Abstract

Background: In the last decade studies have reported an increasing number of proteins that lack a tertiary structure. These proteins have been defined as intrinsically disordered proteins (IDPs) which maintain important functions like signalling and regulation, which are complementary to the functions of proteins of known structure. The current challenges that the bioinformatics field faces revolve around the understanding of the fundamental principles that govern IDPs, how they contribute to the functioning of living organisms and how their physic-chemical principles can be utilized in adjacent fields like drug discovery or biotechnology. The topic of intrinsic disorder is particularly interesting since numerous human diseases have been linked to IDPs. In this thesis we focus on the disease-disorder relationship and assume that mutations within intrinsic disorder can cause disease. Distinguishing between pathogenic and functionally neutral point mutations has been in the limelight of bioinformatics and it is increasingly being exploited. The originality in this thesis is that we are focusing on the mutations within disordered regions in a systematic way that is easy to follow for researchers and paves the way for further research into this area (i.e. incorporation within a larger mutation predictor for structured proteins).

Results: We use the nonsynonymous single nucleotide polymorphisms (nsSNP) found in the SwissVar portal and correct the dataset to only include human diseases. We first identify which mutations fall within intrinsic disorder via a well-known disorder predictor IUPred. Once the data is pre-processed we design two classifiers: one based on the difference of empirical probability density functions; and the second one based on the average of the class labels (in our case the label identifies if the mutation is disease/non-disease causing). In order to better classify new data, the scoring functions were estimated in three ways, first by linear interpolation, second by fitting splines and third by fitting weighted splines. The results were in line with our hypothesis (some mutations within disorder are deleterious) and showed that based solely on disorder scores, our classifiers can roughly fit the data (area under the ROC curve equal to 65%).

Conclusions: We conclude that there is a significant link between disease and mutations within intrinsic disorder. The classical scoring functions with few degrees of freedom were not enough to capture the complex interactions between the variables so we successfully fitted spline surfaces which were more flexible. From a biological point of view this finding leads to further research questions that might help shed light on the current human diseases without efficient treatment.

Acknowledgments

I would like to thank my supervisor Dr. Julian Gough for all his help and support throughout this research task. His feedback has been of tremendous help and kept me in line with the objectives of the work. I would also wish to thank Matt Oates and Hashem Shihab for their input and original ideas into answering the research question.

I also wish to thank my Course Supervisor, Dr. Tim Kovacs who had faith in me and allowed me to follow this Advaced Computing Master.

Lastly I would like to thank my boyfriend for supporting me throughout this project, encouraging me to trust my findings.

Table of Contents

1 Introduction

1.1 Motivation

Proteins are very important molecules that perform vital functions within cells. Up until around a decade ago it was believed that protein sequence follows structure which follows function. Nevertheless, scientists found a large number of proteins that lacked a stable structure and still maintained vital functions within the cell. With time, the structure-function paradigm has been shaken up and academia now accepts the existence of intrinsically disordered proteins that have well defined functions which are complementary to those of structured proteins, yet are devoid of a stable tertiary structure. What makes these disordered proteins so special is the fact that they have been linked to a number of serious human diseases. A growing body of researchers have dedicated time in trying to tackle the challenges intrinsic disorder brings to the table. In this thesis we wish to improve the knowledge on missense mutations, but this time in disordered proteins. These sorts of point mutations appear to be more prevalent than first expected, thus understanding their effects and functional impact on IDPs is a new question that has yet to be broached.

1.2 Research questions and hypothesis

This thesis has a strong research orientation and endeavours to answer a series of questions concerning our topic. The main questions that we are addressing are:

- How common are point mutations within IDPs and what functional impact do they have?
- Are disease-causing mutations more prone to fall within ordered or disordered protein regions?
- Is deterioration of an IDP's function, to the point of becoming disease prone because of missense mutations or due to other processes like posttranslational modifications (PTM)?

The hypothesis that we base our research questions on is that point mutations found within intrinsically disordered proteins/regions have a higher propensity of being disease causing. With this in mind we envisage a mutation classifier that will determine whether there is a link between mutations within disorder and disease. It may not be the best performing classifier, but as long as it proves the link then we are satisfied as, in essence, that is what we are researching.

In order to answer these research questions we are going to first determine the correct data needed, adapt it for our research project and then conduct an analysis that will shed light on the aforementioned questions.

1.3 Aims and objectives

The link between disease and mutations within disordered regions has yet to be (dis)proven via an extensive analysis. The only articles that have inquisited into the matter are (Vacic & Iakoucheva 2012) and (Hu et al. 2011). The first authors acknowledged the idea that disease mutations within IDPs may have a higher impact than first envisaged. They base their arguments on observations extracted from simple statistics extracted from the analyzed dataset (from the UniProtKB/SwissProt database) that highlight the level of disorder within the data. The authors observe that there are considerable more cases of disorder to order transitions within the mutated protein sequences caused by missense point mutations than the other way around. This led to the formalization of a ranking of the amino acid substitutions that are involved in these sorts of transitions. Their analysis stops there and they advocate for further, more thorough research to be done, suggesting to also look at other type of mutations in conjunction with missense mutations.

The latter authors have a different approach and start their hypothesis that the single nucleotide variants (SNVs) that cause disorder to order transitions are deleterious. Compared to the previous study, these authors focus on single nucleotide variants rather than single nucleotide polymorphisms. The first types of mutations are aberrations that occur only in one individual in a given population while the second type is a mutation that occurs in the same locus for all members of a given species. Thus, from the beginning we are dealing with a different set of data. Moreover, the authors use simulated data from the 1000 Genome Project. The authors work with the data at a gene level and construct a function that will measure the change in disorder prediction caused by the mutation. This research also proved to be unable to answer a generalized question concerning disease causing mutations within intrinsic disorder.

Lastly, the extensive research that has been done into mapping SNPs to a particular disorder based on complex classifiers cannot be utilized on intrinsic disorder since the fundamental feature of protein structure cannot be utilized with intrinsic disorder. Moreover, the amino acid charges and chemical attributes of intrinsically disordered proteins are vastly different from those of structured proteins, making it impossible for researchers to correctly assess the impact mutations that fall in these regions can have.

Based on these short falls we have investigated a more generalized approach to disease and mutations within disorder. The aims of the work were therefore to:

- Explore the link between disease and mutations within intrinsic disorder based on an in-depth statistical analysis.
- Present the results in a classifier form in order to dismiss or accept the working hypothesis.

The above mentioned aims follow the next set of objectives:

1. *Literature review*: Further study the advancements made within the bioinformatics field on intrinsically disordered proteins. Assess the best performing disorder predictors based on the task at hand and highlight the differences between the existing mutation predictors and what a mutation within disorder must incorporate.
2. *Statistical analysis*: Analyze the best statistical approach of identifying relationships between variables given the constraints of our dataset.
3. *Evaluate the one-feature classifier:* evaluate the classifier based on the novel information it puts forward. Concentrating on accuracy and efficiency levels of the classifier are not the main targets of this research since the data information used is scarce and the one feature analysis is not strong enough to build a standalone

classifier. The key goal here is to establish the strength of the link between disease and mutations within intrinsic disorder at a generalized level.

4. *Discussion for further research*: Determine further theoretical and implementation avenues that could strengthen (or dismiss) the hypothesis based on incorporating more biological related information.

1.4 Organization

As mentioned earlier, this thesis is of investigative type. In general terms, the paper has three main areas: the first is an extensive background comprising the first 6 chapters on intrinsically disordered proteins, point mutations, disorder and SNP predictors as well as a short overview of the limited research done in our area of interest. The second part of the thesis comprises the 7th and 8th chapter where the data pre-processing techniques are presented and a comprehensive statistical analysis of the data is explained. The last main part of the thesis comprises chapters 9 and 10 which present and discuss the results while the latter chapter gives a critical evaluation of the work performed and the achievements of this thesis. Lastly, an insight into the further research that can be done given a larger time frame will be discussed. The thesis will conclude with a summary of what we have achieved.

2 Intro to Intrinsically Disordered Proteins (IDPs)

This background chapter will provide the necessary information concerning structured proteins versus intrinsically disordered proteins that is needed for understanding the work and scope of this thesis. The following section 2.1 will describe what structured proteins are, whilst 2.2 will focus on disordered proteins in terms of the definition, classification, function and occurrence. Section 2.3 will highlight the link academia has associated between disordered proteins and human disease.

2.1 Proteins and protein structures

Proteins are macromolecules that have a pivotal role in the structure and functioning of living cells; they represent the basis of life. Humans have identified their importance from the beginning of modern civilization, thus the etymology of the word protein from the Greek "protos" which means first. Proteins are involved in a variety of activities throughout the organism; here we mention transport of oxygen, regulation of cellular functioning, but also gene replication.

Depending on their molecular structure, proteins can be categorized into fibruous, globular and conjugated proteins. However, according to their function, proteins can be categorized in antibodies, enzymes, messengers or proteins that act as structural components or transporters.

A protein's structure and function are heavily correlated as function is derived from the molecular structure. The structural organization of a protein spans over 4 levels. The primary structure is the linear sequence of amino acids (or residues) that constitute the protein. The secondary structure is defined by the hydrogen bonds within the peptide backbone which can form helixes or sheets. The tertiary structure represents the more complex level of folding observed within the secondary structure. Lastly, the quaternary structure is a combination of two or more amino acid chains that will form a full unit. Figure 2.1 offers a graphical representation of all four levels of protein structure.

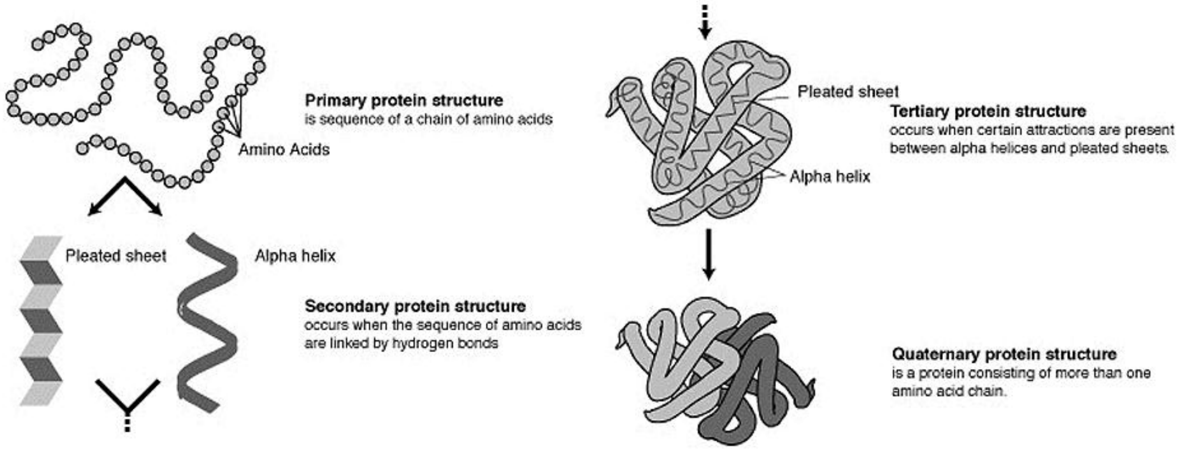

Primary protein structure
is sequence of a chain of amino acids
Amino Acids
Pleated sheet
Alpha helix
Secondary protein structure
occurs when the sequence of amino acids are linked by hydrogen bonds
Pleated sheet
Tertiary protein structure
occurs when certain attractions are present between alpha helices and pleated sheets.
Alpha helix
Quaternary protein structure
is a protein consisting of more than one amino acid chain.

Figure 2.1 The four levels of structure for ordered proteins. (source: http://www.umass.edu/molvis/workshop/prot1234.htm)

As mentioned earlier, the function of a protein is discovered based on its 3D structure. Nevertheless, this paradigm has proven not to stand in the case of intrinsically disordered proteins which lack a tertiary structure, and yet perform vital functions within the cellular machinery. The following section will further develop this paradox.

2.2 Intrinsically disordered proteins

The past decade has led to the discovery that proteins in cells do not always display a unique tertiary structure when isolated, yet still maintain a specific function under physiological conditions. This discovery was only possible because of advancements in X-ray crystallograpy, NMR spectroscopy and hydrodinamics (Serdyuk 2007). This discovery raised a series of questions, among which *why does a cell need intrinsically disordered proteins; what proportion of such proteins is out there or what determines the intrinsic disorder?* The aim of this section is to provide a general definition and classification of these sort of proteins, describe the functions these proteins possess and determine thir occurence in nature.

2.2.1 Definition and classification

After the decoding of the human genome, it was discovered that intrinsically disordered proteins (IDP) are actually highly abundant in nature, with a higher rate in eukaryotes than prokaryotes. This sort of proteins lack a stable tertiary structure yet still maintain a proper function, jeopardizing the paradigm that structure follows function. The idea that a unique tertiary structure is needed for function dates back from over 100 years ago when Emil Fischer discovered that molecules follow a key and lock pattern while conducting experiments on brewer’s yeast extract (*Saccharomyces cerevisiae*). 40 years later, Mirsky and Pauling advanced the idea that the function of an enzyme is directly related to its unique configuration. After the development of macromolecular crystallograpy in the late 1950s it has become apparent that not all proteins fold to a stable tertiary structure making it evident that disorder is a recurring theme within a protein’s structure (Serdyuk 2007). Currently, the scientific field is concentrating on correctly identifying these macromolecules; a commonly accepted definition of intrinsically disordered proteins is that they are proteins that lack both a hydrophobic core and a secondary and/or tertiary structure making them highly flexible and soluble (Orosz & Ovádi 2011). Moreover, in terms of amino acid sequences, IDPs differ immensely from folded or structured proteins. For instance, IDPs mostly lack bulky hydrophobic and aromatic residues responsible for stabilizing the hydrophobic core of a structured protein; but, on the other hand, are abundant in polar and structure breaking residues (Uversky et al. 2008). These are the main attributes that existing classifiers use in order to determine whether a protein is disordered or not. As for classification, intrinsic disorder can be categorized either as collapsed (in the form of molten globules) or as extended (in the form of coils or pre-molten globules) (Uversky et al. 2008).

2.2.2 Function

Collectively, the scientific field accepts that IDPs have a complementary role to that of structured proteins. Among the most important functions we mention regulatory functions related to molecular recognitions and signal transduction (Uversky et al. 2009). Previous research (Biology & Uversky 2011) has shown that IDPs are functionally extremely different from structured proteins as the sample the researchers used almost did not contain enzymes and yet performed vital cellular roles. More precisely, the specific functions that these proteins perform can be classified into four groups: *(a)* molecular recognition, *(b)* molecular assembly, *(c)* protein modification and *(d)* entropic chain activities (Uversky et al. 2008). Molecular recognition involves regulatory and signalling interactions which require a high specificity and low affinity interactions with a multitude of partners. In this sense, IDPs are much more promiscuous than structured proteins as they serve as nodes or hubs in protein interaction networks. Also, compared to their structured counterparts, IDPs can fold in order to perform a certain function by interacting with various partners (proteins, nucleic acids, membranes). This ability to fold makes IDPs highly vulnerable to dysfunction which scientists have identified as being the cause for a number of pathological conditions like cancer, neurodegenerative or cardiovascular diseases.

2.2.3 Occurrence

As mentioned before, intrinsic disorder is widespread in all the three kingdoms but with a larger prevalence in eukaryotes with precautious estimates of 5 to 15% of the proteins being fully intrinsically disordered and up to 35%-50% of the proteins having at least one intrinsically disordered region (IDR) (Orosz & Ovádi 2011). Larger IDRs and a higher number of IDPs can be found in eukaryotes generally because of their complexity and of the higher need for regulation and signalling, which are top functions of intrinsic disorder.

2.3 IDPs and their link to disease

Proteins are the building blocks of life; thus, protein dysfunction can lead to the development of a series of pathological conditions. The causes for this pathogenesis can be protein misfolding, loss of regular function or acquiring of toxic function or protein aggregation. The misfolding and malfunction within these proteins can be caused by either point mutations or because of exposure to internal/external toxins or posttranslational modifications (set of chemical modification to the polypeptide chain after biosynthesis that extends the range of functions of the protein), degradation, lost binding partners or blocked trafficking (Uversky et al. 2009).

Uversky *et. al* (2009) introduced the disorder in disorders or D^2 concept which claims that there is an abundance of intrinsic disorder in proteins associated with various diseases. The authors presented an extensive review of the literature that linked IDPs to diseases varying from cancer and AIDS to cardiovascular and neurodegenerative diseases (Parkinson’s disease, Huntington’s disease) via specific case studies. We synthesise a few of the known disordered proteins that have been acknowledged to be involved in a series of human diseases.

Cancer

α-Fetoprotein (AFP) is an oncofoetal glycoprotein that has been identified as marker for the development of cancer or foetal abnormalities. P-53 is another protein extensively studied by researchers thanks to its capabilities of suppressing cancerous tumours. Lastly, the infamous BRCA-1 has been linked to breast cancer, as well as other types of cancer due to its involvement in many biological processes as signalling, transcription or oncogenesis (Uversky et al. 2008).

Neurodegenerative diseases

A-Synuclein and Huntigtin are two disordered proteins linked to Parkinson's Disease and Huntington Disease. These proteins are present in neurons and have a high propensity to aggregate with other partners; which means that chances of it folding into a stable structure are increased, which later can lead to malfunction (Uversky et al. 2008).

Cardiovascular diseases

Hirudin is an inhibitory protein that acts as a blood anticoagulant. Its C-terminal is essential for thrombin binding. Research has shown that this protein, when bound to thrombin, folds into a helix (an ordered structure) indicating the importance of understanding intrinsic disorder for protein-protein interactions (Uversky et al. 2008).

This small overview of some of the major disease causing proteins identified to being intrinsically disordered highlights the connection already established by scientists between disease and intrinsically disordered proteins. The authors continued their analysis and plotted the percentage of disorder within known disease causing proteins. The outcome can be observed below:

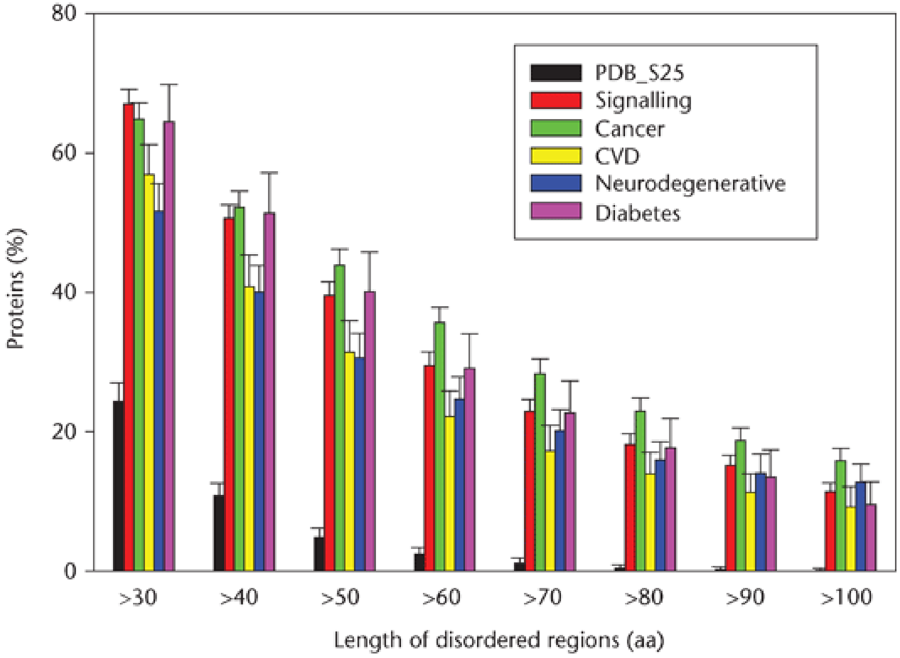


Figure 2.3 IDPs linked to disease (source: (Uversky et al. 2008)*)*

It can be observed that 60% of the sampled proteins have at least 30 consecutive amino acids that fall within a disorder region. Overall, intrinsic disorder is widespread across cancer, diabetes, neurodegenerative diseases when compared to signalling proteins and structured proteins from PDB.

