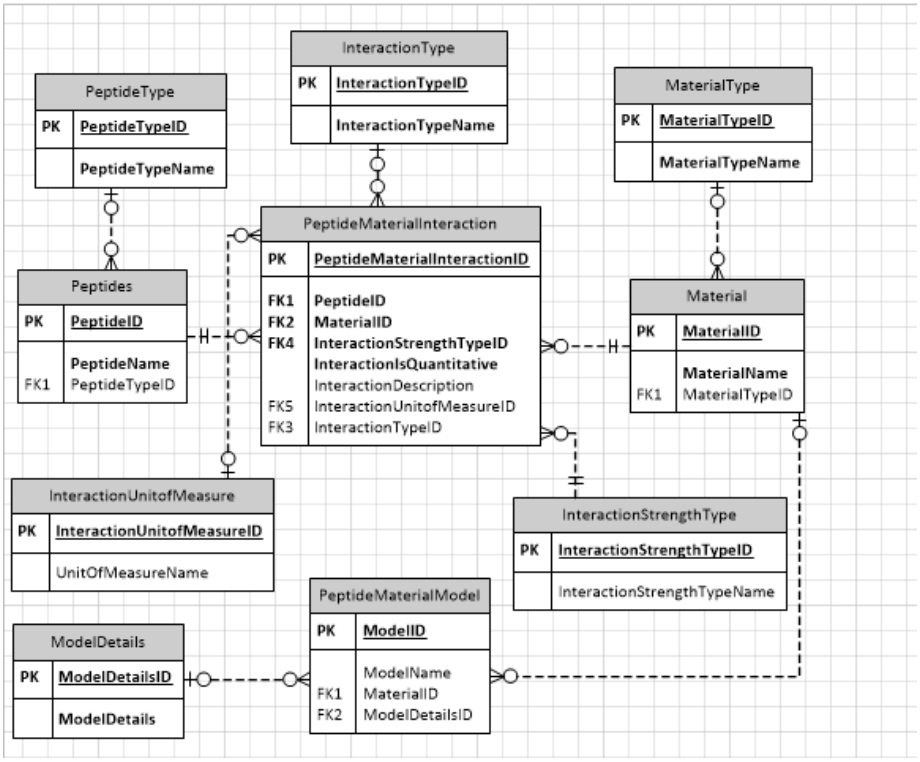


Figure 9: Entity relationship diagram for the peptides and materials database.

2.2.4 Knime for Generating Data Pipelines, Automation & Reporting

As demonstrated in the overview of software we will need various software like R, C++, Python, SQL Server, molecular modeling tools, and a variety of interdependent processes amongst those tools to model a combination of data driven design, rational design and combinatorial data. Thus we would require a tool that can stitch together all these aspects in a user-friendly manner and allow for simple execution of all the processes based on certain requirements from the model. KNIME (Konstanz Information Miner) which is an open source data analytics, reporting, and integration platform allows us to do exactly that[xxi]. It integrates a variety of components for machine learning and data mining applications through its modular data pipelining concepts. It is easy to use graphical interface allows for assembly of nodes for a variety of mundane tasks like data preprocessing (ETL: Extraction, Transformation, Loading), modeling, data analysis, and visualization[xxii]. There is a great community that supports this platform, and it hosts numerous workflow examples where people have already tried to integrate multiple programs for a broad range of applications ranging from bioinformatics to image processing and predictive analytics in finance.

Another benefit of using Knime is that it integrates with other essential tools that are potentially useful. These tools include software like Marvin (a molecule visualization software), Schrodinger biologics and materials science Suite (Integrated solutions for atomic-scale simulation of chemical and biological systems), ImageJ (Image processing tool), Chemistry Development Kit (a suite of open source programs in bioinformatics and cheminformatics) for biologics and materials modeling[xxiii,xxivxxv] . Knime also integrates efficiently with big data databases like MongoDB, scalable computing packages like Spark, in addition to Python, R, C++, SQL Server, Tableau interfaces and offers reporting capabilities via Business Intelligence and Reporting Tool [xxvi,xxii].

Knime thus offers an ability to tap into features from other tools molecular modeling and visualization that might be required in future without changing the underlying architecture. Figure 9 below showcases how we used Knime to create data pipelines to perform data preprocessing, visualization and model parameter tuning by integrating Python, SQL Server and C++ applications in Knime data pipelines.

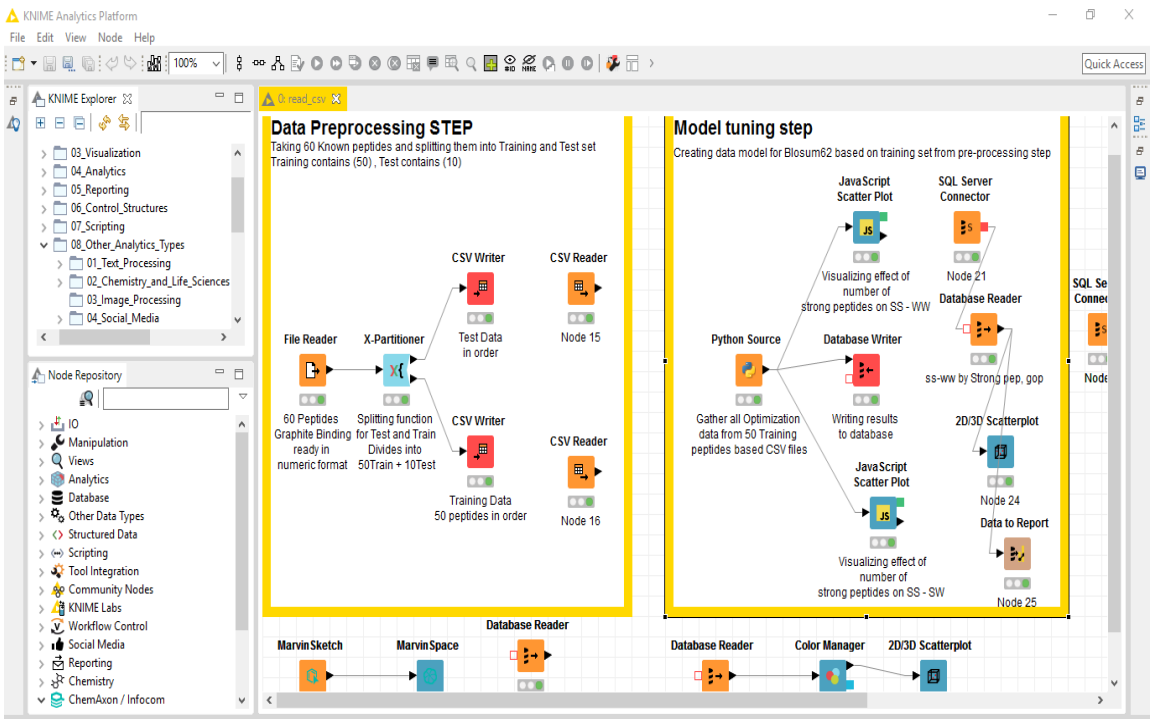


Figure 10: Sample data pipeline showcasing integration of Python, SQL Server and C++ applications in Knime

2.2.5 Tableau for Data Visualization and Dashboards

A common and probably the most useful task in data driven designs and predictive modeling is to make quick and easy displays of information for exploratory analysis. In this space, Tableau seems to be a pretty viable tool, owing to its ability to transform data into visually appealing, interactive visualizations and dashboards[xxvii]. Even though R and Python do offer excellent display capabilities and flexibility, they usually require rigorous programming to be able to make even small changes to interactive visuals. In tableau, it takes only seconds or minutes to achieve interactive graphing capabilities rather than days or weeks as in the case of R and Python, and this is achievable through the use of an easy to use drag-and-drop interface in Tableau. Another advantage with Tableau is that it allows easy connectivity to all the key data sources like SQL Server, Excel, CSV Files, and Knime thus enabling us to access data from multiple sources and join them in Tableau to gain quick insights without much hassle. Tableau also permits the creation of dashboards that are publishable on the network or the internet where other people who might not be familiar with the modeling techniques and details used can access the data and derive insights from it quickly and efficiently.

In current studies, Tableau is used majorly for exploratory analysis of data which is either passed to it through SQL Server or read via CSV Files that python scripts generate and for the creation of visual dashboards.

Chapter 3 - Data-Driven Modelling of Binding Characteristics

The data-driven design of material-binding peptides encompasses combinatorial selection using phage or cell display technologies, bio-panning and binding strength characterization and proceeds to classification model creation using bioinformatics based scoring algorithms and eventually goes on to predictions, model validation, and model refinement. This chapter highlights some critical details involved in the data driven design involved in Graphite binding peptides design. Graphite binding peptides (GrBPs) as shown in Table 1 are found via a combination of cell surface & phage display methods described below.

Peptide Name	Peptide Sequence	Charge	pI	MW	Binding	SD
R2.E4.6	THPLPIHANELT	-1	5.92	1342.52	10.74734	2.775916
R1.E2.3	LPILTPPPDMYS	-1	3.8	1442.73	15.551	4.328121
R4.E2.3	LIHAPPLPVTAT	0	6.74	1229.49	23.71996	1.499474
R2.E2.5	ASGQLHHGYSYD	-1	6	1334.37	25.09771	4.412626
R2.E3.1	LPPLVPPIIHPK	1	8.76	1320.68	32.05668	6.297724
R4.E2.1	PRPSPKMGVSVS	2	11	1241.47	38.3895	0.407905
R2E2.2	PTTAPHITQPPV	0	7.17	1258.44	39.04496	1.248834
R2.E1.3	EPSLEVYHNPDL	-3	4.13	1412.52	40.66001	0.785926
R2.E3.3	PGPPATSMHSPV	0	7.17	1177.34	43.4979	5.754803
R4.E3.6	QSTRLNLTLSSS	1	9.75	1306.44	48.56168	5.772186
R3.B3	TWPRHHTTDALL	0	6.61	1447.62	48.9526	1.349985
R2.E2.4	GSSPATSVPLSS	0	5.24	1032.12	51.15778	0.523611
R1.E1.1	VVAVTDHIRTTT	0	6.71	1312.49	51.74838	1.198891
R2 B3	VPWPSPQSVSLV	0	5.49	1295.5	53.66393	0.333941
R4.E3.2	SSDHSRNMASIT	0	6.46	1305.38	54.46627	4.026499
R1.E4.1	GSTWEPLVARPN	0	6	1326.47	56.7673	1.408535
R4.E3.7	VTWERLEHVPKF	0	6.73	1540.79	56.95549	2.135717
R1 E3.5	QSKGWTTLMNST	1	8.75	1353.51	57.4503	0.675208
R4.B1	STNLYDRYTHSN	0	6.46	1470.52	58.50913	5.064327
R2.B2	SVSAVELNSQKV	0	5.72	1260.41	61.24235	4.478694
R4.E1.10	QLVHTNRVPDSA	0	6.74	1336.47	63.41012	0
R3.E1.5	APPLIYHPSYPF	0	6.78	1401.63	64.91517	2.244914
R1.E2.5	FLHLLPQTRQVT	1	9.76	1452.72	65.36319	5.346573
R2.E4.5	PGRVMHTGNTHV	1	10.2	1305.48	66.40509	1.635942
R3.E4.3	RNPLDNWSLPPV	0	5.84	1407.59	66.7003	1.635942
R1.E3.3	SPYGLHQELNPA	-1	5.22	1258.37	66.935	4.5989
R4.E3.8	KLITTSYSPKTI	2	9.7	1351.61	67.16201	2.309606
R4 B7	STNNAPTPYSTL	0	5.24	1265.34	67.49034	0.17252
R4.E1.5	LTISTSGVEAKT	1	6	1206.36	67.49034	0.42209
R1.E3.1	STFYNSTSSLPS	0	5.24	1290.35	67.54004	1.518503
R1.E2.2	PSPLARLSFWST	1	10.2	1361.56	68.04558	0.331132
R2.E2.1	PQLTNPMFPLQD	-1	4.3	1350.55	69.36993	2.902247
R1 E2.1	IPSYVSTWNPTN	0	5.52	1378.5	70.13667	2.614288
R3 B5	KANHAGKLPWPL	2	10	1331.58	70.66162	3.750523
R4.E4.10	GALHLNSVPTPH	0	6.96	1185.35	70.84548	2.357939
R3.E2.3	PGFRPKGQPTIL	2	11	1310.56	71.36645	1.175394
R2.E1.1	RRDYTKHDNAPA	1	8.6	1443.54	72.09927	0.447094
R1.E3.2	TSYSLASPMSTN	0	5.18	1258.37	72.52876	4.903371
R1.E3.4	SCSVQRSQVHLN	1	7.99	1357.51	72.65622	2.101563
R1.E1.3	YVAGLPLLFTPV	0	5.52	1289.58	72.75211	1.624716
R1.E1.2	APIRGTDSPQTQ	0	5.88	1270.36	74.1231	3.578603
R1.E4.4	DSKLLHDNLQKA	0	6.75	1381.55	74.1636	0.438986
R3 E1.1	TAAAISGPHTPY	0	6.4	1185.3	74.40452	1.183106
R4.E1.7	KLTSSSQVNPLI	1	8.75	1286.49	76.58457	2.842355
R3.E3.4	PYQFVPMPLPST	0	5.95	1376.63	77.74479	1.65674
R4.B3	RELLRTTHYAQN	1	8.75	1501.67	78.05398	2.101824
R3.E2.2	ASGHRITTDQQS	0	6.79	1300.35	78.09454	1.569782
R2E2.3	ADLTVPSLVPLT	-1	3.8	1225.45	78.61235	0.217051
R4.E4.8	NGLDHIPQSVIS	-1	5.08	1279.42	78.86006	0.724204
R2 E4.3	HIANLELSSPSD	-2	4.64	1282.37	79.43791	0.779458
R3 B4	ASPFSADPVLNM	-1	3.8	1248.42	80.49209	1.855444
R2.E4.4	PWDATQGARSHT	-1	5.08	1267.32	81.594	1.521195
R3.E4.8	TLMNPFPPLTLN	0	5.19	1357.63	82.39395	1.758101
R1.E1.4	PSTAHNQPNESS	-1	5.25	1268.26	82.43972	0.999088
R3 B6	AQHRPVDSSTAN	0	6.79	1282.34	82.74783	2.855665
R3.E1.2	NRRLMNSFSLHD	1	9.61	1489.67	83.20036	0.778266
R3.E3.2	LPSRFPFQYSPS	1	8.75	1425.61	83.58011	5.486088
R3.E1.4	PLPLSAKSPGYY	1	8.9	1292.5	84.51377	0.311364
R3.E3.3	PMMARPWISSST	1	10.2	1363.61	85.88664	1.316203
R4.E4.5	IMVTESSDYSSY	0	3.67	1381.47	86.34833	0.731831

Table 1: Sequence, charge, pI, molecular weight, Binding (% Coverage) and SD of graphite binding peptides. (Courtesy: Deniz Yucesoy and Carolyn Gresswell GEMSEC Labs (University of Washington) via ref 7)

3.1 Combinatorial Peptide Selection through Bio-panning.

As mentioned earlier, Cell surface & phage display technologies are the main combinatorial selection strategies which allow for identification of a peptide that binds to particular materials. Figure 11 showcases a schematic diagram for these

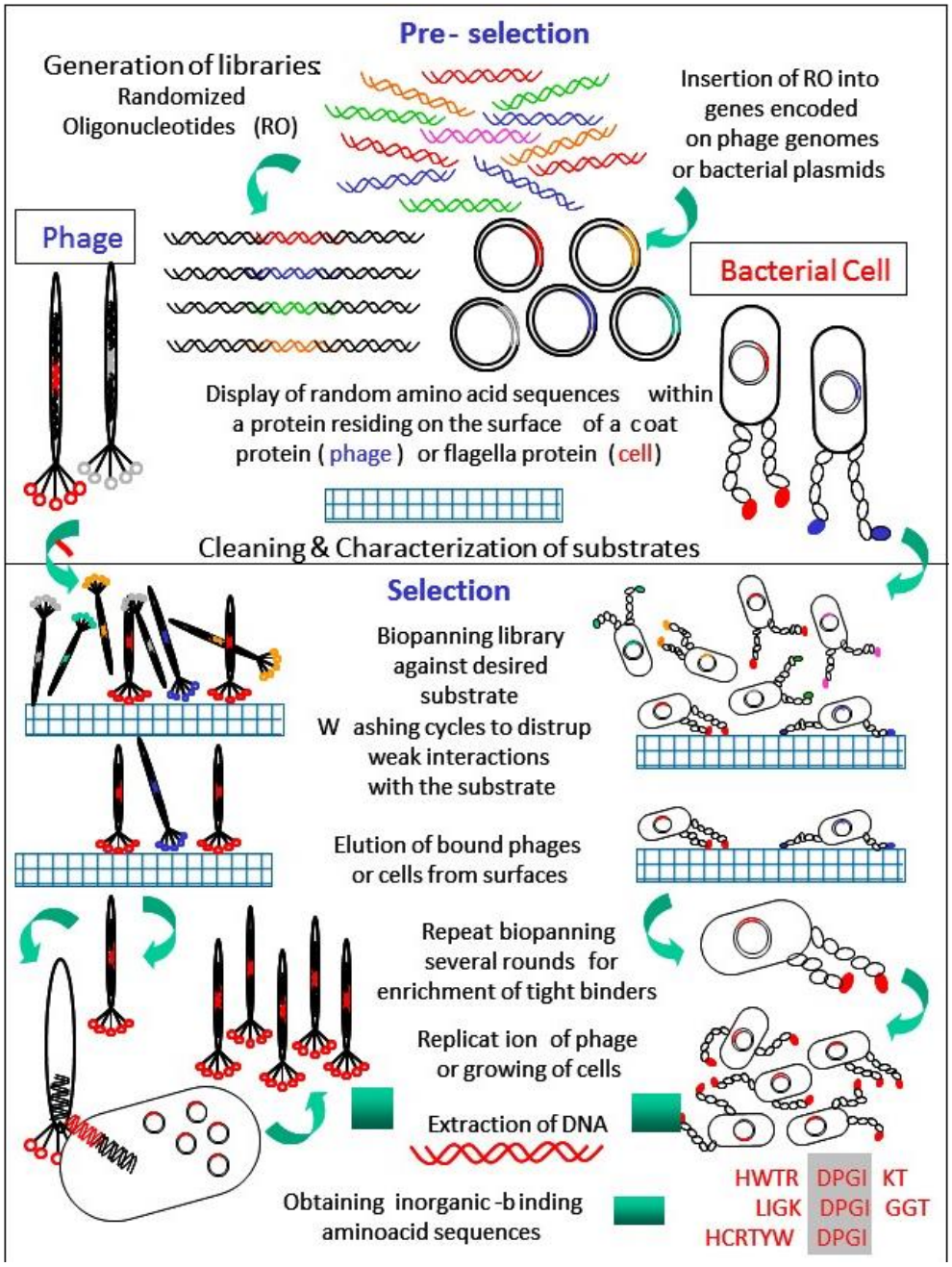


Figure 11: Schematic diagram showing combinatorial selection via a cell surface and Phage display methods. (Courtesy of Dr. Mehmet Sarikaya (GEMSEC, University of Washington))

A typical **cell surface display approach** involves the use of bacterial cell surface systems such as FliTrx library system developed by companies such as Invitrogen or Thermo Fisher Scientific. Here Fli stands for flagellar proteins and Trx stands for thioredoxin. These systems involve a library of random peptides grown in bacterial cells like E. coli and which is being tested for binding against a substrate like graphite. The flagellar proteins (Fli part) ensure peptides are exposed outside of the cells, and the thioredoxin (Trx part) ensures peptides are displayed, by creating a disulfide bridge near the thioredoxin molecules which ensures visibility of random peptide fragments[xxviii] . Once these libraries have been prepared, they are subjected to a binding surface like graphite, and this step is called panning. Panning involves washing away unbound cells from the solid multiple times so as to ensure only those cells which have strong affinity stay bound to the material of choice. The strongly bound cells are then eluted from the material using techniques like changing pH which affect the binding property with materials, and the DNA of these eluted cells is cloned and amplified to ensure production of large amounts of clones of these bacteria that contain these binding peptides.

