2007－BIO－10 (7)
2007／9／14

機械学習を用いたDNA修復タンパク質の識別と分類

ジェービー・ブラウン　　阿久津 達也
京都大学 化学研究所 バイオインフォマティクスセンター

DNA修復とは、紫外線や複製ミスなどの細胞損傷を修復するプロセスである。細胞の機能を維持する役割は大切だと思われているが、以前のDNA修復研究はバイオインフォマティクスでほとんど行われていない。そこで、我々は機械学習を用いて、DNA修復タンパク質の識別と分類という基本的な問題に取り組む。結果として機械学習によってうまく識別と分類ができるということを示す。

DNA Repair Protein Detection and Classification via Machine Learning

J.B. Brown　　Tatsuya Akutsu
Bioinformatics Center, Institute for Chemical Research, Kyoto University

DNA repair is a critical process in cells, repairing internal and external damage resulting from UV radiation and DNA replication errors, to name only a few. Despite its important role, bioinformatics research on DNA repair is limited. In this talk, we examine two basic problems for the application of bioinformatics to DNA repair: detection of DNA repair-related proteins, and subsequent classification. Results will show that machine learning techniques are highly capable in these two tasks.

1 Introduction

Cells of living organisms are constantly under attack from a myriad of destructive factors. Tobacco smoke, UV radiation, and chemical alteration (such as chemotherapy) are three exogenous sources of damage, while DNA replication errors, standard metabolism producing destructive free oxygen radicals, and hydrolysis (addition of water to break up a molecule) are endogenous types of damage that are constantly occuring. A text with extensive details on DNA damange can be found in Friedberg *et. al* [8].

In response to the many types of DNA damage that occur, there are equally a multitude of ways in which DNA repairs itself, listed in Table 1. These repair mechanisms greatly improve the stability of DNA, cells, and life. For example, the DNA mismatch repair system improves the error rate when copying DNA from one mistake per 10^7 nucleotides to one mistake per 10^9 nucleotides [1].

Table 1: An assortment of DNA repair mechanisms, organized by major category in accordance with the listing in [8].

Repair Category	Repair Mechanisms
Excision	Base excision repair Nucleotide excision repair Transcription-coupled nucleotide excision repair Alternative excision repair Mismatch repair
Strand Break	Single-strand break repair Double-strand break repair
Direct Reversal	Reversal of base damage

Research on how and what damages DNA, as well as what genes (and hence proteins) are responsible for repair has been ongoing for many decades. Every year new knowledge on DNA repair is becoming available, giving researchers the sense that the field still has much to be explored. In 2005, Kimball reported that there were at least 11 DNA polymerases encoded by our genes [10], though only a year

later Linn gave a talk in which he discussed the presence of at least 17 DNA polymerases [11]. So while each of these repair categories has been observed and characterized in laboratories, there still remain many open questions in regard to the complete and complex process known as DNA repair.

The discovery of new DNA repair knowledge via bioinformatics techniques is still at a relatively primitive stage, and here we present a bioinformatics-based framework and results for two fundamental problems relating to DNA repair: detection or discrimination of DNA repair-related proteins, and functional classification of proteins known to be related to DNA repair. By using information processing techniques, we can automatically identify repair-related proteins in unannotated genomes, and undertake large-scale analyses of DNA repair for many organisms.

2 Problem Definition and Methods

The two problems we consider are detection (or recognition) of DNA repair proteins, and functional classification of DNA repair proteins. Let us first consider the detection problem.

2.1 DNA Repair Protein Detection

Detection of a DNA repair protein is a simple question: given a protein, does it belong to the class of proteins which are DNA repair-related or not? The basis of our detection approach is machine learning; in particular, the Support Vector Machine (SVM) has been shown to learn patterns very well [6, 3]. Formally, in an experiment, we are given the three datasets D_+, D_-, and D_0, where D_+ represents a set of known DNA repair proteins, D_- is a set of proteins known to be unrelated to or not involved with DNA repair, and D_0 contains the set of proteins on which we wish to decide for each protein in the set whether or not it is a DNA repair-related protein. Each dataset D contains a finite number of proteins $p_1, p_2, \ldots, p_{|D|}$, initially given in terms of their amino acid sequence information in FASTA format. Since FASTA format sequences cannot be directly input into a SVM, we use the simple spectrum kernel transformation to convert a protein's amino acid sequence data into a numerical vector.