Nevertheless, what are the reasons for this high correlation between intrinsic disorder and disease? Basically, IDPs possess specific features that make them susceptible to pathogenicity like induced folding (protein – protein interaction where binding is coupled with at least one of the partners folding), alternative splicing (process that leads to the creation of multiple mRNAs from a single precursor RNA strand through the inclusion or omission of certain segments), functional regulation via PTMs, binding promiscuity (many hub proteins – those that interact with many partners; are actually intrinsically disordered) or binding plasticity (IDPs can be found in protein-protein interaction in two different modes: one-to-many signalling where one IDP can bind to many partners or many-to-one signalling where multiple IDPs can bind to one site on a partner).

The main reason why the link between intrinsic disorder and human disease is explored is for the creation of new drugs that will inhibit the action/reaction of such a disease causing intrinsically disordered protein or it will limit the protein-protein interactions that could cause a certain disease to remain dormant.

2.4 Conclusion

Proteins are the basic components for maintaining life; they are molecule composed of polymers of amino acids, which determines the structure and function of the protein. However, this paradigm has proven not to hold when speaking of intrinsically disordered proteins (which lack a tertiary structure, making them unable to fold into something known, yet preserving vital functions). These sorts of proteins are highly soluble, capable of interacting with more than one partner, as well as folding when bond with an ordered protein. In the past decade, academia has accepted the existence of these proteins and recently has linked them to a series of grave human diseases. A plausible cause for the IDPs pathogenicity is protein misfolding, which can be caused, among other things, by single point mutations. The following chapter will go further in-depth into these types of mutations.

3 Single nucleotide polymorphisms

Non synonymous single nucleotide polymorphisms (nsSNPs) or single amino acid polymorphisms (SAAPs) are point mutations which are deleterious to gene function and often are labelled as disease causing. In this chapter we will provide a definition of SNPs, their link to disease and we will present the existing SNP databases.

3.1 Definition

In general, a single nucleotide polymorphism represents a variation in the DNA sequence when a single nucleotide (A, T, C or G) in the sequence is altered. These mutations are the most frequent type of variation found in nature and offer a plethora of information on genetic determinants of complex diseases. In humans, SNPs account for more than 90% of sequence difference with an average frequency of about one per 1000 bases (Wang & Moult 2001). SNPs can fall within a coding or non-coding region. The phenotype is affected only when the SNP falls within a coding region and the SNP is either a missense or nonsense mutation.

When talking about missense and nonsense mutations though, we talk about single amino acid polymorphisms (SAAP) which represent point mutations at the amino acid level within a protein sequence; they are also called non-synonymous single nucleotide polymorphisms (nsSNP). Compared to nucleotides, there are 20 amino acids with various physicochemical properties; a single amino acid mutation can lead to a change in function at the protein level which can prove to be deleterious to the organism. Non-synonymous SNPs can be regarded as genetic markers of phenotypic traits like diseases; they may also be subject to natural selection.

A general definition of non-synonymous SNPs is of a point mutation where a single nucleotide is changed which then results in a codon that will code for a different amino acid (a codon is a three nucleotide combination that encodes the information for one of the 20 amino acids existing in nature). nsSNPs are categorised as disease causing or neutral. More specifically, non synonymous SNPs are usually disease causing whilst synonymous ones do not change the structure/function of the protein, i.e. the amino acid substitution is neutral in that the substituted residue has the same characteristics as the original. General statistics find that 50% of mutations fall within non-coding regions, 25% lead to an amino acid substitution whilst the remaining 25% are silent (Shastry 2003). In the end, SNPs capture the diversity within a population and the individual’s responsiveness or susceptibility to disease.

3.2 Disease causing mutations

It is imperative to mention that SNPs do not cause disease directly yet they affect disease development. Moreover, SNPs are an extraordinary source of information concerning the evolutionary history of the human population, its development and its diseases. Alleles represent the forms a gene or genetic locus can take. Genetic disorder occurs when an individual acquires two recessive alleles for a genetic trait (single point mutations occur in these alleles).

Since the start of the 1000 Genome Project, scientists have struggled to develop SNP maps for purposes as evolutionary biology studies, gene discovery, diagnostic testing, prediction and responsiveness of various drugs. SNPs can be used for homogeneity testing and pharmacogenetic studies that could lead, at first to the identification of common diseases like heart pressure or diabetes and later, after a thorough understanding of intrinsically disordered proteins, of more complex diseases like cancer or neurodegenerative and cardiovascular diseases (Shastry 2003).

Missense mutations can occur both in structured as well as in intrinsically disordered proteins; NsSNPs in structured proteins can influence the folding of the amino acid chain, the whole stability and structure of the protein or its solubility level, while a mutation within an IDP can lead to loss/gain of interaction within the protein network, PTMs, or disruptions within the structure (causing a disordered region to become ordered) (Vacic & Iakoucheva 2012). These modification, in turn can lead to a loss or gain of function that may be pathogenic. The figure below graphically emphasizes the consequences missense mutations can have on both types of proteins.

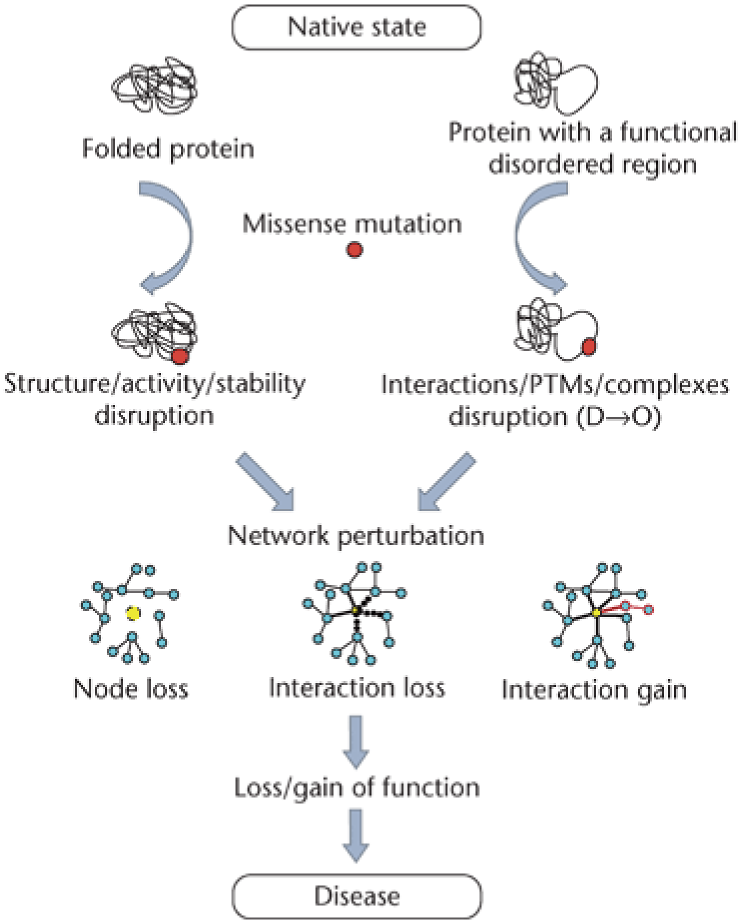


Figure 3-1Non-synonimous single nucleotide polymorphisms(nsSNP)

Finding cohesive data sources on point mutations is not as straightforward as one might think however. There are a multitude of available databases; nevertheless, most of them present the data either at gene level, or the annotations included are not of the high-quality needed for this project. The following two sections will briefly describe the databases used as data source.

3.2.1 UniProtKB/SwissProt database

UniProtKB/SwissProt is a highly specialized, manually curated protein sequence and knowledge database that concentrates on protein entries from humans (Boeckmann 2003). It is an extremely useful source of information for each protein as the manual annotations are extracted from scientific literature and use standardized nomenclature.

The UniProtKB/SwissProt database is the starting point for finding the relevant data for this thesis as this repository contains the unprocessed data needed for answering our research question.

3.2.2 SwissVar

The SwissVar portal is a collection of single amino acid polymorphisms and diseases retrieved from the UniProtKB /Swiss-Prot databases. It provides a unique environment for researchers to investigate the relationship between human variants and phenotypes. Compared to other repositories, SwissVar offers the possibility of searching the database for similar diseases and not just by the query protein sequence. Another aspect that makes SwissVar more attractive for our research is the inclusion of disease and phenotype information associated with a particular mutation; the other databases proved to be relatively unstructured and offering only general gene-centric information. At present, the database includes 3300 diseases and 66,397 human protein variations with 22,892 known to be disease variants, 37,159 netrual variants and 5,346 unclassified (Mottaz et al. 2010). The accession codes for each protein that contains point mutations are extracted and used in the UniProtKB/SwissProt database to acquire the protein sequences (based on the SwissVar accession codes).

4 IDP predictors

So far we have presented what intrinsically disordered proteins and non-synonymous single nucleotide polymorphisms are. We have discussed the functions and structures of these entities, as well as the databases where they can be downloaded from. The goal of this thesis was to analyze whether there is a direct relationship between point mutations and intrinsically disordered proteins in terms of pathogenicity based on a single feature, i.e. the conservation status of the amino acid mutation. Since this question has yet to be answered within the bioinformatics field, it is imperative to present the tools needed to identify these mutations. At the moment there are over 60 intrinsically disorder protein predictors, but the methodology is still in its infancy. There is no commonly accepted criterion on selecting these IDPs and the best so far only distinguishes between long or short IDPs because of the different characteristics of such regions. In fact, the predictive algorithms that work properly on long disordered regions or IDPs perform poorly on short ones (Peng et al. 2005). Nevertheless, protein structure prediction is one of the most important fields of study in bioinformatics and shifting from the saturated structured protein market onto the newly intrinsically disordered one will be beneficial from a medical point of view since the link with human disease has already been acknowledged by the scientific community. In this chapter we will present the predictors for intrinsic disorder trialled for this thesis.

The existing IDP predictors use features such as the amino acid composition corroborated with the physicochemical properties of amino acids to determine whether the query sequence falls within a disordered region or not. Others use the amino acid sequence scores derived from PSI-BLAST (bioinformatics algorithm to compare protein sequences for similarities) or from protein secondary structure information. The most popular classifiers are based either on Neural Networks or on Support Vector Machines learning models. However, not all of the following methods use the same criteria for disorder. Moreover, the existing predictors should be taken at face value since all of them present both advantages and disadvantages and none can be deemed superior over another at a significant level. It is actually recommended to combine two or more of these predictors in order to avoid pitfalls like when few homologues are available or high false positives due to too many homologues being identified. Nevertheless, the choice of disorder predictor is highly dependent on the types of research questions the scientists are trying to answer.

To help the scientific field, a series of experiments called Critical Assessment of protein Structure Prediction (CASP) have been developed since 1994 in order to test all predictors in an unbiased matter. Every year, the new predictors are benchmarked against the old ones and new features are scrutinized for their information gain. Based on the current academic preferences for disorder predictions we will present the algorithms used by DISOPRED2, PONDR VSL2 and IUPred.

4.1 DISOPRED2

The DISOPRED2 server uses a knowledge-based algorithm to predict IDPs from the amino acid sequence. Just like the first predictor, DISOPRED 2 trains a neural network to analyse the sequence profiles generated by running iteratively PSI-BLAST 3 times. Basically, each amino acid is profiled from the PSI-BLAST and classified using the neural network. Then, the actual disorder predictor uses a support vector machine algorithm that outputs a probability for the amino acid to fall in a disordered region or not (Jones & Ward 2003).

The training set included those proteins with available X-ray structures in the PDB database. In order to correctly identify the disordered regions within the protein sequence, the amino acid sequence (SEQRES records) was aligned with the ATOM records (or alpha-carbon coordinates); the SEQRES and ATOM records represent the sequence of the protein and the amino acid coordinates respectively in the PDB format. A region was deemed disordered if and only if the amino acids found in the SEQRES records did not correspond to those in the ATOM records.

The performance of the predictor is benchmarked to the targets derived from CASP5 (2005). The authors did indeed notice that the data are slightly overfitted when the disordered region with less than 0.8 confidence are included in the SVM predictor. Given the binary nature of the answer (that the protein is disordered or not) and the disparity between the sizes of the structured and disordered protein classes, the Matthews Correlation Coefficient is used (Jones & Ward 2003).

The DISOPRED predictor was developed with the aim of identifying short disordered proteins/regions using the amino acid sequence after scoring with the Position Specific Scoring Matrix (PSSM) utilized by PSI-BLAST to determine a profile of the amino acids in the query protein sequence. The CASP5 experiments deemed it significant for identifying short regions devoid of secondary structure.

4.2 PONDR family

PONDR or Predictor of Naturally Disordered Regions is a set of algorithms used for disorder prediction within protein regions. The PONDR family works primarily from primary sequence data. The predictors use a number of standard feed forward neural networks that incorporate sequence information derived from sliding windows of 9 to 21 amino acids (http://www.pondr.com/).

The PONDR family includes 9 meta-predictors with the most famous ones PONDR VL-XT and PONDR VSL2. The first one is actually three predictors merged into one, one trained on Variously Long disordered regions and two trained on X-ray characterized Terminal disordered regions (http://www.pondr.com/pondr-tut2.html).

The PONDR VL-XT neural network includes a small number of attributes among which: amino acid composition, amino acid net charge and average coordination number (number of neighbours a residue has). The algorithm will predict the order-disorder class for every amino acid in the query sequence. The algorithm is particularly useful in identifying the regions that are undergoing disorder to order transitions (Dosztányi et al. 2010).

The PONDR VSL2 is a combination of two predictors optimized both for long (>30 residues) and short (<30 residues) disordered regions. This predictor is constructed as a linear support vector machine compared to previous versions which used logistic regression models. PONDR VSL2 successfully addressed the length-dependency problem that most IDP predictors faced and proved to outperform both the VL-XT and the DISOPRED2 predictors at per residue values (Peng et al. 2006).

The PONDR family encompasses a series of disorder predictors that work on short, long and both types of disorder regions; all algorithms output scores for individual residues facilitating the analysis of the data at a per residue level. Unfortunately, all PONDR predictors are computationally expensive in terms of the time needed to output the results per protein as it uses PSI-BLAST for each queried protein.

4.3 IUPred

IUPred is a very effective disorder predictor that has a different approach from the rest of the classifiers out there. It addresses the causes that lead a protein not to fold; basically, if an amino acid cannot form enough intra-chain contacts, then it will not be able to adopt a stable locus in the 3D structure of the chain (Dosztányi et al. 2010). If such amino acids cluster together for a portion or the whole of the amino acid chain then that protein will be deemed intrinsically disordered. The algorithm the authors implement is based on *statistical potential* (energy functions derived from the analysis of known protein structures found in the Protein data Bank). More specifically, the pairwise energy of a protein is a function of its amino acid sequence and its conformation. They construct a 20x20 interaction matrix based on all the amino acids existent in nature and assign each pair a number signifying their energy content (Dosztányi et al. 2005).

The algorithm developed allows the authors to assess the foldability of proteins even when the structural model is absent. There is no training involved as the delineation between disordered and ordered proteins is done based on the energy content of each amino acid pair. The scoring for each position is computed by only taking into consideration interaction partners that are 2 to 100 residues apart in each direction. The score is then smoothed over a sliding window of 21 amino acids. This new predictor was tested on a set of globular proteins and proved to outperform both DISOPRED2 and PONDRVL3H (another algorithm from the PONDR family that specializes in identifying long disordered regions, i.e. larger than 30 residues) (Dosztányi et al. 2005).

The predictor itself is straightforward to use; it takes as input the individual amino acid sequence and outputs a score between 0 and 1 for each amino acid; everything that has a score above 0.5 is considered to be disordered. The computational time is extremely fast, making this predictor a viable candidate for our data analysis.

4.4 Conclusion

Different empirical studies have proven that no one single disorder predictor will correctly reveal the structure of a protein. However, in combination and depending on the end result needed for the research, some predictors perform quite well. For instance, if the purpose of a study is to identify large disordered regions, then PONDR VL3 will prove optimal (Dosztányi et al. 2010). DISOPRED2 is useful for identifying smaller disordered segments. PONDR VL-XT and VSL2 together with IUPred are preferred for identifying both short and long disordered regions.

In order to select the best disorder predictor for our thesis we started off by identifying what was actually needed for answering our research question. Since we are investigating disease causing point mutations within disordered proteins/regions we needed a predictor that would output a single score per residue. All of the above mentioned predictors give this output. However, we also needed an algorithm that would be fast and easy to integrate within our Perl script. All of the predictors were implemented in C, but the decisive factor that led us into choosing IUPred was its speed. Both IUPred and PONDR VSL2 need the extra step of accessing PSI-BLAST and identifying homologues which is time consuming.

In conclusion, we considered that IUPred is easiest to adopt given our task and is extremely efficient, time wise at outputting a disorder score.

5 SNP predictors

Earlier we have identified the importance of understanding intrinsically disordered proteins and their link to disease. IDPs are more susceptible to misfolding or malfunction because of, for instance, binding to a different partner or alternative splicing or posttranslational modifications or because of a missesnse mutation. In this chapter we will identify the current state of the art SNP predictors and discuss the ones that have incorporated features for measuring the level of disorder within a protein.

Non synonymous SNPs have been in the limelight ever since the whole human genome was completely mapped. Since then, scientists have concentrated on developing predictors that will determine when an nsSNP is deleterious or neutral. Most of the available tools identify and annotate the function altering and disease associated amino acid substitutions based only on either the protein sequence or the protein structure; they fail to identify the changes brought up by molecular functions introduced or altered by a missense mutation. Based on this caveat, tools concentrating solely on predicting the disease causing nsSNPs have been developed; SIFT (P. C. Ng 2003) and PolyPhen2 (Ramensky et al. 2002)(I. a Adzhubei et al. 2010) are the ones mostly used on average. Apart from these though, different methodologies have arisen, but the most widely used ones are *1)* empirical rule-based, *2)* probabilistic models and *3)* machine learning approaches.

Nevertheless, it must be mentioned that all of the existing predictors are built for structured proteins. Unfortunately, the properties of folded proteins are different from those of IDPs and implying that these predictors will perform well in disordered regions is simply erroneous. Some of the more recent nsSNP predictors have incorporated a special attribute into the classifier that verifies, with the help of disorder predictors, whether the mutation falls within a disordered region or not. This inclusion is a step up towards recognizing the importance of disordered proteins for human pathogenicity.

In this chapter we will present the most common SNP predictors available, together with some new classifiers that incorporate disorder among their features.