Similar to cell surface display, **in phage display** a peptide library kit called Ph.D. - 12 is used. It typically allows a billion different randomized peptide sequences to be tried for binding studies. Each of the sequences is amplified once to yield approximately 100 copies of each sequence in 10 µl of the supplied phage [xxix]. In this technique, random dodecapeptides (peptides with 12 amino acids) are fused to minor coat proteins such as (pIII) of a virus like M13's phage and are expressed at the N-terminus followed by a short spacer in between the peptide and the wild-type pIII sequence [xxx]. These phages are exposed to materials of choice such as graphite and subjected to panning technique of washing, elution and amplification cycles to select strong binders to the material. Amplified phages are then subjected to DNA sequence analysis, and finally, individual clones are characterized by quantitative fluorescent microscopy employing tools such as Nikon Eclipse TE-2000U. This is same technique as described by Yucesoy et. al [xxxi] .

3.2 Quantification of Binding Strength and Initial Classification

The cloned bacterial cells or virus phages identified to bind to material from combinatorial selection methods described above are subjected to fluorescent labeling and characterized for binding strength via quantitative fluorescent microscopy using instruments such as Nikon Eclipse TE-2000U fluorescent microscope and analyzed through software such as METAMORPH [xxxi]. The binding affinity of these clones is determined by calculating the average number of adherent cells or phages on the graphite surface in triplicate samples. The cells or phages that get attached to the material surface have a fluorescent label attached to them which gets showcased as brightly colored material under the microscope while wild-type sample which doesn't contain bacteria or phages that can attach to the material is non-fluorescent and helps in baselining of data. The percentage coverage is then used to classify these peptides into three categories namely Strong, Moderate and Weak binders. Figure 12 below showcases the binding strength which is expressed in terms of percentage of surface coverage for 60 Graphite binding peptides selected via phage display method. Here peptides above 80% surface coverage are classified as Strong Binders (10 in number), peptides below two-third surface coverage (<66.67%) are classified as Weak Binders (24 in number), and the rest are classified as Moderate binders (26 in number).

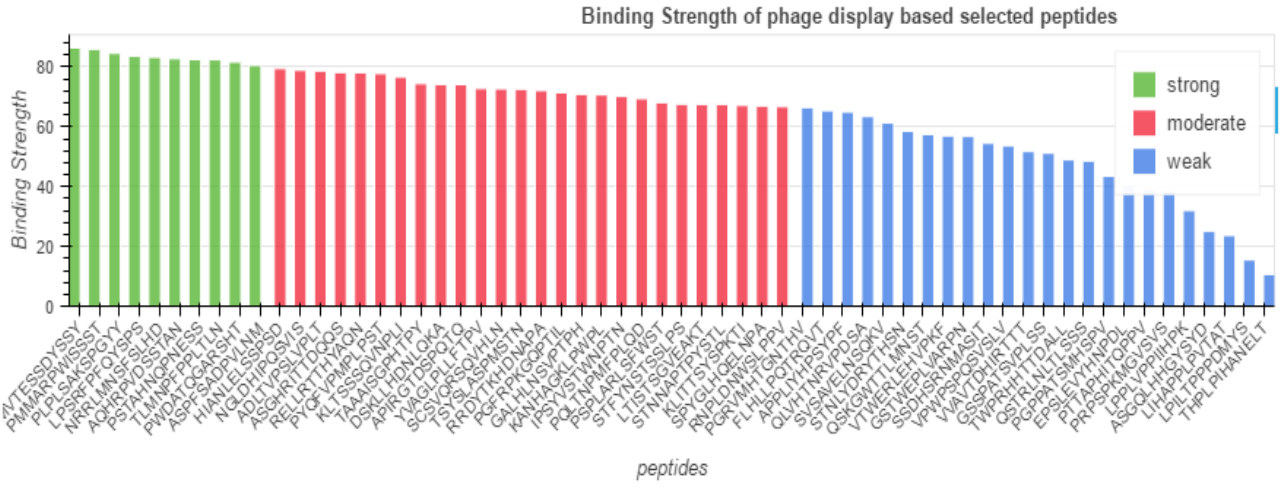


Figure 12: Histogram showing Binding Strength (% surface Coverage) of various combinatorially selected graphite binding peptides and their classification into strong, moderate and weak binders.
. (The original data supplied by ref 7 and M. Sarikaya Lab, University of Washington)

This classification into categories like strong, moderate and weak is typically governed via scoring models which tend to capture cutoff points for these classifications based on how similar and strongest peptides can remain in one group while having maximum surface coverage. The models are explained in detail in sections 3.3 and 3.4.

3.3 Similarity Analysis Using Needleman-Wunsch Algorithm

Similarity analysis is a bioinformatics technique where a biomolecule sequence like DNA, RNA or protein is studied using multiple analytical methods with an aim to understand features of the molecule, mainly by comparing its features to other similar sequences. The motivation for similarity analysis type approach for material-binding peptide design stems from the success of similar approaches in the field of proteomics and on a simple observation that in the wild proteins which are responsible for similar functions usually have similar sequences [xxxii]. This sequence function similarity is perhaps due to evolutionary and environmental constraints and thus exploring the sequence to structure, and function relationship is an important study to understand the interaction between materials. Oren *et. al.* in 2007, have demonstrated that quartz-binding peptides can be successfully predicted using a similar technique from a sequence of 1 million randomly generated peptides [viii]. Their studies underscored the hypothesis that peptides generated via directed evolution through *in vivo* selection could recognize solid materials if they possess sequences similar to strong binders just like evolutionarily related proteins behave. These similarly sequenced proteins are known to perform very similar functions due to evolutionary and environmental constraints they face.

The scoring matrix which captures amino acid replacement scores along with variables such as gap and mismatch penalties are the typical constituents of the underlying model used for these kinds of analyses. Figure 13 shows BLOSUM62 a simple scoring matrix. The similarity score, an output of such models, is obtained by scoring match-mismatch alignments of amino acids in two protein sequences. An

overall high score means there is a high similarity between sequences. BLOSUM (Block substitution matrix) and PAM (Point Accepted Mutation) based matrices are two most commonly utilized scoring matrices for peptide similarity analysis and have been derived from naturally occurring similarity amongst protein sequences. These matrices have been extensively studied and used to understand homology or evolutionary relationships found in nature for nucleotide and protein sequence comparisons.

BLOSUM62

	Ala	Arg	Asn	Asp	Cys	Gln	Glu	Gly	His	Ile	Leu	Lys	Met	Phe	Pro	Ser	Thr	Trp	Tyr	Val
Ala	4																			
Arg	−1	5																		
Asn	−2	0	6																	
Asp	−2	−2	1	6																
Cys	0	−3	−3	−3	9															
Gln	−1	1	0	0	−3	5														
Glu	−1	0	0	2	−4	2	5													
Gly	0	−2	0	−1	−3	−2	−2	6												
His	−2	0	1	−1	−3	0	0	−2	8											
Ile	−1	−3	−3	−3	−1	−3	−3	−4	−3	4										
Leu	−1	−2	−3	−4	−1	−2	−3	−4	−3	2	4									
Lys	−1	2	0	−1	−3	1	1	−2	−1	−3	−2	5								
Met	−1	−1	−2	−3	−1	0	−2	−3	−2	1	2	−1	5							
Phe	−2	−3	−3	−3	−2	−3	−3	−3	−1	0	0	−3	0	6						
Pro	−1	−2	−2	−1	−3	−1	−1	−2	−2	−3	−3	−1	−2	−4	7					
Ser	1	−1	1	0	−1	0	0	0	−1	−2	−2	0	−1	−2	−1	4				
Thr	0	−1	0	−1	−1	−1	−1	−2	−2	−1	−1	−1	−1	−2	−1	1	5			
Trp	−3	−3	−4	−4	−2	−2	−3	−2	−2	−3	−2	−3	−1	1	−4	−3	−2	11		
Tyr	−2	−2	−2	−3	−2	−1	−2	−3	2	−1	−1	−2	−1	3	−3	−2	−2	2	7	
Val	0	−3	−3	−3	−1	−2	−2	−3	−3	3	1	−2	1	−1	−2	−2	0	−3	−1	4

PAM250

	Cys	Gly	Pro	Ser	Ala	Thr	Asp	Glu	Asn	Gln	His	Lys	Arg	Val	Met	Ile	Leu	Phe	Tyr	Trp
Cys	12																			
Gly	-3	5																		
Pro	-3	-1	6																	
Ser	0	1	1	1																
Ala	-2	1	1	1	2															
Thr	-2	0	0	1	1	3														
Asp	-5	1	-1	0	0	0	4													
Glu	-5	0	-1	0	0	0	3	4												
Asn	-4	0	-1	1	0	0	2	1	2											
Gln	-5	-1	0	-1	0	-1	2	2	1	4										
His	-3	-2	0	-1	-1	-1	1	1	2	3	6									
Lys	-5	-2	-1	0	-1	0	0	0	1	1	0	5								
Arg	-4	-3	0	0	-2	-1	-1	-1	0	1	2	3	6							
Val	-2	-1	-1	-1	0	0	-2	-2	-2	-2	-2	-2	-2	4						
Met	-5	-3	-2	-2	-1	-1	-3	-2	0	-1	-2	0	0	2	6					
Ile	-2	-3-	2-	-1	-1	0	-2	-2	-2	-2	-2	-2	-2	4	2	5				
Leu	-6	-4	-3	-3	-2	-2	-4	-3	-3	-2	-2	-3	-3	2	4	2	6			
Phe	-4	-5	-5	-3	-4	-3	-6	-5	-4	-5	-2	-5	-4	-1	0	1	2	9		
Tyr	0	-5	-5	-3	-3	-3	-4	-4	-2	-4	0	-4	-5	-2	-2	-1	-1	7	10	
Trp	-8	-7	-6	-2	-6	-5	-7	-7	-4	-5	-3	-3	2	-6	-4	-5	-2	0	0	17

Figure 13: BLOSUM62 and PAM250 scoring matrices highlighting scores for replacement on Amino acids. [xxxiii]

There are two main categories of sequence alignments namely pair-wise alignment and a more complex and computationally expensive method called multiple sequence alignment. While pairwise alignment methods are used to find motif's (domain's) among a pair of sequences, the multiple sequence alignment methods are used to find common motif's sequences (domains) for a given set of sequences [xxxiv]. A common way to perform multiple sequence alignment is via a progressive alignment approach. In this method, pairwise alignments can be conducted for all the sequences, and a tree can be built as a guide for the multiple alignments. This method cannot guarantee optimal alignments and is thus not always very reliable. Optimal alignments are however achievable and guaranteed via dynamic programming approaches which are common approaches for pairwise alignment. This method can be applied to an N-dimensional matrix for all sequences to achieve multiple sequence alignment kind of

objectives as well. We will be employing dynamic programming based pairwise sequence alignments strategy for multiple sequences in this study. This approach is not efficient if the matrix space increases beyond a certain limit, however, for our case we only have 100 or fewer peptides containing only 12 amino acids and thus this strategy is suitable for optimal alignments in the present study. Two important dynamic programming algorithms for pairwise alignment of sequences are the Needleman-Wunsch algorithm and the Smith-Waterman algorithm [xxxv,xxxvi] . Both algorithms use similar approaches and variables like the scoring matrix, gap penalties and traceback methods for optimal alignments and the difference is in global vs. local alignment strategies employed in them.

Needleman-Wunsch tries to achieve the best global alignment, i.e. alignments over the entire input while Smith-Waterman tries to achieve best local alignment. Since global alignment algorithms attempt to align every residue in every sequence, they are of great use when sequences are very similar and equal size. We will be using the Needleman-Wunsch algorithm in our studies as we have all sequences of the same size of twelve amino acids.

As highlighted above, the scoring matrices are the key to modeling this sequence similarity and thus it is important to get these matrices to capture the peptide materials interactions. Traditional matrices like BLOSUM and PAM are derived based on similarities and mutations that occur in naturally occurring proteins and thus, they are not the best models for our use cases. However, they do serve as good seed matrices to generate a better scoring matrix for capture similarities amongst materials binding peptides. To create these new scoring matrices, we need a model to capture the similarities between strongly binding peptides and create a huge separation between active binders with weak binders. The total similarity score (TSS) between two sets of groups of peptide binders for example (A and B) is used to achieve this separation and TSS is defined as below. This is adapted from the work of Oren *et. al.* [viii].

$$TSS_{A-B} = \frac{1}{NA \cdot (NB - \delta_{AB})} \sum_{i=1}^{NA} \sum_{j=1}^{NB} PSS_{ij}(1 - \delta_{ij}\delta_{AB})$$

Here A can be a group of Strong, Medium or Weak binders, and similarly, B can be the Strong, Medium or Weak binders group. NA and NB are the total numbers of sequences in sets A and B. PSS_{ij} is the Pairwise similarity score value for the i^{th} sequence of set A and j^{th} sequence of set B. $\delta_{ij} = 1$ when i=j and $\delta_{ij} = 0$ when i≠j.

Now to create a separation between strong binders and weak binders a model or scoring matrix that can maximize the difference ($TSS_{S-S} – TSS_{W-W}$) will be ideal.

To create new scoring matrices, the seed matrices which can be BLOSUM or PAM based matrices initially can be perturbed using a greedy procedure where all the elements in a 20x10 diagonal matrix are changed one at a time in small increment or decrement of 1, and new TSS scores are calculated. The perturbations that lead to an increase in $TSS_{S-S} – TSS_{W-W}$ score are kept, and this is iterated over a fixed number of perturbations or until a sufficient separation between TSS values is obtained. These newly generated matrices can then be used in prediction of novel peptides which might have very high or low binding affinities based on scores from a pool of randomly generated ones.

Section 3.3.1 below highlights the Classification model built for Graphite Binding peptides based on commonly used Scoring matrices like BLOSUM62. Section 3.3.2 highlights how parameter tuning and model refinement is done to select optimal parameters to model Graphite binding peptides.

3.3.1 Classification Model Using Common Amino Acid Substitution Matrices.

As described in earlier sections common methods for sequence similarity analysis of protein sequences typically measure similarity by using a substitution matrix which helps in assigning scores for all possible exchanges of one amino acid with another. In evolutionary biology and bioinformatics, these substitution

matrices have traditionally been used to describe the rate at which one character (Amino Acid) in a protein sequence changes to another character state over time [xxxvii]. Over periods of evolutionary cycles amino acids in protein sequences generally, tend to mutate into various other amino acids but follow certain environmental constraints which allow only certain types of amino acid replacement to be favorable without much compromise in function of proteins. For example, hydrophilic amino acids like arginine are more likely to be replaced by hydrophilic amino acids like glutamine, versus them to be mutated into a hydrophobic amino acid like leucine. Mutating an amino acid to another with significantly different properties usually tends to affect the folding pattern in proteins, and thus their activity for a particular function could be lost, and thus there is an evolutionary pressure not to favor such mutations to maintain similar function. The most widely used scoring matrices in bioinformatics like PAM and BLOSUM categories tend to account for these selective pressures and are thus an excellent starting point for any study where amino acid substitutions can be a variable.

The **PAM (Point Accepted Mutation)** class of scoring matrices developed by Margaret Dayhoff in the 1970s is one of the earliest and most widely used amino acid substitution matrices in bioinformatics studies [xxxviii]. This matrix was derived by observing around 1500 observed mutations in the phylogenetic trees of some 71 families of closely related proteins which were selected by based on high similarity with their predecessors. Only those protein alignments which displayed at least 85% identity were included, and it was assumed that the aligned mismatches occurred because of single mutation events, rather than several mutations at the same location. The numbers that are usually at the end of PAM matrices like PAM1 and PAM30 are based on estimates of what rate of substitution is expected if 1 or 30 out of 100 amino acids had changed in a given evolutionary interval [xxxix, xl]. The PAM1 matrix is a generator matrix, and it is used for calculating others. It is assumed while generating new matrices that repeated mutations would follow a similar pattern as observed in the PAM1 matrix. It is also assumed that multiple substitutions can occur on same the site. PAM250 matrix is calculated from PAM1 using the Markov chain model of protein mutation [xxxix, xl]. This model works well for closely

related protein sequences but fails to do well in evolutionary divergent protein sequences where BLOSUM Model is useful. It is described in detail in the following paragraph.

BLOSUM (Block Substitution Matrix), is another commonly used substitution matrix class in bioinformatics and was first created by Henikoff and Henikoff in 1992 by using multiple alignments of evolutionarily divergent proteins. They derived these substitution matrices from around 2000 blocks of aligned sequence segments and characterized more than 500 groups of related proteins [xli, xlii] . The probabilities used to generate the matrix were derived using blocks of conserved sequences found via multiple protein alignments. These regions tend to have functional importance within related proteins, but they can sometimes be a cause of bias, which can be reduced by clustering the segments in blocks which contain a sequence identity above a fixed threshold and by giving weight to each such cluster. For example in BLOSUM62, the most commonly used BLOSUM type matrix, this threshold is set at 62% [xli , xlii]. Higher numbered BLOSUM matrices are typically used for aligning closely related sequences whereas, the lower numbered matrices are used for divergent sequences.

For our purpose, both PAM and BLOSUM matrices can be employed as seed matrices which can be perturbed to generate a better classifier matrix for specific material-binding peptides. Now, even though our aim is to generate new matrices, it is important to find out specific parameters in seed matrices that can lead to best separation of strong binding peptides and weak binding peptides as they will help us identify parameters for generating new matrices. Figure 14 below showcases various parameters that influence the new scoring matrix generation and eventually prediction of new vigorous and weak binders for a specific material. The most influential parameters for a particular scoring matrix-based model are the number of amino acids in peptide sequences(TotalAAs), the peptide sequences used(file_All), the number of peptides classified as strong(TotalStr), medium(TotalMod) and weak(TotalWeak). Other relevant input parameters are the gap opening and gap extension penalties(gop and gep) and type of objective function used (difx). The variables, number of

perturbations allowed(nmax) and the number of loops allowed (maxloop) are used to generate a new matrix and thus are not required until the new matrix is being generated. The effects of various parameters for BLOSUM62 and PAM250 matrices are discussed in section 3.3.2 below.

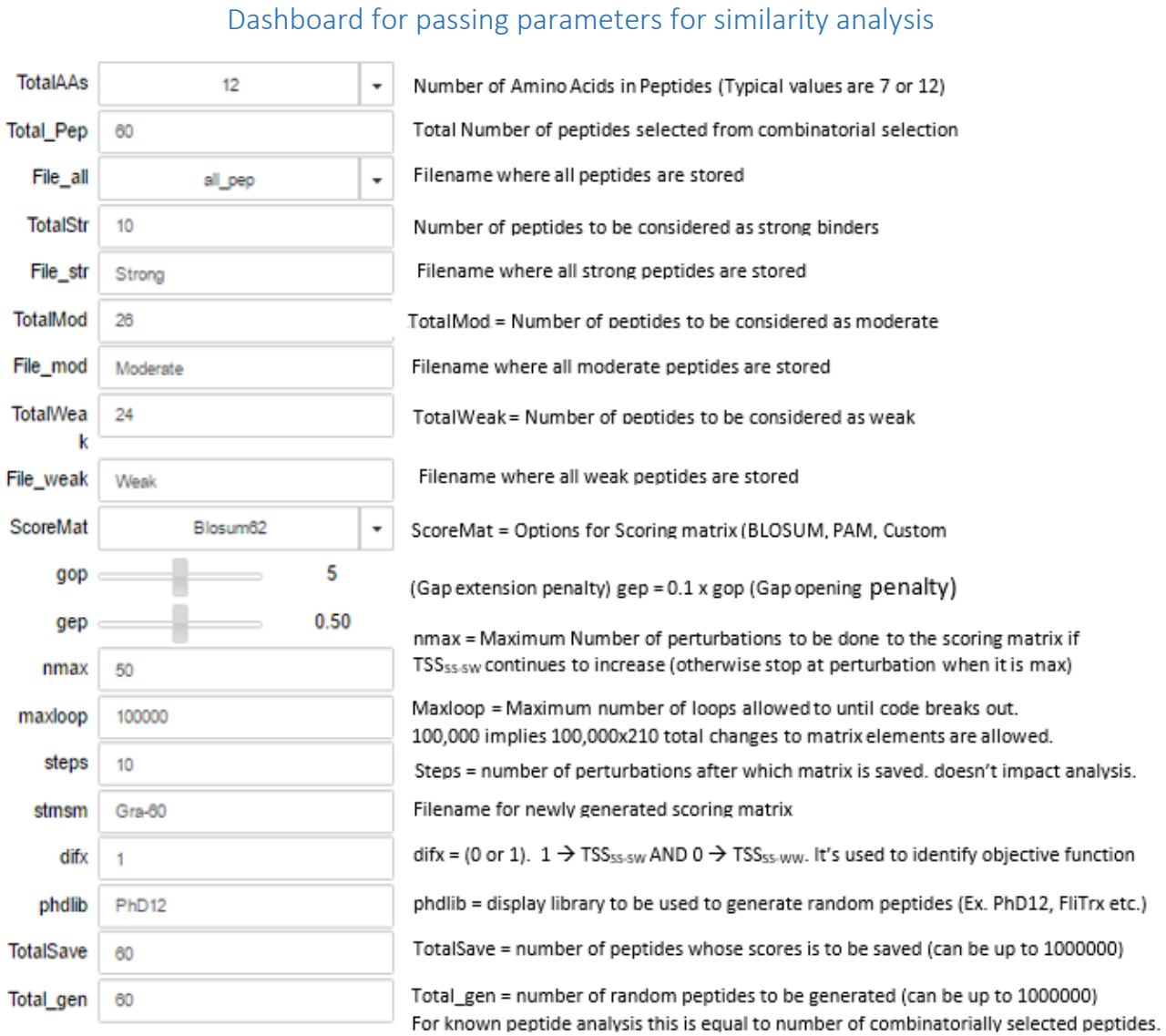


Figure 14: A sample dashboard for setting parameters for similarity analysis of graphite binding peptides.

