# SPECIFICITY AND CONSTRAINTS IN PEPTIDE-PROTEIN BINDINGS IN THE MOUSE PROTEOME

## Report for 3rd Year Research Project

February 21, 2016

---

Dhruv SHARMA

# Contents

Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome

# Part I

# Introduction

This report details the work done towards the fulfillment of the requirements of the department of Physics at Ecole Polytechnique. I undertook this research project under the guidance of Dr. Remi Monasson at the Laboratoire de Physique Theorique at the Ecole Normale Superieure in Paris.

The aim of the project was to study the interactions between short peptide chains and a specific section of signaling proteins called PDZ Domains. One of the aspects that we study here is the specificities of interactions between peptides and PDZ domains. It is well known that macromolecules such as proteins and enzymes interact in a specific manner with other macromolecules and biomolecules. What interested us over the course of the study are the constraints present in the peptide sequences due to the specificity of their interactions with PDZ domains. We will also have an occasion to understand similar constraints on the PDZ domain sequences.

This report is organized as follows. After a brief introduction to the biological importance of PDZ Domains, we explain the experiments performed by **Insert reference here**. Using these experiments, Stifler et al created a model which is capable of predicting whether a peptide will bind to a PDZ domain given the sequence of the peptide. We shall explain the data that Stifler et al have provided. The first two models that we propose utilise the data provided by Stifler et al.

Once the data presented and the biological context established, we present a first model which seeks to understand the constraints imposed on the peptide sequences under the effect of mutations. We present the results derived from this model and discuss the limitations. A second improved model is then proposed which considers error rates as probabilities. We present some interesting observations on the basis of this model. In particular, we show how certain positions are particularly constrained over all peptides and present a simple way of the calculating the level of constraint.

Finally, to render the study of peptide-PDZ domain specificity complete, we explain how we could integrate the PDZ Domain sequences into the modeling. This is done by a regression method called the *Lasso*. We shall have the chance to present the lasso method in more detail in the relevant section.

We conclude with a summary of our findings and possible directions of further improvements.
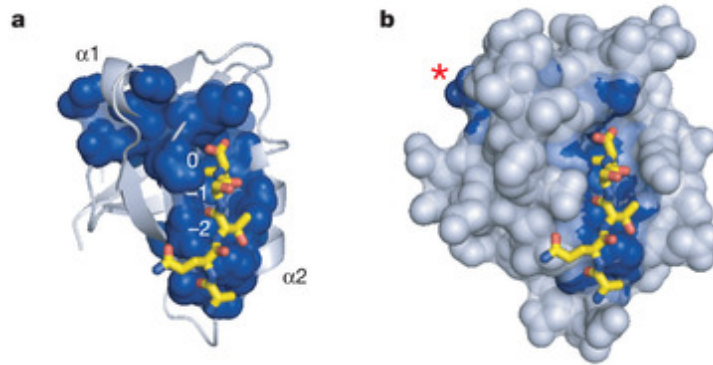
*Figure 1:* **Left:** *We show the PDZ domain (in blue) with the target peptide (yellow). The C-terminal of the target peptide is numbered 0 whilst the preceding amino acids are numbered in descending order.* **Right:** *The position of the PDZ-Domain within the parent protein is shown. The position marked with a little star shows that during the binding process there are other parts of the PDZ domain which are involved and not just the binding pocket*

# 1 PDZ Domains

Let us begin by explaining PDZ Domains, their importance, their structure and how they bind to other macromolecules. PDZ domains are short sections of proteins composed of 80-90 amino acids. PDZ Domains are usually found in signaling proteins where they regulate processes such as the separation of cell membranes. PDZ itself is an acronym for the first three letters of three proteins: Post-synaptic density protein (PSD95), Drosophilia disc large tumor suppressor (Dlg1) and zonula occludens-1 protein (zo-1)

PDZ Domains are usually composed of 80-90 amino acids and many PDZ domains can be present within a single protein. Multiple domains within the same protein can have similar functions or different functions, each of which will bing to a different part of the target protein or a different protein altogether.

PDZ Domains always bind to the C-terminal of certain specific sections of the target proteins. The binding occurs within a binding pocket formed by the domain. The binding pocket are usually formed by the first 20 amino acids. A diagram showing the positions of the binding pocket as well as the target peptide are shown below.

# 2 Explanation of the data of Stifler et al

Now that we have given a small introduction to PDZ domains we shall explain the work done by Stifler and co-authors which we sought to extend and improve upon.

The article that we have considered is titled "PDZ domain binding specificity is optimized across

the mouse proteome". In this article, Michael Stiffler and co-authors sought to characterize the selectivity between 157 PDZ domains and 217 target peptides found in the mouse proteome. To this end, they created a model which could predict whether a given PDZ domain would bind to a target peptide given the sequence of the target peptide.

In their model, the last 5 positions near the C-terminal of the target peptide are considered. The model can be summarized in the following simple equation:

$$\phi_i = \sum_{p,q} A_{p,q} \theta_{i,p,q} > \tau_i \tag{1}$$

where for a given PDZ Domain $i$, we can calculate a binding score $\phi_i$. $A$ is an indicator of the peptide sequence : $A_{p,q} = 1$ if the amino acid at position $p$ of the peptide is $q$ and $A_{p,q} = 0$ otherwise. The numbers $\theta_{i,p,q}$ are the parameters that need to be fit to the experimental data. $\tau_i$ is a scoring threshold which is different for each domain $i$. The values $\theta_{i,p,q}$ are to be interpreted in the following fashion: $\theta_{i,p,q} > 0$ if the PDZ Domain $i$ prefers amino acid $q$ at position $p$ more than the other PDZ domains, negative if it prefers it less, and 0 if it has no bias relative to the other domains. $\tau_i$ is defined as the $m$th percentile of the $\phi_i$'s for all the peptides in the model that bound to the domain $i$. To make predictions using this model, it suffices to know the sequence of the target peptide and compute $\phi_i - \tau_i$ for the domain $i$. If the domain binds with the target protein then this number is positive and negative otherwise.

The data used to fit the model was obtained by protein micro-array assays. Each PDZ Domain-peptide pair under consideration was verified for binding during the experiment. The researchers were able to model 74 domains among the 157 domains experimented upon. The experimental results of these experiments are available to us in the form of an interaction matrix which tells us whether a PDZ domain $i$ binds to a target peptide $j$ or not.

For each of the 74 domains successfully modeled, we also possess the parameters $\theta_{i,p,q}$ allowing us to test the model on newer peptide sequences. We also possess the error rates for the model. These are given in terms of false positive and false negative rates. These error rates interpreted as empirical probabilities were used in one of the models presented later in the report.

# 3   Questions asked and answered

With the biological context established and the various data that we possess explained, let us now briefly the discuss the questions that we sought to answer over the course of the project.

Firstly, we sought to understand the effect of mutations on the specificity of domain-peptide interactions. It is well known that PDZ domains bind to only certain specific target peptides and not to others. What if the target peptide suffered a point mutation? Will the PDZ domain still bind to the peptide?

Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome

A related question concerns the level of constraint imposed on the peptides due to the specificity of their interactions with PDZ domains. If the peptide were to suffer a point mutation and if it continued to bind to the PDZ domain, it then becomes interesting to consider the extent to which the peptide can undergo mutations before it stops binding to the peptide. Inversely, if the peptide doesn't accept many mutations, it is again interesting to analyze whether there are certain positions or amino acids which impose constraints on the sequence.

All such questions were modeled, analyzed and understood in this project.

# Part II

# First Model

## 4   Features of the model

Our first model sought to study the effects of the introduction of mutations in the target peptide sequences conserving all the while the interaction matrix. The mutations considered were point mutations whereby one amino acid at a particular position was muted into another one. To this end, we define a parameter $\alpha$ which corresponds to binding or non-binding for the domain-peptide pair in the interaction matrix, $\alpha = 1$ if PDZ domain $i$ binds to peptide $j$ and $\alpha = -1$ otherwise. Since we want to introduce mutations that conserve the interaction matrix, we begin with the natural sequence of a given peptide and introduce mutations in the natural sequence.

Now, we start from a natural sequence of the peptide $l$. Once a point mutation has been introduced in the sequence, we calculate the binding score $\phi$ for this new sequence using (1). We then define a probability of binding and non-binding in the following manner:

$$p(\alpha_{i,l} = 1) = \frac{1}{1 + e^{-\phi}} \tag{2}$$

$$p(\alpha_{i,l} = -1) = \frac{1}{1 + e^{\phi}} \tag{3}$$

for the domain $i$ and peptide $l$. Here we continue to use the label $l$ for the peptide even though the *new* binding score $\phi$ is calculated for a mutated sequence and not the natural sequence for the peptide $l$

Furthermore, we define an *energy* for each peptide $l$ as:

$$\text{E} = \sum_i \log(1 + e^{-\alpha_{i,l}\phi_{i,l}}) \tag{4}$$

where the sum is over all domains $i$ in our data set. The energy defined can thus be defined for any sequence derived by introducing point mutations into the base sequence of the peptide $l$. In practice, we have a natural sequence, we then introduce a mutation into the sequence, calculate the probabilities $p(\alpha = 1)$ or $p(\alpha = -1)$ for a given domain $i$ and sum the natural logs of these probabilities over all the 74 domains.

Once the mutations have been introduced and an energy defined for a new sequence derived from some natural peptide sequence, we now need a method to determine whether we accept or reject these mutations. To this end, we adopt the Metropolis algorithm, used for performing

Monte Carlo simulations. The choice of performing a Monte Carlo simulation is a natural one since the sequence space consists of $20^5$ sequences and rather than sampling the whole space uniformly, it is more economical to perform a Monte Carlo (hereby MC) simulation.

The general flow of the simulations was as follows:

1. We begin with a single peptide and introduce point mutations starting with the natural sequence of the peptide.

2. We then compare the energies, as defined in (4), between two sequences $E_{old}$ and $E_{new}$.

3. We then use the Boltzmann factor to attribute probabilities to each of the energies $E_{old}$ and $E_{new}$, where the boltzmann factor is

$$p(E) \propto e^{-\beta E} \tag{5}$$

where $\beta$ is called the inverse temperature and the factor of proportionality is the partition function. Since we will take ratios of probabilities, we need not compute the partition function explicitly.

4. We define the acceptance probability as :

$$p_{\text{accept}} = (min(1, \frac{p_{new}}{p_{old}})) \tag{6}$$

where $p_{new}$ and $p_{old}$ are computed using the Boltzmann factor.

5. Given $u \in U[0,1]$ where $U[0,1]$ is the uniform probability distribution, the new sequence is accepted if $u < p_{\text{accept}}$

We chose the value of $\beta = 1$. We performed 10 cycles of Monte Carlo runs, whereby during each run we affected 1000 mutations to each of the 217 peptides in our data set.

# 5   Results

The first quantity that we computed was the initial distribution of energies i.e. energies computed for each of the peptides using only their natural sequences.

We compare this distribution to the distribution of energies of those sequences which were accepted during the Monte Carlo runs

We see a reduction by a factor of 5 in the spread of energies before and after the Monte Carlo runs.

The next quantity that we considered was the probability of having a given amino acid at any position for any peptide. To this end, we computed the frequency matrix which tells us how often a given amino acid was accepted during the Monte Carlo runs.

Specificity and constraints in
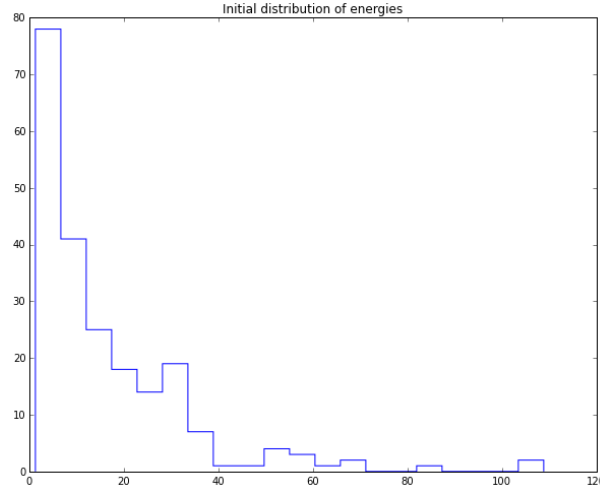Peptide-Protein bindings in the mouse
proteome



*Figure 2: The distribution of initial energies of the peptides. We observe that the energies are well spread out, with the majority in the range from 0-5 and the maximum energy close to 100*
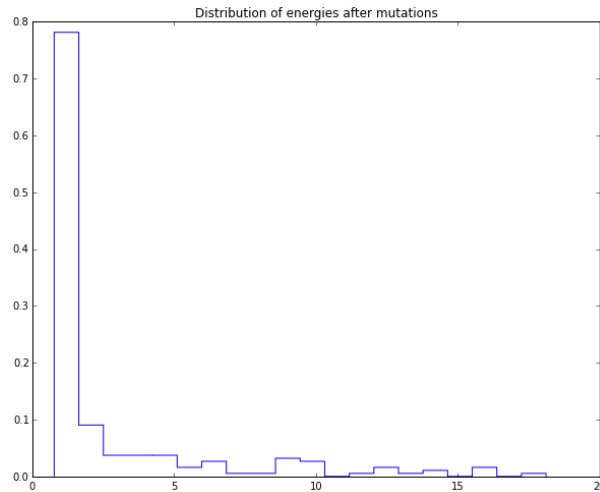


*Figure 3: Here the spread is reduced. The maximum is close to 19.*

Here the peptide considered **SERCA2A** has the sequence **PAILE**. We observe that the most frequent amino acid during the MC runs was Proline(P) and not Alanine (A). Thus we can already see that this peptide is susceptible to undergo mutations. This claim is further supported with the spread over all the 20 amino acids.