The spectrum kernel [13] is a transformation technique which counts the number of occurrences of each key (in a set of keys) that appear in a query object. While such a definition implies that it could be applied to chemical graphs or other structural analyses, the most common use of the spectrum kernel is in sequence analysis. As a concrete example, if we are dealing with sequences, have the keys $\Sigma = \{A, B\}$, and we query against the sequence $S = ABBAB$, then the basic spectrum kernel $K_s(\Sigma, S)$ results in the vector

$$K_s(\{A, B\}, ABBAB) = (2, 3)$$

because there are two As and three Bs. More generally, if we consider the *k-spectrum kernel*, we can consider all combinations of exactly k input symbols $\sigma \in \Sigma$, denoted by Σ^k. The meaning of K_s remains the same: $K_s(\Sigma^k, S)$ is the $|\Sigma|^k$-length vector representing the number of occurrences of each possible sequence $\sigma_a\sigma_b \ldots \sigma_k$ for $\sigma_a, \ldots, \sigma_k \in \Sigma$ that is exactly k input symbols long. To extend the previous example, consider the 2-spectrum kernel for $\Sigma = \{A, B\}$. We now have $\Sigma^2 = \{AA, AB, BA, BB\}$, since this represents all possible combinations of inputs of exactly length 2. The calculation of K_s is then easily verified upon manual inspection or computation to be

$$K_s(\Sigma^2 = \{AA, AB, BA, BB\}, ABBAB) = (0, 2, 1, 1).$$

When performing detection of DNA repair proteins, we consider the 1-, 2-, and 3-spectrum kernels on amino acid data, though results are shown for 1- and 3-spectrum kernels only. Hence, for each amino acid sequence representation of a protein input, we output a numerical vector of 20, 400, or 8000 dimensions. It is intuitive that as the dimensionality of the numerical vector increases, the SVM has more dimensions at its disposal when deriving a decision function and can hence improve its performance. After the training databases D_+ and D_- are converted to numerical format, the Support Vector Machine performs learning on the databases and derives a decision function. Using that decision function, we evaluate each transformed protein in D_0 to arrive at a decision as to whether it is related to DNA repair or not.

Cross validation is used to create the datasets D_+, D_-, and D_0. In preliminary tests, results were similar for 5- and 10-fold cross validation. Experimental results in Section 4 are given for 5-fold cross validation tests.

Table 2: DNA repair protein classification types used in experiments. The types listed in "Mechanism Category" are the keywords matching the major DNA repair protein mechanisms identified in Table 1. The types listed in "Supplement Category" are the remaining keywords selected for classification but not listed in Table 1.

Category	Protein Functionality
Mechanism Category	Excision, Mismatch
Supplement Category	Atpase, Cross-link, Putative, Polymerase, Helicase, Nuclease, Recombination, Radical

In order to evaluate the SVM's *relative* ability to correctly distinguish between DNA repair and non-repair proteins, we compare its performance to a similar classification task done by using BLAST [2]. In the case of BLAST, the sequences are not transformed; a database D_+ of FASTA format DNA repair protein sequences is provided to build a local BLAST database, and then each protein in D_0 is BLASTed against that database, providing an expectation value (e-value) which measures the confidence that the query sequence is similar to the sequences in the database (hence similar to DNA repair proteins). For BLAST detection, we only use a database of positive (DNA repair) examples for the reason that building a negative (non-repair) database and testing non-repair examples against that database does not use the DNA repair protein database as examples, and therefore cannot appropriately distinguish repair from non-repair. To evaluate how well BLAST detects non-repair proteins, however, non-repair proteins *are* BLASTed against a positive database. As with the SVM method, results via BLAST are confirmed using cross validation.