3.3.2. Parameter Tuning and Model Refinement Using Substitution Matrices

The first step in parameter tuning for any model is exploratory data analysis of dataset and data-preprocessing to make the dataset suitable for analytics. For preprocessing, we first convert amino acid

alphabets to numbers using the table in figure 15, to speed up calculations in code. We also split the dataset into test and training sets as shown in figure 15 to avoid overfitting the data on the entire dataset.

(a)

amino acids	Numbers assigned
C	0
S	1
T	2
P	3
A	4
G	5
N	6
D	7
E	8
Q	9
H	10
R	11
K	12
M	13
I	14
L	15
V	16
F	17
Y	18
W	19

(b)

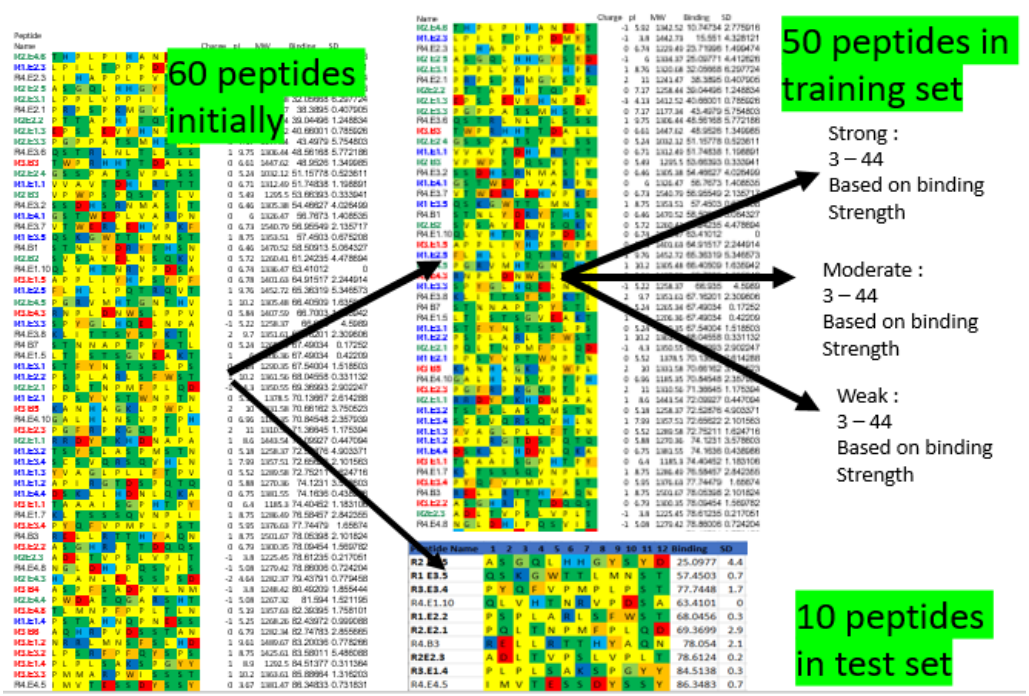


Figure 15: (a) amino acid to number conversion key for faster processing of data in algorithms. **(b)** Splitting the dataset into training and test datasets containing 50 peptides in training set and 10 in the test set.

Figure 16 below showcases the peptides that were randomly selected to be in the test dataset.

Peptide Name	1	2	3	4	5	6	7	8	9	10	11	12	Binding	SD
R2 E2 5	A	S	G	Q	L	H	H	G	Y	S	Y	D	25.0977	4.4
R1 E3.5	Q	S	K	G	W	T	T	L	M	N	S	T	57.4503	0.7
R3.E3.4	P	Y	Q	F	V	P	M	P	L	P	S	T	77.7448	1.7
R4.E1.10	Q	L	V	H	T	N	R	V	P	D	S	A	63.4101	0
R1.E2.2	P	S	P	L	A	R	L	S	F	W	S	T	68.0456	0.3
R2.E2.1	P	Q	L	T	N	P	M	F	P	L	Q	D	69.3699	2.9
R4.B3	R	E	L	L	R	T	T	H	Y	A	Q	N	78.054	2.1
R2E2.3	A	D	L	T	V	P	S	L	V	P	L	T	78.6124	0.2
R3.E1.4	P	L	P	L	S	A	K	S	P	G	Y	Y	84.5138	0.3
R4.E4.5	I	M	V	T	E	S	S	D	Y	S	S	Y	86.3483	0.7

Figure 16: Peptides that were randomly chosen to be a part of test data set (Data Supplied by the Sarikaya Lab).

Now, once we have done all necessary data preprocessing, the next steps are model tuning for a specific scoring matrix. For this purpose, we first have to split the training dataset into strong, moderate and weak

binders. We do this by first arranging the peptides in descending order of binding strength in training set and then select top "TotalStr" number as strong binders. This number is chosen to be greater than 3 and less than (total peptides in training set – 6) to involve at least three strong, three moderate and three weak binding peptides to perform multiple sequence comparisons. Similarly, after top "TotalStr" have been filtered out we then select "TotalMod" numbers from remaining peptides in sequential order, and again this number is kept > 3. Finally, remaining sequences are chosen as weak binders. This way we vary the sequences in each category from 3 to (total peptides in training set – 6) and observe their effects on $TSS_{SS\text{-}SW}$ our objective function that we want to maximize. Now, we choose a specific scoring matrix BLOSUM62 and showcase how effects of various other parameters influence our objective function $TSS_{SS\text{-}SW}$. Figure 17 and 18 below showcase the effects of the number of strong binders with top "TotalStr" sequences on Average $TSS_{SS\text{-}SW}$ across all observations of Gop and Gep (which are the third dimension in this picture). It is evident from this analysis that strong binders in the range of 3-8 are probably the most optimal choice to maximize $TSS_{SS\text{-}SW}$ where six seems to be an ideal choice.

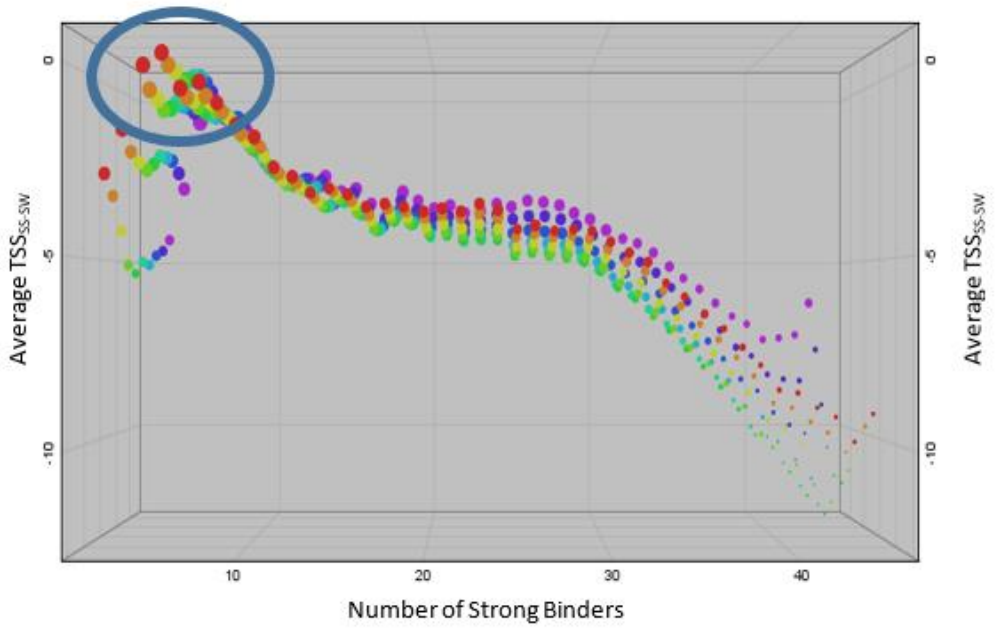


Figure 17: 3D representation of the effect of the number of strong binders on average TSS (hidden axis is Gop/Gep).

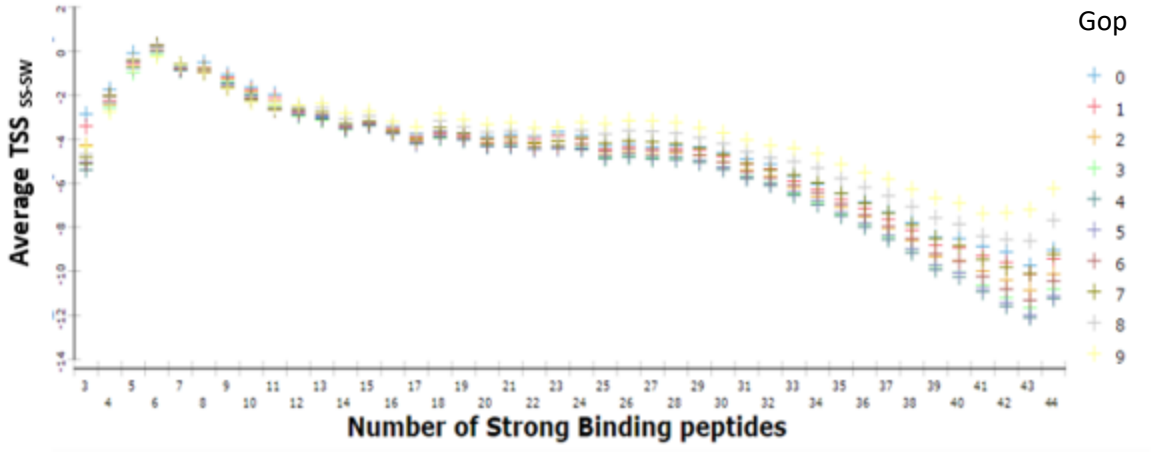


Figure 18: 2D representation of Effect of the number of strong binders on average $TSS_{ss\text{-}sw}$.

Figure 19, showcases the effect of Gop and Gep which are used by the affine formula of g(k) = -gop - (k -1)gep. Here k is the gap length and gep, and gop are related by the relation gep = 0.1(gop). Here for the Strong peptide number in range 4-8, we see that gop of 5-7 and gep of 0.5-0.7 are most optimal.

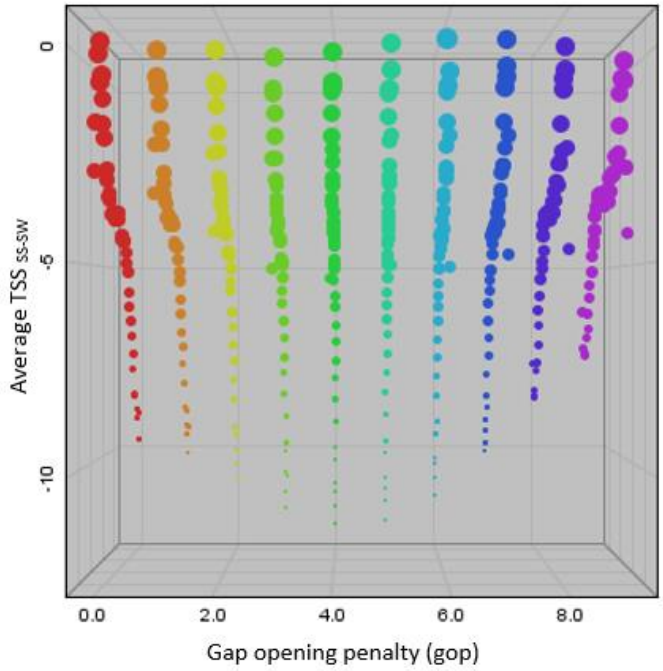


Figure 19: 3D representation of the effect of gap opening penalty (gop) on $TSS_{ss\text{-}sw}$ (hidden axis is strong binders).

Thus, from the above two observations, a combination of 6 strong binders and a gop value of 0.6 seem to maximize the Average values of TSS_{SS-SW}. Next, we identify an optimal number of weak binders by using ratio (strong binders / weak binders) vs. Average TSS_{SS-SW} as shown in figure 20 below. Figure 20 points us an optimal point of 26 weak binders or 42 weak binders. Now taking 41 weak binders might not be a great solution since that way are left with only three moderate binders because the total number of strong + moderate + weak binders must be equal to peptides in the training set (50).

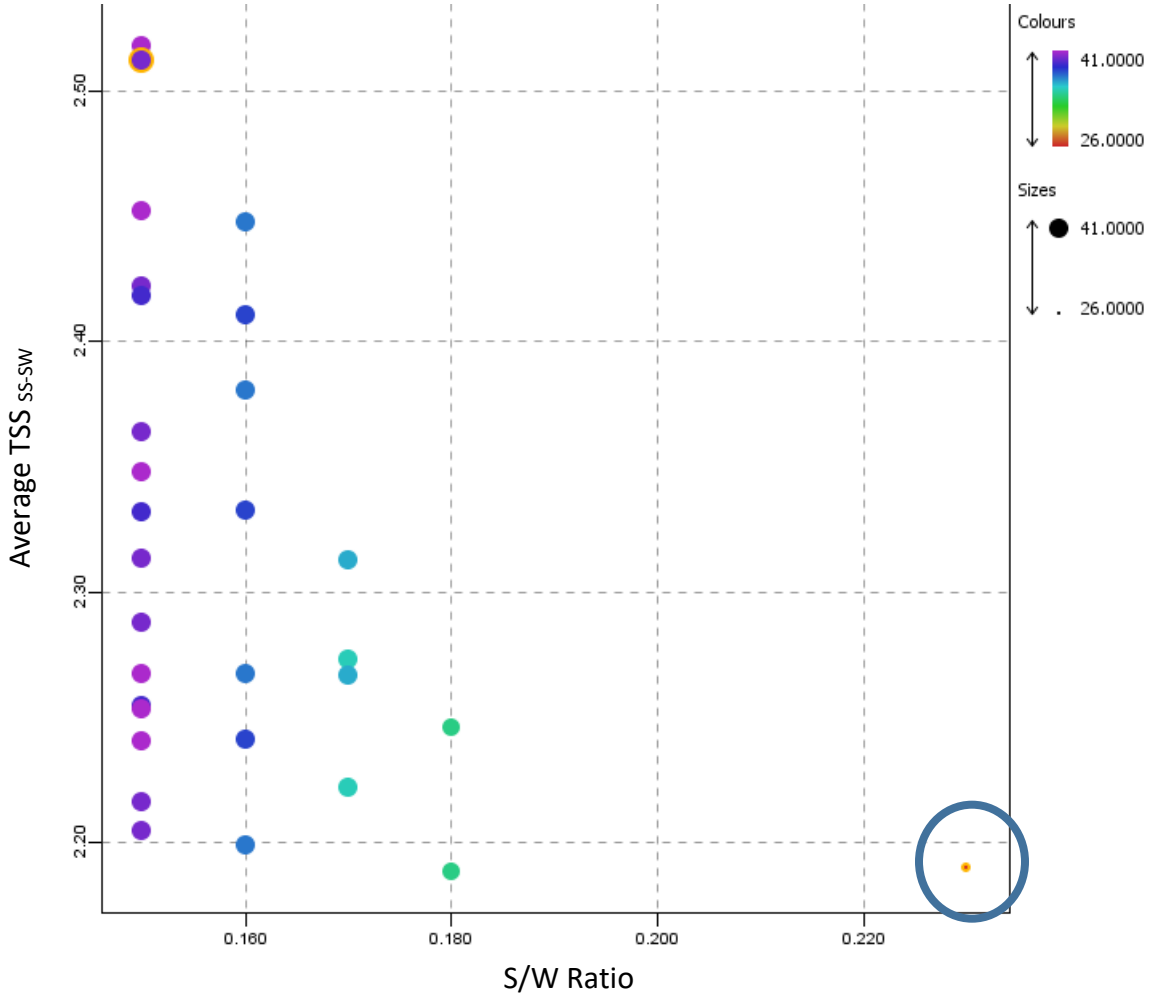


Figure 20: Effect of ratio of strong binders to weak binders on TSS_{SS-SW} scores. (color and size represent Weak binders).

Thus, we take 26 as a more reasonable and optimal number of weak binders. This observation leaves us with six heavy binders, 26 weak binders and 18 moderate binders and a Gop value of 0.5 or 0.6 for Blosum62. We have now found best parameters for BLOSUM62 Matrix that can be used for generation of

the new scoring matrix and for prediction of heavy binders from the test set that we had left out for training our model.

Figure 21 shows, the similarity scores obtained on the training established by using BLOSUM62 matrix and parameters of 6 strong binders (colored in green), 18 moderate binders (colored in red), 26 weak binders (colored in blue) with a Gop of 0.5 and gep of 5 units.

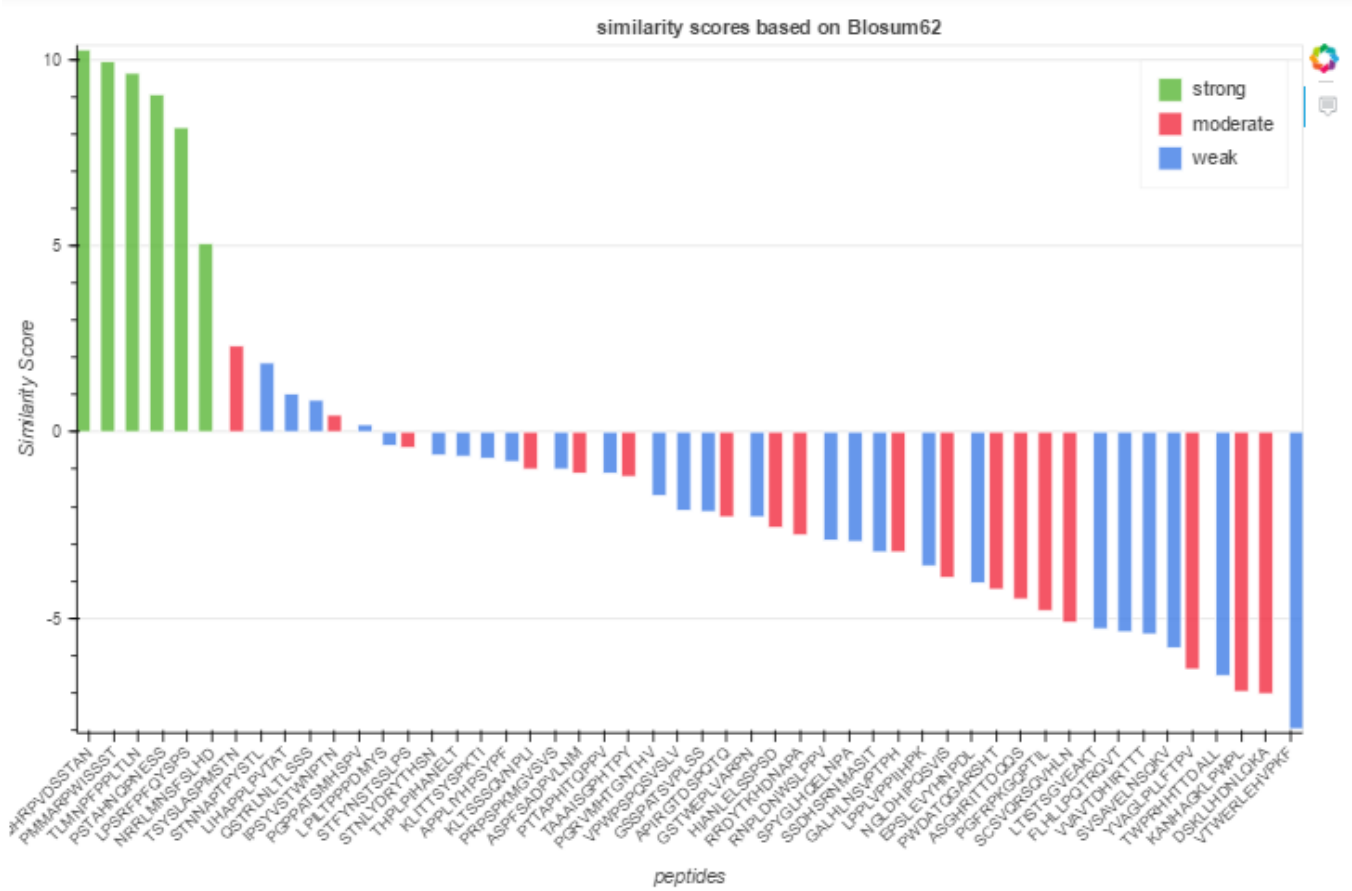


Figure 21: Similarity Scores for training set based on optimized parameters for the BLOSUM62 scoring matrix.