5.1 SIFT

SIFT (Sorting Intolerant From Tolerant) is a classifier that through the aid of homology verifies whether an amino acid substitution could alter the protein function and hence the phenotype (P. C. Ng 2003). The SIFT algorithm utilizes only the protein sequence to infer its predictions, yet, it registers performances similar to those predictors that use structure (Pauline C Ng & Steven Henikoff 2002). The algorithm follows two steps: first, the homologues of the query sequence are obtained through multiple sequence alignment by performing PSI-BLAST on the SWISS-PROT database with two iterations; secondly, the amino acid substitution is given a probability score based on the alignment for that given position obtained in the previous step. The algorithm verifies the obtained probability against a threshold in order to infer that the amino acid substitution indeed altered the protein function.

The SIFT algorithm only uses one key feature, and that is conservation which is computed for each position of the residues for every potential sequence alignments. Then, the median conservation over all the amino acid positions within a sequence is computed and if it does not fall under a user-defined threshold then this protein hit is preserved and added to the final alignment. This process continues until the threshold is reached. The final alignment will include all protein hits with conservation levels above the threshold. To diminish noise from pseudo genes, all sequences which are more than 90% identical to the query sequence are removed from the final alignment.

For the second step, the alignment obtained in the first step is converted into a Position Specific Scoring Matrix (PSSM). A PSSM is a 20x L (length of the protein sequence) matrix where each cell contains the probability of an amino acid a found at position c (p_{ca}) to be replaced by any of the remaining 19 amino acids. The formula the authors developed is:

$$p_{ca} = \frac{N_c}{N_c + B_c} * g_{ca} + \frac{B_c}{N_c + B_c} * f_{ca}$$

Where N_c is the total number of sequences in the alignment; B_c is the total number of pseudocounts; g_{ca} is the sequence-weighted frequency that amino acid a will appear at position c in the alignment; f_{ca} pseudocouns computed from Dirichlet mixtures (P C Ng & S Henikoff 2001).

SIFT is a very useful tool for identifying exactly how protein function is altered given an amino acid substitution/missense mutation.

5.2 PolyPhen2

PolyPhen2 (polymorphism phenotyping) is a server for functional annotation of coding nsSNPs (Ramensky et al. 2002). It also provides a method for predicting damaging missense mutations. A Naive Bayes classifier is used to determine whether a missesnse mutation is deleterious or not. Compared to the first PolyPhen algorithm, PolyPhen2 has been greatly improved in terms of predictive features, alignment pipeline and method of classification (I. a Adzhubei et al. 2010).

The automated sequence alignment pipeline follows a 6 step procedure:

1. The user inputs an nsSNP or sequence;
2. With the query sequence in place, PSI-BLAST is performed on the Uniref100 database to retrieve the homologues;
3. Then the amino acid sequences found in the previous step are aligned using MAFFT (Katoh et al. 2002); MAFFT aligns homologues through Fast Fourier Transform;
4. To minimize the noise contained in the alignments because of the poor-quality segments, the LEON software is used; homology is predicted only from complete sequences (Thompson et al. 2004);
5. The remaining sequences are clustered together and only the cluster containing the query sequence is preserved; this step is done using the Secator algorithm (Wicker et al. 2001);
6. Lastly, the Position Specific Independent Counts (PSIC) profile is computed and the final alignment scores are obtained (S. R. Sunyaev et al. 1999); the profile score indicates how likely it is for a certain amino acid to occupy a specific position within the protein sequence.

From the homologues found through multiple sequence alignment, the characteristic features of the classifier are extracted. The top feature with the highest information gain verifies how likely the mutation is to occur given the multiple sequence alignment and how distant is the protein that had the first point mutation from the wild-type.

PolyPhen2 uses a total of 11 features automatically selected with the aid of a greedy algorithm. These features are then implemented to train a Naive Bayes classifier in identifying the likelihood of a missense mutation to affect the protein function and phenotype. Naive Bayes performed equally well as other classifiers like logistic regression or decision trees, but because of the simplicity and reduced computation time Naive Bayes was chosen (I. A. Adzhubei et al. n.d.). Moreover, PolyPhen2 is trained on two different datasets (HumDiv and HumVar).

5.3 SySAP

SySAP (System-level predictor of deleterious Single Amino Acid Polymorphisms) is the newest predictor of deleterious single amino acid polymorphisms available. It was developed in 2010 and brought in the form of a server to the academic world in 2012. SySap is a new classifier where each Single Amino Acid Polymorphism is coded using 472 features derived from the transformed scores of the amino acid index, PSSMs, structural features and betweenness and the KEGG enrichment scores of the protein neighbours (the KEGG pathway is a network specific features and describes the molecular interaction and reaction networks for human diseases) (Huang et al. 2010). All 472 features are further analysed using the Maximum Relevance Minimum Redundancy (mRMR) and Incremental Feature Selection (IFS) in order to obtain the optimal number of features to be used. After this, the prediction method for the server that the authors choose is the LinlineaR machine learning approach because it performs best in a short time and on large sample datasets (Huang et al. 2012).

Apart from protein network features, SySAP also included features of the disorder scores. These scores were calculated using the PONDR VSL2 algorithm. On top of the disordered scores for each amino acid comprised in the sequence, the authors also identified whether the mutation falls within a disordered region or not. They run BLAST on the query sequence against the DisProt database and based on the hits generated by BLAST they transferred the annotated disorder region to the query sequence (Huang et al. 2010).

Out of the 472 features the original paper included, only 9 were disorder related features. Unfortunately, the authors do not mention how many of these disorder features survived after applying the mRMR algorithm to determine the features with the highest information gain. There is a high chance that these features were dropped partially from the final list of features that was actually fed to the classifier, possibly leaving disorder out of the classification.

The authors benchmarked their results against the SIFT predictor and reported that SySAP performed better (around the 80% mark) in comparison to SIFT (around 71%). Nevertheless, it must be mentioned that SIFT bases its prediction solely on the conservation scores derived from PSSM whilst SySAP includes more hundreds of features that are bound to overfit the data.

5.4 Conclusion

Apart from SySAP, no other mutation predictor has included features that measure the level of disorder within a protein. Unfortunately, this is mostly because the links between disease causing mutations and disordered proteins has yet to be explored. In this thesis we started from the hypothesis that point mutations within disordered regions that have an impact on the level of disorder may be disease causing. Answering this question is useful for identifying which features must be included in larger scale mutation predictors in terms of identifying intrinsic disorder. This could explain why the disorder features incorporated by the SySAP authors did not prove conclusive.

IDPs exhibit a different evolution rate and amino acid substitution patterns than structured proteins; because of this, when applying SIFT or any other deleterious SNP predictor these mutations can be wrongly labelled as neutral (because of the different conservation status) (Bagchi et al. 2011).

Both SIFT and PolyPhen2 are two credible SNP predictors that have credibility since they are used as benchmark for newer predictors. Incorporating disorder features into SNP predictors of this calibre could add another dimension to the area of point mutation and possibly help classify mutations that before had a great deal of uncertainty attached to them.

6 Extant research on disease mutations and disorder

This chapter will give an overview of the current academic literature on the topic of disease and mutations within intrinsic disorder. The field is in its infancy which would explain the scarcity of research done.

6.1 Disease mutations in disordered regions

The first paper published in 2012 looks at the disease causing mutations found in disordered regions and analyze the impact these mutations would have on the protein disorder (Vacic & Iakoucheva 2012). The central theme of the paper is to identify the functional impact of mutations in disordered regions. The authors pinpoint the idea that structured protein SNP predictors will classify mutations within non-conserved regions (or disordered regions) as benign or tolerant, thus missing potentially disease causing mutations. They identify the effects mutation could have on IDPs in the form of hypotheses that mutations may impact disordered regions at the protein-protein, protein-DNA, protein-RNA and protein – ligand interactions level; they can also affect posttranslational modifications, as well as signalling and regulatory networks.

To support their allegations, the authors downloaded all the disease associated mutations within the UniProt database and ranked the substitutions according to their frequency in mutant proteins. They identified a series of amino acid substitutions that cause a disordered protein to become more ordered and vice-versa. They also only concentrated on certain amino acid substitutions that they have found to be recurrent within their dataset. Moreover, they have extracted a series of known disordered proteins based on the DisProt database and analysed their disorder to order transitions. The authors focused more on particular cases refraining from making overall conclusions since their analysis was more theoretically rooted.

They did not incorporate their findings into a classifier but advocate for further research to be done in the area of mutations (be them missense, nonsense, splice or indels) within intrinsic disorder for a better understanding of the functional changes these mutations can have at the protein level.

6.2 Changes in predicted protein disorder tendency contributes to risk disease

The second paper looks to identify those single nucleotide variants (SNV) which lead to an amino acid substitution that will change the function and disorder tendency of a protein (Hu et al. 2011). The data they used was gene centric and consisted of simulated autosomal genes as a preview to what scientists would expect to find analysing real data. The authors first mapped the mutations within the amino acid sequence and then verified a mutation's ability to alter disorder by adjusting the disorder score obtained through the PONDR VSL2 algorithm. the *mutation-induced change in disorder prediction scores* (ΔDS) is computed to identify whether a significant change in a protein caused by a mutation leads to malfunctions and disease development. Again, the paper did not construct an actual predictor of disease causing mutations within disordered regions, but merely adjusted the disorder score for the given amino acid substitution obtained via an IDP predictor. They also did not determine a sensible threshold for separating harmful mutations from harmless ones.

The two papers above are the precursors of the future deleterious SNP predictors within disordered regions. They offered a good starting point in terms of approaches in analyzing the potential of change in structure from Disorder-to-Order that a mutation can have on disordered regions. Most significant method that could prove useful for this project is the *mutation-induced change in disorder prediction scores* ($\Delta DS = DS_{min} - DS_{maj,}$ where DS_{min} and DS_{maj} are the disorder scores for minor and major alleles respectively) which measure the ability of a mutation to structurally change a region to form disorder/structure.

7 Data Preprocessing

This chapter acts as a guide into the preparation of the dataset for its further analysis. We first determine the data sources and then describe the method used for parsing the proteins for obtaining both the wild and mutant protein sequences into the right format.

7.1 Data sources

One of the most challenging tasks of our project was finding the right data and pre-processing it for further analysis. Because of its high quality and extensive manual annotations we chose the UniProtKB/SwissProt database as our main source of information, together with the SwissVar portal which hosts all the mutations within the SwissProt database. Since we chose to deal only with human disease, we used the humsavar.txt file, which is a variation of the SwissVar file that contains the protein accession codes and amino acid changes only for human mutations.

Before going more in-depth we will explain what wild and mutated proteins are. A wild protein sequence refers to the linear amino acid chain in its original state. A mutated protein sequence is the linear amino acid chain that at a certain locus (i.e. where the point mutation occurs) the amino acid differs from the one found in the wild protein sequence.

We used two main raw data files: the uniprot_sprot.fasta file containing all the wild (not mutated) protein sequences from the UniProtKB/SwissProt database and humsavar.txt containing a list of wild proteins for which their mutations have been annotated as disease carrying or not.

The UniProt/SwissProt file contains approximately 4 million rows of amino acid strings (or more than 800 thousand protein sequences) so we extracted only the relevant 66,397 protein sequences which are annotated in humsavar.txt.

Table 7-1 shows a sample of rows from humsavar.txt. There are three relevant columns from this table that we used: Swiss-Prot AC which contains the accession code that we used to extract the protein sequences from the Uniprot/Swissprot file, AA change which encodes the mutation that the wild protein suffered, and the Type of variant column which labels the mutation as being disease causing, non-disease causing (polymorphism) or unclassified.

Row #	Swiss-Prot AC	AA change	Type of variant	Disease name
1	P04217	p.His52Arg	Polymorphism	-
2	P04217	p.His395Arg	Polymorphism	-
17	Q9NPC4	p.Met183Lys	Unclassified	-
18	Q9NPC4	p.Gly187Asp	Polymorphism	-
23	A6NGZ7	p.Gly192Arg	Polymorphism	-
24	A6NLB4	p.Trp263Arg	Polymorphism	-
26	Q9NRG9	p.Gln15Lys	Disease	Achalasia-addisonianism-alacrima syndrome
28	Q9NRG9	p.Ser263Pro	Disease	Achalasia-addisonianism-alacrima syndrome
45	Q2M2I8	p.Gln533His	Polymorphism	-
53	Q16613	p.Ala129Thr	Disease	Delayed sleep phase syndrome
75	Q6ZMQ8	p.Leu97Val	Unclassified	A lung adenocarcinoma sample
217	Q99758	p.Glu801Asp	Unclassified	A breast cancer sample

726	O95255	p.Arg1314Trp	Disease	Arterial calcification of infancy, generalized
1139	P00519	p.Arg166Lys	Unclassified	A melanoma sample
1304	Q15027	p.Lys114Arg	Unclassified	A breast cancer sample
1305	Q15027	p.Arg129Gln	Unclassified	A colorectal cancer sample
1371	Q99798	p.Ser112Arg	Disease	Infantile cerebellar-retinal degeneration
1458	Q9ULC5	p.Lys388Arg	Unclassified	A colorectal cancer sample
1459	Q9ULC5	p.Gly466Asp	Unclassified	A colorectal cancer sample
6143	Q92560	p.Ala95Asp	Unclassified	A lung cancer sample
6144	Q92560	p.Gly178Val	Unclassified	A lung cancer sample
6799	O00238	p.Arg149Trp	Polymorphism	-
6899	P15056	p.Gly466Val	Disease	Lung cancer
6900	P15056	p.Leu597Arg	Disease	Lung cancer
6903	P15056	p.Gly464Glu	Disease	Colorectal cancer
7011	P38398	p.Glu23Lys	Disease	Familial breast-ovarian cancer type 1
13389	P12111	p.Arg1395Gln	Disease	Ullrich congenital muscular dystrophy
66245	P51508	p.Ser179Asn	Disease	Mental retardation X-linked type 45
66399	Q8IYH5	p.Pro456Ser	Unclassified	A colorectal cancer sample

Table 7-1 Sample rows and selected columns from humsavar.txt

7.2 Parsing the protein sequences

To give a better explanation of the data preparation process we take as an example the first row from Table 7-1. The accession number is P04217 which we will use to extract its protein sequence from the UniProtKB/SwissProt file. The actual extracted protein sequence is shown in Table 7-2. The first row follows the standard fasta format containing the accession number P04217 and other annotations. The fasta format is a text-based format for representing a protein sequence, in our case at the amino acid level in a single letter format. The first line contains representative information about the protein like the database it is found in, its accession code, the name of the protein, the species etc.

P04217.fasta
>sp|P04217|A1BG_HUMAN Alpha-1B-glycoprotein OS=Homo sapiens GN=A1BG PE=1 SV=4 MSMLVVFLLLWGVTWGPVTEAAIFYETQPSLWAESESLLKPLANVTLTCQAHLETPDFQL FKNGVAQEPVHLDSPAIKHQFLLTGDTQGRYRCRSGLSTGWTQLSKLLELTGPKSLPAPW LSMAPVSWITPGLKTTAVCRGVLRGVTFLLRREGDHEFLEVPEAQEDVEATFPVHQPGNY SCSYRTDGEGALSEPSATVTIEELAAPPPPVLMHHGESSQVLHPGNKVTLTCVAPLSGVD FQLRRGEKELLVPRSSTSPDRIFFHLNAVALGDGGHYTCRYRLHDNQNGWSGDSAPVELI LSDETLPAPEFSPEPESGRALRLRCLAPLEGARFALVREDRGGRRVHRFQSPAGTEALFE LHNISVADSANYSCVYVDLKPPFGGSAPSERLELHVDGPPPRPQLRATWSGAVLAGRDAV LRCEGPIPDVTFELLREGETKAVKTVRTPGAAANLELIFVGPQHAGNYRCRYRSWVPHTF ESELSDPVELLVAES

Table 7-2 Example file P04217.fasta containing the amino acid sequence of the protein with the accession number P04217, extracted from the wild protein sequence flat file uniprot_sprot.fasta

Having the wild protein sequence we now need to obtain the mutated version. To do this we use the mutation code in the column AA change: p.His52Arg. This code is interpreted as a mutation at position 52 in the amino acid string from His to Arg. We cannot use the three letter amino acid codes in their original format so we had to transform them into one letter codes by using Table 7-3.

From:	To:	From:	To:
Ala	A	Leu	L
Arg	R	Lys	K
Asn	N	Met	M
Asp	D	Phe	F
Cys	C	Pro	P
Glu	E	Ser	S
Gln	Q	Thr	T
Gly	G	Trp	W
His	H	Tyr	Y
Ile	I	Val	V

Table 7-3 Conversion table from three letter amino acid codes to one letter amino acid codes (aminoacid_codes.txt)

With this information we can now proceed to create the file with the mutated variant by changing the H amino acid at position 52 with T. The resulting mutated sequence can be seen in Table 7-4.

P04217_HR52.fasta

```
>sp|P04217|A1BG_HUMAN Alpha-1B-glycoprotein OS=Homo sapiens GN=A1BG PE=1 SV=4
MSMLVVFLLLWGVTWGPVTEAAIFYETQPSLWAESESLLKPLANVTLTCQARLETPDFQL
FKNGVAQEPVHLDSPAIKHQFLLTGDTQGRYRCRSGLSTGWTQLSKLLELTGPKSLPAPW
LSMAPVSWITPGLKTTAVCRGVLRGVTFLLRREGDHEFLEVPEAQEDVEATFPVHQPGNY
SCSYRTDGEGALSEPSATVTIEELAAPPPPVLMHHGESSQVLHPGNKVTLTCVAPLSGVD
FQLRRGEKELLVPRSSTSPDRIFFHLNAVALGDGGHYTCRYRLHDNQNGWSGDSAPVELI
LSDETLPAPEFSPEPESGRALRLRCLAPLEGARFALVREDRGGRRVHRFQSPAGTEALFE
LHNISVADSANYSCVYVDLKPPFGGSAPSERLELHVDGPPPRPQLRATWSGAVLAGRDAV
LRCEGPIPDVTFELLREGETKAVKTVRTPGAAANLELIFVGPQHAGNYRCRYRSWVPHTF
ESELSDPVELLVAES
```

Table 7-4 Example file P04217_HR52.fasta containing the amino acid sequence for the protein with accession number P04217 containing a point mutation at position 52 from H to R (AA change = p.His52Arg).

In order to prepare the data for usage in further analysis we change the labels from the column ‘Type of variant’ to class numbers based on Table 7-5.

Class Text Label	Class Numeric Label
Disease	1
Polymorphism (= No Disease)	0
Unclassified	-1

Table 7-5 Class labels and corresponding numbers used for building the classifier

As mentioned in chapter 4, we have decided to use the IUPred disorder predictor since it gives an output in the form we require and it is computationally fast and easier to integrate within the Perl script than the other two predictors presented earlier.

After obtaining the wild and mutated protein sequences we can feed them into IUPred to obtain the disorder scores for each of the two variants. Table 7-6 shows a selection of the results for our example protein with accession number P04217. The left half of the table shows the disorder scores for each amino acid in the wild protein and the right half shows the scores for the mutated protein. We extracted the scores at the mutation position, the average of a symmetrical 11 amino acid window and a 21 amino acid window (5 respectively 10 amino acids on each side of the mutation locus).