The peptide **Cftr** presents a rather particular case for us. Its frequency matrix is presented here. We remark very clearly how, except for the last position, the other positions are not susceptible at all to undergo mutations. Even the mutations that do occur in the last position are not drastic. The natural sequence of **Cftr** is **QETRL**. On the last position, we see that the only mutation accepted is a transformation from Leucine (L) to Isoleucine, which would not be sterically important. Thus we find an interesting case of the existence of strong constraints on the sequence of the peptides in the face of mutations. During the MC runs, **Cftr** oscillates
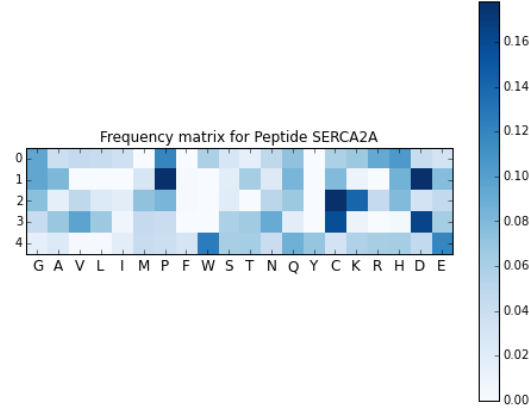
Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome



*Figure 4: The frequency matrix gives a very simple and visual way of deciding whether a given peptide
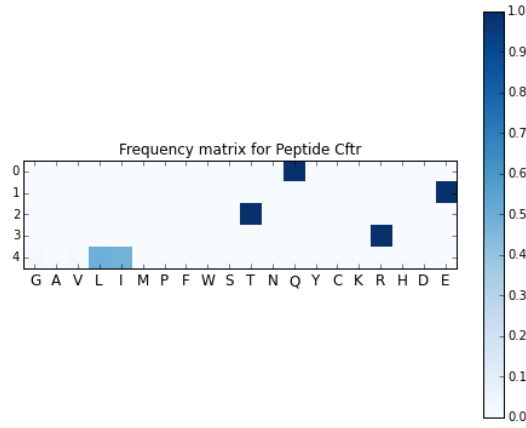is susceptible of undergoing mutations.*



*Figure 5: The sequence for **Cftr** is very strictly constrained accepting mutations only at the last
position.*

between two sequences: natural sequence **QETRL** and a mutated sequence **QETRI**

# 6 Limitations

This model however is not without its limitations. Another particular case that we consider is
the peptide **APC** which has the natural sequence **LVTSV** and has a natural energy of 108.
When we consider the evolution of the energy over the course of any one MC cycle, we observe
that it has a tendency to very quickly reduce its energy as is evident from the plot below

This behavior is related to the particular nature of the peptide **APC** as well as the errors
inherent in the model. We ascribe this behavior to the erroneous predictions of the model
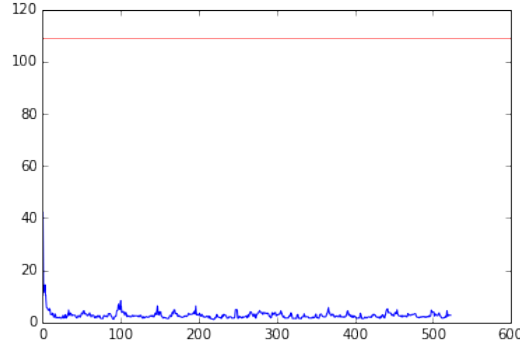which needs to be accounted for in a newer model. This is the subject of the next section.

*Figure 6: The evolution of the energies of the mutated sequences of **APC** during one MC cycle. Out of 1000 mutations possible, it accepts close to 530 mutations, showing a great susceptibility to undergo mutations. Red line shows the natural energy.*

# Part III

# Improvements over first model: Bayesian Modeling

The case of the peptide **APC** tells us that we cannot consider all the predictions of the model to be true and we need a way to integrate the error rates which are present in any predictive model. To this end, we propose a refined model based on conditional probabilities.

## 7    Error rates as probabilities

Any predictive model is seldom 100% perfect and there are always cases where the predictions of the model are in contradiction with the experimental facts. This is the case with the model used by Stifler et al. The information concerning the error rates for their model is known to us and thus we can integrate this extra information to refine our model.

To do this, we need to understand the error rates that we possess. These error rates are reported as true positive rate, false positive rate, true negative and false negative rates. To be more precise, considering that the model predicts binary values 0(False) or 1(True):

1. **True Positive Rate (TP)**: $\frac{\text{Number of events predicted to be True}}{\text{Number of events actually True}}$

2. **True Negative Rate (TN)**: $\frac{\text{Number of events predicted to be False}}{\text{Number of events actually False}}$

3. **False Positive Rate(FP)**: $\frac{\text{Number of events predicted to be True}}{\text{Number of events actually False}}$

4. **False Negative Rate(FN)**: $\frac{\text{Number of events predicted to be False}}{\text{Number of events actually True}}$

Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome

Table 1: Error rates from Stifler et al.

| Error Rate | Value |
|------------|-------|
| TP | 96% |
| FN | 4% |
| FP | 15% |
| TN | 85% |

For the model proposed by Stifler, these values are given as:

The way these error rates are defined reminds us of the definition of conditional probabilities. And thus, we can interpret these error rates as conditional probabilities. If the model makes a prediction on a new piece of data, these rates then tell us with what probability the experimental value was +1 or -1. Thus we have,

$$\text{TP} = \mathbb{P}(\text{Model} = 1|\text{Experiment} = 1) \tag{7}$$

$$\text{FN} = \mathbb{P}(\text{Model} = -1|\text{Experiment} = 1) \tag{8}$$

$$\text{FP} = \mathbb{P}(\text{Model} = 1|\text{Experiment} = -1) \tag{9}$$

$$\text{TN} = \mathbb{P}(\text{Model} = -1|\text{Experiment} = -1) \tag{10}$$

We can then recast Table 1 in the following manner:

Table 2: Error rates as conditional probabilities

| Model | Exp | P(Model|Exp) |
|-------|-----|--------------|
| +1 | +1 | 0.96 |
| -1 | +1 | 0.04 |
| +1 | -1 | 0.15 |
| -1 | -1 | 0.85 |

# 8   Updated Bayesian model

Having cast error rates as probabilities we can now use Bayes rule to invert the Table 2 i.e. calculate the probability $\mathbb{P}(\text{Exp}|\text{Model})$. This can be done using the following equation:

$$\mathbb{P}(\text{Exp}|\text{Model}) = \frac{\mathbb{P}(\text{Model}|\text{Exp})\mathbb{P}(\text{Exp})}{\mathbb{P}(\text{Model})} \tag{11}$$

where $\mathbb{P}(\text{Model})$ is a normalising factor that needs to be calculated from the experimental data and $\mathbb{P}(\text{Exp} = 1)$is for a given peptide, number of domains that it binds to, and similarly for

$\mathbb{P}(\text{Exp} = -1)$. We can thus calculate a the *posterior* probabilities for each peptide: by using the interaction matrix for calculating $\mathbb{P}(\text{Exp})$, with $\mathbb{P}(\text{Model})$ is calculated by normalising the probability $\mathbb{P}(\text{Exp}|\text{Model})$.