Since our model was built to distinguish Strong Binders from Weak and didn't have much input about moderate binders it does a decent job at giving a high score to all strong binders while not being able to identify weak binders and moderate binders very accurately. Also, we had a choice of either assuming 41 weak binders and three moderate binder and six heavy binders or 26 weak binders and 18 moderate binders and six strong binders which meant that the model might be able to classify moderate as weak

binders. This analysis on training set leads to satisfactory results however the real test of a model is on the test dataset.

Figure 22 showcases how this model performs in the classification of peptides in the test set. Based on the similarity scores for training dataset we can see that strong binding peptides typically have a similarity score of greater than 4. Moderates peptides can be roughly put in a range of -4 to 4 and the region below the score of -4 is usually dominated by weak peptides.

	peptide	similarity score	
0	PQLTNPMFPLQD	4.5	✗
1	PYQFVPMPLPST	1.4333	✓
2	PSPLARLSFWST	0.3	✓
3	QSKGWTTLMNST	-0.2	✓
4	QLVHTNRVPDSA	-1.2667	✓
5	RELLRTTHYAQN	-3.5333	✓
6	ADLTVPSLVPLT	-3.5667	✓
7	PLPLSAKSPGYY	-3.7	✗
8	IMVTESSDYSSY	-4.0333	✗

Peptide Name	1	2	3	4	5	6	7	8	9	10	11	12	Binding	SD
R2 E2 5	A	S	G	Q	L	H	H	G	Y	S	Y	D	25.0977	4.4
R1 E3.5	Q	S	K	G	W	T	T	L	M	N	S	T	57.4503	0.7
R3.E3.4	P	Y	Q	F	V	P	M	P	L	P	S	T	77.7448	1.7
R4.E1.10	Q	L	V	H	T	N	R	V	P	D	S	A	63.4101	0
R1.E2.2	P	S	P	L	A	R	L	S	F	W	S	T	68.0456	0.3
R2.E2.1	P	Q	L	T	N	P	M	F	P	L	Q	D	69.3699	2.9
R4.B3	R	E	L	L	R	T	T	H	Y	A	Q	N	78.054	2.1
R2E2.3	A	D	L	T	V	P	S	L	V	P	L	T	78.6124	0.2
R3.E1.4	P	L	P	L	S	A	K	S	P	G	Y	Y	84.5138	0.3
R4.E4.5	I	M	V	T	E	S	S	D	Y	S	S	Y	86.3483	0.7

Criterion : SS > 4 → Strong, SS between (-4, 4) → Moderate, SS < -4 → Weak

Figure 22: Results of BLOSUM62 based scoring model's classification on the test dataset.

As we see in figure 22 above, there are three wrong predictions out of 9 which means the accuracy of this model is at 66.67%. Another notable aspect of this model is that it does fine to classify moderate and weak peptides in their respective regions, but fails badly in the classification of two powerful peptides which have the binding strength of 86% and 84% respectively. Also, there is a significant limitation in the model that the training dataset is minimal, and thus the training set might not have been supplied ample information about sequences like R3.E1.4 and R4.E4.5 which might be similar to each other but might not have been similar to peptides that landed in the training set.

A side experiment without creating a test data set was also done by passing in top 10 binders from the bio-panning data set containing all 60 peptides selected by ordering the peptides in decreasing order of

binding affinity as strong. The next 26 were labeled as moderate and the last 24 as weak as discussed in section 3.2 and shown in figure 12. The experiment yielded a scoring pattern that was more in alignment with the binding strength, but involved overfitting as the model was trained on same data. This model, however, gave very similar scores to R3.E1.4 and R4.E4.5 which might imply that since these kinds of peptides were missing in that initial dataset, the training model had no idea that these could be strong binders. This experiment even though futile for testing purpose does underscore the importance that more variety in initial dataset might be needed for this kind of models to work correctly.

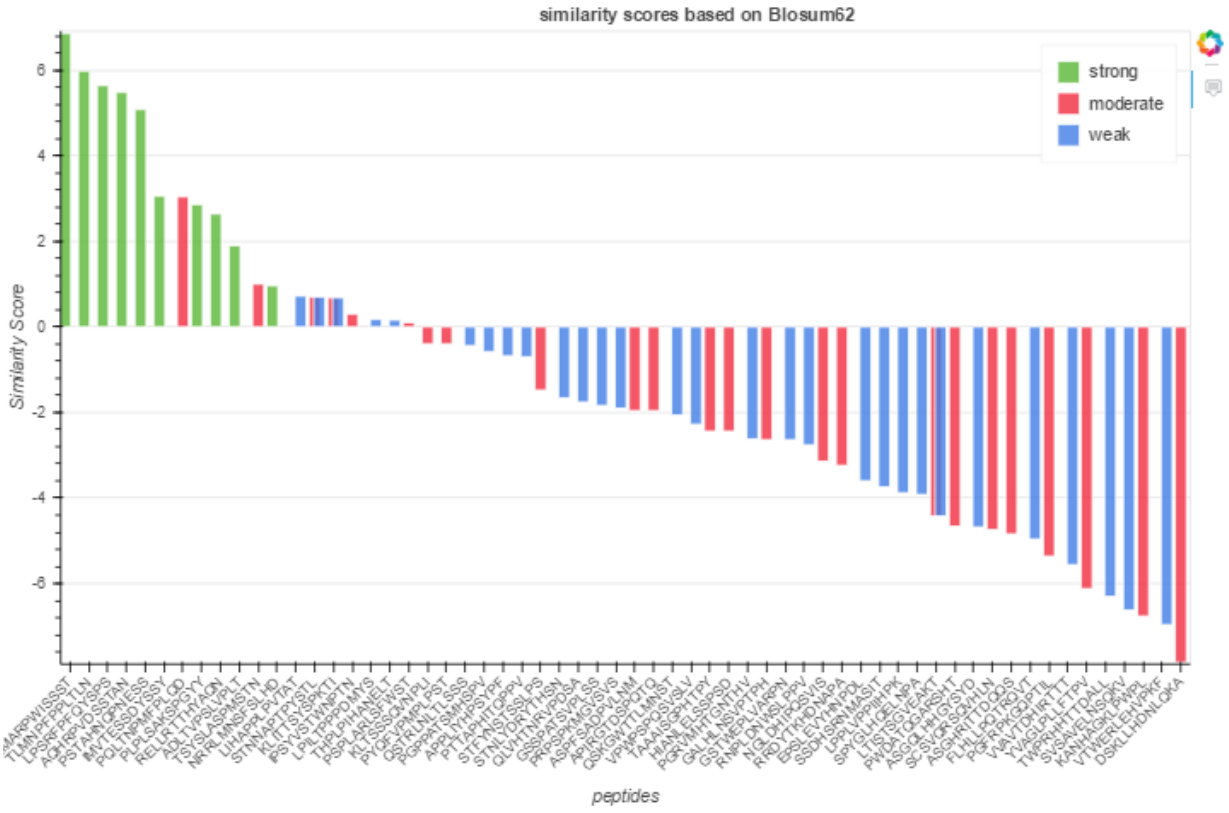


Figure 23: Similarity score calculation by using BLOSUM62 Matrix with all combinatorially selected peptides as input

Having optimized the model on BLOSUM62 and studies various nuances lets now move on to perturbing these matrices and generating new matrices that might be better predictors of graphite binding capabilities.

Section 3.4 covers this aspect of how the matrices are being generated from a preceding matrix. It also includes a discussion on the parameters tuning and results of these new matrices. Out training data remains the same 50 that were used to tune the BLOSUM62 model.

3.4. Generation of Material Specific Scoring Matrix

We use the greedy algorithms to perturb the seed matrix such as BLOSUM62 to create a new matrix that maximizes the $TSS_{SS\text{-}SW}$ difference. In general, the **greedy algorithms** involve problem-solving heuristics to find locally optimal choices at every stage while expecting to find a global optimum eventually. However, in many situations, this strategy does not produce an optimal solution[xliii, viii] . Nonetheless, greedy heuristics are usually very efficient at yielding locally optimal solutions which are a decent approximate to the optimal global solution and do that within a short time. Since for our use case we want to perturb the seed matrix to obtain a new matrix that maximizes the $TSS_{SS\text{-}SW}$ difference, any perturbation that increases $TSS_{SS\text{-}SW}$ is accepted, and all others are rejected. To demonstrate this idea let's use BLOSUM62 as seed matrix. Since we have optimized all other parameters, we know that by using BLOSUM62 the maximum $TSS_{SS\text{-}SW}$ is obtained when gep/gop are 0.5/5, and we use 6,18 and 26 strong, moderate and weak binders respectively. These parameters yield a $TSS_{SS\text{-}SW}$ value of 0.27. Now we want to generate matrices that can increase this value beyond 0.27 mark. To do this, we change all the elements of the scoring matrix containing 210 elements across the diagonal by either adding 1 or subtracting 1 and see if we can get a higher score of $TSS_{SS\text{-}SW}$ if we do find it we keep the new perturbed matrix and again try changing the elements of the new matrix. In this approach, we are always keeping a locally optimal choice that increased the difference in $TSS_{SS\text{-}SW}$ and discarding anything that reduces it. One perturbation of BLOSUM62 to account for graphite binding peptides is as shown below in figure 24. This perturbation only affects first row and column but leads to increase in $TSS_{SS\text{-}SW}$ from 0.27 to 0.675.

BLOSUM62

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	4	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	5	-1	0	-2	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-2
P	-3	-1	-1	7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	0	-1	4	0	-2	-2	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-3
G	-3	0	-2	-2	0	6	0	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2
N	-3	1	0	-2	-2	0	6	1	0	0	1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	-1	-1	-2	-1	1	6	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	-1	-1	-1	-2	0	2	5	2	0	0	1	-2	-3	-3	-2	-3	-2	-3
Q	-3	0	-1	-1	-1	-2	0	0	2	5	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8	0	-1	-2	-3	-3	-3	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5	-1	-3	-2	-2	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5	1	2	1	0	-1	-1
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4	2	3	0	-1	-3
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4	1	0	-1	-2
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	2
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

1st perturbation to BLOSUM62

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	8	-2	-2	-4	-1	-4	-4	-4	-5	-4	-4	-4	-4	-2	-2	-2	-2	-3	-3	-3
S	-2	5	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-2	1	5	-1	0	-2	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-2
P	-4	-1	-1	7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	-1	1	0	-1	4	0	-2	-2	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-3
G	-4	0	-2	-2	0	6	0	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2
N	-4	1	0	-2	-2	0	6	1	0	0	1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-4	0	-1	-1	-2	-1	1	6	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-5	0	-1	-1	-1	-2	0	2	5	2	0	0	1	-2	-3	-3	-2	-3	-2	-3
Q	-4	0	-1	-1	-1	-2	0	0	2	5	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-4	-1	-2	-2	-2	-2	1	-1	0	0	8	0	-1	-2	-3	-3	-3	-1	2	-2
R	-4	-1	-1	-2	-1	-2	0	-2	0	1	0	5	2	-1	-3	-2	-3	-3	-2	-3
K	-4	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5	-1	-3	-2	-2	-3	-2	-3
M	-2	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5	1	2	1	0	-1	-1
I	-2	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4	2	3	0	-1	-3
L	-2	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4	1	0	-1	-2
V	-2	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4	-1	-1	-3
F	-3	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6	3	1
Y	-3	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	2
W	-3	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

Figure 24: BLOSUM62 and 1st perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1, max(TSS_{SS-SW})= 0.675.

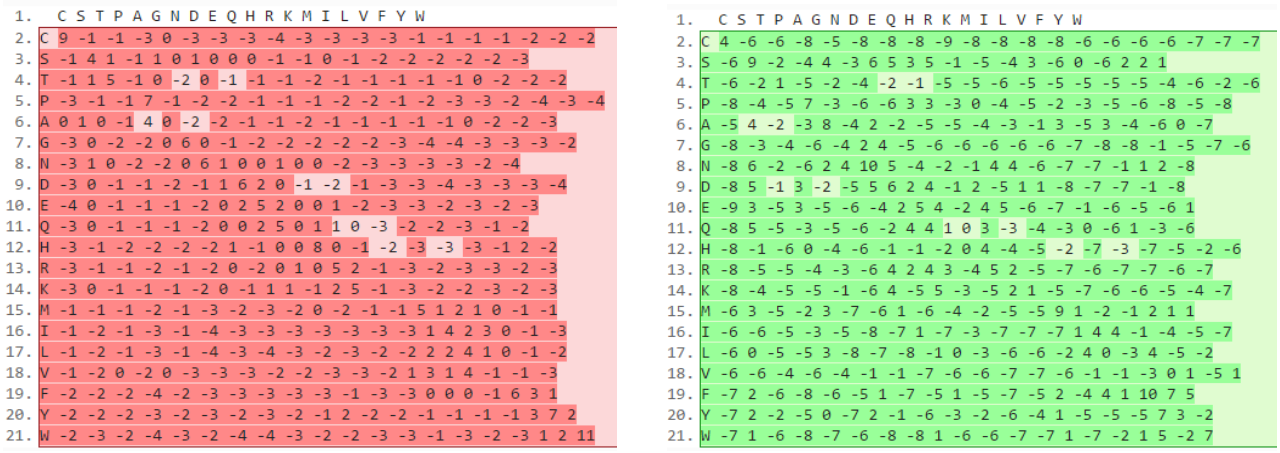
Figure 25 shows, an intermediate matrix obtained by 158 perturbations to BLOSUM62. Using same parameters as previously used. Now we see many values have changed, but most of the changes from BLOSUM62 only seem to be different by ± 1 unit. However, the TSS_{SS-SW} has risen manifolds here to 4.54 from an initial value of o.27 and 0.675 after the first perturbation.

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	4	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	5	-1	0	-2	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-2
P	-3	-1	-1	7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	0	-1	4	0	-2	-2	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-3
G	-3	0	-2	-2	0	6	0	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2
N	-3	1	0	-2	-2	0	6	1	0	0	1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	-1	-1	-2	-1	1	6	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	-1	-1	-1	-2	0	2	5	2	0	0	1	-2	-3	-3	-2	-3	-2	-3
Q	-3	0	-1	-1	-1	-2	0	0	2	5	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8	0	-1	-2	-3	-3	-3	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5	-1	-3	-2	-2	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5	1	2	1	0	-1	-1
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4	2	3	0	-1	-3
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4	1	0	-1	-2
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	2
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	8	-2	-2	-4	-1	-4	-4	-4	-5	-4	-4	-4	-4	-2	-2	-2	-2	-3	-3	-3
S	-2	5	0	-2	2	-1	2	1	1	1	-2	-2	-1	0	-3	-1	-3	-1	-1	-2
T	-2	0	4	-2	1	-3	-1	0	-2	-2	-3	-2	-2	-2	-2	-1	-3	-1	-3	-1
P	-4	-2	-2	6	-2	-3	-3	0	0	-2	-1	-3	-2	-1	-2	-2	-3	-5	-2	-5
A	-1	2	1	-2	5	-1	-1	-3	-2	-2	-3	-2	0	0	-2	0	-1	-3	-3	-4
G	-4	-1	-3	-3	-1	5	1	-2	-3	-3	-3	-3	-3	-4	-5	-5	-2	-2	-4	-3
N	-4	2	-1	-3	-1	1	7	2	-1	1	0	1	1	-3	-4	-4	-2	-2	-1	-5
D	-4	1	0	0	-3	-2	2	5	3	1	-2	-1	-2	-2	-2	-5	-4	-4	-2	-5
E	-5	1	-2	0	-2	-3	-1	3	4	3	-1	1	2	-3	-4	-2	-3	-2	-3	-2
Q	-4	1	-2	-2	-2	-3	1	1	3	4	-1	2	0	-1	-4	-1	-3	-2	0	-3
H	-4	-2	-3	-1	-3	-3	0	-2	-1	-1	7	-1	-2	-3	-4	-4	-4	-2	1	-3
R	-4	-2	-2	-3	-2	-3	1	-1	1	2	-1	6	3	-2	-4	-3	-4	-4	-3	-4
K	-4	-1	-2	-2	0	-3	1	-2	2	0	-2	3	4	-2	-4	-3	-3	-2	-3	-4
M	-2	0	-2	-1	0	-4	-3	-2	-3	-1	-3	-2	-2	6	0	1	0	1	0	0
I	-2	-3	-2	-2	-2	-5	-4	-2	-4	-4	-4	-4	0	3	3	2	-1	-2	-4	
L	-2	-1	-2	-2	0	-5	-4	-5	-2	-1	-4	-3	-3	1	3	3	0	1	-2	-1
V	-2	-3	-1	-3	-1	-2	-2	-4	-3	-3	-4	-4	-3	0	2	0	3	-2	-2	-2
F	-3	-1	-3	-5	-3	-2	-2	-4	-2	-2	-2	-4	-2	1	-1	1	-2	7	4	2
Y	-3	-1	-1	-2	-3	-4	-1	-2	-3	0	1	-3	-3	0	-2	-2	-2	4	6	1
W	-3	-2	-3	-5	-4	-3	-5	-5	-2	-3	-3	-4	-4	0	-4	-1	-2	2	1	10

Figure 25: BLOSUM62 and 158th perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 158, max(TSS_{SS-SW})= 4.54.

Figure 26, showcases the final matrix obtained by 534 perturbations to BLOSUM62. The perturbations beyond 534 perhaps decreased the $TSS_{SS\text{-}SW}$ or held it constant, and thus they are not used and the program terminated at perturbation 534 even though the maximum allowed perturbations be set at 1000. Using same parameters as previously used. Now we see the values in the scoring matrix have changed significantly, and many have changed by five units like C -> C was 9 in BLOSUM62 which has now become 4. The $TSS_{SS\text{-}SW}$ has risen even more now and reached to 14.23 from an initial value of 0.27 and 0.675 after the first perturbation and 4.54 after 158th perturbation.