2.2 Classification of DNA Repair Proteins

Classification of DNA repair proteins is done in a one-versus-rest (1vR) format. By transforming the problem in this way, we can use the same testing framework developed for detection of DNA repair proteins, but for classification D_+ represents one particular class of DNA repair proteins, and D_- represents all other types of repair proteins in a given database excluding class D_+. The classes selected for DNA repair classification are in Table 2, and were selected upon manual inspection of annotation information for the repair-specific databases listed in Table 3. D_0 in this context is the subset of DNA repair proteins we have reserved for testing the ability to recognize that a protein belongs to a certain class. As with detection, D_0 is created in the cross validation technique.

Identical to detection experiments, we use the spectrum kernel transformation in conjunction with the SVM method, and the unchanged amino acid sequences for BLAST. Technique performance is evaluated via cross-validation for classification as well.

3 Materials and Experiments

Databases of DNA repair proteins and non-repair proteins were obtained from several sources. We used the KEGG database [9] API to obtain a set of proteins with DNA repair in their annotation, as well as a set of histones, a similar type of protein which also binds to DNA and resides in the nucleus. We added an additional histone dataset provided by the NIH [12]. In order to test a more realistic multifunction group of non-repair proteins, we queried the UniProt database for "nuclear AND NOT dna repair". To isolate each technique's DNA repair-specific performance by using another DNA repair database, we queried the UniProt database for "DNA repair", and downloaded the resulting query hits.

Upon their initial download, the datasets are not filtered to remove identical or similar sequences, which may occur if identical or similar proteins are present in the genomes of two related organisms. Since filtered datasets better test the abilities of the SVM and BLAST methods, we filter the source datasets at various levels of similarity. The BLAST suite offers a program titled *blastclust* for generating clusters of specified sequence similarity. The source and resulting dataset sizes from filtration are given in Table 3. Also note that the databases downloaded are not for any particular organism, but instead constitute sequence knowledge across all organisms.

Table 3: DNA repair and non-repair data sets obtained from multiple sources, and their resulting dataset sizes after filtration at various levels of similarity. Datasets filtered at 30%, 40%, and 60% similarity had a *blastclust* overlap threshold of 50%, while 80% and 100% similarity clusters had an overlap threshold of 100%.

Data source	Original Size	Sequence similarity threshold and resulting dataset size				
		100%	80%	60%	40%	30%
KEGG DNA Repair	3742	3374	2743	1690	854	526
KEGG Histone	2648	2117	1608	792	414	335
NIH Histone	2189	1370	553	77	11	6
UniProt Non-repair	4519	4366	3293	2080	1513	1182
UniProt DNA Repair	3165	2936	2415	1478	766	473
Combined Repair	6907	4751	3851	2309	1155	688
Combined Non-repair	9356	7574	5321	2900	1914	1494

We performed a number of experiments in a logical fashion in order to test the SVM and BLAST methods with different datasets. By testing with many datasets, we remove bias toward any one particular dataset. Inspired by the work of Bhasin *et. al* [3] which used SVMs to identify histones, we started by performing detection experiments for DNA repair proteins versus histones. After testing both KEGG and NIH histone databases, we changed the negative dataset to the more realistic and multifunctional non-repair dataset obtained from UniProt. Next, we replace the KEGG repair dataset used in the first three experiments by the UniProt repair dataset, while keeping the negative dataset fixed on multifunctional non-repair proteins. To conclude our detection experiments, we combined the two repair sources and the three non-repair sources into aggregated files, filtered the resulting datasets, and ran subsequent experiments using the two aggregated files as positive and negative databases.

Since the classification experiments require the use of a DNA repair-only database as the data source, we use the KEGG and UniProt databases for classification. The filtered datasets are used for classification purposes as well, testing the 1vR generalization capability of SVM and BLAST on increasingly smaller datasets. A relatively simple script divides the database into subsets based on the fixed list of keywords given in Table 2 that appear in annotation information.

4 Results

4.1 Performance Assessment 1: Accuracy

The most intuitive way to measure the performance of the machine learning techniques is to measure the accuracy of the technique when cross validated. Results for detection can be found in [4], with the SVM achieving 90-99% accuracy when tested using KEGG DNA repair proteins against KEGG and NIH histone datsets. SVM classification experiments reported in [5] indicate approximately 95% accuracy for the ten types of proteins listed in Table 2.