We can create a posterior probability matrix for the complete data set, by summing over all the peptides in the data set. We thus have

*Table 3: Posterior Probability Matrix*

| Exp | Model | P(Exp\|Model) |
|-----|-------|---------------|
| +1  | +1    | 0.1818        |
| -1  | +1    | 0.8181        |
| +1  | -1    | 0.0016        |
| -1  | -1    | 0.9983        |

We observe that the value $\mathbb{P}(\text{Exp}=1|\text{Model}=1)$ is very small. This is because for the whole data set $\mathbb{P}(\text{Exp}=1)$ is very small $\sim 0.03$.

For the purposes of introducing mutated sequences into the model, we would now like to compute the probability of binding (non-binding) given a mutated sequence. To this end, we use the chain rule to get the following expression for $\mathbb{P}(\text{Exp}|\text{Sequence})$:

$$\mathbb{P}(\text{Exp}|\text{Sequence}) = \sum_{\text{Model}=\pm 1} \mathbb{P}(\text{Exp}|\text{Model})\mathbb{P}(\text{Model}|\text{Sequence}) \tag{12}$$

where

$$\mathbb{P}(\text{Model}|\text{Sequence}) = \frac{1}{1 + e^{-y_{\text{model}}\phi}} \tag{13}$$

and where $y_{\text{model}} = \pm 1$ are the values predicted by the model (+1 for binding and -1 for non-binding) and $\phi$ is the score calculated from (1). We once again compute an energy for each peptide, this time for the updated probability (12). Using the variable $y$ for the values that Exp and Model can take, we write the energy for the peptide $l$ as :

$$\text{E} = -\sum_i \log(\mathbb{P}(y_{\text{Exp}}^{i,l}|\text{seq})) \tag{14}$$

where the sum runs over all the domains $i$. Using this redefined energy, we perform MC simulations with the same parameters as before.

Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome

# 9   Results

1. We first compare the distribution of energies in the updated model. In the updated model, the spread of energies is limited between 0 and 20. More importantly, the distribution of energies once the mutations have been affected is similar. We thus see already a great improvement in the model whereby we have prevented drastic drops in energy like the one we saw in the case of **APC**
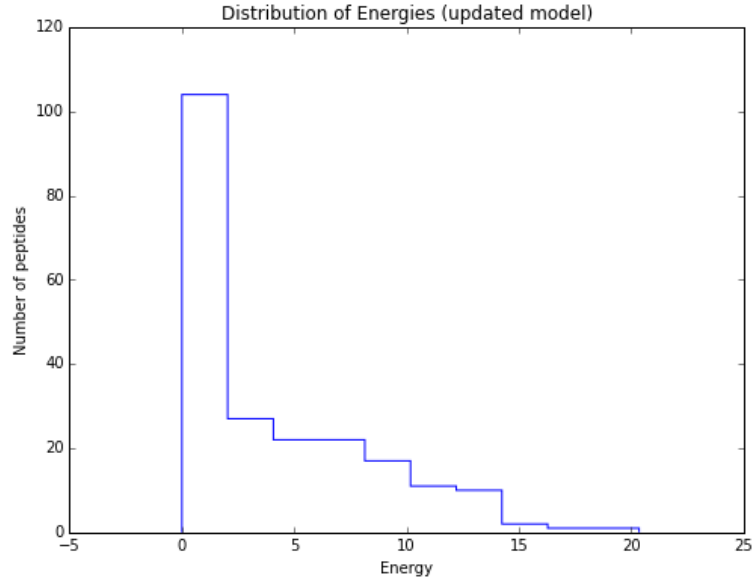


2. This time we had more confidence in the model with respect to the previous one, we computed other relevant quantities that could give us information about the constraints present in the sequences of the peptides. One of the first quantities that we computed was the entropy of the peptide from the frequency matrix. We define the entropy of the peptide $l$ for the position $k$ in its sequence as :

$$S_k^l = - \sum_{a=1}^{a=20} p_{k,a} \log(p_{k,a}) \tag{15}$$

where the sum $a$ runs over all the 20 amino acids and $p_{k,a}$ is the probability of finding amino acid $a$ at position $k$. This probability can readily be computed from the frequency matrix for the peptide $l$. The entropy is a particularly relevant quantity since it provides a direct way of determining the degree of constraint that is present in the position $k$ for any peptide. In particular, a value of $S = 0$ tells us that the peptide doesn't accept any mutations on the position considered and is thus highly constrained.

Furthermore, we can also compute the quantity $S^l = \sum_k S_k^l$ which is the sum of the entropies over all positions for the peptide l. This quantity tells us how much liberty we do have while performing mutations of the natural peptide sequence. To facilitate
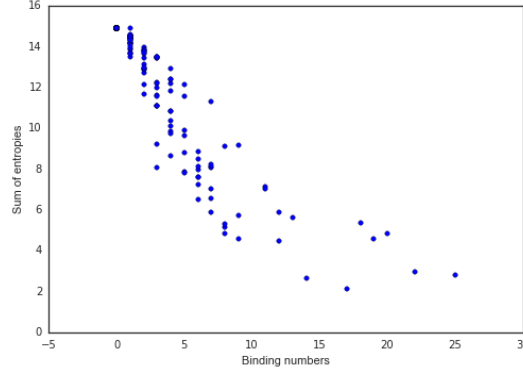
*Figure 7: Dependence between entropy and binding number*

our understanding and provide a simple way of distinguishing between different types of peptides, we divide all the 217 peptides based on a quantity called the binding number bn, which is just the number of domains that a peptide binds to.

As a first indication of the dependence of the behavior of peptides as a function of their binding number, we trace the number $S^l$.

We remark that the entropy varies inversely with an increase in binding number. This shows that higher the binding number, higher are the constraints present over the natural sequence of the peptide. Thus, we are led to a closer analysis of the position wise entropies $S_k^l$ for different values of the binding number.