P04217.iupred	**P04217_HR52.iupred**
# sp|P04217|A1BG_HUMAN	# sp|P04217|A1BG_HUMAN
1 M 0.0044	1 M 0.0045
…	…
42 L 0.1137	42 L 0.1162
43 A 0.2034	43 A 0.2064
44 N 0.2364	44 N 0.2399
45 V 0.2436	45 V 0.2470
46 T 0.2609	46 T 0.2645
47 L 0.1449	47 L 0.1476
48 T 0.1528	48 T 0.1583
49 C 0.1528	49 C 0.1554
50 Q 0.1298	50 Q 0.1349
51 A 0.1298	51 A 0.1349
52 H 0.1206	52 R 0.1251
53 L 0.1881	53 L 0.1914
54 E 0.1554	54 E 0.1611
55 T 0.1275	55 T 0.1323
56 P 0.1823	56 P 0.1881
57 D 0.1942	57 D 0.1969
58 F 0.3019	58 F 0.3087
59 Q 0.2364	59 Q 0.2399
60 L 0.3053	60 L 0.3087
61 F 0.2193	61 F 0.2224
62 K 0.2645	62 K 0.2680
…	…
495 S 0.3872	495 S 0.3872

Table 7-6 Output from IUPred for P04217.fasta (left) and P04217_HR52.fasta (right)

Repeating this process for all the variants from the file humsavar.txt we obtained the main three files that we used for our analysis. A representative fragment of two of these files can be seen in Table 7-7. The files contain the accession and mutation codes, the disease numeric labels, the wild disorder scores and the mutated disorder scores.

disorder_w11.txt				**disorder_w21.txt**			
Accession	Disease	scoreW	scoreM	Accession	Disease	scoreW	scoreM
P04217_HR52	0	0.152	0.156	P04217_HR52	0	0.193	0.197
P04217_HR395	0	0.673	0.659	P04217_HR395	0	0.579	0.567
…				…			
Q9NPC4_MK183	-1	0.012	0.016	Q9NPC4_MK183	-1	0.014	0.018
…				…			
Q9NRG9_HR160	1	0.035	0.036	Q9NRG9_HR160	1	0.047	0.048
…				…			

Table 7-7 Two of the three files that we will use for analysis containing: accession number with a mutation code, disease label, IUPred score of the original (wild) protein and IUPred score of the mutated protein. The file in the left contains average disorder scores for a window of 11 amino acids and the one in the right for a window of 21 amino acids.

The data preprocessing was a challenging scripting task which we implemented in Perl as the datasets are not readily available; the wild protein sequences needed to be extracted from the larger UniProtKB/SwissProt database, while the mutated sequences need to be obtained one by one through the method described above in a fasta format. Lastly, IUPred accepts only one sequence at a time, thus constructing an efficient script to run all proteins and store the disorder amino acid scores correctly was vital.

8 Data Analysis

This chapter contains a detailed analysis of the data and an explanation of the classifiers that we designed with. The actual performance of these classifiers will be presented in chapter 9.

8.1 Levels of disorder within the dataset

The starting point in analysing our data is to build a cross table and count the samples based on several cases: wild disorder score greater or less than 0.5, mutated disorder score greater or less than 0.5, disease equal to 1 or 0. Table 8-1 shows these counts for the case when we extract only the mutated amino acids disorder score from IUPred (amino acid window = 1). Ordered proteins have been extensively more studied in the literature than intrinsically disordered proteins and this can be seen from the table as well: here are four times more ordered (score wild < 0.5) cases in the dataset than disordered (score wild > 0.5) based on the wild disorder score. The same situation is found if we analyse the counts based on the mutated disorder score but the ration is slightly higher. Important findings are that 19.77% of mutations fall in disordered regions and 19.13% of mutations from disordered regions are disease causing. What is interesting is that it happens very rarely that a protein goes from disorder to order state or order to disorder state when suffering a mutation: 8.02% of the cases present disorder to order transitions and 1.50% of the cases are order to disorder transitions.

Amino acid window = **1**		**Score wild (sw)**		**Score mutated (sm)**		**Disease (d)**	
Total sequences = 60,051		< 0.5	> 0.5	< 0.5	> 0.5	= 0	= 1
Score wild (sw)	< 0.5	47,926	0	47,207	719	27,320	20,606
	> 0.5	0	12,125	972	11,153	9,839	2,286
Score mutated (sm)	< 0.5	47,207	972	48,179	0	27,558	20,621
	> 0.5	719	11,153	0	11,872	9,601	2,271
Disease (d)	= 0	27,320	9,839	27,558	9,601	37,159	0
	= 1	20,606	2,286	20,621	2,271	0	22,892

Table 8-1 Sample counts for several cases of the variables wild disorder score, mutated disorder score and disease for the case when the amino acid window is equal to one. In total there are in total 60,051 sequences presented in this table.

Table 8-2 shows the same attributes as Table 8-1 for the average scores for a window of 11 amino acids. The total number of protein sequences in this table is smaller due to the fact that some of the mutations occurred too close to the ends of the proteins, therefore we did not have enough data to calculate the average of the amino acid window. What we are most interested in is if there are any significant changes in the counts due to the increase in the window size. It seems there are no significant differences most likely due to the fact that a point mutation not only changes a single disorder score but it also changes the disorder scores around it: 19.33% of mutations fall in disordered regions, 18.69% mutations from disordered regions are disease causing, 7.65% disorder to order transitions and 1.61% order to disorder transitions.

Amino acid window = **11**		**Score wild (sw)**		**Score mutated (sm)**		**Disease (d)**	
Total sequences = 59,876		< 0.5	> 0.5	< 0.5	> 0.5	= 0	= 1
Score wild (sw)	< 0.5	48,184	0	47,410	774	27,386	20,798
	> 0.5	0	11,692	894	10,798	9,601	2,091
Score mutated	< 0.5	47,410	894	48,304	0	27,578	20,726

(sm)	> 0.5	774	10,798	0	11,572	9,409	2,163
Disease (d)	= 0	27,386	9,601	27,578	9,409	36,987	0
	= 1	20,798	2,091	20,726	2,163	0	22,889

Table 8-2 Sample counts for several cases of the variables wild disorder score, mutated disorder score and disease for the case when the amino acid window is equal to eleven. In total there are in total 59,876 sequences presented in this table.

Table 8-3 shows the same attributes as Table 8-1 for the average scores for a window of 21 amino acids. The number of total sequences is actually equal to the ones in Table 8-2 which means that there were no point mutations at a distance from any end of the sequence between 5 and 10 amino acids. However, the disorder scores have shifted leading to more cases of ordered sequences based both on the wild and mutated disorder scores. The increase in the proportion of ordered sequences is due to the fact that by increasing the window size to 21 amino acids from 11 we now include more disorder scores smaller than 0.5 from amino acids which were not involved in the mutation. The percentages of interest are not too different from the previous two tables but the direction of their change can be easily explained for some of the cases: 18.87% mutations fall now in disordered regions (due to the lower proportion of disordered sequences), 18.45% mutations from disordered regions are disease causing (this happened probably because more sequences with disease have become ordered due to the increase in window size), 7.79% disorder to order transitions and 1.45% order to disorder transitions (the total number of transitions being similar to Table 8-2).

Amino acid window = **21**		**Score wild (sw)**		**Score mutated (sm)**		**Disease (d)**	
Total sequences = 59,876		< 0.5	> 0.5	< 0.5	> 0.5	= 0	= 1
Score wild (sw)	< 0.5	48,384	0	47,684	700	27,549	20,835
	> 0.5	0	11,492	895	10,597	9,438	2,054
Score mutated (sm)	< 0.5	47,684	895	48,579	0	27,774	20,805
	> 0.5	700	10,597	0	11,297	9,213	2,084
Disease (d)	= 0	27,549	9,438	27,774	9,213	36,987	0
	= 1	20,835	2,054	20,805	2,084	0	22,889

Table 8-3 Sample counts for several cases of the variables wild disorder score, mutated disorder score and disease for the case when the amino acid window is equal to twenty one. In total there are in total 59,876 sequences presented in this table

8.2 Initial graphical analysis

In this section we will present scatter plot and bivariate distribution analyses which will provide the motivation for the direction of the rest of this thesis.

While giving us a general idea and a way to check if there are any errors in our data, the tables with counts were not detailed enough for our purpose. We went further to look at the data by plotting scatter plots. The top of Figure 8-1 contains a scatter plot of the wild disorder scores (x axis) versus the mutated disorder scores (y axis) with colour labelling of the disease class (blue for polymorphism or not disease and green for disease) for the amino acid window size equal to one. What is first evident from this scatter plot is that there are no large transitions in the disorder scores after a mutation because all the points are situated close to the 45 degree line. This result is to be expected as a single mutation cannot completely change a disorder score from one extreme to the other. The density of points is larger in the region with order (where $sw < 0.5$ and $sm < 0.5$) which is in line with what we saw in the count tables.

Another feature of the data is that although diseased and not diseased proteins are found throughout the range of scores, there is large clustering of diseased proteins in the region of ordered proteins (with $sw < 0.5$ and $sm < 0.5$). Not diseased proteins seem to be more prevalent in the region of disordered proteins (with $sw < 0.5$ and $sm < 0.5$), but there is also a strong presence of diseased proteins at the very extreme of disorder (where $sw \approx 1$ and $sm \approx 1$).

We now want to check the symmetry of the transitions from disorder to order and order to disorder. The lower part of Figure 8-1shows a scatter plot where we now have the y axis representing the difference between the mutated and wild disorder scores ($sm - sw$). The transitions look very symmetrical abut we do see a small difference around the region with a very large wild disorder score (where $sw \approx 1$). Generally the findings from the scatter plots are in line with what we have seen in the tables but they do bring extra precision in our analysis.

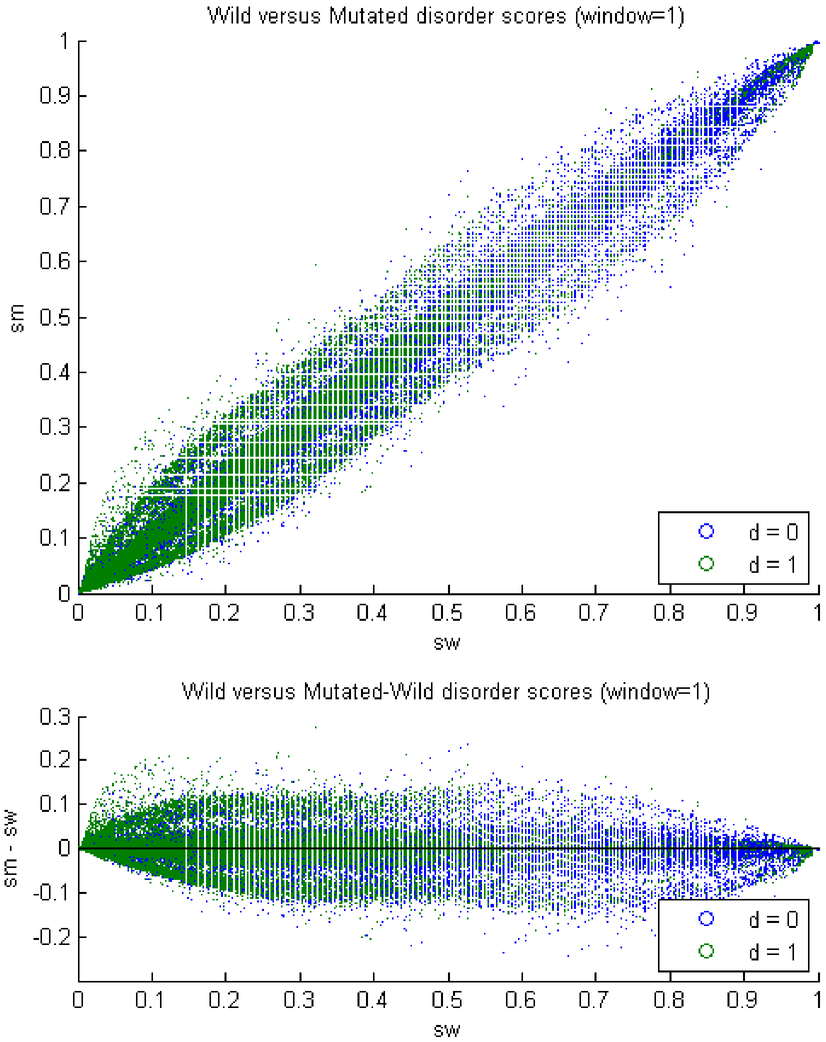


Figure 8-1 The first figure is a scatter plot of the wild disorder score (x axis) versus the mutated disorder score (y axis) for the amino acid window size equal to one. The second figure presents on the y axis the difference between the mutated disorder scores and the wild disorder scores. The diseased instances are marked with green whereas the polymorphisms (no disease) are marked with blue.

We further want to check if there is any significant difference in the general distribution of the data in the case where the amino acid window is increased. Due to space constraints we only show the case with a window size of 21 in Figure 8-2. The general conclusion is that apart from a difference in the total sample size (fewer points in this case) there is no significant difference evident from these scatter plots.

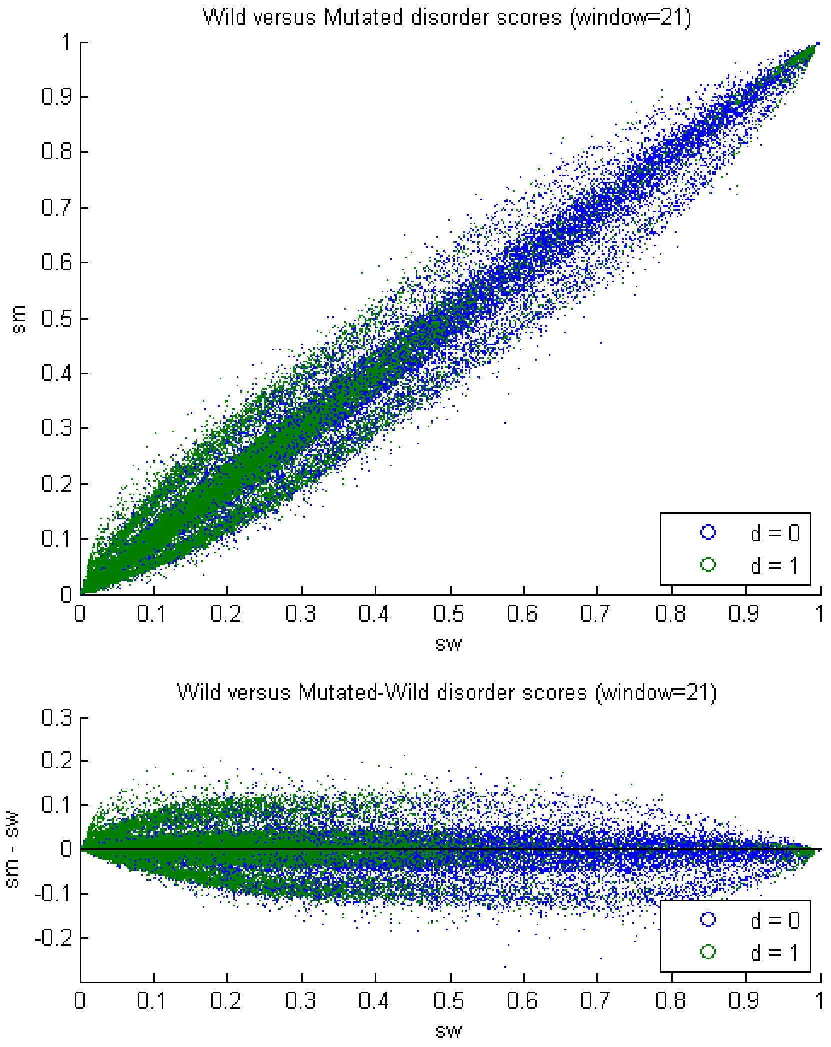


Figure 8-2 The first figure is a scatterplot of the wild disorder score (x axis) versus the mutated disorder score (y axis) for the amino acid window size equal to twenty one. The second figure presents on the y axis the difference between the mutated disorder scores and the wild disorder scores. The diseased instances are marked with green whereas the polymorphisms (no disease) are marked with blue.

To go further in our analysis we now wanted to be able to see what the frequency of the data for each region in the sw, sm space was. We achieved this by plotting the bivariate histograms of the disease and polymorphism cases in Figure 8-3. Starting with the upper part of the graph, representing the case where the window size is equal to one, we observe that there is an extremely large number of sequences (≈ 5000) with disorder scores approximately equal to zero (where $sw \approx 0$ and $sm \approx 0$). This is in line with the general trend for scientists to study mostly structured proteins which have an extremely low disorder score. The diseased ($d = 1$) proteins show a peak of frequency on the first diagonal with the highest frequencies at the very extremes of the sw, sm space. At the same time, the polymorphism instances also show higher frequency on the first diagonal but with smoother transition, leading to a lower peak proportionally at the maximum of the sw, sm space ($sw \approx 1$ and $sm \approx 1$). This is very valuable information for our project as there seems to be a higher probability of disease especially where $sw \approx 1$ and $sm \approx 1$. The lower part of the figure containing the case where the window size is equal to 21 supports the same conclusion, the only difference being that there are fewer observations and the extreme at ($sw \approx 0, sm \approx 0$) is less pronounced.

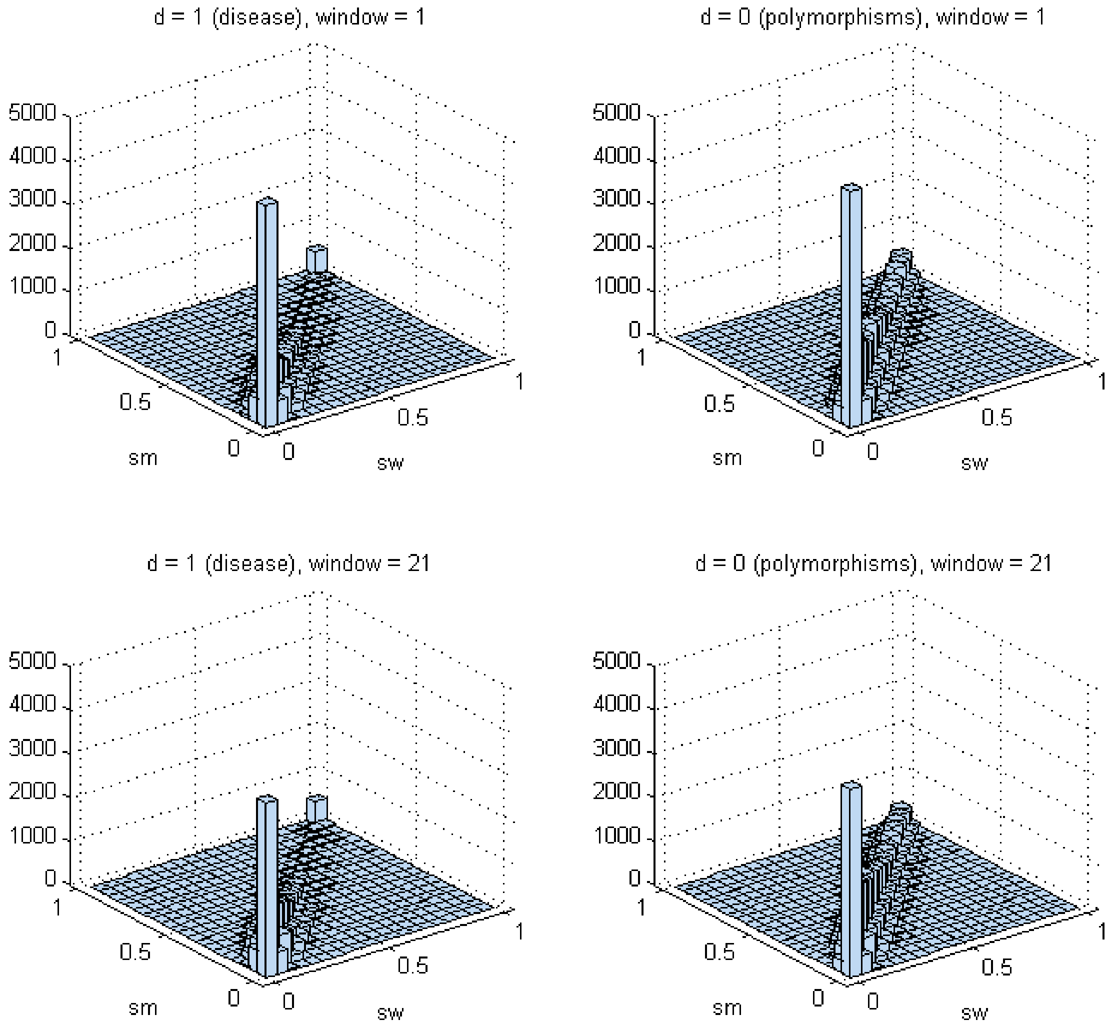


Figure 8-3 Bivariate histograms of the disease (d=1, left) and polymorphisms (d=0, right) cases for window sizes 1 (top) and 21 (bottom). The x axis represents the wild disorder score (sw) and the y axis the mutated disorder score (sm) the z axis represents the counts.