	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	9	-1	-1	-3	0	-3	-3	-3	-4	-3	-3	-3	-3	-1	-1	-1	-1	-2	-2	-2
S	-1	4	1	-1	1	0	1	0	0	0	-1	-1	0	-1	-2	-2	-2	-2	-2	-3
T	-1	1	5	-1	0	-2	0	-1	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-2
P	-3	-1	-1	7	-1	-2	-2	-1	-1	-1	-2	-2	-1	-2	-3	-3	-2	-4	-3	-4
A	0	1	0	-1	4	0	-2	-2	-1	-1	-2	-1	-1	-1	-1	-1	0	-2	-2	-3
G	-3	0	-2	-2	0	6	0	-1	-2	-2	-2	-2	-2	-3	-4	-4	-3	-3	-3	-2
N	-3	1	0	-2	-2	0	6	1	0	0	1	0	0	-2	-3	-3	-3	-3	-2	-4
D	-3	0	-1	-1	-2	-1	1	6	2	0	-1	-2	-1	-3	-3	-4	-3	-3	-3	-4
E	-4	0	-1	-1	-1	-2	0	2	5	2	0	0	1	-2	-3	-3	-2	-3	-2	-3
Q	-3	0	-1	-1	-1	-2	0	0	2	5	0	1	1	0	-3	-2	-2	-3	-1	-2
H	-3	-1	-2	-2	-2	-2	1	-1	0	0	8	0	-1	-2	-3	-3	-3	-1	2	-2
R	-3	-1	-1	-2	-1	-2	0	-2	0	1	0	5	2	-1	-3	-2	-3	-3	-2	-3
K	-3	0	-1	-1	-1	-2	0	-1	1	1	-1	2	5	-1	-3	-2	-2	-3	-2	-3
M	-1	-1	-1	-2	-1	-3	-2	-3	-2	0	-2	-1	-1	5	1	2	1	0	-1	-1
I	-1	-2	-1	-3	-1	-4	-3	-3	-3	-3	-3	-3	-3	1	4	2	3	0	-1	-3
L	-1	-2	-1	-3	-1	-4	-3	-4	-3	-2	-3	-2	-2	2	2	4	1	0	-1	-2
V	-1	-2	0	-2	0	-3	-3	-3	-2	-2	-3	-3	-2	1	3	1	4	-1	-1	-3
F	-2	-2	-2	-4	-2	-3	-3	-3	-3	-3	-1	-3	-3	0	0	0	-1	6	3	1
Y	-2	-2	-2	-3	-2	-3	-2	-3	-2	-1	2	-2	-2	-1	-1	-1	-1	3	7	2
W	-2	-3	-2	-4	-3	-2	-4	-4	-3	-2	-2	-3	-3	-1	-3	-2	-3	1	2	11

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	4	-6	-6	-8	-5	-8	-8	-8	-9	-8	-8	-8	-8	-6	-6	-6	-6	-7	-7	-7
S	-6	9	-2	-4	4	-3	6	5	3	5	-1	-5	-4	3	-6	0	-6	2	2	1
T	-6	-2	1	-5	-2	-4	-2	-1	-5	-5	-6	-5	-5	-5	-5	-5	-4	-6	-2	-6
P	-8	-4	-5	7	-3	-6	-6	3	3	-3	0	-4	-5	-2	-3	-5	-6	-8	-5	-8
A	-5	4	-2	-3	8	-4	2	-2	-5	-5	-4	-3	-1	3	-5	3	-4	-6	0	-7
G	-8	-3	-4	-6	-4	2	4	-5	-6	-6	-6	-6	-6	-7	-8	-8	-1	-5	-7	-6
N	-8	6	-2	-6	2	4	10	5	-4	-2	-1	4	4	-6	-7	-7	-1	1	2	-8
D	-8	5	-1	3	-2	-5	5	6	2	4	-1	2	-5	1	1	-8	-7	-7	-1	-8
E	-9	3	-5	3	-5	-6	-4	2	5	4	-2	4	5	-6	-7	-1	-6	-5	-6	1
Q	-8	5	-5	-3	-5	-6	-2	4	4	1	0	3	-3	-4	-3	0	-6	1	-3	-6
H	-8	-1	-6	0	-4	-6	-1	-1	-2	0	4	-4	-5	-2	-7	-3	-7	-5	-2	-6
R	-8	-5	-5	-4	-3	-6	4	2	4	3	-4	5	2	-5	-7	-6	-7	-7	-6	-7
K	-8	-4	-5	-5	-1	-6	4	-5	5	-3	-5	2	1	-5	-7	-6	-6	-5	-4	-7
M	-6	3	-5	-2	3	-7	-6	1	-6	-4	-2	-5	-5	9	1	-2	-1	2	1	1
I	-6	-6	-5	-3	-5	-8	-7	1	-7	-3	-7	-7	-7	1	4	4	-1	-4	-5	-7
L	-6	0	-5	-5	3	-8	-7	-8	-1	0	-3	-6	-6	-2	4	0	-3	4	-5	-2
V	-6	-6	-4	-6	-4	-1	-1	-7	-6	-6	-7	-7	-6	-1	-1	-3	0	1	-5	1
F	-7	2	-6	-8	-6	-5	1	-7	-5	1	-5	-7	-5	2	-4	4	1	10	7	5
Y	-7	2	-2	-5	0	-7	2	-1	-6	-3	-2	-6	-4	1	-5	-5	-5	7	3	-2
W	-7	1	-6	-8	-7	-6	-8	-8	1	-6	-6	-7	-7	1	-7	-2	1	5	-2	7

Figure 26: BLOSUM62 and 534th perturbation to BLOSUM62 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1000, max($TSS_{SS\text{-}SW}$)= 14.23.

Similar, perturbation on PAM250 with a gap and gop of 5 and 0.5 yielded 450th perturbation to be highest before program terminated. The max $TSS_{SS\text{-}SW}$ value obtained of 10.809 thus implying that BLOSUM250 based perturbation might be better with the Gep/Gop of 0.5/5 and strong, moderate and weak binding peptides value of 6,18 and 26 as shown in figure 27 below.

We will be using the matrix generated via the 534th perturbation to BLOSUM62 as our new similarity matrix with same parameters to see if this matrix can predict the binding characteristics in a better manner.

PAM250

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	12	0	-2	-3	-2	-3	-4	-5	-5	-5	-3	-4	-5	-5	-2	-8	-2	-4	0	-8
S	0	2	1	1	1	1	1	0	0	-1	-1	0	0	-2	-1	-3	-1	-3	-3	-2
T	-2	1	3	0	1	0	0	0	0	-1	-1	-1	0	-1	0	-2	0	-3	-3	-5
P	-3	1	0	6	1	-1	-1	-1	-1	0	0	0	-1	-2	-2	-3	-1	-5	-5	-6
A	-2	1	1	1	2	1	0	0	0	0	-1	-2	-1	-1	-1	-2	0	-4	-3	-6
G	-3	1	0	-1	1	5	0	1	0	-1	-2	-3	-2	-3	-3	-4	-1	-5	-5	-7
N	-4	1	0	-1	0	0	2	2	1	1	2	0	1	-2	-2	-3	-2	-4	-2	-4
D	-5	0	0	-1	0	1	2	4	3	2	1	-1	0	-3	-2	-4	-2	-6	-4	-7
E	-5	0	0	-1	0	0	1	3	4	2	1	-1	0	-2	-2	-3	-2	-5	-4	-7
Q	-5	-1	-1	0	0	-1	1	2	2	4	3	1	1	-1	-2	-2	-2	-5	-4	-5
H	-3	-1	-1	0	-1	-2	2	1	1	3	6	2	0	-2	-2	-2	-2	-2	0	-3
R	-4	0	-1	0	-2	-3	0	-1	-1	1	2	8	3	0	-2	-3	-2	-4	-4	2
K	-5	0	0	-1	-1	-2	1	0	0	1	0	3	5	0	-2	-3	-2	-5	-4	-3
M	-5	-2	-1	-2	-1	-3	-2	-3	-2	-1	-2	0	0	6	2	4	2	0	-2	-4
I	-2	-1	0	-2	-1	-3	-2	-2	-2	-2	-2	-2	-2	2	5	2	4	1	-1	-5
L	-8	-3	-2	-3	-2	-4	-3	-4	-3	-2	-2	-3	-3	4	2	8	2	2	-1	-2
V	-2	-1	0	-1	0	-1	-2	-2	-2	-2	-2	-2	-2	2	4	2	4	-1	-2	-6
F	-4	-3	-3	-5	-4	-5	-4	-6	-5	-5	-2	-4	-5	0	1	2	-1	9	7	0
Y	0	-3	-3	-5	-3	-5	-2	-4	-4	-4	0	-4	-4	-2	-1	-1	-2	7	10	0
W	-8	-2	-5	-6	-6	-7	-4	-7	-7	-5	-3	2	-3	-4	-5	-2	-6	0	0	17

450th perturbation to PAM250

	C	S	T	P	A	G	N	D	E	Q	H	R	K	M	I	L	V	F	Y	W
C	8	-4	-6	-7	-6	-7	-8	-9	-9	-9	-7	-8	-9	-9	-6	-12	-6	-8	-4	-12
S	-4	6	-3	5	5	1	5	4	4	-1	-1	-2	-4	0	-3	-1	-5	-1	-1	0
T	-6	-3	-1	-4	-1	-2	-4	-2	-2	-5	-3	-3	-4	-1	-4	-6	-4	-5	-3	-9
P	-7	5	-4	4	-1	-5	-3	3	1	-4	2	-4	-3	0	2	-5	-5	-9	-9	-10
A	-6	5	-1	-1	4	-3	4	-2	-4	-4	1	-6	-5	3	-4	-1	-1	-1	-2	-9
G	-7	1	-2	-5	-3	2	-3	0	-1	-2	-5	-6	-5	-6	-6	-3	-4	-8	-8	-10
N	-8	5	-4	-3	4	-3	5	3	0	2	-1	3	-2	1	-5	-6	-3	-1	1	-7
D	-9	4	-2	3	-2	0	3	5	0	5	-2	2	3	0	1	-7	-5	-9	-7	-10
E	-9	4	-2	1	-4	-1	0	0	1	5	-2	0	3	-1	-5	-4	-5	-4	-7	-10
Q	-9	-1	-5	-4	-4	-2	2	5	5	1	0	-2	-2	0	-1	1	-5	-8	-7	-8
H	-7	-1	-3	2	1	-5	-1	-2	-2	0	3	1	-3	-1	-5	-1	-5	-5	-3	-6
R	-8	-2	-3	-4	-6	-6	3	2	0	-2	1	5	6	-3	-5	-4	-5	-1	-7	1
K	-9	-4	-4	-3	-5	-5	-2	3	3	-2	-3	6	2	-3	-5	-6	-5	-8	-7	-6
M	-9	0	-1	0	3	-6	1	0	-1	0	-1	-3	-3	9	-1	5	-1	-1	-1	-7
I	-6	-3	-4	2	-4	-6	-5	1	-5	-1	-5	-5	-5	-1	2	3	1	2	-4	-2
L	-12	-1	-6	-5	-1	-3	-6	-7	-4	1	-1	-4	-6	5	3	5	-1	3	-4	-1
V	-6	-5	-4	-5	-1	-4	-3	-5	-5	-5	-5	-5	-5	-1	1	-1	1	-2	-5	-3
F	-8	-1	-5	-9	-1	-8	-1	-9	-4	-8	-5	-1	-8	-1	2	3	-2	12	10	3
Y	-4	-1	-3	-9	-2	-8	1	-7	-7	-7	-3	-7	-7	-1	-4	-4	-5	10	13	-3
W	-12	0	-9	-10	-9	-10	-7	-10	-10	-8	-6	1	-6	-7	-2	-1	-3	3	-3	14

Figure 27: PAM250 and 450th perturbation to PAM250 using, gop = 5, gep =0.5, strong binders=6, moderate binders = 18, weak binders = 26, max perturbation = 1000, max($TSS_{SS\text{-}SW}$) = 10.809.

3.4.1 Validation on Experimental Datasets

We used 534th perturbation derived scoring matrix, and a Gep/Gop of 0.8/8 was found to be the best fitting model which reasonably maximized the $TSS_{SS\text{-}SW}$ and classified the peptides correctly without over-penalizing the moderate and weak binders. The results for this model on training set as shown below in figure 28.

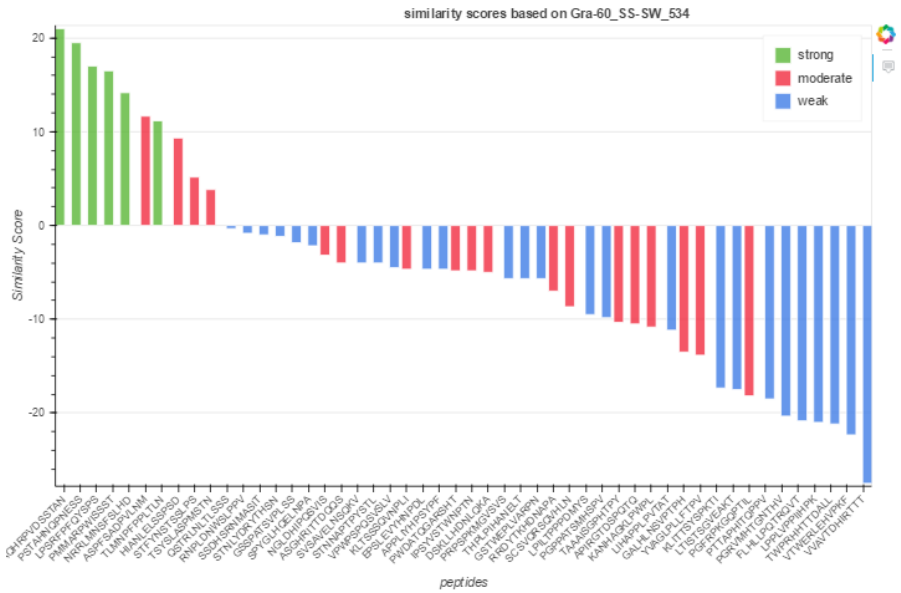
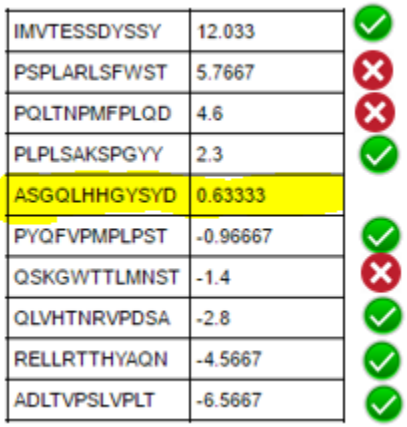


Figure 28: Similarity Scores for training set based on optimized parameters for 534th perturbation on BlOSUM62 derived scoring matrix with Gep/Gop of 0.8/8 units.

Figure 29 below, showcases the predictions made on the test dataset using the newly generated Graphite matrix derived through 534th Perturbation of BLOSUM62 Matrix. For this analysis, we see that now 6 out of 9 peptides are now correctly classified as well. We leave out ASGQLHHGYSYD because it was not accounted for in the last prediction as probably the binding data for this peptide may have been incorrectly reported as 25.0977. Moreover, this matrix somehow classifies IMVTESSDYSSY and PLPLSAKSPGYY correctly as strong binders. The chosen cutoff for strong binders according to this model is any peptide with a score of greater than -1. Moderate peptides can be in the range of -7 to -1 and weak peptides below -7. The only incorrect predictions made by this model are PSPLARLSFWST and PQLTNPMFPLQD which should have been in the moderate category of less than -1 but greater than -7 and QSKGWTTLMNST which should have been less than -7 and in a weak category. This matrix thus gets correct predictions for stronger binding peptides if we tend to relax the range a bit and misclassifies some moderate or weak binders as strong peptides.



IMVTESSDYSSY	12.033
PSPLARLSFWST	5.7667
PQLTNPMFPLQD	4.6
PLPLSAKSPGYY	2.3
ASGQLHHGYSYD	0.63333
PYQFVPMPLPST	-0.96667
QSKGWTTLMNST	-1.4
QLVHTNRVPDSA	-2.8
RELLRTTHYAQN	-4.5667
ADLTVPSLVPLT	-6.5667

Peptide Name	1	2	3	4	5	6	7	8	9	10	11	12	Binding	SD
R2 E2 5	A	S	G	Q	L	H	H	G	Y	S	Y	D	25.0977	4.4
R1 E3.5	Q	S	K	G	W	T	T	L	M	N	S	T	57.4503	0.7
R3.E3.4	P	Y	Q	F	V	P	M	P	L	P	S	T	77.7448	1.7
R4.E1.10	Q	L	V	H	T	N	R	V	P	D	S	A	63.4101	0
R1.E2.2	P	S	P	L	A	R	L	S	F	W	S	T	68.0456	0.3
R2.E2.1	P	Q	L	T	N	P	M	F	P	L	Q	D	69.3699	2.9
R4.B3	R	E	L	L	R	T	T	H	Y	A	Q	N	78.054	2.1
R2E2.3	A	D	L	T	V	P	S	L	V	P	L	T	78.6124	0.2
R3.E1.4	P	L	P	L	S	A	K	S	P	G	Y	Y	84.5138	0.3
R4.E4.5	I	M	V	T	E	S	S	D	Y	S	S	Y	86.3483	0.7

Figure 29: Results of Newly generated matrix from 534th perturbation of BLOSUM62 based scoring model's classification on the test dataset.

Next, we see the effect of taking all 60 peptides into training the model for newly generated matrix in figure 30 below.

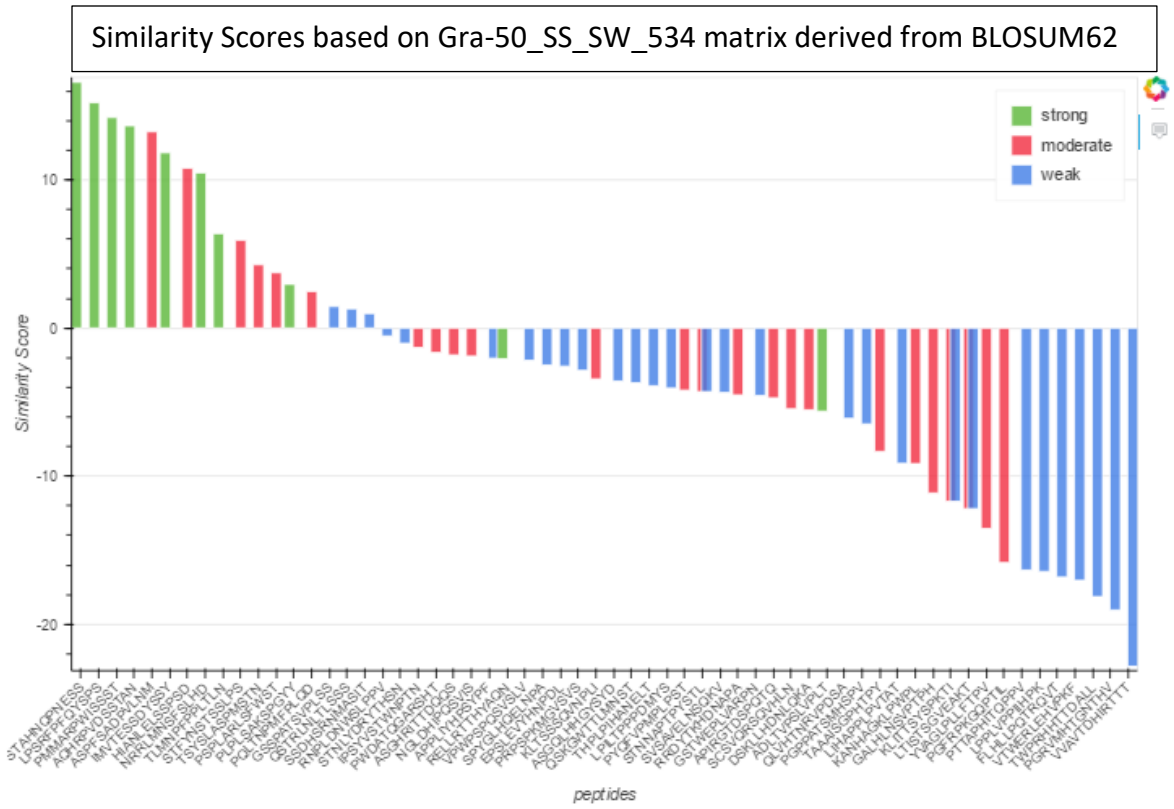


Figure 30: Similarity scores of all combinatorially selected peptides based on Newly generated BLOSUM62's 534th perturbation matrix with gop and gep values of 8 and 0.8 and ten strong, 26 moderate and 24 weak binders.

We can see from this classifier that strong peptides green colored fall into similarity scores of -1 and above except one, while a lot of weak binders have scores below -7.

3.5. Data-Driven Design of Material-Binding Peptides Using New Scoring Matrix

Now, that we have generated a new matrix that seems to work better at the identification of strong binders we now make some prediction on some new strong binders and weak binders as shown in figure 31 (a) and (b) below. Figure on the right (c) illustrates the distribution of similarity scores based on 534th perturbation of graphite binding matrix derived from BLOSUM62.