3. We know that if $S_k^l = 0$ for any position $k$ then it implies that the position $k$ is immune to mutations. After dividing the peptides by their binding numbers, we searched for any peptides which showed such behavior. Remarkably and contrary to what we would have believed initially, such peptides are present over a large range of binding numbers: we would accept that a peptide with bn=25 would have some position constrained owing to its extremely low entropy value. However, we also observed constraints for binding numbers as low as bn=3. The peptide with bn=3 showing such a behavior is **NMDAR2C**

4. After having observed the presence of constraints, we wanted to see whether any non-trivial correlations existed between the two positions. To determine which two positions could be potentially correlated, we computed the mutual information matrix, defined as follows:

$$\text{MI}_{i,j} = \sum_{a=0,b=0}^{a=20,b=20} p_{i=a,j=b} \log\left(\frac{p_{i=a,j=b}}{p_{i,a}p_{j,b}}\right) \tag{16}$$

where $p_{i=a,j=b}$ is the joint probability of finding the amino acids $a$ and $b$ at positions $i$ and $j$ respectively and $p_{i,a}$ is the probability of finding amino acid $a$ at position $i$. The mutual information intuitively tells us how much information is considered in the variable
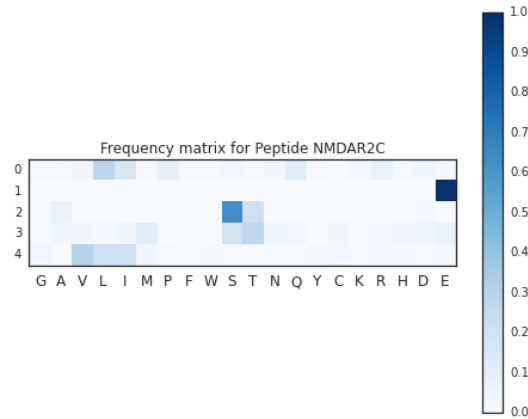
*Figure 8: The 2nd position in the sequence of **NMDAR2C** is clearly constrained. Even for other positions we see that the space of possible mutations isn't spread out over many amino acids.*
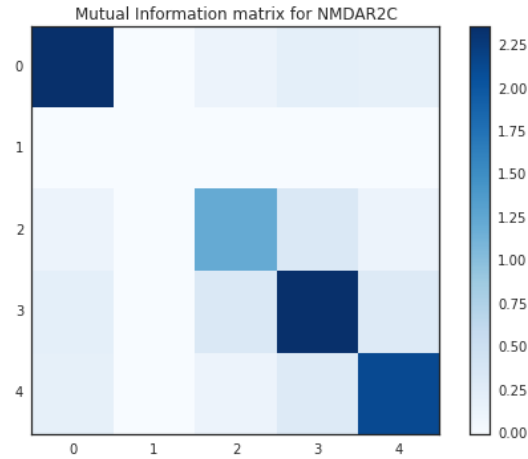


*Figure 9: The mutual information matrix for NMDAR2C. The second position (1 in the figure) being completely constrained is not correlated with any of the other positions.*

X about the variable Y. In our case, by computing the mutual information matrix, we would be able to determine the correlated positions in the peptide sequence.

The constraint on the position numbered 1 in the figure (the numbering used in the figures starts with 0, common in programming languages) is manifest in the mutual information matrix as well. It is completely independent of the other positions, implying that it is completely uncorrelated with any other position as well. This is just another way of saying that the position numbered 1 must conserve the amino acid found in the natural sequence irrespective of the other positions.

From the mutual information matrix, we see that positions numbered 2 and 3 might be correlated. However, the computed correlations are very small, suggesting that the positions are only weakly correlated. This trend was observed for other peptides with different bn's as well.
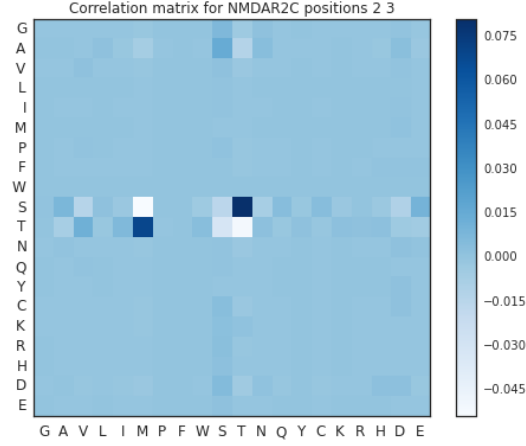
*Figure 10: The normalized correlation coefficient matrix for correlations between the different pairs of amino acids. Barring a few amino acids, most amino acids are completely uncorrelated.*

5. From the frequency matrix we can compute the most probable sequence. This sequence is constructed by computing the amino acid with the highest probability of occurrence at any given position. We present here the most probable sequences for each of the 18 peptides which showed highly constrained behavior during the MC runs.

*Table 4: Peptides with constrained positions*

| Name | MPS | Base Sequence | Hamming Distance | Binding Number |
|------|-----|---------------|------------------|----------------|
| Caspr2 | KRWIC | KEWLI | 3 | 5.0 |
| Cftr | QETRL | QETRL | 0 | 17.0 |
| Cnksr2 | LETHV | IETHV | 1 | 25.0 |
| CRIPT | KQTSV | KQTSV | 0 | 12.0 |
| ctransprtr | NATRL | NATRL | 0 | 8.0 |
| Nav1. | RESNV | RESIV | 1 | 9.0 |
| NMDAR2B | IESDV | IESDV | 0 | 8.0 |
| PKC | LQAAV | LQSAV | 1 | 8.0 |
| PMCA1 | IETSL | LETSL | 1 | 19.0 |
| Sapk3 | LETAL | KETAL | 1 | 20.0 |
| SSTR2 | ILAWV | IIAWV | 1 | 22.0 |
| TAZ | FLTWL | FLTWL | 0 | 18.0 |
| TRPC4 | NHTRL | VTTRL | 2 | 6.0 |
| NMDAR2C | LESTV | LESEV | 1 | 3.0 |
| Kv1.5 | RETDQ | RETDL | 1 | 7.0 |
| Kv1.7 | MVTTV | MVTEV | 1 | 13.0 |
| TRPM6 | NHTRL | DHTRL | 1 | 5.0 |
| TRPV3 | PETSV | PETSV | 0 | 12.0 |

6. An interesting observation that we made during the search for the presence of constraints was that the last position, i.e. C-terminal amino acid was never constrained. This is surprising because PDZ domains bind to the C-terminal of the target peptides. The lowest values for the entropy and thus the most constraints were seen for positions numbered 2 and 3 in the diagrams, which would corresponds to -3 and -2 in the sequence of the peptide. No constraints were found for last position that is -4 either.

7. Finally, we attempted to see whether by the removal of one PDZ domain would change the behavior of the peptides or not. It was found that it did not. However, on average the peptides accepted more mutations that with the presence of the domain. But these mutations were energetically unfavorable.

**Some Caveats**

1. The peptide sequences considered here are made up only 5 amino acids. Thus, our modeling is necessarily local in nature and we are not aware of the relevant steric and chemical effects that might influence the binding between peptides and PDZ-domains. Furthermore, we know that the binding pocket is not the only part of the PDZ domain which is implicated in the binding, other positions far away from the binding pocket are involved too. Thus, by considering only those sections of the target molecules which are present near the binding pocket, our analysis can only be limited.

2. There are many peptides for whom we didn't see any constraints whatsoever. This however should not mean that constraints are completely absent from these peptides. Far from it, this serves as a reminder of the limited scope of our study. With only 74 domains to play with, we cannot generalize the behavior of the peptides that we studied.