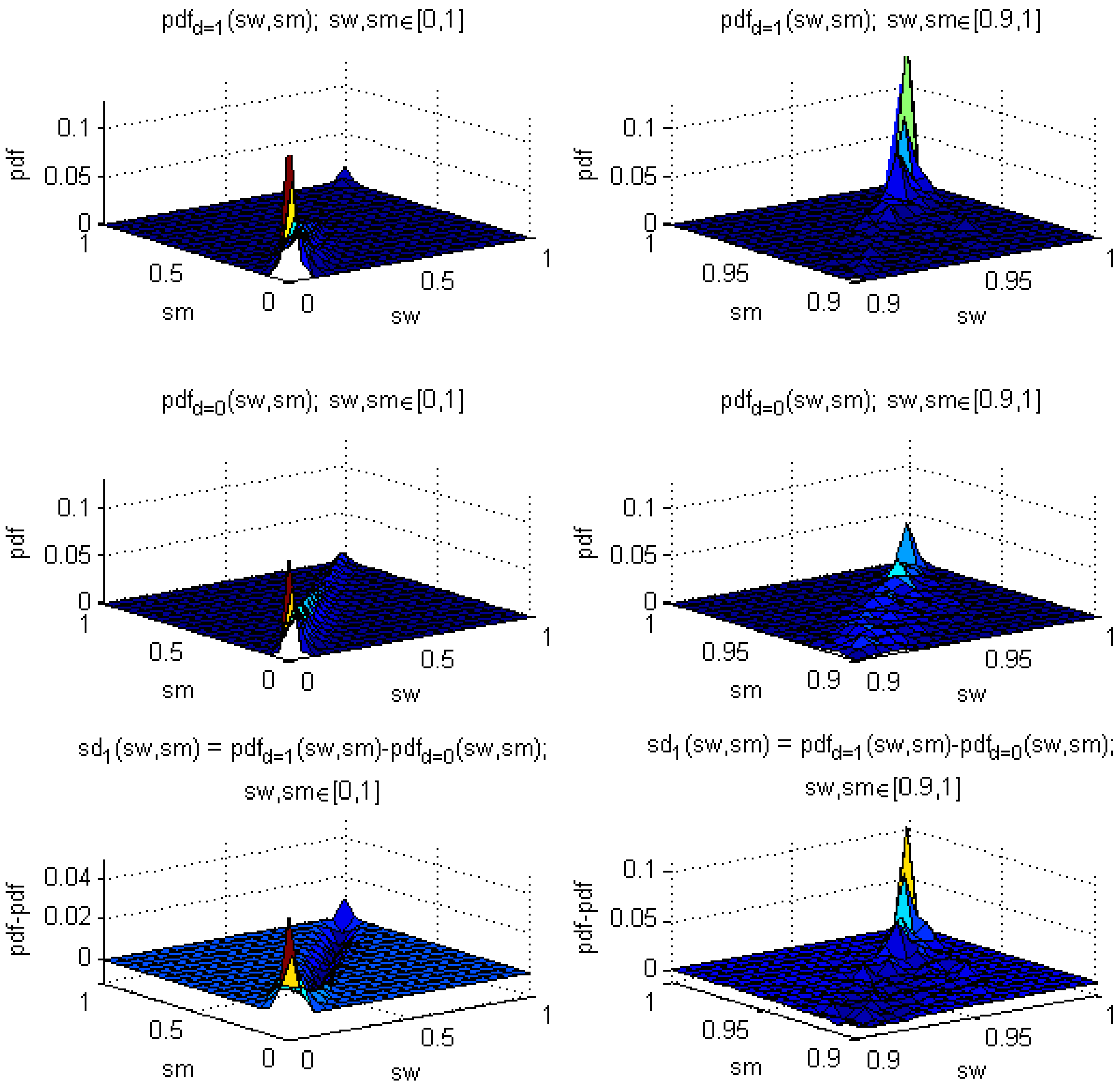
8.3 The first classifier: difference of probability density functions

In this section we will present our first idea of a scoring function which can be used to classify proteins based on their wild and mutated disorder scores.

At this stage we are already seeing quite detailed features of the data. However the total counts are not directly comparable as they have different scales. Therefore we proceed in scaling the counts with their total sum obtaining the empirical probability density functions.

Figure 8-4 displays six different probability density functions for the case with amino acid window equal to one. We start the discussion of the figure with the left half which corresponds to the data sample with $\mathrm{sw} \in [0,1]$ and $\mathrm{sm} \in [0,1]$. The first surface plot corresponds to the probability density function for the disease instances and it shows the usual peaks at the extremes of the sw, sm space with high probability along the first diagonal. The second surface plot corresponds to the polymorphisms and the third plot is equal to their difference. The shape of the difference paints a clear picture: along the diagonal there is a higher probability of the mutation being a polymorphism ($d = 0$), at the extremes where $(sw, sm) \approx (0,0)$ and $(sw, sm) \approx (1,1)$ there is a very high chance of a disease causing mutation ($d = 1$) and in the rest of the (sw, sm) space we have no data so we are uncertain. In order to have a clearer picture of the local variation of the data we also analysed the points around $(sw, sm) \approx (1,1)$ by plotting the same probability density functions in the right half of the figure for the data sample with $\mathrm{sw} \in [0.9,1]$ and $\mathrm{sm} \in [0.9,1]$. There are important findings from the three surface plots from the right side of the figure: the probability density functions are not smooth in the region and the more we zoom in on the point $(sw, sm) \approx (1,1)$ the more we see that there is an extremely high probability of having a disease causing mutation ($d = 1$) in a very small region around that point. This already gives us direction for the next steps of our analysis. First, we should use very fine gridding of the data as there is a lot of information in small regions. Second, when we build a classifier we need a function with many degrees of freedom as opposed to a very simple function which could not capture the detailed variation in the data.

At this step of the analysis we observed that the difference between the empirical distribution functions that we presented can by itself already act as a classifier. The classifier scoring function would be $sd_1(sw, sm) = \widehat{pdf}_{d=1}(sw, sm) - \widehat{pdf}_{d=0}(sw, sm)$ where the hat symbolizes the estimate of the probability density function. The threshold of the classifier would be zero therefore the classifier will predict disease ($d = 1$) if $sd(sw, sm) \geq 0$ and it will predict polymorphism if ($d = 0$) if $sd(sw, sm) < 0$.



$pdf_{d=1}(sw,sm)$; $sw,sm \in [0,1]$

$pdf_{d=1}(sw,sm)$; $sw,sm \in [0.9,1]$

$pdf_{d=0}(sw,sm)$; $sw,sm \in [0,1]$

$pdf_{d=0}(sw,sm)$; $sw,sm \in [0.9,1]$

$sd_1(sw,sm) = pdf_{d=1}(sw,sm)\text{-}pdf_{d=0}(sw,sm)$; $sw,sm \in [0,1]$

$sd_1(sw,sm) = pdf_{d=1}(sw,sm)\text{-}pdf_{d=0}(sw,sm)$; $sw,sm \in [0.9,1]$

Figure 8-4 Empirical probability density functions for the case where the amino acid window is equal to one. The x axis corresponds to the wild disorder score sw*, the y axis corresponds to the mutated disorder score* sm *and the z axis corresponds to the empirical probability density function* $pdf(sw,sm)$*. The left half of the figure corresponds to the data sample with* $sw \in [0,1]$, $sm \in [0,1]$*, and the right half of the figure corresponds to the data sample with* $sw \in [0.9,1]$, $sm \in [0.9,1]$*. The first row in the figure corresponds to the data sample with proteins classified as diseased* ($d = 1$)*, the second row corresponds to the data sample with proteins classified as polymorphisms* ($d = 0$)*, and the third row corresponds to the difference between the two probability density functions* $pdf_{d=1}(sw,sm) - pdf_{d=0}(sw,sm)$*. The empirical probability density functions are obtained by dividing the instances into a grid of size* $20 \cdot 20$ ($sw = 0, 0.05, 0.1, \dots 1$, $sm = 0, 0.05, 0.1, \dots 1$)*, counting the instances inside each grid element and then dividing all the counts to the sum of all counts obtaining probabilities.*

8.3.1 Interpolation and smoothing

In this subsection we will introduce the techniques that we used to estimate a continuous scoring function which can be used for classifying new data points.

We now have a first idea of a classifier inspired directly from a detailed analysis of the data. In order to classify new points we need to use a variation on interpolation. We decided to use linear interpolation as a first trial version. Figure 8-5 displays the $\mathrm{sd}_1^{\mathrm{i}}(\mathrm{sw}, \mathrm{sm})$ scoring function which was estimated on a grid of $100 \cdot 100$ and then linearly interpolated. You can easily observe the high confidence close to the extreme values of (sw, sm) that the instances will be classified as disease ($\mathrm{sd}_1^{\mathrm{i}}(\approx 0, \approx 0) > 0$ and $\mathrm{sd}_1^{\mathrm{i}}(\approx 1, \approx 1) > 0$).

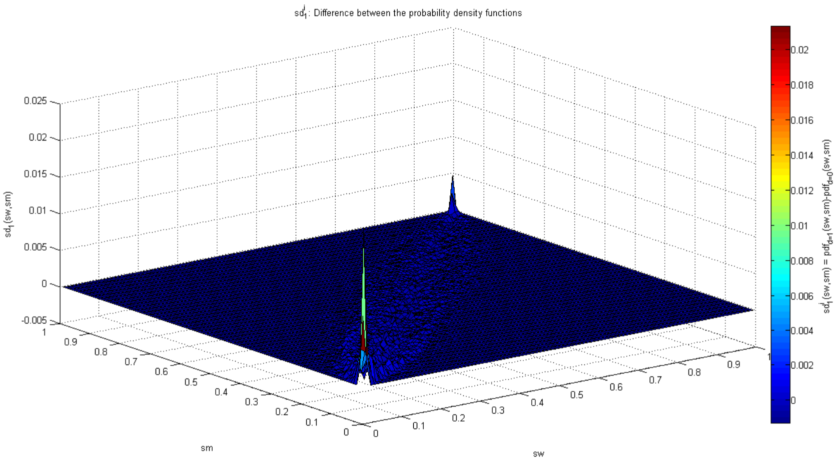


Figure 8-5 The first scoring function for predicting disease. We estimated a continuous scoring function $sd_1^i(sw, sm)$ by simple linear interpolation such that we could classify points which were not exactly on the grid.

We could use only the interpolated version of our scoring function, which would probably give good forecasting results, but that is not a good idea if we want to extract the main features of the data in order to draw more general conclusions.

At the same time a smoother scoring function would give better results when testing its forecasting accuracy on new data. During the process of working for this thesis we experimented with multiple methods for generating a smoother scoring function. We started with nonparametric models like scatter plot smoothing techniques (locally weighted scatter plot smoothing or LOWESS), but the disadvantage was that it did not behave well along the edges of the areas where we had no observations. Another method that we tried was Multivariate adaptive regression splines, but again this type of dataset was not regular enough for the method to work well. Other methods that we tried were linear regression and logistic regression but the problem with these types of models is that they have too few parameters to be able to capture the variations of our dataset.

Despite being unsuccessful with those methods we were however successful in fitting splines. This method has many features which were very advantageous to our project: they are flexible enough to include detailed variation in the dataset but at the same time their smoothness can be controlled according to our preference for generality over specificity of our model. We experimented with a few spline fitting approaches, including the ones already included in MATLAB like spaps.m which is based on (Reinsch n.d.).

Nevertheless, the best method for our purposes was based on (Garcia 2010). We also used the implementation of the spline fitting algorithm from the same paper. The main features of the algorithm that were very useful to us were the fact that it could iterate until it found the best smoothing parameter by minimizing the generalized cross-validation score, and that we could give higher weights to points that had higher density. In order to fit the spline surfaces we applied the algorithm to the uniformly sampled data points obtained by calculating the frequencies in each element of a $100x100$ grid of the (sw, sm) space.

We used the spline fitting with two different types of parameterisations. First, we fitted a spline surface with a smoothing parameter s which minimized the generalized cross validation criterion: $s = argmin_s\left(GCVscore(s)\right)$. Second, we fitted a spline surface by giving a higher weight to points in the grid which were calculated using a larger sample (the weight was equal to the sum of the empirical probability density functions for the disease and polymorphisms: $w(sw, sm) = pdf_{d=1}(sw, sm) + pdf_{d=0}(sw, sm)$).

The first case, without the weights is displayed in Figure 8-6. It is evident that the surface is much smoother than in Figure 8-5 and better summarizes the main features of the data.

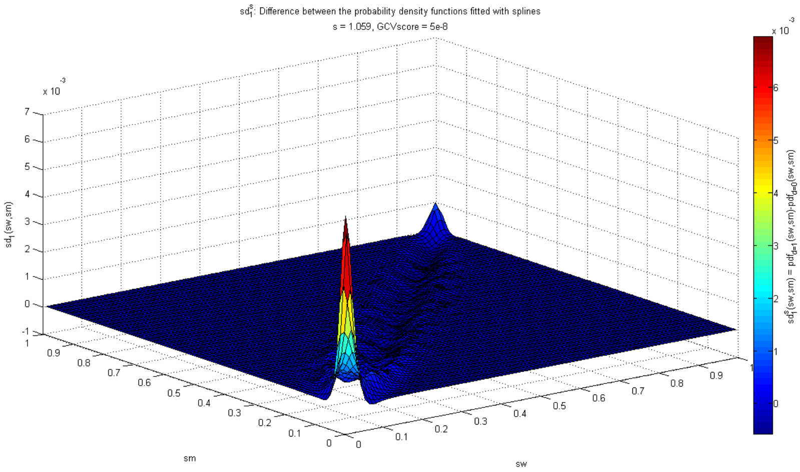


Figure 8-6 The first scoring function for predicting disease. We estimated a continuous scoring function $sd_1^s(sw, sm)$ by fitting a spline surface with a smoothing parameter $s = 1.059$ which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. The minimized GCVscore was equal to 5e-8.

The second case, with the weighting is displayed in Figure 8-7. This surface is the smoothest out of the three, having fewer degrees of freedom. It also has a higher prediction accuracy for new data points as measured by the smaller generalized cross validation score than in Figure 8-6 (GCVscore = 2e-8 versus 5e-8) which is perfectly in line with Occam’s Razor principle. The main conclusion that we can draw from the figure is that disease is predicted in the corners: $sd_1^{s,w}(\approx 0, \approx 0) > 0$ and $sd_1^{s,w}(\approx 1, \approx 1) > 0$.

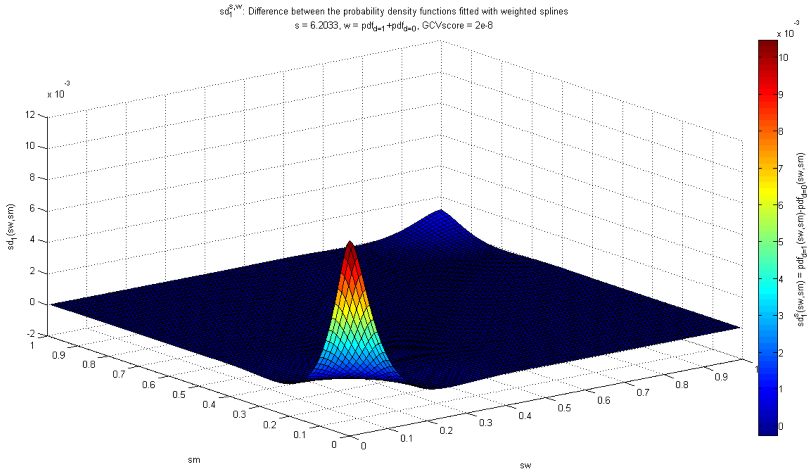


Figure 8-7 The first scoring function for predicting disease. We estimated a continuous scoring function $sd_1^{s,w}(sw, sm)$ by fitting a spline surface with a smoothing parameter $s = 6.2$ which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. The minimized GCVscore was equal to 2e-8. The smoothing was carried out in this case by giving a higher weight to points in the grid which were calculated using a larger sample (the weight was equal to the sum of the empirical probability density functions for the disease and polymorphisms: $w(sw, sm) = pdf_{d=1}(sw, sm) + pdf_{d=0}(sw, sm)$). The result is a much smoother surface, with fewer degrees of freedom, which has a higher prediction accuracy for new data points as measured by the smaller GCVscore than in Figure 8-6 which is perfectly in line with Occam's razor principle.

8.4 The second classifier: average of class labels

In this section we will explain how we arrived at our second classifier $sd_2(sw, sm)$. We will again use interpolation ($sd_2^i(\mathrm{sw}, \mathrm{sm})$) and spline surface fitting ($sd_2^s(\mathrm{sw}, \mathrm{sm})$ and $sd_2^{s,w}(\mathrm{sw}, \mathrm{sm})$) in order to be able to classify new data which is not directly overlapping the grid points.

We start by viewing the data in a different way. In the previous section the value of the disease score $sd_1(sw, sm)$ depended on the distribution of the whole data set as we were calculating the difference between two probability density functions of the data. Now we will look at the average of the class labels $d(sw, sm)$ inside each grid element. This value will be always between zero and one and the uncertainty threshold will be equal to 0.5. Therefore our second disease score will classify an instance as being disease if $sd_2(sw, sm) > 0.5$ and not disease (polymorphism) if $sd_2(sw, sm) < 0.5$.