With a high level of prediction accuracy, it may appear that the SVM is clearly the technique of choice for detection and recognition of DNA repair proteins. However, the approaches in [4] and [5] suffer several fallbacks which impose the need for additional verification of the technique's usefulness.

The first is that there is no comparison to another machine learning or inference technique. In other words, it may be possible that Hidden Markov Models [7] or BLAST can achieve equal performance. Second, the techiques are only being evaluated at a single threshold, and do not indicate how the techniques perform over a multitude of thresholds. As both the SVM and BLAST techniques output a score indicating a level of confidence in the decision (for either detection or classification), we can further filter out only the "highly confident" proteins by raising the necessary output score requirement (threshold). By doing this we can have greater confidence that the machine's decision is the correct one, though we may miss some genuine DNA repair proteins that simply did not have a high output score. Conversely, we can lower the threshold to be sure to collect all DNA repair proteins, but we introduce the risk of including proteins which are not truly DNA repair-related.

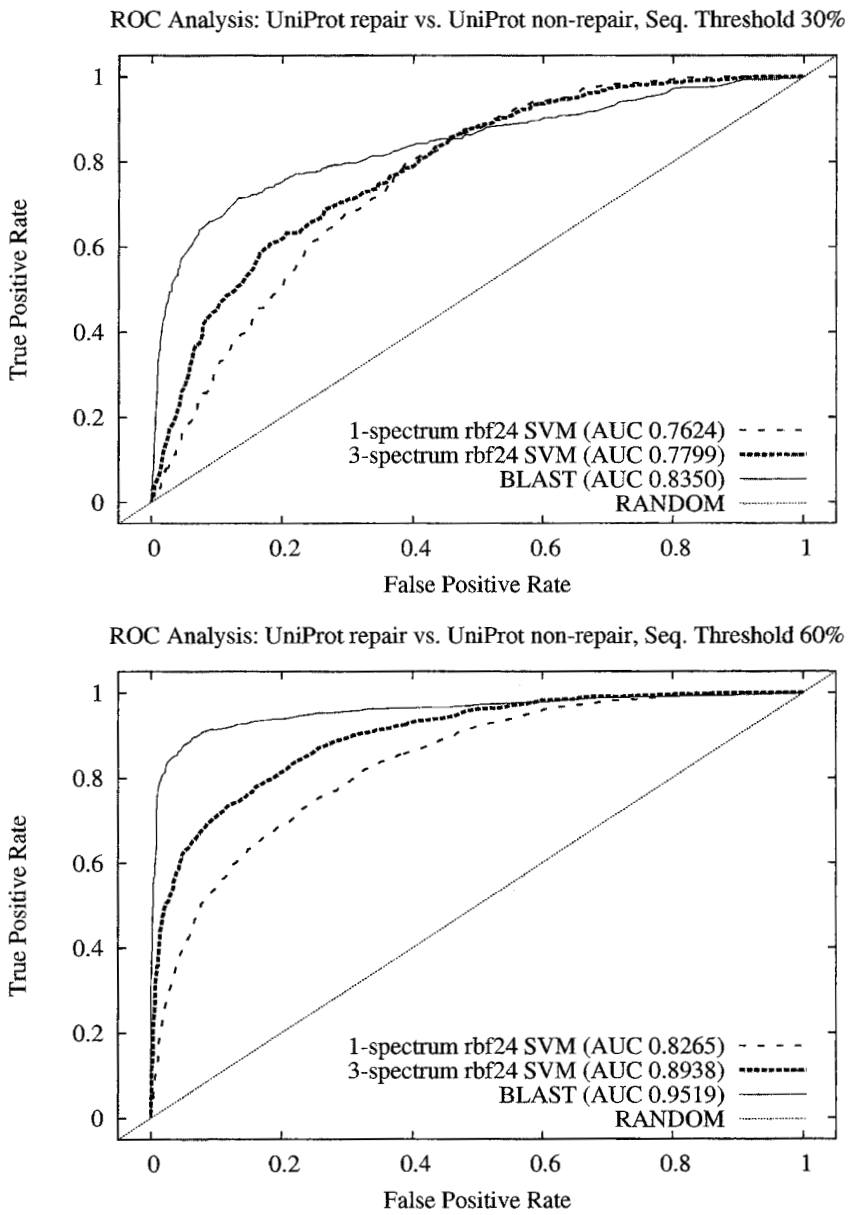

Figure 1: Detection experiments using the UniProt DNA repair protein database and the UniProt multi-function non-repair protein database. SVM experiments are performed using 1- and 3-spectrum kernels before input to the SVM, which uses the RBF kernel with a gamma value of 2.4 for similarity calculations. BLAST experiments are performed as described in Section 2. The total performance AUC statistic shows that BLAST is the better technique, though this is biased because of the use of *blastclust* to generate dissimilar datasets. For the 30%-filtered dataset, the SVM method can achieve 100% true positive detection both quicker and with a lower false positive rate than the BLAST method. As more data becomes available in the 60% dataset, the BLAST technique performs better.