# Part IV

# Integrating PDZ Domain sequences

After having studied the peptides present in our data set, we sought to integrate the sequences of the PDZ domains. We wanted to be able to propose a model using the sequences of the PDZ Domains which would be able to recreate the parameters $\theta_{i,p,q}$ of the original model (1). To this end, we decided to use a simple linear model. However, this linear model is necessarily under-constrained. For instance if we consider one domain $i$, and we assume that we will consider only the first 20 amino acids in the sequence, it gives us a possible 400 parameters (20 amino acids $\times$ 20 positions) parameters to fit for 100 unknown variables. (For each domain $i$, we have 100 values for $\theta_{i,p,q}$: 5 positions on the peptide $\times$ 20 amino acids). An ordinary least squares regression would not be very efficient. However, the method **LASSO** or Least Absolute Shrinkage and Selection Operator is best adapted to our needs. This is the method that we decided to finally use.

## 10  The LASSO

The LASSO is a method which forces the sum of the absolute value of the regression coefficients to be less than a fixed value, leading to certain regression coefficients to be set to zero. The condition of forcing the absolute sum of the coefficients to be less than a certain limit is known as **L1 Regularization**.

The Lasso can be written as an optimization problem set in the form:

$$\min_{\beta_0,\beta} \left\{ \frac{1}{N} \sum_{i=1}^{N} (y_i - \beta_0 - X\beta)^2 \right\} \text{ subject to } \sum_{j=1}^{p} \|\beta_j\| \leq t \tag{17}$$

However, we can recast this equation in the Lagrangian form as follows:

$$\min_{\beta \in \mathbb{R}^p} \left\{ \frac{1}{N} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1 \right\} \tag{18}$$

.

Here, $y$ is the output of the model and $X$ is the data. In our case, the output variables are the $\theta_{i,p,q}$ for each of the domains $i$ and the $X$ correspond to a matrix of dimensions $n \times 20$ where $n$ is the length of the PDZ domain sequence that we consider. The matrix $X$ is defined as follows: $X_{n,a} = 1$ if the amino acid $a$ is at position $n$ and $X_{n,a} = 0$ otherwise.

For a practical implementation, we considered 70 domains for whom we could find the complete sequence. For each of the domains the $\theta_{i,p,q}$ were collapsed into a vector of dimensions $100 \times 1$,

and the $X^i$ were collapsed into a vector of size $(n \times 20) \times 1$. Thus, the output variable $y$ had dimensions $70 \times 100$ while $X$ had dimensions $70 \times (n \times 20)$. The lasso was finally implemented with the open source library scikit-learn in Python.

# 11    Results

Starting with the sequences, we first performed an alignment since the domain sequences were collected from different sources. Once aligned we were left with two configurations, one with gaps and one without gaps. We decided to remove those positions from the sequences for which there were gaps in more than 50% of the positions in the multiple sequence alignment. Once these positions were removed, we could begin testing our model using the lasso method.

1. We first tested the validity of the model and its predictive power. A 80-20 split was performed: 80% of the data set was used for training the model and 20% was used for testing it. For the lasso method, we have one free parameter $\lambda$ in (18). To decide the optimal value for $\lambda$ we tracked the training and testing error for different values of $\lambda$. The training error and test error were simply defined as the L$^2$ norms of the difference between the model prediction and the experimental values.

   $\lambda$ is not the only parameter that is at our disposition. A critical choice needs to be made about the length of the sequence $n$ of the domain that we want to model. Thus we tracked the evolution of the training and testing error for different values of the sequence length.
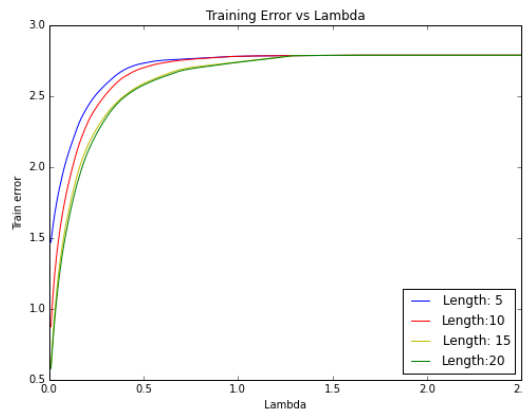


*Figure 11: The evolution of training error for various values of $\lambda$*

   We remark that the training error for different values of $n$ has the same behavior. The training error increases as the value of $\lambda$ increases for all values of $n$. This is to be expected: for small values of $\lambda$ there is essentially a situation of overfit whereby the model is completely fit to the data that we have provided but will not generalize to the new test data. For higher values of $\lambda$, the regularization factor in (18) makes a majority of terms zero and thus the model is very poorly fit. Thus, we expect to find an optimal value of $\lambda$ which will serve the need for a predictive model: fit existing data well and

Specificity and constraints in
Peptide-Protein bindings in the mouse
proteome

generalize well to new data. To this end, we need to track the evolution of test error, that is the error committed by the model when it is used to predict values that were not used to fit the parameters of the model. This situation is best illustrated in the following diagram
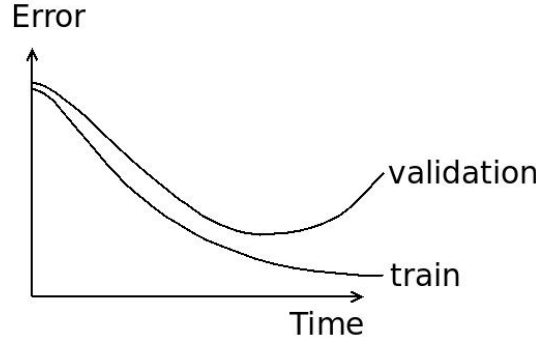


*Figure 12: The evolution of training and test error for a predictive model.*

We see that if the validation or test error increases while the training error decreases then a case of overfitting might have occurred. However, the best predictive and fitted model would be where the validation error has its global minimum.

Below we have plotted the variation of the test error for various values of $\lambda$ for different values of $n$. We observe that for small values of $n$ like 5 or 10, there is a global minimum for test error indicating us that our model can be used to predict on new values. However, for higher values of $n$, the test error shows monotonic behavior implying that there is no global minimum for the test error. Thus, we fear that for higher values of $n$, our model might not be used to predict on unknown test data.
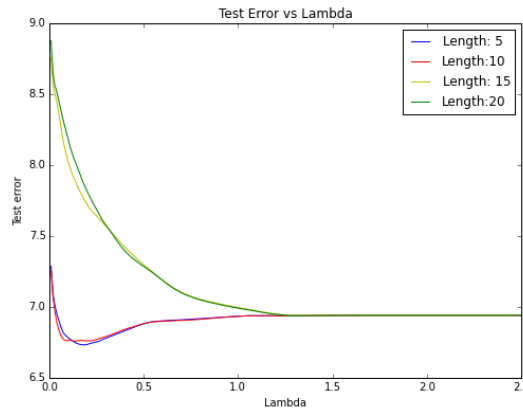