Figure 8-8 displays three heat maps. The one on top plot the average of the disease label $d(sw, sm)$ for each grid element, versus the wild disease score sw and the mutated disease score sm. For our purposes we had to fill in the grid elements where we had no counts therefore we decided to set the values to 0.5 which is equal to the uncertainty threshold. This detailed heat map is color coded such that the grid elements which contain only diseased instances appear as red and the ones which contain only no disease (only polymorphism) instances appear as blue. This type of scoring provides more variation as due to the independent treatment of each grid element region we can have cases where a blue square appears right next to a blue square. The main features evident from this heat map are that proportionally there are more disease instances close to the origin ($sd_2(< 0.5, < 0.5) > 0.5$) and more polymorphisms close to the disordered area ($sd_2(> 0.5, > 0.5) < 0.5$). At the same time the diseased proteins are most present where there is a strong change in the disorder after the mutation: maximum $abs(sw - sm)$. We will use these values as a basis for our second disease scoring function. We now go on to analyze the other two plots. The heat map in the bottom left plots a measure of the counts of the disease instances inside each grid element. For display purposes, the actual color code is calculated as $\log_{10}(n_{d=1} + 1)$, where $n_{d=1}$ is the number of disease instances inside that grid element. We used the log function due to the fact that there is exponential variation between the counts and we added the one so we do not end up with $\log_{10}(0) = -\infty$. As a consequence the grid points with counts equal to zero will have the value 0 displayed on the color map legend and will have the color blue. The conclusions from the bottom left plot is that there are an extremely large number of disease proteins at $(sw, sm) \approx (0,0)$ and $(sw, sm) \approx (1,1)$ and a large number of disease proteins in the region $(sw < 0.5, sm < 0.5)$ while there are large regions (colored in blue) which contain no data. The subplot on the bottom right is built exactly in the same way except for the fact that it displays polymorphism (no disease) counts. The conclusions are that the polymorphisms are distributed in a similar way but are more evenly spread out. We will use these counts presented in the bottom two graphs for weighting the spline fit to the disease label average matrix (for obtaining the estimate of $sd_2^{s,w}(\mathrm{sw}, \mathrm{sm})$).

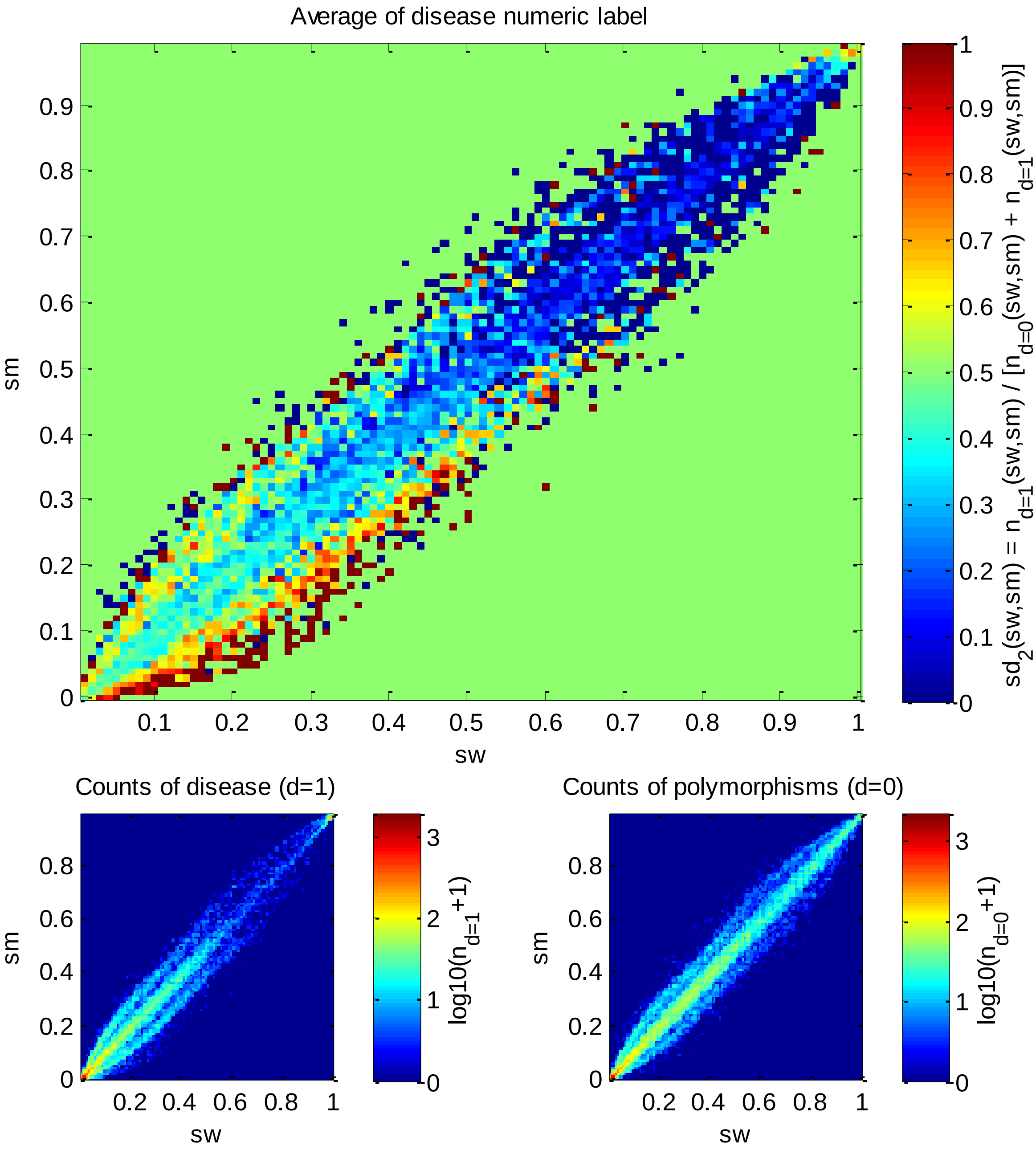


Figure 8-8 ***Top****: heat map of the average of the disease label d(sw,sm) for each grid element, versus the wild disease score sw and the mutated disease score sm. Where we had no data set the values to 0.5 which is equal to the uncertainty threshold. This detailed heat map is color coded such that the grid elements which contain only diseased instances appear as red and the ones which contain only no disease (only polymorphism) instances appear as blue.* ***Bottom left****: heat map of a measure of the counts of the disease instances inside each grid element. For display purposes, the actual color code is calculated as $log_{10}(n_{d=1}+1)$, where $n_{d=1}$ is the number of disease instances inside that grid element. The grid points with counts equal to zero will have the value 0 displayed on the color map legend and will have the color blue.* ***Bottom right****: heat map built exactly in the same way as the one in the bottom left except for the fact that it displays polymorphism (no disease) counts.*

8.4.1 Interpolation and smoothing

In this section exactly as in section 8.4.1 we will use interpolation (to obtain $sd_2^i(\mathrm{sw},\mathrm{sm})$) and spline surface fitting (to obtain $sd_2^s(\mathrm{sw},\mathrm{sm})$ and $sd_2^{s,w}(\mathrm{sw},\mathrm{sm})$) in order to be able to classify new data which is not directly overlapping with the grid points.

Figure 8-9 displays the $sd_2^i(sw,sm)$ scoring function which was estimated on a grid of $100x100$ points and then linearly interpolated. Here we can observe that although on average we can see similarities with the first scoring function, the surface is extremely noisy because we make the calculations inside each grid element independently from the other regions of the (sw,sm) space. This surface is very likely to be over fitted which can lead to very poor out of sample performance of the classifier. We proceed to fit a smoother surface with fewer degrees of freedom.

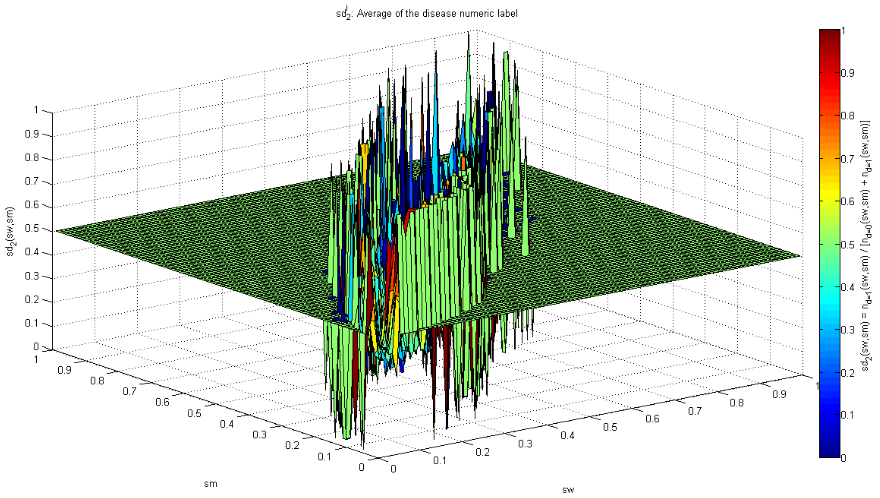


Figure 8-9 The second scoring function for predicting disease. We estimated a continuous scoring function $sd_2^i(sw,sm)$ by simple linear interpolation such that we could classify points which were not exactly on the grid.

We again used the spline fitting algorithm published by (Garcia 2010) with two different types of parameterisations. First we fitted a spline surface with a smoothing parameter s which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. Second we also included a weight for each point (equal to the sum of the grid counts for the disease and polymorphisms: $w(sw,sm) = n_{d=1}(sw,sm) + n_{d=0}(sw,sm)$).

The first case, without the weights is displayed in Figure 8-10. It is evident that the surface is much smoother than in Figure 8-9 and better summarizes the main features of the data. We see that this function will classify instances as being disease mainly if they are on the edges of the area where our data is situated and it will classify as polymorphism (not disease) the data is situated towards the centre, close to the main diagonal of the (sw,sm) space. We see very high confidence in disease in the area of small sw and very small sm. This can be interpreted as: disease is found where the change in disorder scores after a mutation is highest, or $(sw,sm) = \mathrm{argmax}_{sw,sm}(abs(sw - sm))$.

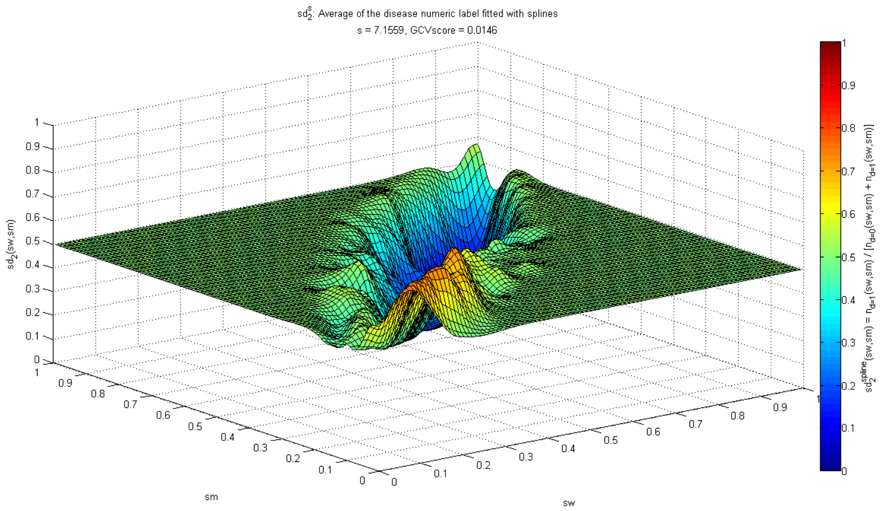


Figure 8-10 The second scoring function for predicting disease. We estimated a continuous scoring function $sd_2^s(sw, sm)$ by fitting a spline surface with a smoothing parameter $s = 7.15$ which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. The minimized GCVscore was equal to 0.014.

Our last formulation, with the weighting is displayed in Figure 8-11. This surface is the smoothest out of the last three, having fewer degrees of freedom. It also has a higher prediction accuracy for new data points as measured by the smaller generalized cross validation score than the case presented in Figure 8-10 (GCVscore = 3e-5 versus 0.014) which is perfectly in line with Occam's Razor principle. The main conclusion that we can draw from the figure is that disease is predicted towards the (0,0) corner: $sd_2^{s,w}(\mathrm{sw} \approx 0, \mathrm{sm} \approx 0) > 0$ and polymorphisms are predicted close to the (1,1) corner: $sd_2^{s,w}(\mathrm{sw} \approx 1, \mathrm{sm} \approx 1) < 0$.

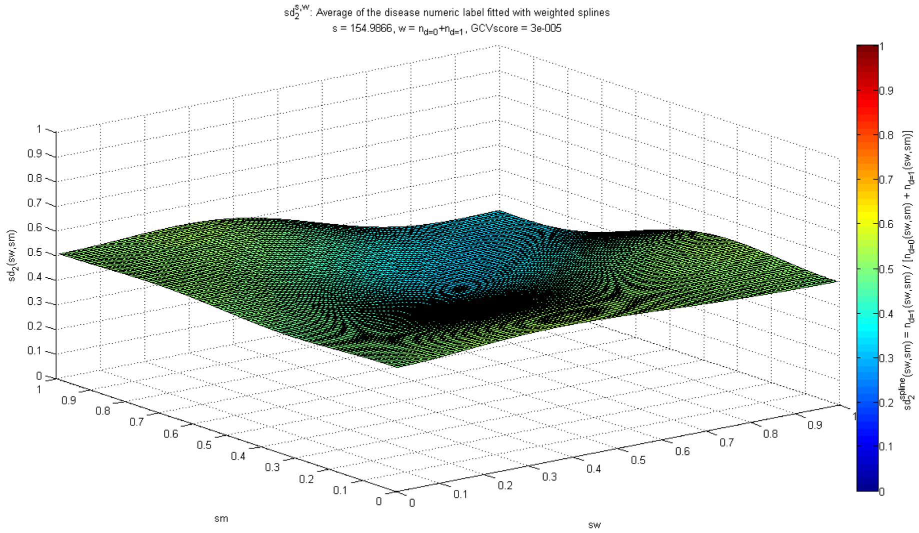


Figure 8-11 The second scoring function for predicting disease. We estimated a continuous scoring function $sd_2^{s,w}(sw, sm)$ by fitting a spline surface with a smoothing parameter $s = 154$ which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. The minimized GCVscore was equal to 3e-5. The smoothing was carried out in this case by giving a higher weight to points in the grid which were calculated using a larger sample (the weight was equal to the sum of the grid counts for the disease and polymorphisms: $w(sw, sm) = n_{d=1}(sw, sm) + n_{d=0}(sw, sm)$). The result is a much smoother surface, with fewer degrees of freedom, which has a higher prediction accuracy for new data points as measured by the smaller GCVscore than in Figure 8-10 which is perfectly in line with Occam's razor principle.

8.5 Conclusion

Via this thorough statistical analysis we have managed to answer to our first research questions which concentrated on identifying the level of disorder within our dataset; the disorder-to-order and order-to-disorder transitions and whether missense mutations are more prone to fall within disordered regions or not. The tables in section 8.1 give an exact overview of the level of disorder within the dataset based on an amino acid window of 1, 11 and 21 residues. The scatter plots in section 8.2 graphically show that disorder to order transitions are more frequent and that the dataset is symmetrical in the levels of deleterious mutations (we can see throughout all the graphical analysis that there are two major clusters of disease mutations, one within structured proteins and one within disordered ones). This concludes that disorder is conserved within the dataset, meaning that mutations that happen to fall within disordered regions can sometimes be conserved across species.

9 Results

Even though our main purpose was not to design and implement the best predictor for disease, we do want to establish whether there is a relationship between wild disorder scores, mutated disorder scores and the occurrence of disease. To achieve our main goal we tested several scoring functions that have as factors the wild and mutated disorder scores. To be able to make a high quality inference based on the data we would prefer a smooth model with as few parameters as possible which also matches the data well. Therefore our focus in this chapter will be to select out of the best performing predictors the one that would allow us to make the most intuitive inference on the relationship between the variables. The smoothed versions of classifiers one and two are already good candidates that we had in mind and presented earlier. However, we will give all the methods an equal chance and we will build a ROC curve for each of them. The main performance indicator that we will use to compare all the models is the area under the ROC curve.

We will now give a concise description of a selection of all the models which we tried that have good enough performance to be presented in this thesis. We will use the same order in our presentation as in Table 9-1:

1. $\widehat{\mathrm{sd}}_1^{\mathrm{i}}(\mathrm{sw}, \mathrm{sm})$
2. $\widehat{\mathrm{sd}}_1^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$
3. $\widehat{\mathrm{sd}}_1^{\mathrm{s,w}}(\mathrm{sw}, \mathrm{sm})$
4. $\widehat{\mathrm{sd}}_2^{\mathrm{i}}(\mathrm{sw}, \mathrm{sm})$
5. $\widehat{\mathrm{sd}}_2^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$
6. $\widehat{\mathrm{sd}}_2^{\mathrm{s,w}}(\mathrm{sw}, \mathrm{sm})$
7. $\widehat{sd}_3(sw, sm) = sw - sm$, where $sw \geq 0.5$
8. $\widehat{sd}_4(sw, sm) = \mathrm{sm} - \mathrm{sw}$, where $sm \geq 0.5$
9. $\widehat{sd}_5(sw, sm) = -\mathrm{abs}(sw - sm)$, where $sw \geq 0.5$
10. $\widehat{sd}_6(sw, sm) = -\mathrm{abs}(sw - sm)$, where $sm \geq 0.5$
11. $\widehat{sd}_7(sw, sm) = \mathrm{sm}$, where $sm \geq 0.5$
12. $\widehat{sd}_8(sw, sm) = \mathrm{sw}$, where $sw \geq 0.5$

Area under the ROC curve for each disease score function						
Disease score function		Data subsample	Estimation	Amino acid window		
				1	11	21
1	$\widehat{\mathrm{sd}}_1^{\mathrm{i}}(\mathrm{sw}, \mathrm{sm})$	all data	interpolated	0.653	0.656	0.658
	$\widehat{\mathrm{sd}}_1^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$	all data	spline	0.645	0.652	**0.655**
	$\widehat{\mathrm{sd}}_1^{\mathrm{s,w}}(\mathrm{sw}, \mathrm{sm})$	all data	spline, weighted	0.628	0.633	0.635
2	$\widehat{\mathrm{sd}}_2^{\mathrm{i}}(\mathrm{sw}, \mathrm{sm})$	all data	interpolated	0.656	0.658	0.659
	$\widehat{\mathrm{sd}}_2^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$	all data	spline	0.654	0.656	**0.659**
	$\widehat{\mathrm{sd}}_2^{\mathrm{s,w}}(\mathrm{sw}, \mathrm{sm})$	all data	spline, weighted	0.633	0.640	0.643

3	$\widehat{sd}_3(sw, sm) = sw - sm$	$sw \geq 0.5$	direct calculation	0.513	0.515	0.515
4	$\widehat{sd}_4(sw, sm) = \mathrm{sm} - sw$	$sm \geq 0.5$	direct calculation	0.539	0.536	0.527
5	$\widehat{sd}_5(sw, sm) = -\mathrm{abs}(sw - sm)$	$sw \geq 0.5$	direct calculation	0.527	0.544	0.549
6	$\widehat{sd}_6(sw, sm) = -\mathrm{abs}(sw - sm)$	$sm \geq 0.5$	direct calculation	0.522	0.529	0.539
7	$\widehat{sd}_7(sw, sm) = sm$	$sm \geq 0.5$	direct calculation	0.530	0.536	0.545
8	$\widehat{sd}_8(sw, sm) = sw$	$sw \geq 0.5$	direct calculation	0.531	0.561	0.563

Table 9-1 Area under the ROC curve for each disease score

The conclusion from Table 9-1 is that we can already cross out the scoring functions three to eight. These scoring functions have too few degrees of freedom to capture the complex interactions between the variables; this would explain the very low AUC. We therefore concentrate only on the first and second classifiers.

Based on the above decision, we further show the ROC curves only for classifiers one and two. Figure 9-1 displays the ROC curves for the amino acid window size equal to one. We first notice that the curve is convex so we do not need to operate any classifier on the convex hull. We also notice that there is no intersection with the random line so we do not need to build a hybrid classifier (like we would have had to do had we used the scoring functions from 3 to 8). However there is potential for combining the classifiers for achieving maximum performance by using Scott's maximum realisable ROC but that is outside the scope of this thesis. It is evident from the plot that the six versions of the first two classifiers have a very similar performance.