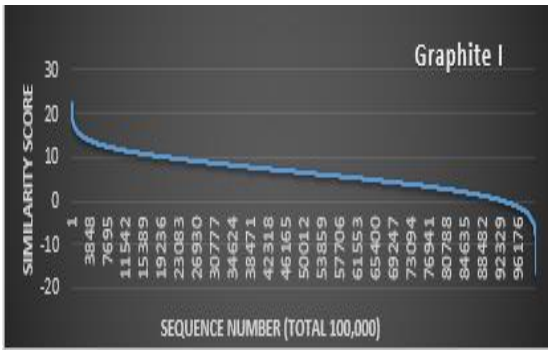
(a)

	peptide	similarity score
0	DSELEFSNPSSN	20.66
1	PSSRPNSSEQSS	20.48
2	PMNSQPSFSPSM	17.02
3	PSPFSSPDLSPM	16.6
4	QRNMNPFPLSSS	16.4
5	LQQNAPVDSSAN	16.04
6	MSSAMDYMDSSA	15.96
7	LVNAPNMESSSS	15.9
8	PAAQHSSSSSSQ	15.88
9	SSDNSPSSSPQY	15.48
10	FSSAMPRPSQSS	14.78
11	LPSPYSPDPLSS	14.58
12	EPSTASFNSSDA	14.32
13	SDSFSNEEGHAN	14.24
14	ANELQPMQNASN	14.18
15	PWDSSPVPQNLS	14.12
16	LPSAANSNLSQM	13.84
17	PSYSFSMPPVAS	13.74
18	AINEWESASNVS	13.1
19	SMGPLDASSSSL	13.1

(b)

	peptide	similarity score
0	HHRTLTTRTPPC	-24.84
1	GCGVSTKLLLKK	-25.08
2	ITHIGTMLLLVV	-25.12
3	VLTGYITTITFL	-25.3
4	TLLTLVTMALCI	-25.32
5	GCLGTMTKFVHV	-25.5
6	TTHTRERRKVKG	-25.54
7	TLLWLTTYGLVV	-25.62
8	RHHVTVCLTHLV	-25.72
9	VHTHRTGHKTKE	-25.76
10	IVIVTTLWHHVT	-25.78
11	CTMQKVVKGRHV	-25.84
12	CTTTAGIRLVTT	-25.98
13	WGTGTVTQTRKT	-26.38
14	VHTVATTTKWTT	-26.4
15	LGLTLTGVGLGT	-26.54
16	LITTLLCWLLTG	-26.56
17	TTVEGGTTHTTP	-26.62
18	YTYTCTTVVLLV	-26.68
19	HQKCKTTGVTHC	-26.76

(c)



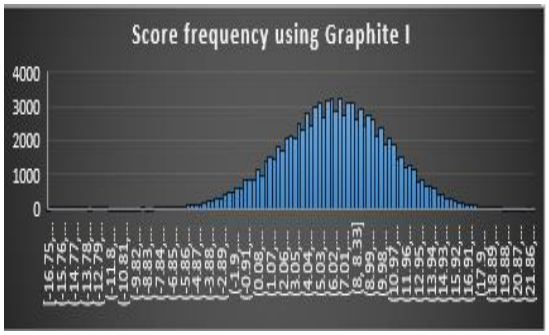


Figure 31: Predictions on 100,000 randomly generated peptides binding properties based on the newly generated scoring matrix. (a) top 20 strongest binders (b) top 20 weakest binders and (c) score distribution of 100,000 random peptides.

As we can see, BLOSUM62 matrix and matrix derived via the 534th perturbation on BLOSUM62 did a reasonable job at predicting strong binders or weak binders. However, it was wrong in predicting around 33% of them correctly. However, if we combine those two predictions via and simplify the classification into strong and weak binders (more than 0 score is strong, and less than 0 is weak), we might be able to get slightly better predictions and avoid losing out on few peptides that can be strong binders.

Similarly, a combination of other strategies that can better predict the strong binders can be used in addition to this technique to generate a better classifier. Section 3.6 discusses some alternative prediction models that can be combined with the scoring matrix method to generate better ensemble method based classifiers.

3.6. Discussion of Alternative Techniques for Modeling Binding Behavior

Ensemble methods are utilized in statistics and predictive analytics to obtain a better predictive performance out of machine learning algorithms. Ensemble methods employ a combination of algorithms which if used in conjunction would yield a better result than the results of any constituent learning algorithms used alone. A similar strategy can be utilized in our studies, and following two methods can probably enhance the performance of the scoring matrix-based model.

The first method can be using a weighted hybrid of position-specific scoring matrix (PSSM), and standard scoring matrices like BLOSUM62.to generate a better seed matrix. PSSM can be an advantageous method because BLOSUM62 type matrices do not consider the order of amino acid sequences in material-binding peptides and thus it might not be the best seed matrix for the greedy algorithm that relies heavily on seed matrix to find an optimal solution for new materials specific scoring matrix. Figure 32 showcases how a simple weighted matrix addition can accomplish a better design of seed matrix.

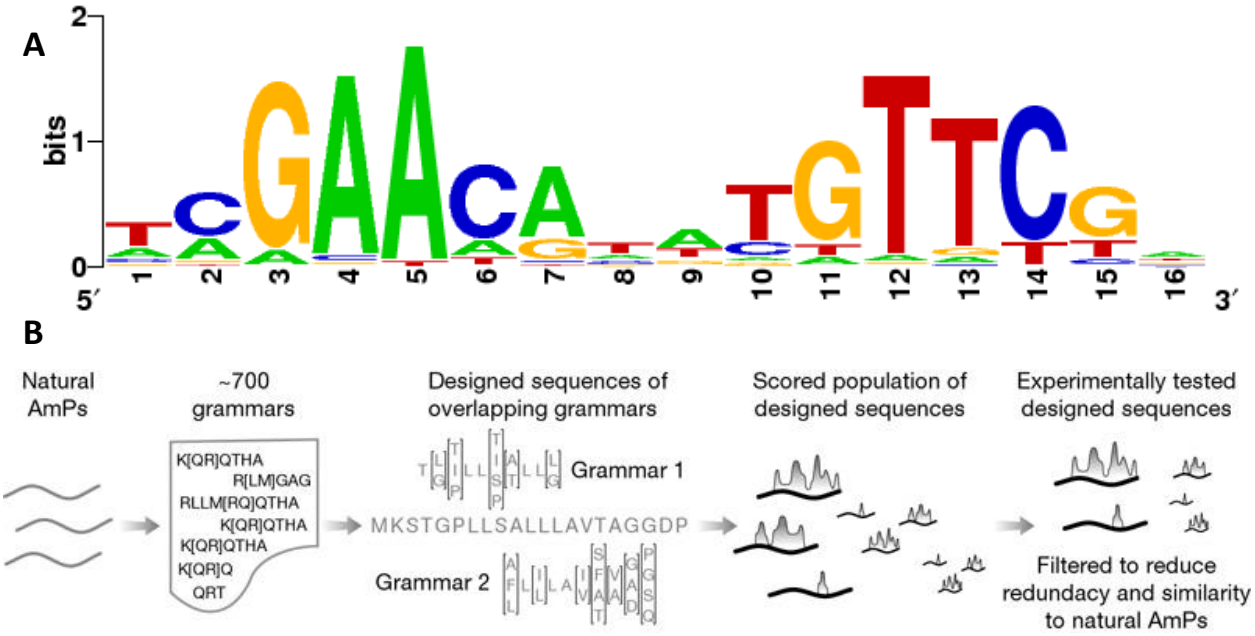


Figure 32: Alternative methods that can supplement the scoring matrix based classification of peptides (A) PSSM Model, (B) grammer of peptides type model from computational linguistics.
(Adapted from Gnomehacker [xliv] and B) adapted from work by Loose *et. al.* [xlv])

The second method that can supplement the scoring matrix is the utilization of n-grams type method from natural language processing for document similarity analysis approach and use sequence of amino acids instead of a bag of words. This method will enable identification of recurring amino acid domains that are significantly responsible for binding in peptides. A similar approach had been employed by Loose et. al in 2006 to design antimicrobial peptides [xlv]. We can also create a decision tree model to add a bonus to similarity scores calculated using the similarity matrix. This way we can also incorporate structural information of peptides in the model.

Chapter 4 - Rational Design of Self-Assembling Peptides

Rational design of self-assembling peptides in the current context is an approach for peptide design which can be used in conjunction with the data-driven design of material-binding peptides to design multifunctional peptides which are capable of both ordered self-assembly and strong binding to a specific material.

This approach encompasses identification of factors responsible for peptide assembly, the study of effects of carefully designed mutations in peptide sequences and their effects in self-assembly behavior of peptides as a starting point. It then proceeds to the creation of a classification model creation using selection rule-based decision tree model to distinguish peptides capable of assembly and eventually goes on to predictions, model validation, and model refinement like data-driven peptide design. This chapter highlights some key details involved in the rational design of Graphite based self-assembling peptides.

4.1 Initial Dataset and Characteristics of Some Self-Assembling Peptides

Unlike graphite binding peptide datasets, which consisted of some 50-60 peptide sequences, self-assembling peptides on graphite have been studied insufficiently and thus there is very little data of previously known graphite based self-assembling peptides. Thus, careful observation is needed on the characteristics of this limited set to generate a good rational design model for self-assembling peptides. Some reasons why this data is limited are that for self-assembly type studies, AFM like advanced characterization methods is needed, and such methods have only existed during last decade, while binding characterization methods have been there for a much longer time. Another reason is that Phages can be directly used for binding type studies while self-assembly needs peptides.

A reliable dataset available for such studies is as shown in figure 33 below. This dataset consists of carefully designed mutant variants of IMVTESSDYSSY (a peptide that has been demonstrated to be an active self-assembling peptide on graphite surface) that are predicted to be of importance in assembling behavior.

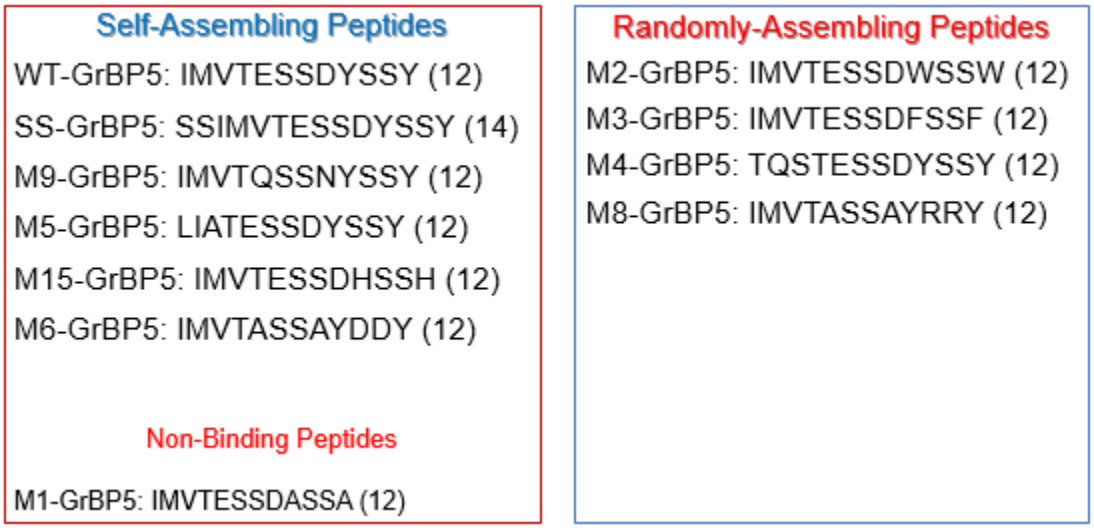


Figure 33: list of available peptides that have been studied for self-assembly behavior on graphite. (Courtesy: Dr. Mehmet Sarikaya, David Starkebaum and Deniz Yucesoy (GEMSEC, University of Washington))

Chris So. *et. al.* in 2012 showcased that surface processes of WT-GrBP5 (IMVTESSDYSSY) self-assembling peptide can be interrogated through rational mutations of the peptide. They identified three chemically distinct domains of (I) hydrophobic (IMV), (II) hydrophilic (TESSD), and (III) aromatic (YSSY) sub-segments that impacted the self-assembly of this peptide as shown in figure 34 below [vi].

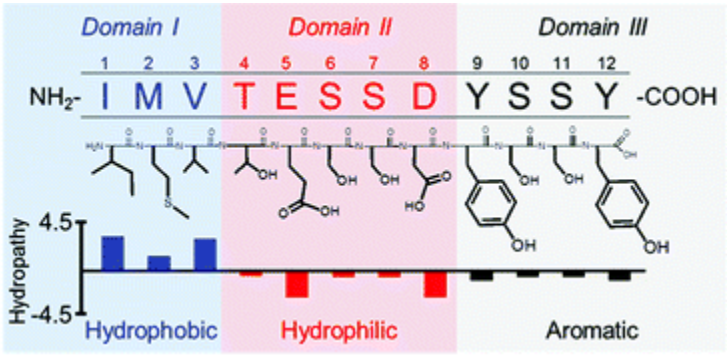


Figure 34: Domains of a GrBP5-WT peptide identified by So. *et al*. which impact self-assembly pattern of GrBP5[vi].

Here aromatic sub-segment consisting of amino acids such as tyrosine (Y) can help in strong interaction with graphitic surfaces through a coupling of π-electrons. The hydrophobic and hydrophilic domains could make up for an amphiphilic tail which can provide intermolecular interactions necessary for long-range order. In M1 mutant of GrBP5, the aromatic parts YSSY was modified to ASSA (A is non-aromatic), to remove the aromatic interactions. ASSA mutation thus led to the loss of binding domain and thus self-assembly on a graphite surface. When this YSSY was replaced by WSSW where W(Tryptophan) is another aromatic amino acid but with better π- interaction with graphite due to a new indole ring, a random self-assembly on graphite surface is observed. Similarly, when YSSY is replaced with FSSF where F is phenylalanine (again aromatic amino acid but with slightly lesser interactivity with graphite due to the absence of OH^- group) there was again random self-assembly, and much weaker interaction existed than in the case of YSSY and WSSW. M15 'IMVTESSDHSSH' tends to self-assemble as well. Thus, all aromatic amino acid cannot be used in general for self-assembly, and it requires specific Y or H amino acids as Interacting moieties. Changing the hydrophobic domain 'IMV' in M4 with similar size but hydrophobic amino acids 'TQS' lead to loss of ordering in self-assembly and thus result in the formation of highly porous and disordered structures. While changing IMV to LIA in M5 mutant restored the hydrophobic domain and created amphiphilic domain but with slightly higher hydropathy. This mutation again led to ordered self-assembly very similar to WT-GrBP5. Thus, it was demonstrated that amphiphilic nature and hydrophobic domain I are crucial for self-assembly in graphite binding peptides. Figure 35 showcases the effects of amphiphilic domain substitutions on self-assembly Pattern studied via time lapsed AFM characterization.

In mutants, M6 and M8 effects of charged aromatic domains were studied. In M6, YSSY was replaced by YDDY to increase the charge in the aromatic domain, which was counterbalanced by replacing E and D with A's in the hydrophilic domain to keep the net charge of peptide same as WT-GrBP5.

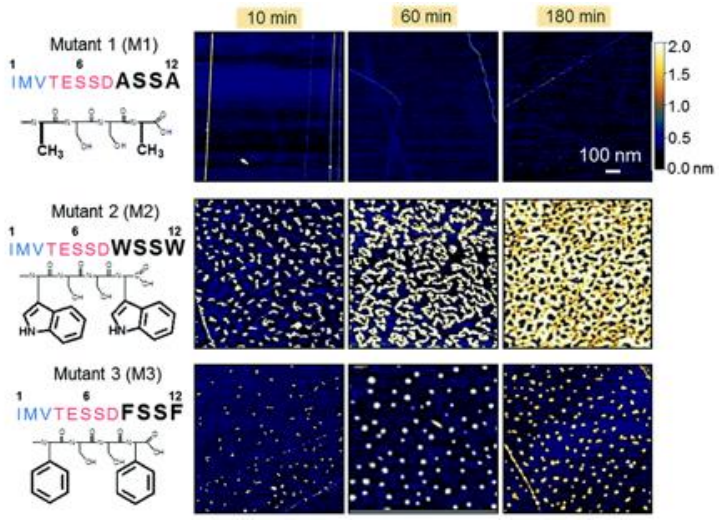


Figure 35: Effects of substitutions changing the amphiphilic domains (regions I and II) in GrBP5 and mutants. (Adapted from work by So. *et. al* [vi])

This strategy worked on MoS_2 for M6, as ordered assembly like WT-GrBP5 occurred, but it did not work as expected for M8 and there was disordered assembly thus showcasing that small variation in residues even though similarly charged can affect the assembly pattern, but it might not affect the assembly process altogether. M6 showed weak self-assembly in graphite too.

In M9 David Starkebaum *et. al.* displaced two carboxylate residues (E and D) from WT-GrBp5 with polar, but non-charged, amide residues (Q and N). This mutant peptide formed crystalline nanostructures on graphite. This simple mutation eliminated the sensitivity of GrBP5-WT to buffer and pH changes [xlvi].

4.2 Overview of Effects of Hydropathicity, Amphiphilicity, Total Charge and Aromaticity in Peptide Self-Assembly.

As can be inferred from the dataset description above, few factors can lead to organized self–assembly while others can result in amorphous assembly. The main factors that were identified to be most influential in the assembly behavior experiments related to self-assembly of peptides on single layer materials include Hydropathicity profile of peptides, the total charge on peptides, the presence of specific

aromatic domains and presence of amphiphilic domain in peptides besides effects of small mutations. The effects of each of these are described below and helps in the creation of rational design discussed in section 4.3. Hydropathicity is the measure of relative hydrophobicity or hydrophilicity of an amino acid residue in peptides. **Kyte-Doolittle scale,** defined by Kyte and Doolittle in 1982, is a widely used analytic scale to detect and quantify the hydropathy of various segments in proteins. Regions with a positive value are hydrophobic, and ones with negative values are hydrophilic [xlvii]. As demonstrated by experiments on mutations M4, M5 and GrBP5-WT peptide an amphiphilic domain is crucial for ordered self-assembly of peptides on graphite. Based on the design of M4, M5, and WT-GrBP5 it can be said that if a peptide shall have following constituents in three domains to be a desirable candidate for self-assembly. It shall have three amino acids forming a hydrophobic domain, five amino acids forming hydrophilic domain and four amino acids forming the binding domain out of these four at least two shall be aromatic amino acids of type Y. These domains will thus be one of the most important considerations when we try to design a slightly flexible but similar rational design.

The surface charge of peptides that interacts with a material has also been shown to impact the assembly of peptides greatly. In this regard experiments involving M6, M9, M8, and WT-GrBP5 are critical. We see that when a peptide has a total negative charge as in the case of WT-GrBP5 (Net Charge = -2 at pH = 7) and M6 (Net Charge = -2 at pH = 7) this had a positive impact on ordered self-assembly. Whereas, when total Peptide charge was Positive as in the case of M8 (Net Charge = +2 at pH = 7) the assembly was random. The neutral charged species M9 shows signs of self-assembly but it was not always ordered, and thus negative total charge might have some positive influence on self-assembly pattern of peptides on materials like graphite, and we will take this into account in building the rational design as well.

The type of aromatic amino acids present in the domain III also seems to have had a significant impact. It was shown in the case of WT-GrBP5, M2, M3, and M15 that aromatic amino acids like ‘H’ and ‘Y’ had a positive impact on self-assembly whereas ‘F’ and ‘W’ had a negative consequence. Another important fact

about the domain III is that it is also responsible for binding and thus it is an essential domain for self-assembly. A rational design would thus include all the observations above and try to be a little more flexible so as to allow some variations that might lead to even better self-assembly behavior. Section 4.3 describes how a selection rule and decision tree based model can encompass such observations in creating a rational design for self-assembly of peptides on graphite.

4.3 Creation of Rational Model Using Selection Rule-Based Decision Tree Method

As described in section 4.2 the main components that need to be accounted for the creation of the rational design of self-assembling peptides on single layer material like graphite are the presence of three domains (hydrophilic, hydrophobic and aromatic (binding capable)). Also, we need to consider effects of total charge, length, constituents, and hydropathicity of these domains.

Thus, a proposed rational design which rigidly defines the first domain as hydrophobic, second domain as a hydrophilic and third domain as aromatic but allows for some mutations is shown in figure 36 below.