*Figure 13: The evolution of test error for various values of $\lambda$*

2. To understand the behavior of intermediate values of $n$ as we varie $\lambda$, we tracked the variation of train and test error as a function of $\lambda$. In the plot below, we observe that the lowest values of test error are achieved for sequence lengths $n = 6$ and $n = 7$, whilst for

other values of $n$, we see a minimum but the overall test error is higher. The optimum is achieved for $\lambda \sim 0.2$. We have thus concluded that till $n = 10$ we do have a minimum and thus our model has predictive ability.
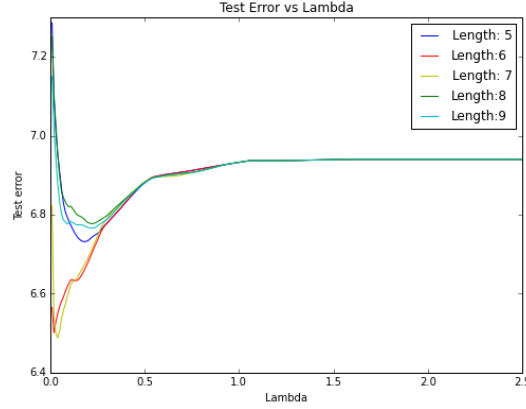


*Figure 14: The evolution of test error for various values of $\lambda$ for $n = 5 \rightarrow n = 9$*

In the next figure we plot the test error for values of $n$ from 10 to 14. We observe that for $n = 10, 11, 12$, a minimum is achieved for $\lambda \sim 0.2$. However, for $n = 13$ there is no minimum and the difference between the curves for $n = 13$ and other curves is pretty striking. We thus conclude that the maximum sequence length that we can hope to model faithfully is restricted till $n = 12$ and that for higher values, our model may not be predictive.
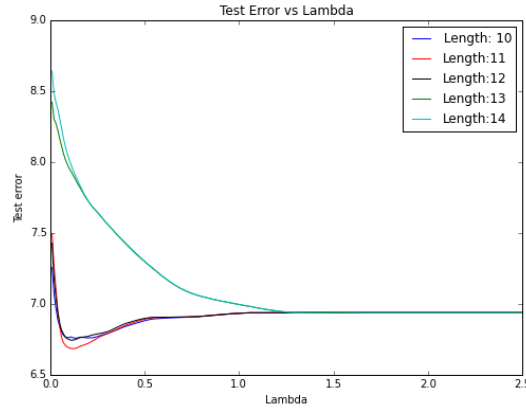


*Figure 15: The evolution of test error for various values of $\lambda$ for $n = 10 \rightarrow n = 14$*

3. Finally we analyse the lasso parameters that we retrieved from fitting the model to the data. For a sequence of length 10, the matrix of parameters fit to the data is a matrix of dimensions $100 \times 210$ which is a relatively sparse matrix. Furthermore, the interpretation of each of the parameters is difficult, especially since we are interested in the correlations between positions in the peptide sequence and positions in the PDZ sequence. Thus to facilitate our understanding, we undertake a pooling operation: we sum the absolute
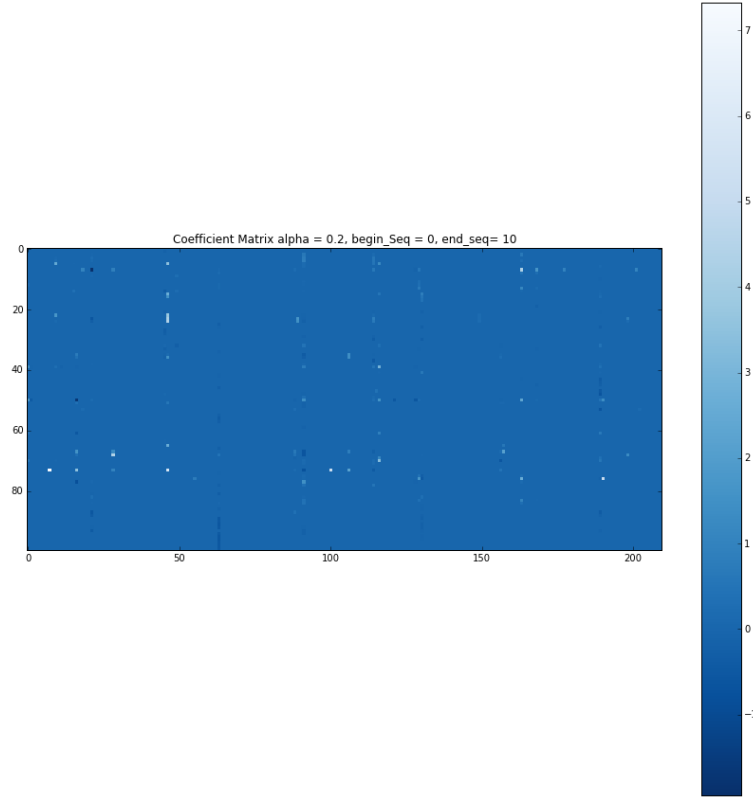
*Figure 16: The matrix of coefficients fit to the data by the lasso method. Here the size of the sequence taken is 10. The interpretation of different parameters isnt very clear, especially since the matrix is sparse*

values for each $20 \times 21$ block in the coefficient matrix. This pooling operation provides a nice, visual way of representing the details of the coefficient matrix. If some of the entries in the pooled matrix are zero, then we can be sure that the corresponding submatrice in the coefficient matrix were zero, and thus are unimportant for predicting binding affinities between a peptide and a PDZ domain.

4. Observing the matrix below, we remark that the maximum values in the pooled matrix correspond to the positions (1,2) and (3,0). Let us have a closer look at the submatrices that contribute to these entries.