We do observe that our classifier of choice for this thesis $\widehat{\mathrm{sd}}_2^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$ performed reasonably well compared to the other classifiers while providing valuable insight into the complex interactions between the variables.

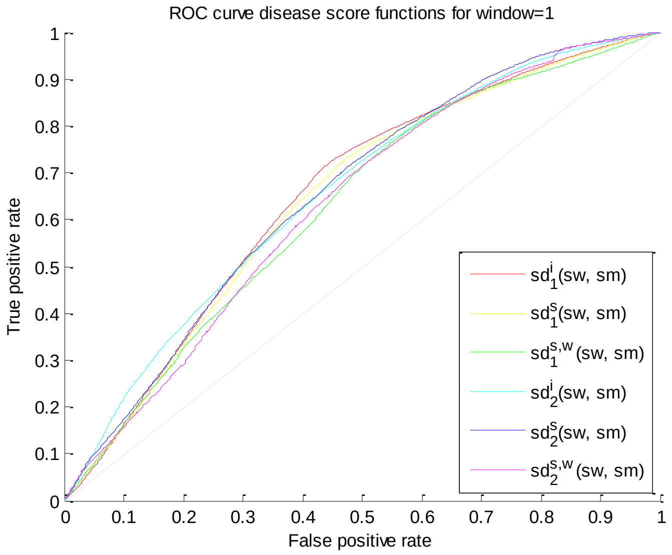


Figure 9-1 ROC curve for the disease score functions one and two for the amino acid window size equal to one

Now if we look at the results for the amino acid windows of 11 and 21 from Figure 9-2 and Figure 9-3 we can reach similar conclusions. This is to be expected as the disorder scores in all cases are quite similar.

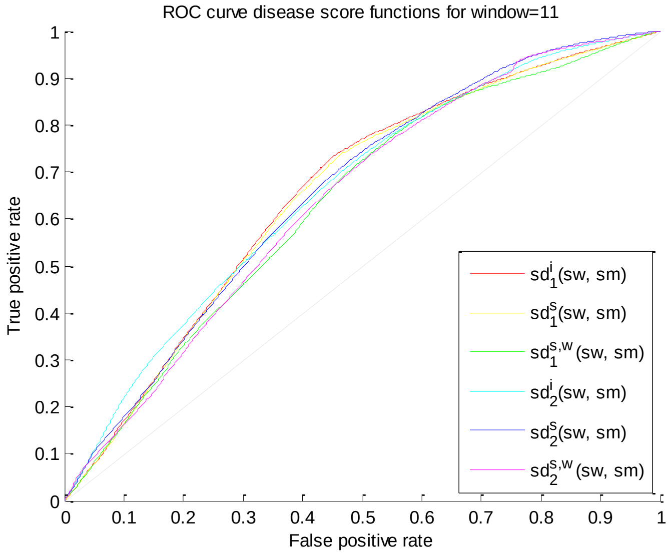


Figure 9-2 ROC curve for the disease score functions one and two for the amino acid window size equal to eleven

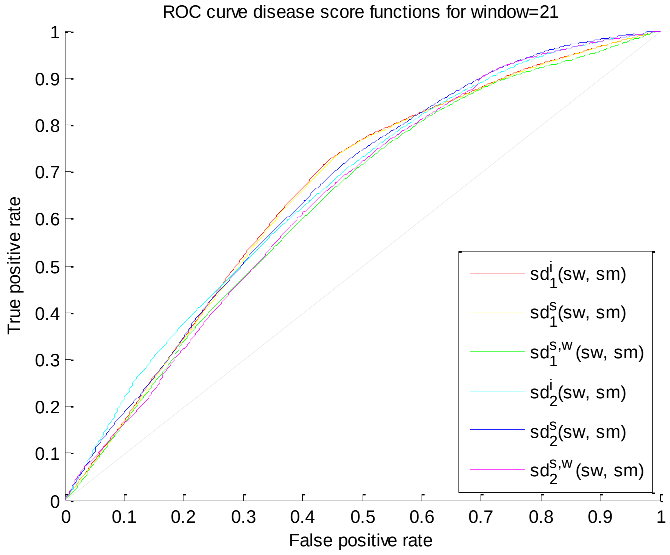


Figure 9-3 ROC curve for the disease score functions one and two for the amino acid window size equal to twenty one

We conclude that our classifier of choice for this thesis will remain $\widehat{sd_2^s}(sw, sm)$ for a window of 21 amino acids because it performed reasonably well compared to the other classifiers in terms of area under the ROC curve and generalized cross-validation score while providing valuable insight into the complex interactions between the variables.

We again present in Figure 9-4 the second classifier fitted with splines without weights, $\widehat{\mathrm{sd}}_2^{\mathrm{s}}(\mathrm{sw}, \mathrm{sm})$ viewed from the top in order to better analyse its results.

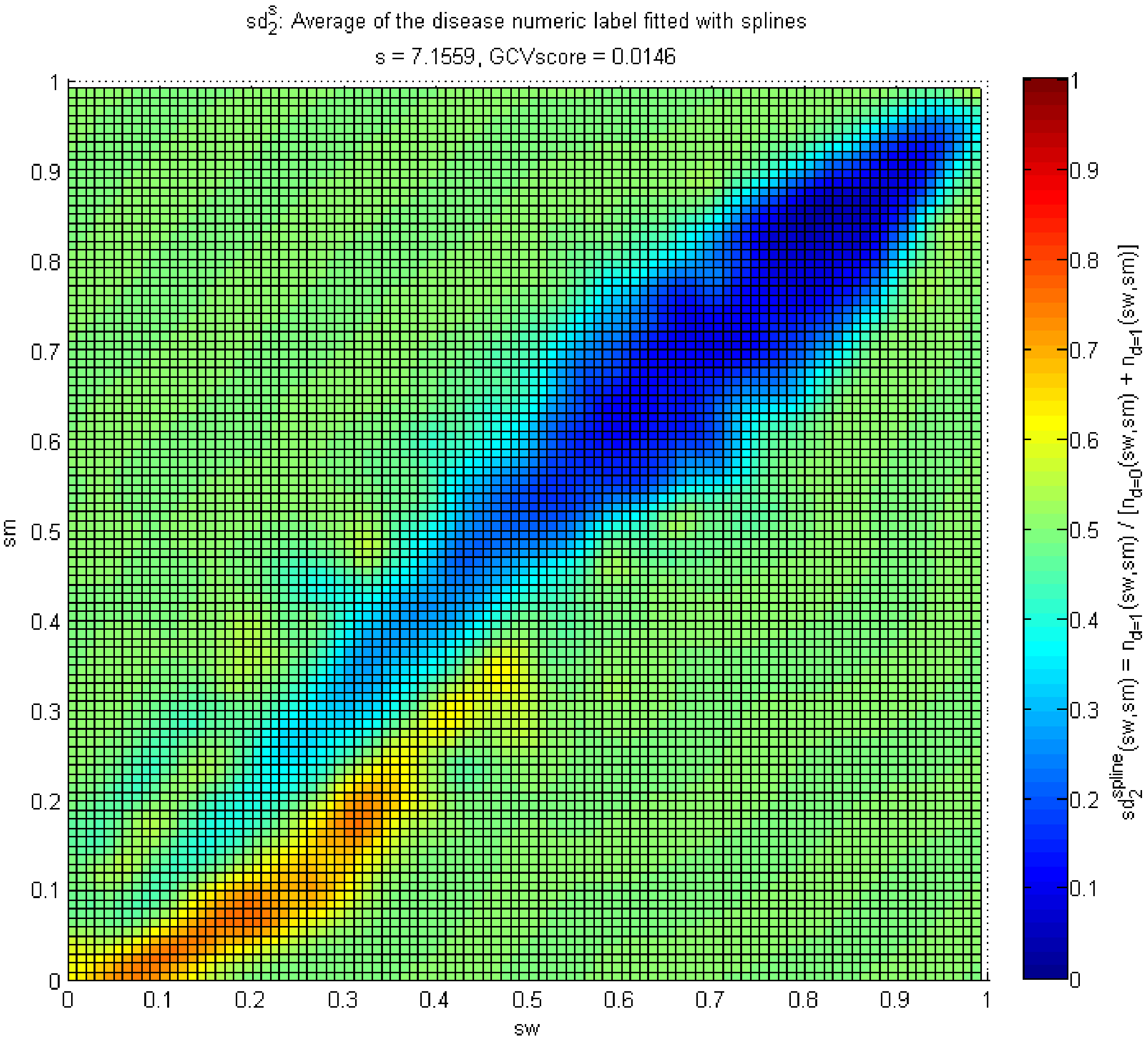


Figure 9-4 The second scoring function for predicting disease. We estimated a continuous scoring function $sd_2^s(sw, sm)$ by fitting a spline surface with a smoothing parameter $s = 7.15$ which minimized the generalized cross validation criterion: $s = argmin_s(GCVscore(s))$. The minimized GCVscore was equal to 0.014.

In conclusion we would like to point out that the two classifiers that we gave implemented have yet to be tested on a dataset of point mutations with the scope of measuring the degree of pathogenicity within disordered regions. The results are consistent with our hypothesis and as can be seen from figure 9.4, a smoothed scoring function performs better than a raw one simply because it allows for more degrees of freedom for capturing the data.

10 Critical evaluation

From an overall perspective, this project had to first investigate the link between disease and mutations within intrinsic disorder and second to develop a straightforward classifier based on one feature that measured the degree of disorder change brought by these point mutations. We strongly believe that these aims were achieved via 4 major objectives as follows.

First of all, this project needed extensive understanding of the bioinformatics field for its success. The first objective was to get acquainted with the concept of intrinsically disordered proteins and assess the bioinformatics tools available for detecting these sorts of proteins in nature. We believe this first objective has been successfully achieved offering a balanced, self standing review of the extant literature on intrinsic disorder and disease.

The second objective was to conduct a statistical analysis on the dataset chosen. The available literature that dabbed into the problem we were trying to assess did not offer a proper comparison of the tools utilized. The biggest challenge was to determine how to effectively assess the relationship between disorder and disease and test its significance. A large proportion of the time was spent understanding and plotting the data. After a few failed attempts based on the direct correlation analysis of the wild-type and mutated type disorder scores we decided that a more in-depth approach is required; the results we first obtained indicated that the more disordered the protein the less susceptible to disease it was; this outcome is blatantly against what academia have presented. Thus, we were adamant that there is causality between mutations within disorder and human disease. We shifted our classical statistical approach from the traditional methods (linear regression and Pearson Correlation matrix which would not work given the nature of the data available) of assessing the dependencies between our variables and shifted our attention towards more complex measurements of variable dependencies. The first classifier we adopted was concentrated on probability density functions and the second used the averages of the disease class labels.

Our third objective was to fit a classifier to our dataset. The focus was not necessarily on prediction performance but on being able to identify the nature of the relationship between disorder scores of the wind and mutated variants and the disease classification. We achieved this objective by using the two classifiers mentioned above. In order to estimate the disease scoring function we first used the classifiers in their original form by linear interpolation. The second approach was to fit splines and the third approach was to fit weighted splines. The results were satisfactory as the splines were able to capture the complex interactions between the variables. The classifier based on averages of disease labels and unweighted splines was our final choice as a good prediction performer and also giving us a good interpretation of the relationship between the data. We believe we have succeeded in achieving this objective also.

Given the limited timeframe available for answering our research question, a few interesting approaches could not be trialled and implemented. This was due to not the most efficient time management, as more time than first envisaged was spent on analysing the data. However, we have compiled a set of further research options to improve our project.

Overall we have answered our research question; we have applied statistical analysis tools that have yet to be tested on disordered proteins although they are widely used within the bioinformatics field. The classifiers designed have some predictive power reinforcing our hypothesis.

11 Further research

Being such a new research field, there are plenty of other avenues that can be taken in order to improve the outcome of the research. Each of the following approaches includes a higher degree of biological information needed to infer strong enough findings. In this section we will attempt to explain further ideas worth exploring.

1. ***Different disorder predictors.*** Given that the plethora of available disorder predictors uses different features depending on the research group that developed them, it would be an interesting side–project to test a series of known well performing disorder predictors on the statistical framework we have developed. Based on a ranking system we could determine which disorder predictors work best with the task of identifying disease causing missense mutations. In this paper we have selected the top three predictors that output an individual score for each residue. We chose the simplest and computationally fastest predictor to incorporate from an implementation view. However, testing more predictors built on different features of intrinsic disorder we can gain more comprehensive understanding of the sequence features or structure properties of such proteins.
2. ***Gene Ontology terms.*** Genes are units of heredity passed from a parent to an offspring; they are short segments of DNA which code for a certain type of protein. Gene ontology is a bioinformatics initiative to help unify gene related information (gene representation and gene product or proteins and RNA attributes) across all species. Just like for intrinsically disordered proteins, there is not a standardized nomenclature describing genes and gene product attributes. This project can be further enriched if given the list of gene ontology disease terms found in the SUPERFAMILY database and the list of disease causing mutations that were correctly identified by our classifier, map them together based on disease and function and compute the statistical significance (after normalization) for each of them. This, in essence would be a purely biological task that would identify which diseases are linked to which IDP function. This task will be further pursued with the aim of getting the paper published, however for our central task this has limited computer science power.
3. ***Extend the research question to include binding sites.*** Another research question that is adjacent to our project would be to verify the impact mutations within binding sites of disordered regions have based on the amino acid sequence. Binding sites function via a disorder to order transition caused by binding to a globular partner. This assumption is very probable given the nature of disordered proteins that are highly promiscuous and flexible. The authors of IUPred extended their research into the binding sites and created the ANCHOR predictor which aims at identifying portions within disordered regions that cannot form enough favourable intra-chain interactions, but do have the energetic capability of gaining such interactions when binding to globular protein partners (Dosztányi et al. 2009). Cancer is one of the diseases where protein-protein interactions and binding sites have often been linked. Corroborating this information with disease-causing mutations makes for an interesting new project. However, this is beyond the scope of our project since we are trying to prove a generalized causality between disease and mutations in disordered region. Neverteless, (Pajkos et al. 2012) have investigated the causality between protein disorder and the increased biological risk in terms of cancer-associated point mutations. Their findings include that genetic mutations have a functional impact on both structured and intrinsically disordered proteins, however in different ways. This conclusion can only reinforce ours.

4. ***Implement classifier within the FATHMM software.*** Functional Analysis Through Hidden Markov Models (FATHMM) is a University of Bristol in house mutation predictor constructed for the SUPERFAMILY pipeline. The procedure of predicting the functional consequences of point mutations concentrates on building a HMM model for the query sequence in order to identify its homologues. Then, the amino acid probabilities within the HMM are interrogated and if there is a reduction in the difference between the wild-type and the mutant residue then the assumption is that the point mutation can have a negative functional impact (Edwards et al. n.d.). A further enhancement of out project would be to incorporate the classifier developed here within the FATHMM framework and verify if the predicted results improve (even by a small percentage) that of FATHMM. This extra task is mostly an implementation task and the rate of success is not guaranteed. Moreover, we decided that the time spent on integrating the codes is better spent in answering our research questions to the fullest.

We have presented 4 major directions for further research that can improve the findings of this project.

12 Conclusion

Through this paper we have provided a framework for analysing an interesting and novel factor of intrinsically disordered proteins. We have started by offering an extensive background on the biological implications of these proteins and the mutations they include. We later tested the working hypothesis that mutations within disordered regions are deleterious on a non-redundant, highly annotated dataset. We adopted a novel approach to the statistical analysis of the data and designed a classifier that was in line with the research question. There is now scope for the scientific field to further research other avenues into identifying as soon as possible the deleterious mutations that cause serious human diseases.

13 Bibliography

13.1 Papers

Adzhubei, I. a et al., 2010. A method and server for predicting damaging missense mutations. *Nature methods*, 7(4), pp.248-9.

Adzhubei, I.A. et al., nature | methods A method and server for predicting damaging missense mutations Ivan A Adzhubei , Steffen Schmidt , Leonid Peshkin , Vasily E Ramensky , Anna Gerasimova , Peer Bork , Alexey S Kondrashov & Shamil R Sunyaev Supplementary figures and text : , 7(4).

Bagchi, A. et al., 2011. NIH Public Access. , 31(3), pp.335-346.

Biology, C. & Uversky, V.N., 2011. The International Journal of Biochemistry Intrinsically disordered proteins from A to Z. *International Journal of Biochemistry and Cell Biology*, 43(8), pp.1090-1103.

Boeckmann, B., 2003. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucleic Acids Research*, 31(1), pp.365-370.

Dosztányi, Z. et al., 2005. The pairwise energy content estimated from amino acid composition discriminates between folded and intrinsically unstructured proteins. *Journal of molecular biology*, 347(4), pp.827-39.

Dosztányi, Z., Mészáros, B. & Simon, I., 2010. Bioinformatical approaches to characterize intrinsically disordered/unstructured proteins. *Briefings in bioinformatics*, 11(2), pp.225-43.

Garcia, D., 2010. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational Statistics & Data Analysis*, 54(4), pp.1167-1178. Available at: http://linkinghub.elsevier.com/retrieve/pii/S0167947309003491 [Accessed July 13, 2012].

Hu, Y. et al., 2011. Changes in predicted protein disorder tendency may contribute to disease risk. *BMC genomics*, 12 Suppl 5(Suppl 5), p.S2.

Huang, T. et al., 2010. Prediction of deleterious non-synonymous SNPs based on protein interaction network and hybrid properties. *PloS one*, 5(7), p.e11900.

Huang, T. et al., 2012. SySAP: a system-level predictor of deleterious single amino acid polymorphisms. *Protein & cell*, 3(1), pp.38-43.

Jones, D.T. & Ward, J.J., 2003. Prediction of disordered regions in proteins from position specific score matrices. *Proteins*, 53 Suppl 6(February), pp.573-8.

Katoh, K. et al., 2002. MAFFT: a novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucleic acids research*, 30(14), pp.3059-66.

Mottaz, A. et al., 2010. Easy retrieval of single amino-acid polymorphisms and phenotype information using SwissVar. *Bioinformatics (Oxford, England)*, 26(6), pp.851-2.

Ng, P C & Henikoff, S, 2001. Predicting deleterious amino acid substitutions. *Genome research*, 11(5), pp.863-74.

Ng, P. C., 2003. SIFT: predicting amino acid changes that affect protein function. *Nucleic Acids Research*, 31(13), pp.3812-3814.

Ng, Pauline C & Henikoff, Steven, 2002. Accounting for human polymorphisms predicted to affect protein function. *Genome research*, 12(3), pp.436-46.

Orosz, F. & Ovádi, J., 2011. Proteins without 3D structure: definition, detection and beyond. *Bioinformatics (Oxford, England)*, 27(11), pp.1449-54.

Peng, K. et al., 2006. Length-dependent prediction of protein intrinsic disorder. *BMC bioinformatics*, 7, p.208.

Peng, K. et al., 2005. Optimizing long intrinsic disorder predictors with protein evolutionary information. *Journal of bioinformatics and computational biology*, 3(1), pp.35-60.

Ramensky, V., Bork, P. & Sunyaev, S., 2002. Human non-synonymous SNPs: server and survey. *Nucleic acids research*, 30(17), pp.3894-900.

Reinsch, C., Smoothing by Spline Functions. *Numerische Mathematik*, 83(x).

Serdyuk, I.N., 2007. Structured proteins and proteins with intrinsic disorder. *Molecular Biology*, 41(2), pp.262-277.