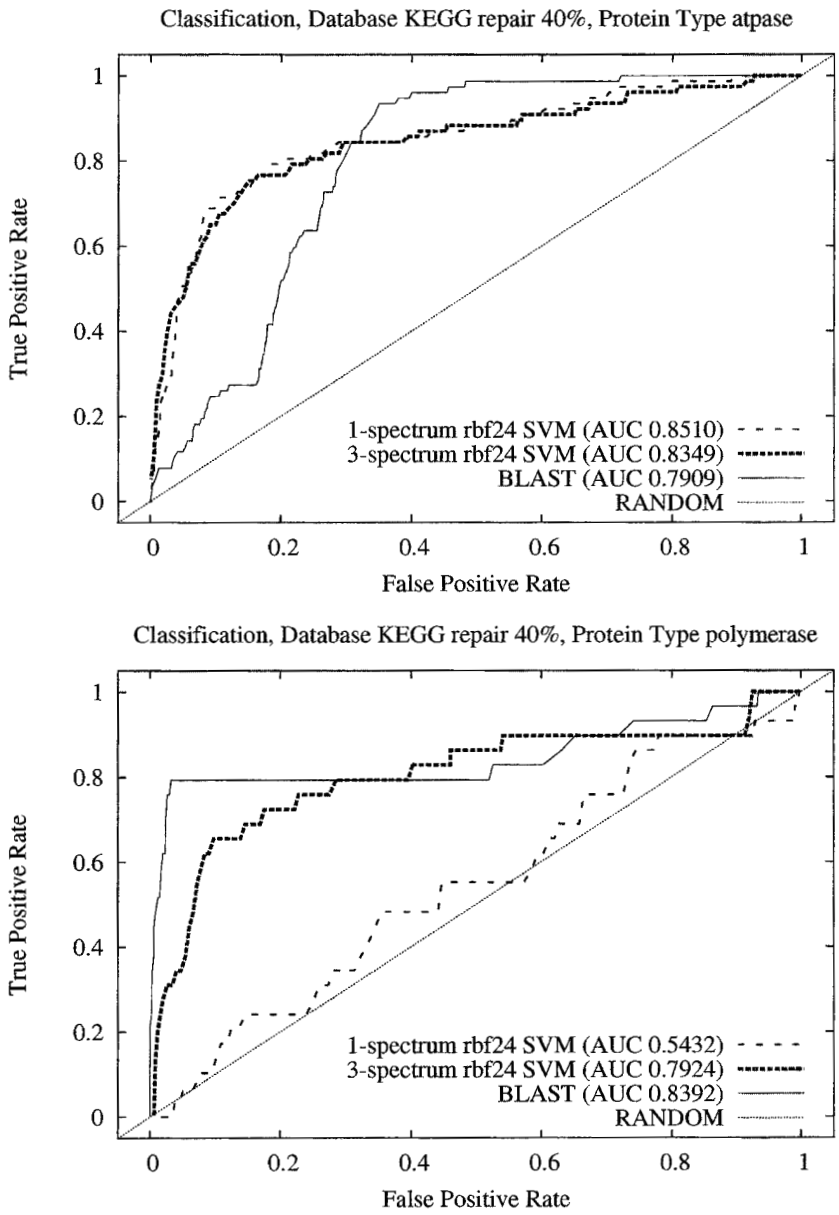

Figure 2: Classification experiments using the KEGG DNA repair protein database filtered at 40% similarity. SVM experiments are performed using the 1- and 3-spectrum kernels, and a RBF kernel with a gamma value of 2.4 for similarity calculations. BLAST experiments are performed as described in Section 2. With respect to AUC, the SVM method outperforms the BLAST method for the protein type atpase, though BLAST is the AUC winner for identifying DNA polymerases. The SVM method produces results on par with or better than the BLAST method despite the KEGG dataset being created and maintained via homology searching similar to BLAST [9], and despite the use of *blastclust* to create dissimilar datasets.

4.2 Performance Assessment 2: ROC and AUC

The tradeoff between identifying true DNA repair proteins while allowing some non-repair proteins to be misjudged, using different threshold levels, is indicated through the use of a Receiver Operating Characteristic (ROC) curve. By using a large range of thresholds (based on output values from the SVM and BLAST methods), we visualize the ability of each technique to recognize and classify DNA repair proteins.

Let us define the four types of outcomes in detection experiments: true positives are proteins that are known to be DNA repair-related and predicted to be DNA repair-related; false negatives are proteins that are known to be DNA repair-related but predicted to be non-repair; true negatives are proteins that are known to be non-repair and predicted to be non-repair; and false positives are proteins known to be non-repair but predicted to be DNA repair-related.

Figure 1 shows ROC curves for the UniProt repair versus UniProt non-repair detection experiment, where datasets have been filtered down to 30% and 60% sequence similarity. The curves show that both techniques are good at detecting DNA repair proteins at both 30% and 60% similarity levels, achieving total Area Under the Curve (AUC) scores of over 0.75 in all experiments (the maximum AUC possible is 1.0 for an ideal method). Upon inspection it can be seen that the BLAST method has higher AUC scores. This is an unfortunate bias due to the way data is prepared, because of the use of *blastclust* to create similarity clusters. Since the BLAST algorithm creates the clusters, it is a matter of course that the BLAST algorithm can better recognize DNA repair proteins from those clusters.