The first submatrix that we analyse is the matrix corresponding to position 1 in the peptide (on the actual sequence, we need to add +1 since the numbering here follows the notation adopted in programming languages) and 2 in the PDZ sequence. We observe that the important coefficients correspond to the amino acid *Isoleucine* on the domain at position 2 while the most important amino acids on the peptide are similar amino acids like *Alanine, Leucine and Isoleucine*. It is quite remarkable that even though it is *Leucine* which is most common at position 2 in the multiple sequence alignment of the domain sequences, it is *Isoleucine* which has non-zero coefficients in our model.
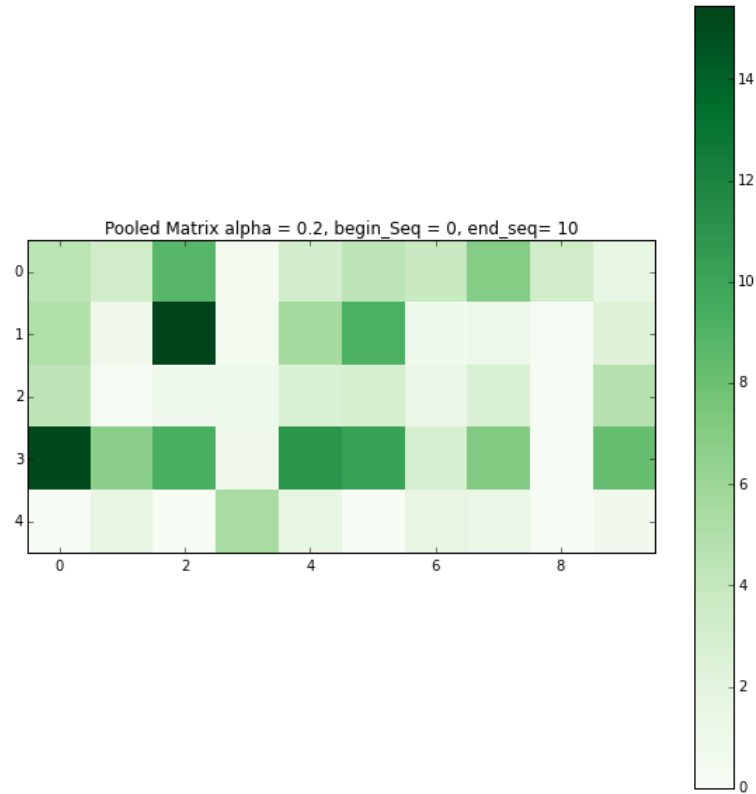
*Figure 17: The matrix received after pooling the entries in the complete coefficient matrix. This provides a better overview of the structure of the coefficient matrix and also provides us information on the relevant terms in the coefficient matrix*
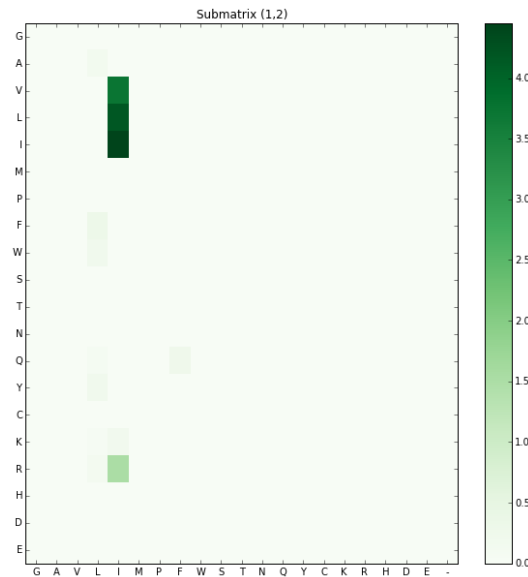


*Figure 18: The terms making up the submatrix (1,2) in the complete coefficient matrix*

Let us now consider the case of the submatrix (3,0). Here the amino acid *Phenylalanine* (F) on the PDZ domain sequence is correlated with *Tyrosine* (Y) on the peptide sequence. Again it is interesting to see that *Phenylalanine* is not the most common amino acid at position 0 in the multiple sequence alignment. Let us now turn our attention to the most important submatrix for our purposes
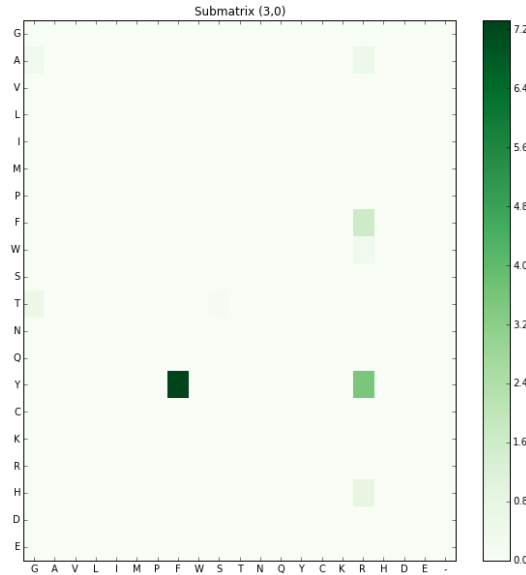


*Figure 19: The terms making up the submatrix (3,0) in the complete coefficient matrix*

In the sub-matrix (4,3) we see a remarkable thing happening: the only amino acid which counts on the PDZ domain sequence is *Glycine*. If we look at the multiple sequence alignment, we see that *Glycine* is found at the position number 3 in all the PDZ domains. Thus it isnt a surprise that non-negligeable terms in the coefficient matrix exist for *Glycine*. What is perhaps more interesting is that apart from the last position on the peptide sequence, none of the other positions have comparable coefficients in the pooled matrix. If we remember from our analyse on peptide sequences, we had observed that there were no constraints on the last position of the peptide sequence. We can now understand this phenomena by the fact that for peptide-domain binding it is the position 3 (or 4 on the actual sequence) on the PDZ domain sequence which is important. Thus, the constraint does exist, but not on the peptide but on the PDZ domain sequence since it is conserved through all the domains. What is also perhaps interesting to observe is the quasi-uniformity of the coefficients for most of the amino acids on the peptide sequence.

Here we have seen a remarkable case of the existence of constraints not on the peptide sequence but on the domain sequence. We thus conclude that it is not just the peptide sequence that is important in determining PDZ Domain-Peptide binding specificity but also the sequence of the PDZ domain itself.

**Some Caveats**

- We have only considered short sections of the PDZ Domain sequence. Thus our model

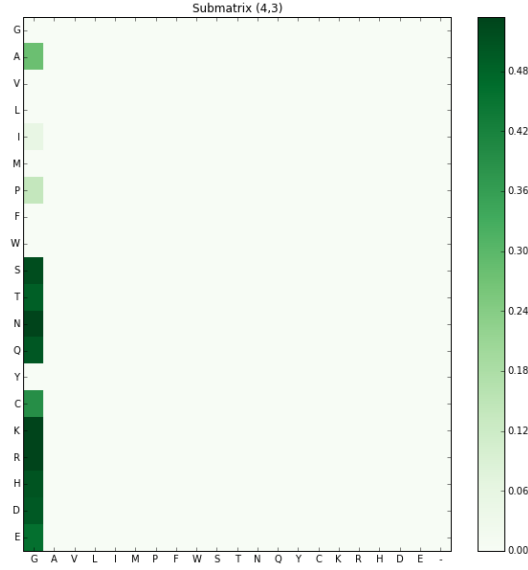*Figure 20: The terms making up the submatrix (4,3) in the complete coefficient matrix*

has still remained relatively local in its scope. By restricing ourselves to about the first 10 amino acids in the PDZ Domain sequence, we have actually not ventured out of the binding pocket. Thus the model is limited because it does not take into account the influence of the amino acids further in the PDZ domain sequence.

- The model proposed here is a linear model and thus will not be able to capture the complete richness of the interactions between a target peptide and a particular PDZ domain. Furthermore, we have seen the applicability of the model depends closely on the length of the sequence concerned. Thus if are to complete integrate the complete sequence of the domain, then it would be a difficult task.

- For lack of time, an analysis of the effects of mutations on the PDZ domain sequences has not been conducted.