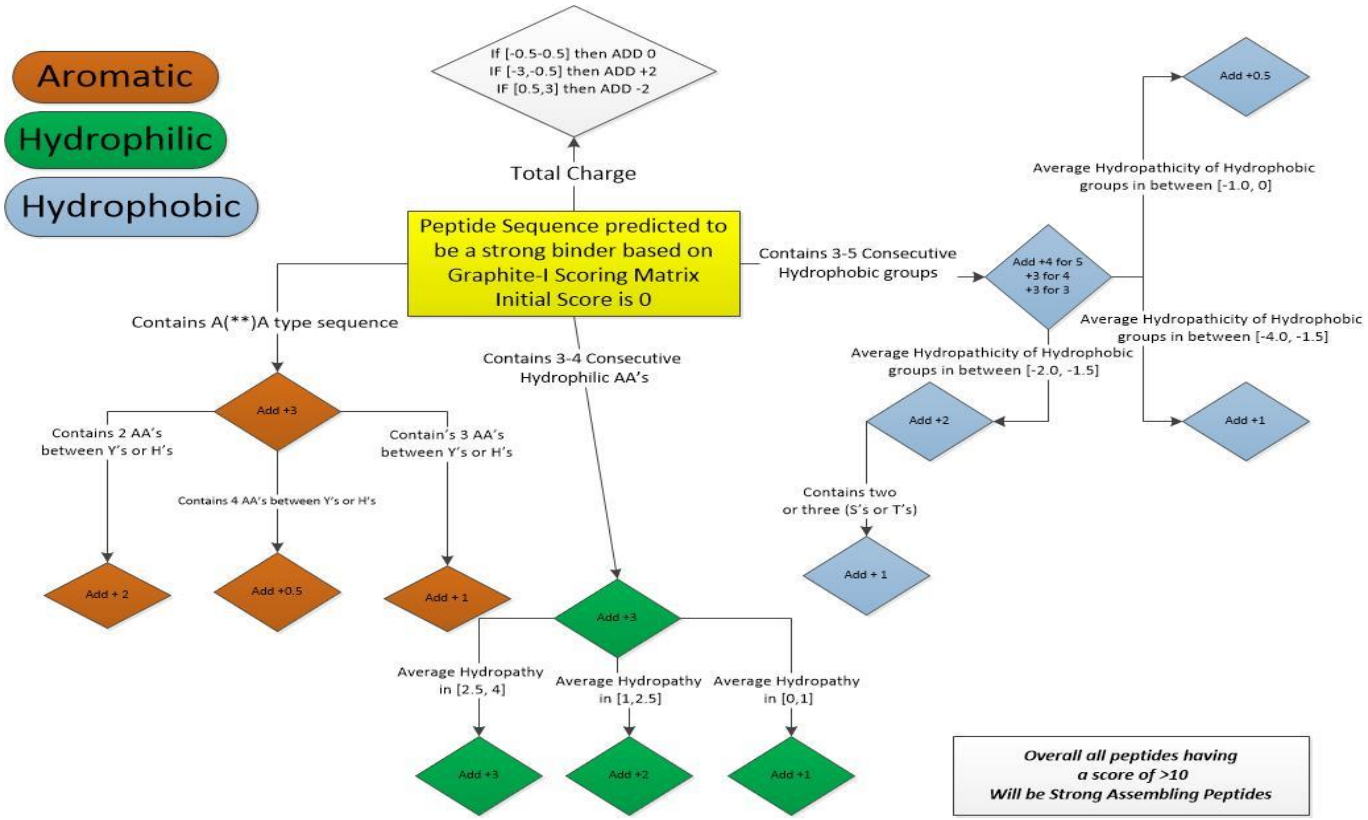


Figure 36: Decision tree and selection rule based rational design to classify self-assembling capabilities of peptides on graphite surface

This model initially assigns a score of 0 to any peptide and then evaluates it for Aromatic domains, hydrophilic domains, hydrophobic domains and total charge. If any peptide is shown to contain most of the desirable characteristics, it is bound to get a very high score based on this model, and any peptide which scores more than ten will be a potential candidate for strong self-assembly. The model details are described below.

For aromatic domain, if a peptide has A**A type sequence in last few amino acids, where A is either 'H' or 'Y' and ** is some gap between those two A's then they get a bonus of +3 points. If in addition to A**A they have exactly 2 Amino acids new score of +2 is added. The addition of +2 is done to be highly selective of YSSY or HSSH type peptide sequences which are known to be good for self-assembly. If there are 3 Amino acids in between A's, we give a slightly lower bonus of +1 point. Similarly, if there are 4 Amino acids between A's we give an even lower bonus of +0.5 points. By adding these flexibilities, we will allow for other variations like the length of the aromatic domain and variety of sequences in between them but still, stick to things that have been demonstrated to give optimal results for assembling studies.

For Hydrophobic domain, the current mutants only consisted of 3 AAs. However, to allow little more flexibility, the proposed model allows for 3-4 AAs amongst first few amino acids. If the peptide sequence contains either 3 or 4 consecutive amino acids of positive hydrophobicity index based on the Kyte-Doolittle scale we add +3 to the score for assembly. If the hydropathicity average of these 3-4 amino acids is in the range of [0, 1], we add 1 point to score. If it is in the range of [1, 2.5], we add +2 to score, and if average hydropathy of this segment is in the range [2.5, 4], we add +3 to score. These additions are done to favor most hydrophobic amino acids over somewhat less hydrophobic ones.

For Hydrophilic domain, we check if there are 3-5 consecutive amino acids in the middle section of the peptide and in case they are all hydrophilic i.e. they all have negative hydrophobicity then we add +4 when there are five consecutive amino acids like that. We add +3 if there are either 3 or 4 consecutive hydrophilic amino acids. In addition to these scores if the hydrophilic domain has average hydropathy

value in the range [-4.0, -2.0] then we add +1. If average is in the range [-2, -1] then we add +2, and if this average is in the range [-1, 0], then we add 0.5 to favor moderate hydrophilicity in middle region over extremely high or very low hydrophilicity values. In addition to this if this middle region has two or three 'S' or 'T' type amino acids then +1 is added to score. +1 is added to score to allow S's and T's to be filler amino acids and allow some conformational flexibility in the middle region.

Finally, for total charge quantification, if the total charge is between [-0.5, 0.5] do nothing. If the total charge is negative and in the range of [-3, -0.5] add +2 to score. If the total charge is positive in the range of [0.5, 3], then subtract -2 from the score. This way we can penalize the positively charged peptides as we predicted based on the observations from experiments done on M6, M8, and M9.

4.4 Application of Rational Design on 100,000 Randomly Generated Peptides

Out of 100,000 random peptides, which are also studied for binding properties based on the BLOSUM62 scoring matrix, the peptides that scored above 10 in decision tree model are MVVENPSPSLPA, LMLGRSRSLSPY, ALFSNSPYAATQ, FVLQPPPSSFSQ, LLFTSYDHYHQM, and LMIPNTDQPAY. All these peptides are shown in figure 37 along with their hydropathicity plots. We can see all these typically contain the tree domains (hydrophobic, hydrophilic and aromatic) in general. These are not perfectly following the domains and thus allow for small variations which may or may not be useful for better self-assembly on graphite. These peptides will now be evaluated for binding based similarity analysis using the perturbed BLOSUM62 matrix that might better classify these peptides and assign a similarity score. Then, Peptides among these proposed strong assemblers which also show predictable strong binding capability by getting a high similarity score with strong binders will be synthesized, characterized and used for model improvement.

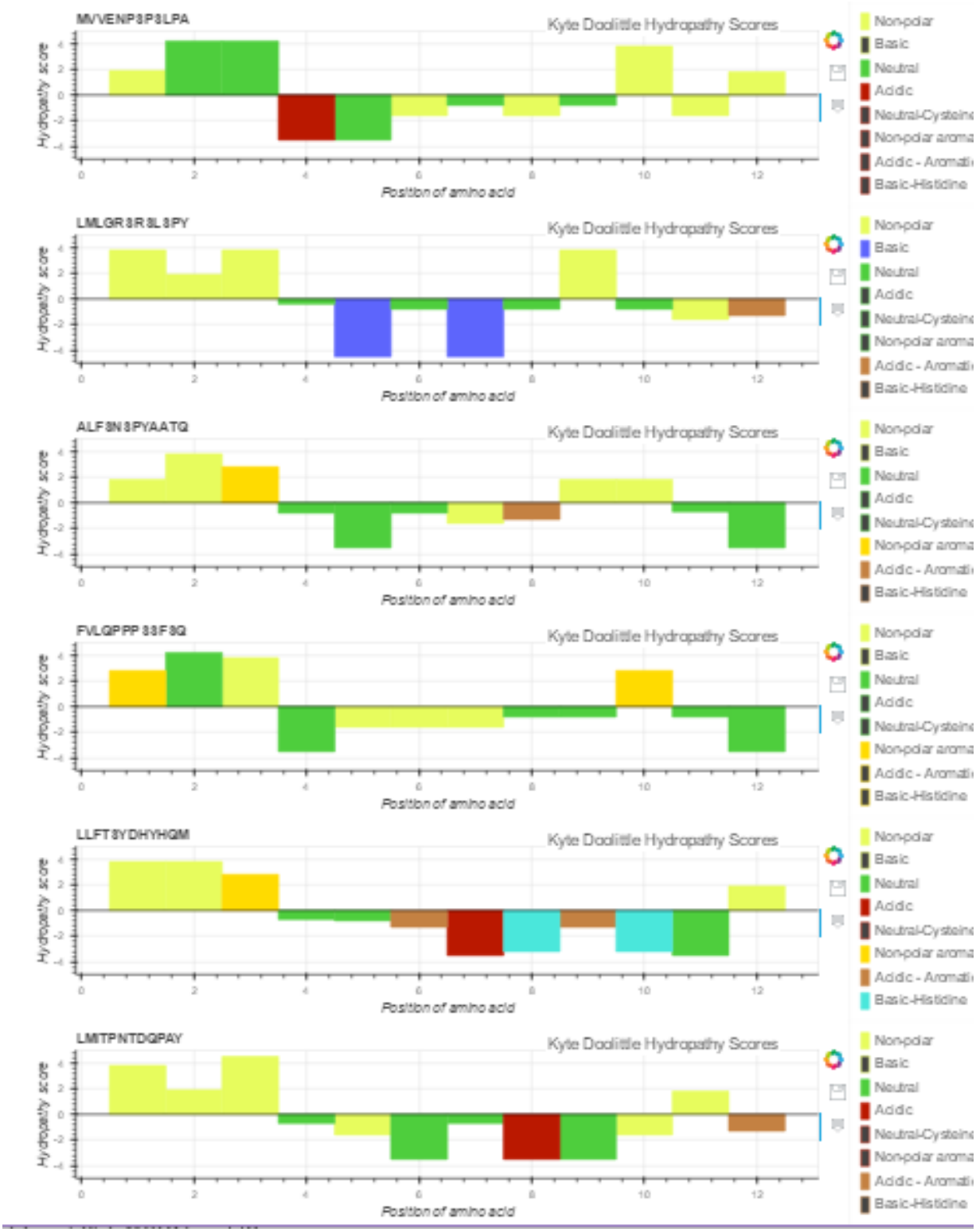


Figure 37: New peptides predicted to be capable of self-assembly out of 100000 random peptides

4.5 Identification of Probable Super Peptides Capable of Binding and Assembly

The best method to identify super peptides capable of binding, as well as self-assembly, is to combine the logical design model for self- assembly with the scoring matrix model used in binding classification. A rational approach will be to generate millions of random peptide sequences and do a similarity analysis for identification of strong binders. Then identification and filtering may be done for few thousands of strong binding capable peptides which will then be filtered for self-assembly based on rational design model. We can thus narrow the results to some tens or at maximum some hundreds of peptides which are capable of both self-assembly and strong binding. We can then synthesize these and validate our predictions or make improvements to these models as shown in figure 6 and 7 in section 1.5.

Another way of doing it, which might be more efficient as well would be to generate some thousands (or millions if possible) of random peptides which score more than 10 in self-assembly model and then filter them out based on strong similarity scores with strong binders. Once peptides with strong binding and self-assembly capabilities have been identified, synthesized and characterized they can be studied for other properties such as anti- bacterial coatings, and novel electronic interfaces thus generating multifunctional super peptides.

4.6 Future Work for Refinement and Validation of Self-Assembling Rational Design

As mentioned in section 4.5, once the similarity scores for these peptides are determined and the strong assemblers have been identified they will then be synthesized and characterized in terms of their binding and assembly behavior using the techniques such as AFM and Fluorescence Microscopy. These results will then be used to validate our predictions and refine the data models. Few refinements that can be applied to self-assembly based rational design are as below.

1. The fractional factorial design of experiments, the focused hierarchical design or the D-Optimal design-based approach can be used to design new mutations and acquire more information out

of a small number of experiments rather than making one mutation at a time and taking a longer and less economical experimental route.[xlviii, xlix]

2. The rational design can be altered to obtain more flexible sequence regarding where the domains lie instead of currently fixed order of having a hydrophobic tail, then hydrophilic interior and aromatic head.
3. The model can be applied to other similar single layer materials like MoS_2 and BN to gain a more fundamental understanding of peptide materials interactions and identification of properties of subdomains that might be responsible for binding and self-assembly.
4. Application of ensemble machine learning methods involving a combination of n-partite graph algorithms, n-grams type approach for amino acids or grammer of peptides type methods from computational linguistics, position-specific scoring matrix, combinatorial dataset, and kernel based other scoring methods can be tried to model assembly behavior when more data in the order of 30-40 mutant peptides are available [l, xlv, li].

Chapter 5 - Future work

Until this point, we had been able to make some general models to predict material binders based on combinatorially selected peptides based and their binding characterization data which typically contained 60-100 peptides. We have also been able to model active self-assembling peptides based on the rational design of some mutants of these heavy binders that are also able to self-assemble on specific substrates. This work can be expanded in multiple directions as mentioned below.

5.1 Model Enhancements to Design Graphene Super Peptides

One way to expand on current work is to design multifunctional super peptides that are not only able to bind and assemble in an ordered manner but also enhance material properties by adding novel electronic, optical, magnetic or biological properties like catalyzing biomineralization or generation of an antibacterial surface on materials. These methods will involve either designing of chimeric peptides that can act as molecular glue for materials with special properties or addition of another domain that can act as linker or property enhancer in addition to the aromatic and amphiphilic domains already present in peptides. One such design could be to incorporate developing a Fast DNA data reader by linking Polymerase like enzyme or polymerase derived peptides and graphene-based FET device as shown in Figure 38 below.

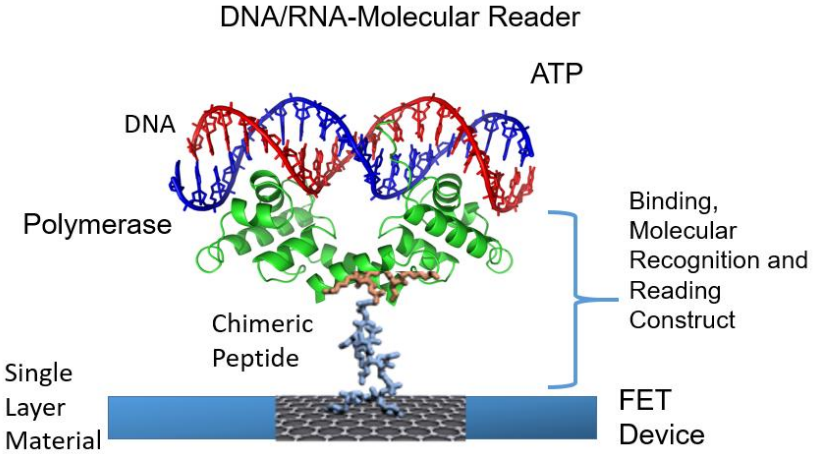


Figure 38: Schematic diagram indicating a potential DNA Reader Construct using graphene-based super-peptide (Courtesy: Dr. Mehmet Sarikaya (GEMSEC Labs, University of Washington))

5.2 Modelling Single Layer Materials and Generation of a Library of Useful Peptides

We can now utilize the modeling techniques used in studies of graphite based super peptides to generate a library and database of models for useful materials like semiconductors, insulators, and conductors. We can start out with single layer material systems like MoS_2, WSe_2, WS_2, $MoSe_2$, and hBN as they will have very similar features with graphene-based peptides. These libraries will help in obtaining more fundamental understanding about what kinds of mutations and commonalities can exist in modeling a variety of different systems. Figure 39 showcases a sample periodic table based demonstration of these useful peptides.

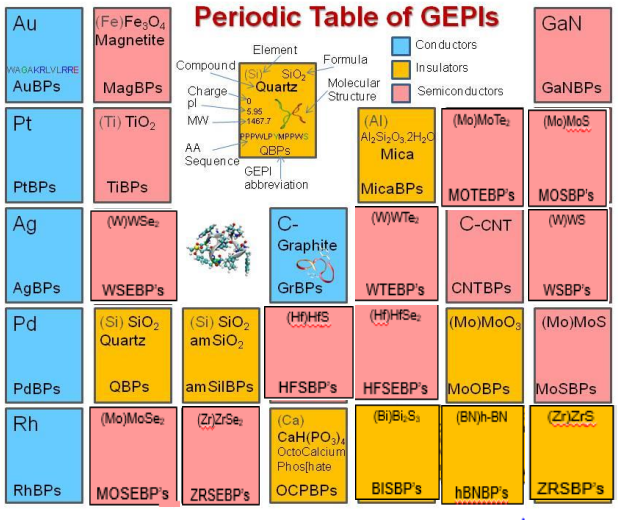


Figure 39: Schematic diagram for periodic table type demonstration of solid-binding peptides. (Courtesy: Dr. Mehmet Sarikaya (GEMSEC Labs, University of Washington)))

5.3 Design of Multifunctional Materials for Bioelectronic Applications

To realize some of the bioelectronic applications we need to integrate devices with biological systems and, thus, it is important to create suitable interfaces between electronic materials and biologically active molecules. Peptides and DNA can be some useful materials for this purpose owing to their unique capabilities of specific interactions with materials via their molecular recognition capacity and their properties to organize and self-assemble in a specific manner on a substrate. One

example of a bioelectronic interface that might have an enormous impact in ways humans interact with electronic systems is the interface between neurons and electronic materials like graphene. This kind of systems can be a foundation stone for creating innovative brain-computer interfaces, and for finding cures for neurodegenerative diseases or acute injuries to the nervous system which might result in loss of neurons and damage of neurites. It has been shown by Li et. al in a study during 2011, that graphene has great biocompatibility with neurites as it does not negatively affect cell viability or morphology and it can promote sprouting and outgrowth of neurites during cell growth stages[lii] . Graphene behaved comparable to tissue culture polystyrene (TCPS) in biocompatibility and outperformed TCPS in neurite sprouting. This study used polylysine peptides, which have high hydrophilicity and a net positive charge to interface the graphene with neurons. In another study, So et. al in 2016 showcased that peptides containing 12 amino acids and specific domains (including one hydrophilic domain) can showcase an ordered self-assembly on graphene and similar single-layered materials [vi]. They also affect the electrical

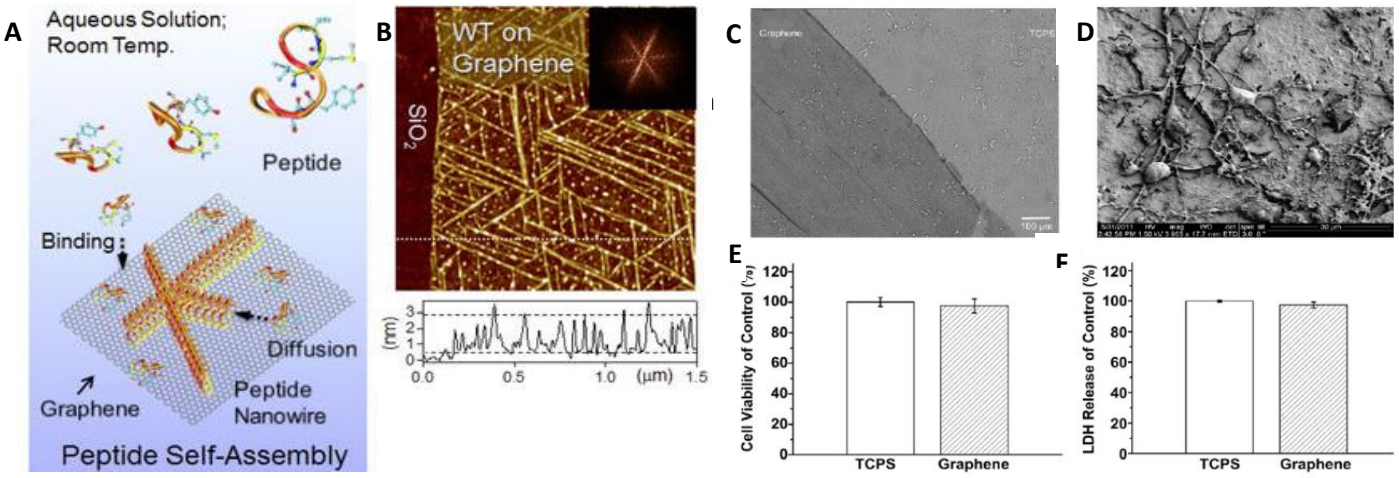


Figure 40: Experiments based on graphene, peptides & neurons to make a biocompatible bio-electronic interface. **A)** Schematic showing the peptide self-organization with a series of the surface processes: binding, diffusion, and self-organization. **B)** AFM image of the self-assembling peptide on single-layer graphene. The bright lines are self-assembled peptides forming organized nanostructures. **C)** An optical image of neurons cultured on the border of graphene (left) and TCPS (right). **D)** Scanning electron microscopy image of neurons on graphene, **E)** MTT-measured viability of neurons cultured on TCPS and graphene after seven days, **F)** LDH activity of neurons after seven days incubation on TCPS and graphene
(A and B are Adapted from the work of So. *et. al* [vi]. C, D, E and F are adapted from works of Li et.al. [lii])

In another study by Urich et. al in 2015, phage display selected peptides were shown to be able to cross the blood-brain barrier while carrying other molecules with them [liii] . The study demonstrates molecules like graphene can be assisted to cross the blood brain barrier with the help of transport peptides. Thus, peptides when ideally designed for the carriage, binding, assembly and electronic doping, have a potential for generating novel bioelectronic interfaces.