Shastry, B.S., 2003. SNP alleles in human disease and evolution. , (2002).

Sunyaev, S.R. et al., 1999. PSIC: profile extraction from sequence alignments with position-specific counts of independent observations. *Protein engineering*, 12(5), pp.387-94.

Thompson, J.D., Prigent, V. & Poch, O., 2004. LEON: multiple aLignment Evaluation Of Neighbours. *Nucleic acids research*, 32(4), pp.1298-307.

Uversky, V.N. et al., 2009. Unfoldomics of human diseases: linking protein intrinsic disorder with diseases. *BMC genomics*, 10 Suppl 1, p.S7.

Uversky, V.N., Oldfield, C.J. & Dunker, A.K., 2008. Intrinsically Disordered Proteins in Human Diseases : Introducing the D 2 Concept. *Proteins*, 2.

Vacic, V. & Iakoucheva, L.M., 2012. Disease mutations in disordered regions--exception to the rule? *Molecular bioSystems*, 8(1), pp.27-32.

Wang, Z. & Moult, J., 2001. SNPs, protein structure, and disease. *Human mutation*, 17(4), pp.263-70.

Wicker, N. et al., 2001. Secator: a program for inferring protein subfamilies from phylogenetic trees. *Molecular biology and evolution*, 18(8), pp.1435-41.

13.2 Code

- heatmap.m from Customizable Heat Maps by Ameya Deoras can be downloaded at: http://www.mathworks.com/matlabcentral/fileexchange/24253-customizable-heat-maps
- smoothn.m Copyright (c) 2010, Damien Garcia, can be downloaded at: http://www.mathworks.com/matlabcentral/fileexchange/25634-robust-spline-smoothing-for-1-d-to-n-d-data
- dctn.m and idctn.m: Narasimha M. et al, On the computation of the discrete cosine transform, IEEE Trans Comm, 26, 6, 1978, pp 934-936.

14 Annex

14.1 Data preprocessing with Perl

A. readme.txt

```
Corina Budeanu - MSc Thesis Advanced Computing (MS51)
cb1987@bristol.ac.uk

The folder structure is as follows:
data/
    mutated/
        in/
            A0AV02_IV281.fasta
            ...
        out/
            A0AV02_IV281.iupred
            ...
        accession.txt
        aminoacid_codes.txt
        humsavar.txt
        mutations.pl
    wild/
        in/
            A0AV02.fasta
            ...
        out/
            A0AV02.iupred
            ...
        accession.txt
    disorder.pl
    disorder_w1.txt
    disorder_w11.txt
    disorder_w21.txt
    uniprot_sprot.fasta
iupred:
    iupred executable
    iupred.c
    ...
filter_by_accession_multifiles.pl
```

B. filter_by_accession_multifiles.pl

```perl
#!/usr/bin/perl -w
use strict;
# perl filter_by_accession_multifiles.pl
my %seqids;
my $flag;
my $fasta = 'data/uniprot_sprot.fasta';
my $list = 'data/wild/accession.txt';
my $outputFolder = 'data/wild/in';
my $outputPath;
print "./fasta_filter_by_accession.pl $fasta $list $outputFolder\n";
open LIST,("$list");
    print "$list\n\n";
    while (<LIST>){
        if (/(\S+)/){
            $seqids{$1}=1;
        }
    }
close LIST;


open FASTA,("$fasta");
    while (<FASTA>){
        if (/^>[s][p][|](\S+)[|]/){
            if (exists($seqids{$1})){
                # print "$1\n";
                $flag=1;
            }else{
                $flag=0;
            }
        }
        if ($flag == 1){
            $outputPath = $outputFolder."/".$1.".fasta";
            open OUTPUT, ">>$outputPath" or die $!;
            # print "$1\n";
            print OUTPUT $_;
            close OUTPUT;
        }
    }
close FASTA;
```

C. disorder.pl

```
#!/usr/bin/perl -w
use strict;
use Scalar::Util qw(looks_like_number);
# perl disorder.pl
my $folderWild = 'wild/out';
my $folderMutated = 'mutated/out';
my $pathAccession = 'mutated/accession.txt';
# my $pathOut = 'disorder.txt';
my $pathOut = 'disorder_w11.txt';
my $window = 5;
my $pathWild;
my $pathMutated;
my $accession;
my $aaFrom;
my $aaTo;
my $aaAt;
my @lineAccession;
my %variantType;
my $disorderWild;
my $disorderMutated;

## Variant Type Codes

$variantType{Disease} = 1;
$variantType{Polymorphism} = 0;
$variantType{Unclassified} = -1;

## ACC
my $i=0;
open ACC,("< $pathAccession") or die $!;
    open OUT,("> $pathOut") or die $!;
        while (<ACC>){
            chomp;
            @lineAccession = split(/\t/, $_);
            ($accession, $aaFrom, $aaTo, $aaAt) = accAaFromAaToAaAt($lineAccession[0]);
            $pathWild = "$folderWild/$accession.iupred";
            $pathMutated = "$folderMutated/".$lineAccession[0].".iupred";
            #$disorderWild = readIupred($pathWild, $aaAt);
            #$disorderMutated = readIupred($pathMutated, $aaAt);
            $disorderWild = readIupredWindow($pathWild, $aaAt, $window);
            $disorderMutated = readIupredWindow($pathMutated, $aaAt, $window);
            print OUT
$lineAccession[0]."\t".$variantType{$lineAccession[1]}."\t".$disorderWild."\t".$disorderMutate
d."\n";
            $i++;
        }
    close OUT;
close ACC;

# SUBS --------------------------------------------------------------------

sub readIupredWindow {
    my $path = shift;
    my $aaAt = shift;
    my $window = shift;
    my $disorder;
    my $disorderAverage = 0;
    my $n = 0;
    my @line;
    open FILE, "< $path" or die $!;
    my @lines = <FILE>;
    for ($i=$aaAt+8-$window; $i<=$aaAt+8+$window; $i++) {
        if ($i<@lines && $i>8) {
            @line = $lines[$i]; #split(/\s/, $lines[$i]);
            #print @line; chomp @line;
            #print "\t\t";
            @line = split(/ +/, $lines[$i]); # \s+
            $disorder = $line[3];
            if(defined $disorder) {
                chomp $disorder;
                if (looks_like_number($disorder)) {
                    print $disorder;
                    print "\n";
                    $disorderAverage += $disorder;
                    $n++;
                }
            }
        }
    }
    close FILE;
    my $result;
    if ($n>0) {
        $result = $disorderAverage/$n;
    } else {
        $result = 'NA';
    }
    return $result;
}

sub readIupred {
    my $path = shift;
    my $aaAt = shift;
    open FILE, "< $path" or die $!;
        my @lines = <FILE>;
        my $disorder = $lines[$aaAt+8];
        chomp $disorder;
    close FILE;
    return $disorder;
}

sub accAaFromAaToAaAt {
    my $str = shift;
    my $accession = substr($str, 0, 6);
    my $aaFrom = substr($str, 7, 1);
    my $aaTo = substr($str, 8, 1);
    my $aaAt = substr($str, 9,length($str)-9);
    return ($accession, $aaFrom, $aaTo, $aaAt)
}
```

D. mutations.pl

```
#!/usr/bin/perl -w
use strict;
# perl mutations.pl

my $fnMutationList = 'humsavar.txt';
my $fnAaCodes = 'aminoacid_codes.txt';
my $fnOut = 'accession.txt';
my $folderWild = "../wild/in";
my $folderMutated = "in"; my $fnWild;
my $fnMutated; my $pathWild; my $pathMutated; my @lineCodes; my @lineMutationList;
my @linesWild; my @linesMutated;
my $sequence; my %aa; my @p; my $accession; my $aaChange;
my $variantType; my $aaAt; my $aaFrom; my $aaFrom2; my $aaTo; my $aaTo2;

## AA Codes
open CODES,("$fnAaCodes") or die $!;
  while (<CODES>){
    chomp;
    @lineCodes = split(/\t/, $_);
    $aa{$lineCodes[0]} = $lineCodes[1];
  }
close CODES;

## Mutation List
open MUTLIST,("< $fnMutationList") or die $!;
  open OUT,("> $fnOut") or die $!;
```

```
    while (<MUTLIST>){

      @lineMutationList = split(/\t/, $_);
      $accession = $lineMutationList[0];
      $aaChange = $lineMutationList[1];
      $variantType = $lineMutationList[2];
      chomp $variantType;
      print $accession."\t".$aaChange."\t".$variantType."\n";
      # Extract only rows containing mutation annotation starting with "p."
      if ($aaChange =~ /^p\./) {
        $aaFrom = $aa{substr($aaChange, 2, 3)};
        $aaTo = $aa{substr($aaChange, length($aaChange)-3, 3)};
        $aaAt = substr($aaChange, 5, length($aaChange)-8);
        if ($aaFrom && $aaTo) {
          print OUT $accession.'_'.$aaFrom.$aaTo.$aaAt."\t".$variantType."\t";
          $fnWild = $accession.'.fasta';
          $fnMutated = $accession.'_'.$aaFrom.$aaTo.$aaAt.'.fasta';
          $pathWild = $folderWild.'/'.$fnWild;
          $pathMutated = $folderMutated.'/'.$fnMutated;
          # Read the WILD and write in the MUTATED fasta file
          open WILD, "< $pathWild" or die $!;
            open MUTATED, ">> $pathMutated" or die $!;
              @linesWild = <WILD>;
              print MUTATED $linesWild[0];
              shift(@linesWild);
              chomp @linesWild;
              $sequence = join('', @linesWild);
              die "Error: seqLen=".length($sequence)."is too small for mutation
position=".$aaAt."\n" unless(length($sequence)>=$aaAt);
              print OUT "seqLen=".length($sequence)."\tactual:".substr($sequence,$aaAt-1,1)."
to ";
              $aaFrom2 = substr($sequence,$aaAt-1,1);
              die "Error: in the wild file aaFrom2=".$aaFrom2."is different from
aaFrom=".$aaFrom."\n" unless($aaFrom2 eq $aaFrom);
              substr($sequence,$aaAt-1,1) = $aaTo;
              print MUTATED $sequence;
            close MUTATED;
          close WILD;
          # Check mutated file
          open MUTATED1, "< $pathMutated" or die $!;
            @linesMutated = <MUTATED1>;
            my $aaTo2 = substr($linesMutated[1],$aaAt-1,1);
            die "Error: aaTo2=".$aaTo2."is different from aaTo=".$aaTo."\n" unless($aaTo2 eq
$aaTo);
            print OUT $aaTo2."\n";
          close MUTATED1;
        }
      }
    }
  close OUT;
close MUTLIST;
```

14.2 Statistical Analysis with MATLAB

A. Section of the main function: main_script.m

```
%% 1. Scatter Wild vs Mutated
figure;
subplot(10,1,1:6)
scatterWM(X.w1,0) % 0 normal;
subplot(10,1,8:10)
scatterWM(X.w1,1) % 1 difference;

%% 2. Counts
X.w1.n = crossCounts(X.w1);
X.w11.n = crossCounts(X.w11);
X.w21.n = crossCounts(X.w21);

%% 3. Histograms bivariate
…

%% 4. sd1: pdf1-pdf0
xyLim = [0 1];
```

```
zLim = [0 0.13];
I = X.w1.sw>=xyLim(1) & X.w1.sm>=xyLim(1);
R.w1.d0 = pdfCalc(X.w1.sw(X.w1.d==0 & I), X.w1.sm(X.w1.d==0 & I), xyLim, 100);
R.w1.d1 = pdfCalc(X.w1.sw(X.w1.d==1 & I), X.w1.sm(X.w1.d==1 & I), xyLim, 100);

figure;
subplot(3,2,1)
surf(R.w1.d1.X, R.w1.d1.Y, R.w1.d1.pdf);
xlim(xyLim); ylim(xyLim); zlim(zLim);
title('pdf_{d=1}(sw,sm); sw,sm\in[0,1]');
xlabel('sw'); ylabel('sm'); zlabel('pdf');

subplot(3,2,3)
surf(R.w1.d0.X, R.w1.d0.Y, R.w1.d0.pdf);
xlim(xyLim); ylim(xyLim); zlim(zLim);
title('pdf_{d=0}(sw,sm); sw,sm\in[0,1]');
xlabel('sw'); ylabel('sm'); zlabel('pdf');

subplot(3,2,5)
surf(R.w1.d1.X, R.w1.d1.Y, R.w1.d1.pdf-R.w1.d0.pdf);
xlim(xyLim); ylim(xyLim); zlim([-Inf Inf]);
title({'sd_1(sw,sm) = pdf_{d=1}(sw,sm)-pdf_{d=0}(sw,sm);', 'sw,sm\in[0,1]'});
xlabel('sw'); ylabel('sm'); zlabel('pdf-pdf');
…

%% 5. sd2: Average
X.w1.sd2 = gridSwSmSd2(X.w1.sw, X.w1.sm, X.w1.d, 0.01);
X.w11.sd2 = gridSwSmSd2(X.w11.sw, X.w11.sm, X.w11.d, 0.01);
X.w21.sd2 = gridSwSmSd2(X.w21.sw, X.w21.sm, X.w21.d, 0.01);
%%
[x,y] = meshgrid(X.w1.sd2.sw, X.w1.sd2.sm);
%% w1: sw 1 and 2 for ROC
X.w1.pdf0 = pdfCalc(X.w1.sw(X.w1.d==0), X.w1.sm(X.w1.d==0), [0 1], 100);
X.w1.pdf1 = pdfCalc(X.w1.sw(X.w1.d==1), X.w1.sm(X.w1.d==1), [0 1], 100);
X.w1.sd1.sd = X.w1.pdf1.pdf-X.w1.pdf0.pdf;
X.w1.sd{1,1} = interp2(x, y, X.w1.sd1.sd, X.w1.sw, X.w1.sm);
X.w1.sd{1,2} = interp2(x, y, smoothn(X.w1.sd1.sd), X.w1.sw, X.w1.sm);
X.w1.sd{1,3} = interp2(x, y, smoothn(X.w1.sd1.sd,X.w1.pdf0.pdf+X.w1.pdf1.pdf), X.w1.sw,
X.w1.sm);
X.w1.sd{2,1} = interp2(x, y, X.w1.sd2.sd, X.w1.sw, X.w1.sm);
X.w1.sd{2,2} = interp2(x, y, smoothn(X.w1.sd2.sd), X.w1.sw, X.w1.sm);
X.w1.sd{2,3} = interp2(x, y, smoothn(X.w1.sd2.sd,X.w1.sd2.d0+X.w1.sd2.d1), X.w1.sw, X.w1.sm);

%% sd1_i: no spline
figure;
surf(X.w1.sd2.sw, X.w1.sd2.sm, X.w1.sd1.sd)
xlabel('sw'); ylabel('sm');  zlabel('sd_1(sw,sm)'); %zlim([0 0.02])
title('sd^i_1: Difference between the probability density functions')
h = colorbar; ylabel(h, 'sd^i_1(sw,sm) = pdf_{d=1}(sw,sm)-pdf_{d=0}(sw,sm)');
%caxis([0, 1])

%% sd1_s: spline
figure;
[sm.z sm.s] = smoothn(X.w1.sd1.sd);
surf(X.w1.sd2.sw, X.w1.sd2.sm, sm.z)
xlabel('sw'); ylabel('sm');  zlabel('sd_1(sw,sm)');
title({'sd^s_1: Difference between the probability density functions fitted with splines',['s
= ' num2str(sm.s) ', GCVscore = 5e-8']})
h = colorbar; ylabel(h, 'sd^s_1(sw,sm) = pdf_{d=1}(sw,sm)-pdf_{d=0}(sw,sm)');

%% sd1_s,w: spline
figure;
w = X.w1.pdf0.pdf+X.w1.pdf1.pdf;
[sm.z sm.s] = smoothn(X.w1.sd1.sd,w);
surf(X.w1.sd2.sw, X.w1.sd2.sm, sm.z)
xlabel('sw'); ylabel('sm');  zlabel('sd_1(sw,sm)');
title({'sd_1^{s,w}: Difference between the probability density functions fitted with weighted
splines',['s = ' num2str(sm.s) ', w = pdf_{d=1}+pdf_{d=0}, GCVscore = 2e-8']})
h = colorbar; ylabel(h, 'sd^s_1(sw,sm) = pdf_{d=1}(sw,sm)-pdf_{d=0}(sw,sm)');

%%
figure;
subplot(3,2,1:4)
heatmap(sd2.z,0.01:0.01:1,0.99:-0.01:0); %imagesc(sd2.z)
xlabel('sw'); ylabel('sm');
title('Average of disease numeric label');
h = colorbar; ylabel(h, 'sd_2(sw,sm) = n_{d=1}(sw,sm) / [n_{d=0}(sw,sm) + n_{d=1}(sw,sm)]');
```

```
subplot(3,2,5)
heatmap(log10(sd2.n1+1),0.01:0.01:1,0.99:-0.01:0);
xlabel('sw'); ylabel('sm');
title('Counts of disease (d=1)');
h = colorbar; ylabel(h, 'log10(n_{d=1}+1)');

subplot(3,2,6)
heatmap(log10(sd2.n0+1),0.01:0.01:1,0.99:-0.01:0);
xlabel('sw'); ylabel('sm');
title('Counts of polymorphisms (d=0)');
h = colorbar; ylabel(h, 'log10(n_{d=0}+1)');


%% sd2_i: no spline
figure;
surf(X.w1.sd2.sw, X.w1.sd2.sm, X.w1.sd2.sd)
zlim([0 1]); xlabel('sw'); ylabel('sm');  zlabel('sd_2(sw,sm)');
title('sd^i_2: Average of the disease numeric label')
h = colorbar; ylabel(h, 'sd_2(sw,sm) = n_{d=1}(sw,sm) / [n_{d=0}(sw,sm) + n_{d=1}(sw,sm)]');
caxis([0, 1])

%% sd2_s: spline
figure;
[sm.z sm.s] = smoothn(X.w1.sd2.sd);
surf(X.w1.sd2.sw, X.w1.sd2.sm, sm.z)
zlim([0 1]); xlabel('sw'); ylabel('sm');  zlabel('sd_2(sw,sm)');
title({'sd_2^{s}: Average of the disease numeric label fitted with splines',['s = '
num2str(sm.s) ', GCVscore = 0.0146']})
h = colorbar; ylabel(h, 'sd_2^{spline}(sw,sm) = n_{d=1}(sw,sm) / [n_{d=0}(sw,sm) +
n_{d=1}(sw,sm)]');
caxis([0, 1])

%% sd2_s,w: spline
figure;
w = X.w1.sd2.d0+X.w1.sd2.d1;
[sm.z sm.s] = smoothn(X.w1.sd2.sd,w);
surf(X.w1.sd2.sw, X.w1.sd2.sm, sm.z)
zlim([0 1]); xlabel('sw'); ylabel('sm');  zlabel('sd_2(sw,sm)');
title({'sd_2^{s,w}: Average of the disease numeric label fitted with weighted splines',['s = '
num2str(sm.s) ', w = n_{d=0}+n_{d=1}, GCVscore = 3e-005']})
h = colorbar; ylabel(h, 'sd_2^{spline}(sw,sm) = n_{d=1}(sw,sm) / [n_{d=0}(sw,sm) +
n_{d=1}(sw,sm)]');
caxis([0, 1])

%% 3. ROC
X.w1.ROC = allROC(X.w1);
% auc = getAUC(X);

%% 3. ROC Plot
plotROC(X.w21)
```