The results of classification are less inclined towards BLAST despite *blastclust* being used to generate the datasets. For clarity purposes, let us again define the four types of outcomes in the classification experiments: true positives are proteins that are known to have a specific function in DNA repair and are predicted to have that function (e.g., a polymerase predicted to be a polymerase); false negatives are proteins known to have a function but predicted not to have that function (e.g., a polymerase predicted to be a non-polymerase); true negatives are proteins known to not have a specific function and are also predicted in such a way (e.g., a non-polymerase predicted to be a non-polymerase); and false positives are proteins predicted to have a certain function but are actually known not to have that function (e.g., a non-polymerase predicted to be a polymerase).

In Figure 2, ROC curves are given for DNA repair atpase and polymerase classification experiments. For atpase, the SVM method produces a better result than BLAST, though more curiously the 1-spectrum kernel is the top performer. We can attribute this to overtraining on a smaller amount of data, as experimental results at the 30% similarity level were similar. For 60%-100% similarity datasets, experimental results (not shown here) show that the 3-spectrum kernel SVM was the best performer among all methods. As mentioned in Section 2, the higher dimensionality of the 3-spectrum kernel provides greater flexibility in deriving a decision function in the presence of more data resulting from higher similarity thresholds.

For the protein type polymerase, BLAST is the winner, as shown in the bottom half of Figure 2. In this situation, the difference between the 1- and 3-spectrum kernel is evident, though BLAST still performs 4% better than the 3-spectrum SVM.

For the remaining eight types of proteins, results vary and largely depend on the level of sequence threshold. *blastclust* does introduce a bias for BLAST, though the SVM does manage to compete well even with the bias. As exemplified by Figure 2, approximately half of the protein types are better classified using the SVM, with the other half being better predicted via BLAST.