5.4 Creation of a Comprehensive Software Toolkit for Biomimetic Intelligence and Bioelectronic Applications for Peptides.

To develop super peptides which have all necessary properties like strong binding capability, ordered self-assembling capability, electronic and optical doping properties and biocompatibility, a series of experiments and simulations need to be performed, and a vast amount of data processing and maintenance will thus need to be involved. To accomplish this data processing, data maintenance and interoperability of tools we can create an integrated suite of applications to help users in designing novel peptides, by combining experimental data, machine learning, predictive modeling, and molecular dynamics. One proposed solution is to create a GEMSEC GEPI and Biomimetic intelligence suite. The suite will consist of tools such as PYMOl for molecular visualization, Schrodinger's materials science and biologics suite for molecular modeling, bioavailability and cheminformatics type of studies, Knime extensions for data pipelining as mentioned in section 1, and some advanced in-house applications. In-house applications will include products like Super pep which will be a library of machine learning algorithms for peptide design and similarity analysis, Bio Intel a digital signal processing and molecular modeling toolkit for peptide and materials based intelligence applications and finally GEPI lib an

interactive interface for peptide materials interaction predictions and properties visualization. Figure 41 showcases a sample schematic of this Suite of applications.

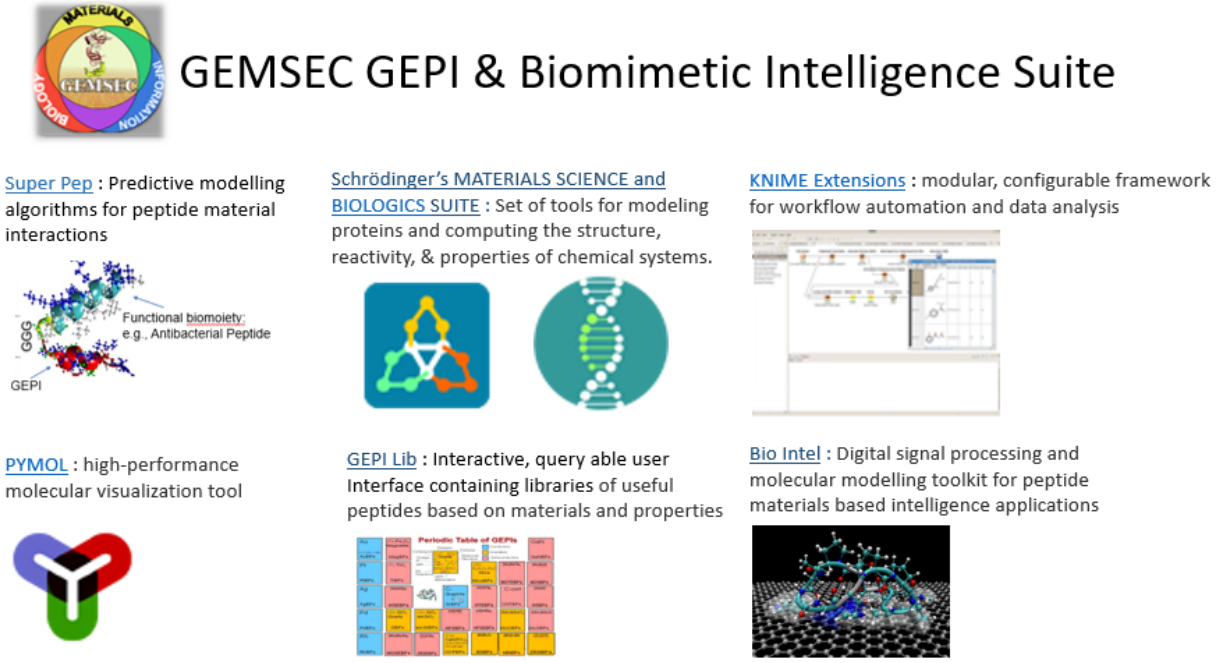


Figure 41: Proposed components of the GEMSEC GEPI and Bio-mimetic intelligence suite for Super peptide design and analysis of material peptide interactions.

IV Summary

In the present study, we have tried to combine a rational design approach for designing self-assembling peptides with the data-driven design and combinatorial selection techniques of solid-binding peptides to narrow our pool of peptides that can be tested via actual synthesis and characterization. We used phage display technique to select graphite binding peptides combinatorially, then applied some Needleman-Wunch type similarity scoring techniques and greedy algorithms to generate a classification model that can identify possibilities of robust and weak interaction based on amino acid sequence present in peptides. We then applied a rational design to model graphene-based self-assembling peptides by creating a decision tree model to allow only certain peptides with specific constructs (hydrophilic, hydrophobic and aromatic binding domains) to be filtered out as self-assembly capable peptides. In this manner, we can first filter out 1000 active binding capable peptides out of 100,000 random peptides and then get down to only six peptides which are also found capable of strong self-assembly. In future experiments, these 6 and a few more peptides will be synthesized, and the model will be validated or enhanced.

Some preliminary experiments with the application of similarity scoring algorithms comprising of conventional scoring matrices like BLOSUM and PAM yielded slightly incorrect results to classify actively binding strong peptides on graphene surface which were slightly rectified by designing a new scoring model and with the addition of more data. The rational design model still needs more testing and data to generate better models.

We also created a software design to model such systems and came up with a design to include a collection of tools like Knime, Python, R, SQL server and C++ in conjunction for modeling such systems. Also, we can now incorporate Schrodinger's Materials Science Suite and Biologics Suite and some in-house algorithms and signal processing techniques being developed at GEMSEC to create a more comprehensive toolkit to model such systems in future.

V Conclusions

This work demonstrates the application of modeling techniques and experimental design to construct novel *de novo* designed spontaneously organizing super peptides on single layer solid-state materials e.g., graphene. The work proposes implementation of such techniques and provides a comprehensive software toolkit to design peptides for practical applications e.g., bio-enabled electronics. The approach creates a hybrid model to identify the best sequence for specific function out of a pool of some 1000s or 10^6s of experimentally determined peptides that provide self-assembly and materials binding characteristics of peptides interacting with graphene. It was found that models that were generic such as BLOSUM scoring Matrix, were not very good at identifying solid-binding peptides but they provided excellent seed models that can be perturbed *via* greedy procedures to generate a novel scoring matrix that can better capture the details of sequence similarity amongst strong binders.

The methodology introduced herein is an intelligent design model that includes a decision-tree type method to limit self-assembling peptides and which can get better with more data on assembly behavior of newly predicted sequences. The approach of combining combinatorial selection, data-driven design, and rational design allows for the inclusion of small random variations that might affect materials property greatly but might be ignored if only strict rational design approaches were considered.

For future, modular design of the peptides are proposed for novel device applications based on the concept of self-organizing peptides on single layer materials. These include a DNA Reader or a Coherent Neural Interface, significantly accelerating DNA sequencing and neural network research. Finally, a comprehensive software suite is proposed that can encompass all available experimental and computational data about these peptides and solid materials including binding strength, chemical properties, self-assembly domains, and modeling approaches by learning from the past observations and create novel futuristic designs in complex technological systems to accelerate research at the cross-sections or physics, biology, informatics, computational modeling, materials science and engineering.

VI References

1. [i] Rode, Bernd Michael. "Peptides and the origin of life 1." *Peptides* 20, no. 6 (1999): 773-786.

2. [ii] "Peptides Vs Proteins." Peptide Glossary. Accessed June 2, 2017.

3. https://www.peptidesciences.com/glossary/peptides-vs-proteins/

4. [iii] Chaparro-Riggers, Javier F., Karen M. Polizzi, and Andreas S. Bommarius. "Better library design: data-driven protein engineering." *Biotechnology journal* 2, no. 2 (2007): 180-191.

5. [iv] "Combinatorial Selection Methods." *Encyclopedia of Cancer*, 2011, 953-57.

6. [v] Gungormus, M. (2012). *Selection, design and applications of solid binding peptides for controlled biomineralization*.

7. [vi] So, Christopher R., Yuhei Hayamizu, Hilal Yazici, Carolyn Gresswell, Dmitriy Khatayevich, Candan Tamerler, and Mehmet Sarikaya. "Controlling Self-Assembly of Engineered Peptides on Graphite by Rational Mutation." *ACS Nano* 6, no. 2 (2012): 1648-656.

8. [vii] Banwell, Eleanor F., Edgardo S. Abelardo, Dave J. Adams, Martin A. Birchall, Adam Corrigan, Athene M. Donald, Mark Kirkland, Louise C. Serpell, Michael F. Butler, and Derek N. Woolfson. "Rational design and application of responsive α-helical peptide hydrogels." *Nature Materials* 8, no. 7 (2009): 596-600.

9. [viii] Oren, Ersin Emre, Candan Tamerler, Deniz Sahin, Marketa Hnilova, Urartu Ozgur Safak Seker, Mehmet Sarikaya, and Ram Samudrala. "A novel knowledge-based approach to design inorganic-binding peptides." *Bioinformatics* 23, no. 21 (2007): 2816-822.

10. [ix] Sarikaya, Mehmet, Candan Tamerler, Alex K. -Y. Jen, Klaus Schulten, and François Baneyx. "Molecular biomimetics: nanotechnology through biology." *Nature Materials* 2, no. 9 (2003): 577-85.

11. [x] Boskey, Adele L. "Matrix Proteins and Mineralization: An Overview." *Connective Tissue Research* 35, no. 1-4 (1996): 357-63.

12. [xi] Smith, G. "Filamentous fusion phage: novel expression vectors that display cloned antigens on the virion surface." *Science* 228, no. 4705 (1985): 1315-317.

13. [xii] Hayamizu, Yuhei, Christopher R. So, Sefa Dag, Tamon S. Page, David Starkebaum, and Mehmet Sarikaya. "Bioelectronic interfaces by spontaneously organized peptides on 2D atomic single layer materials." *Scientific Reports* 6, no. 1 (2016).

14. [xiii] Fourment, Mathieu, and Michael R. Gillings. "A comparison of common programming languages used in bioinformatics." *BMC Bioinformatics* 9, no. 1 (2008): 82.

15. [xiv] Nanz, Sebastian, and Carlo A. Furia. "A comparative study of programming languages in Rosetta Code." In *Software Engineering (ICSE), 2015 IEEE/ACM 37th IEEE International Conference on*, vol. 1, pp. 778-788. IEEE, 2015.

16. [xv] Bassi, Sebastian. "A Primer on Python for Life Science Researchers." *PLoS Computational Biology* 3, no. 11 (2007).

17. [xvi] Ekmekci, Berk, Charles E. Mcanany, and Cameron Mura. "An Introduction to Programming for Bioscientists: A Python-Based Primer." *PLOS Computational Biology* 12, no. 6 (2016).

18. [xvii] Cock, Peter JA, Tiago Antao, Jeffrey T. Chang, Brad A. Chapman, Cymon J. Cox, Andrew Dalke, Iddo Friedberg et al. "Biopython: freely available Python tools for computational molecular biology and bioinformatics." *Bioinformatics* 25, no. 11 (2009): 1422-1423.

19. [xviii] Pirhadi, Somayeh, Jocelyn Sunseri, and David Ryan Koes. "Open source molecular modeling." *Journal of Molecular Graphics and Modelling* 69 (2016): 127-143.

20. [xix] Gentleman, Robert C., Vincent J. Carey, Douglas M. Bates, Ben Bolstad, Marcel Dettling, Sandrine Dudoit, Byron Ellis et al. "Bioconductor: open software development for computational biology and bioinformatics." *Genome biology* 5, no. 10 (2004): R80.

21. [xx] Vries, Andrie De. "Classification of galaxy type from images using Microsoft R Server." *Proceedings of the International Astronomical Union* 12, no. S325 (2016): 139-44.

22. [xxi] Warr, Wendy A. "Scientific workflow systems: Pipeline Pilot and KNIME." *Journal of Computer-Aided Molecular Design* 26, no. 7 (2012): 801-04.

23. [xxii] Berthold, Michael R., Nicolas Cebron, Fabian Dill, Thomas R. Gabriel, Tobias Kötter, Thorsten Meinl, Peter Ohl, Kilian Thiel, and Bernd Wiswedel. "KNIME-the Konstanz information miner: version 2.0 and beyond." *AcM SIGKDD explorations Newsletter* 11, no. 1 (2009): 26-31.

24. [xxiii] Schindelin, Johannes, Curtis T. Rueden, Mark C. Hiner, and Kevin W. Eliceiri. "The ImageJ ecosystem: An open platform for biomedical image analysis." *Molecular Reproduction and Development* 82, no. 7-8 (2015): 518-29.

25. [xxiv] Bochevarov, Art D., Edward Harder, Thomas F. Hughes, Jeremy R. Greenwood, Dale A. Braden, Dean M. Philipp, David Rinaldo, Mathew D. Halls, Jing Zhang, and Richard A. Friesner. "Jaguar: A high-performance quantum chemistry software program with strengths in life and materials sciences." *International Journal of Quantum Chemistry* 113, no. 18 (2013): 2110-2142.

26. [xxv] Beisken, Stephan, Thorsten Meinl, Bernd Wiswedel, Luis F De Figueiredo, Michael Berthold, and Christoph Steinbeck. "KNIME-CDK: Workflow-driven cheminformatics." *BMC Bioinformatics* 14, no. 1 (2013): 257.

27. [xxvi] Karim, Md Rezaul, Audrey Michel, Achille Zappa, Pavel Baranov, Ratnesh Sahay, and Dietrich Rebholz-Schuhmann. "Improving data workflow systems with cloud services and use of open data for bioinformatics research." *Briefings in Bioinformatics* (2017): bbx039.

28. [xxvii] Ali, Syed Mohd, Noopur Gupta, Gopal Krishna Nayak, and Rakesh Kumar Lenka. "Big data visualization: Tools and challenges." *2016 2nd International Conference on Contemporary Computing and Informatics (IC3I)*, 2016.

29. [xxviii] FliTrx Peptide Display. Accessed June 1, 2017. http://utminers.utep.edu/rwebb/html/flitrx_peptide_display.html.

30. [xxix] Devlin, J., L. Panganiban, and P. Devlin. "Random peptide libraries: a source of specific protein binding molecules." *Science* 249, no. 4967 (1990): 404-06.

31. [xxx] Biolabs, New England. "Ph.D.™-12 Phage Display Peptide Library." Ph.D.™-12 Phage Display Peptide Library | NEB. Accessed June 2, 2017. https://www.neb.com/products/e8111-phd-12-phage-display-peptide-library.

32. [xxxi] Yucesoy, Deniz Tanil, and Candan Tamerler. *Genetically engineered solid binding peptides (GEPI) for surface biofunctionalization applications: immobilization of enzymes and antimicrobial peptides on solids*. diss.

33. [xxxii] Koonin EV, Galperin MY. Sequence - Evolution - Function: Computational Approaches in Comparative Genomics. Boston: Kluwer Academic; 2003. Chapter 2, Evolutionary Concept in Genetics and Genomics. Available from: https://www.ncbi.nlm.nih.gov/books/NBK20255/

34. [xxxiii] "User:Hannes Röst." User:Hannes Röst - Wikimedia Commons. Accessed June 2, 2017. https://commons.wikimedia.org/wiki/User:Hannes_R%C3%B6st#/media/File:BLOSUM62.gif.

35. [xxxiv] "Sequence alignment." Wikipedia. June 1, 2017. Accessed June 3, 2017. https://en.wikipedia.org/wiki/Sequence_alignment.

36. [xxxv] Needleman, Saul B., and Christian D. Wunsch. "A general method applicable to the search for similarities in the amino acid sequence of two proteins." *Journal of molecular biology* 48, no. 3 (1970): 443-453.

37. [xxxvi] Smith, Temple F., and Michael S. Waterman. "Identification of common molecular subsequences." *Journal of molecular biology* 147, no. 1 (1981): 195-197.

38. [xxxvii] "Substitution matrix." Wikipedia. June 2, 2017. Accessed June 3, 2017. https://en.wikipedia.org/wiki/Substitution_matrix.

39. [xxxviii] Dayhoff, M. O., R. M. Schwartz, and B. C. Orcutt. "22 A Model of Evolutionary Change in Proteins." In *Atlas of protein sequence and structure*, vol. 5, pp. 345-352. National Biomedical Research Foundation Silver Spring, MD, 1978.

40. [xxxix] "Point accepted mutation." Wikipedia. January 24, 2017. Accessed June 3, 2017. https://en.wikipedia.org/wiki/Point_accepted_mutation.

41. [xl] Sung, Wing-Kin. *Algorithms in bioinformatics: A practical introduction*. CRC Press, 2009.

42. [xli] Henikoff, S., and J. G. Henikoff. "Amino acid substitution matrices from protein blocks." *Proceedings of the National Academy of Sciences* 89, no. 22 (1992): 10915-0919.

43. [xlii] "BLOSUM." Wikipedia. June 2, 2017. Accessed June 3, 2017. https://en.wikipedia.org/wiki/BLOSUM.

44. [xliii] "Greedy algorithm." Wikipedia. June 1, 2017. Accessed June 5, 2017. https://en.wikipedia.org/wiki/Greedy_algorithm.

45. [xliv] By Gnomehacker at English Wikipedia, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=34623350

46. [xlv] Loose, Christopher, Kyle Jensen, Isidore Rigoutsos, and Gregory Stephanopoulos. "A linguistic model for the rational design of antimicrobial peptides." *Nature* 443, no. 7113 (2006): 867-869.

47. [xlvi] Starkebaum, David Alan, and Mehmet Sarikaya. *Controlling long-range ordered self-assembly of solid-binding peptide monolayers on atomically flat layered materials*.

48. [xlvii] Kyte, Jack, and Russell F. Doolittle. "A simple method for displaying the hydropathic character of a protein." *Journal of molecular biology* 157, no. 1 (1982): 105-132.

49. [xlviii] Mee, Roger P., Tim R. Auton, and Phillip J. Morgan. "Design of active analogues of a 15-residue peptide using D-optimal design, QSAR and a combinatorial search algorithm." *The Journal of peptide research* 49, no. 1 (1997): 89-102.

50. [xlix] Muthas, Daniel, Per M. Lek, Johanna Nurbo, Anders Karlén, and Torbjörn Lundstedt. "Focused hierarchical design of peptide libraries—follow the lead." *Journal of Chemometrics* 21, no. 10-11 (2007): 486-495.

51. [l] Giguère, Sébastien, François Laviolette, Mario Marchand, Denise Tremblay, Sylvain Moineau, Xinxia Liang, Éric Biron, and Jacques Corbeil. "Machine learning assisted design of highly active peptides for drug discovery." *PLoS Comput Biol* 11, no. 4 (2015): e1004074.

52. [li] Liu, Geng, Dongli Li, Zhang Li, Si Qiu, Wenhui Li, Cheng-chi Chao, Naibo Yang et al. "PSSMHCpan: a novel PSSM-based software for predicting class I peptide-HLA binding affinity." *Giga Science* 6, no. 5 (2017): 1-11.

53. [lii] Li, Ning, Xuemin Zhang, Qin Song, Ruigong Su, Qi Zhang, Tao Kong, Liwei Liu, Gang Jin, Mingliang Tang, and Guosheng Cheng. "The promotion of neurite sprouting and outgrowth of mouse hippocampal cells in culture by graphene substrates." *Biomaterials* 32, no. 35 (2011): 9374-382.

54. [liii] Urich, Eduard, Roland Schmucki, Nadine Ruderisch, Eric Kitas, Ulrich Certa, Helmut Jacobsen, Christophe Schweitzer et al. "Cargo Delivery into the Brain by in vivo identified Transport Peptides." *Scientific reports* 5 (2015).