5 Discussion and Future Developments

The spectrum kernel based SVM and the standard BLAST methods do a good job of detection and classification, in terms of both accuracy and AUC. Unfortunately, the use of *blastclust* to generate increasingly dissimilar datasets introduces a bias into the performance of BLAST when comparing the SVM technique against it. As a result, future work should have a way to generate smaller datasets that do not suffer the bias presented here. Though less biologically significant, one possible way to generate datasets is to randomly select a fixed size or percentage from the original database, and then perform experiments over a large number of randomized trials. In this way, the techniques can be analyzed for their statistical performance as well.

The framework discussed in this report is a basic framework for application of machine learning to DNA repair. From here, several projects or extensions are possible, of which a few we consider below.

Irrespective of the bias from *blastclust*, the techniques are accurate enough to apply toward the mass scanning of unannotated genomes. It is an interesting research problem to see if there are trends across all genomes with respect to the amounts and types of DNA repair proteins. Such an analysis would be useful because of its real-world application.

Construction of a DNA repair protein interaction network *in silico* for many organisms would open research to examine if the interactivity in DNA repair has properties such as scale-free connectivity, which would be indicated by the presence of hubs, and the identification of several crucial proteins that keep the DNA repair process normally functioning. By completing such a network and analysis, we may be able to design better drugs for the human immune system. The link from this work to the proposed network extension is the use of machine learning techniques to automatically identify the proteins in a database for inclusion in the network.

The spectrum kernel by itself provides good performance in detection and classification of DNA repair proteins. However, it includes no particular biological knowledge. A more useful kernel method for DNA repair should be able to make use of DNA repair specific information, such as a repair protein profile alignment or repair protein structural properties.

References

[1] B. Alberts, D. Bray, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology: An Introduction to the Molecular Biology of the Cell.* Garland Publishing, Inc., 19 Union Square West, New York, 10003, 1998.

[2] S.F. Altschul, W. Gish, W. Miller, E.W. Myers, and D.J. Lipman. Basic alignment search tool. *J. Mol. Biol.*, 215:403–410, 1990.

[3] M. Bhasin, E.L. Reinharz, and P. Reche. Recognition and classification of histones using support vector machine. *Journal of Computational Biology*, 13:102–112, 2006.

[4] J. Brown and T. Akutsu. Dna repair recognition via support vector machines (poster 108). In *The Seventeenth International Conference on Genome Informatics 2006*, 2006.

[5] J. Brown and T. Akutsu. Classification of dna repair proteins via support vector machines (poster 4). In *The Seventh Annual International Workshop on Bioinformatics and Systems Biology*, 2007.

[6] N. Christianini and J. Shawe-Taylor. *Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, Cambridge, England, 2000.

[7] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison. *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids.* Cambridge University Press, Cambridge, England, 1998.

[8] E. Friedberg, G. Walker, W. Siede, R. Wood, R. Schultz, and T. Ellenberger. *DNA Repair and Mutagenesis.* ASM Press, Washington D.C., 2006.

[9] M. Kanehisa, S. Goto, M. Hattori, K.F. Aoki-Kinoshita, M. Itoh, S. Kawashima, T. Katayama, M. Araki, and M. Hirakawa. From genomics to chemical genomics: new developments in kegg. *Nucleic Acids Res.*, 34:D354–357, 2006.

[10] J. Kimball. Dna repair. Online, http://users.rcn.com/jkimball.ma.ultranet/ BiologyPages/D/DNArepair.html, 2005.

[11] S. Linn. History of dna repair - life in the serendipitous lane: Excitement and gratification in studying dna repair. Online, http://videocast.nih.gov/ram/dnarig062006.ram, 2006.

[12] L. Mariño-Ramírez, B. Hsu, A. Baxevanis, and D. Landsman. The histone database: a comprehensive resource for histones and histone fold-containing proteins. *Proteins*, 62:838–842, 2006.

[13] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis.* Cambridge University Press, Cambridge, England, 2004.