Sequence Analysis

Optimizing amino acid groupings for GPCR Classification

Matthew N. Davies[1,*], Andrew Secker[2], Alex A. Freitas[2], Edward Clark[3], Jon Timmis[3] and Darren R. Flower[1]

[1]Edward Jenner Institute, Compton, Newbury, Berkshire, RG20 7NN, U.K.
[2]Department of Computing and Centre for BioMedical Informatics, University of Kent, Canterbury, Kent CT2 7NF, U.K.
[3]Departments of Computer Science and Electronics, University of York, Heslington, York YO10 5DD, U.K.
Associate Editor: Prof. John Quackenbush

ABSTRACT

Motivation

There is much interest in reducing the complexity inherent in the representation of the twenty standard amino acids within bioinformatics algorithms by developing a so-called reduced alphabet. Although there is no universally-applicable residue grouping, there are numerous physiochemical criteria upon which one can base groupings. Local descriptors are a form of alignment-free analysis, the efficiency of which is dependent upon the correct selection of amino acid groupings.

Results

Within the context of G-protein coupled receptor (GPCR) classification, an optimisation algorithm was developed able to identify the most efficient grouping when used to generate local descriptors. The algorithm was inspired by the relatively new computational intelligence paradigm of Artificial Immune Systems. A number of amino acid groupings produced by this algorithm were evaluated with respect to their ability to generate local descriptors capable of providing an accurate classification algorithm for GPCRs.

Contact

m.davies@mail.cryst.bbk.ac.uk

1 INTRODUCTION

The twenty standard amino acids can be grouped or classified using a wide variety of distinct criteria, since each amino acid side chain possesses many different attributes. From an evolutionary perspective, it can be assumed that the presence of twenty different residues confers a selective advantage upon organisms, one that provides sufficient variety to build functional proteins without overcomplicating the transcription of proteins from RNA. It is also possible that proteins were once created from a much smaller set of amino acids. Research into amino acid evolution suggests the abiotic environment may have contained many hydrophobic and charged amino acids but few polar residues (Matthews and Moser 1967; López *et al.*, 2007). Studies into proteins containing predominantly the residues lysine, alanine and isoleucine suggested that is possible to generate stable structures based purely on hydrophobic and electro-

static interactions, provided the protein is stabilised by a Gly-Gly-Tyr C terminus. Moreover, a reduced alphabet is capable of reproducing complex protein structures experimentally (Luthra *et al.*, 2007). The Baker group produced a S3 fold using only five amino acids (Riddle *et al.*, 1997) (Ile-Ala-Glu-Lys-Gly) while Stroud et al. generated a 108 residue protein with a four-helix bundle using only seven different amino acids (Schafmeister *et al.*, 1997).

Presumably, the greater diversity of amino acids has been instrumental in allowing larger and more intricate protein structures to evolve. However, from a computational viewpoint, there are significant advantages in reducing the number of amino acids within a representation. It is more computationally efficient to deal with a smaller number of variables than 20. Moreover, by grouping amino acids into a reduced alphabet, and thus minimising noise, a more accurate protein sequence representation may be created. The grouping may allow conserved structural and functional properties to be identified that are independent of specific motifs. Thus, reduced alphabet approaches have a wide range of potential applications within bioinformatics.

Determining accurate amino acid groupings is extremely difficult due to the astronomically large number of possible ways to group twenty objects. The actual number of groupings can be calculated using Stirling Numbers of the Second Kind (Luthra *et al.*, 2007). There are approximately 5.172×10^{13} possible groupings that can be formed from 20 amino acids. Numerous groupings have been proposed based on the biochemical properties of the amino acids. An obvious grouping separates hydrophilic and hydrophobic residues, as these are fundamental to the behaviour of amino acids in solution (Melo and Marti-Renom *et al.*, 2006). Other obvious groups include acidic residues (Glu and Asp), the basic residues (Lys and Arg) and the alcohols (Ser and Thr). Other residues present properties seemingly unique amongst amino acids (Li *et al.*, 2007): cysteine, which forms disulphide bonds; proline, which forms a bond with its own side chain; and glycine, which is much more flexible than other residues.

Dayhoff's substitution matrix was perhaps the first systematic attempt at grouping. It measured the tendency of one amino acid to be replaced by another (Dayhoff *et al.*, 1978a). Taylor later combined information from substitution matrices with physico-chemical properties to derive amino acid groupings (Taylor

---

*To whom correspondence should be addressed.

1986). More recently, Wang & Wang (1999) classified amino acids using a Miyazawa-Jernigan-like matrix to obtain reduced alphabets based on inter-group energetic interactions. Jing and Wei (2007), who undertook sequence alignment of reduced alphabets, and Li *et al.* (2003) used both alignment scoring and substitution matrices within a Monte Carlo approach to obtain the best grouping. Cannata *et al.* (2002) used the BLOSUM and PAM substitution matrices to evaluate all possible simplified alphabets using a "branch and bound" algorithm.

A principle focus of bioinformatics is the identification and classification of protein structure and function from primary sequence. The GPCR superfamily is a large and diverse multigene superfamily of integral membrane proteins that perform many important physiological functions (Christopoulos and Kenakin 2002; Gether *et al.*, 2002; Bissantz 2003). Approximately 50% of marketed drugs target GPCRs and they are themselves a common target for virtual screening (Flower 1999). Previous work using reduced alphabets to classify GPCRs used functional (four letter), hydrophobic (two letter), chemical (eight letter) and structural (three letter) alphabets to represent their sequences and developed motifs based upon such representations (Gangal and Kumar 2007). The reduced alphabet motifs were shown to perform as accurately as PROSITE (Hulo *et al.*, 2006) and PRINTS (Attwood *et al.*, 2002, Flower and Attwood 2004). Structure is better conserved than sequence within the GPCR superfamily, thus alignment-free approaches have often been more effective at classification than techniques based solely on sequence similarity (Davies *et al.*, 2007a; Davies *et al.*, 2007b). Local descriptors are an alignment-free approach (Cui *et al.*, 2007; Zhang *et al.*, 2007) used previously to classify several protein families. The effectiveness of techniques using local descriptors depends largely on the underlying amino acid grouping. Thus accuracy should improve if the grouping is optimised. Research on reduced alphabets has shown that the number of different groupings is very high and it is impractical to determine which is best *a priori*. To overcome this, we have optimised amino acid groupings, used for local descriptor-based GPCR classification, by improving the quality of the solution over and above the use of pre-defined groupings. This paper proposes optimising the groups in a data driven manner, using a procedure for the optimisation of amino acid grouping based on Artificial Immune Systems, a relatively new computational intelligence paradigm for optimisation and machine learning/data mining. The advantage of such an optimiser is that a classifier may be used to gauge the quality of a solution or solutions at each stage during optimisation. Optimisation of the representation (groupings) is guided by the classification algorithm used during final testing. Therefore, the representation will exploit any bias in that classifier to improve the prediction.
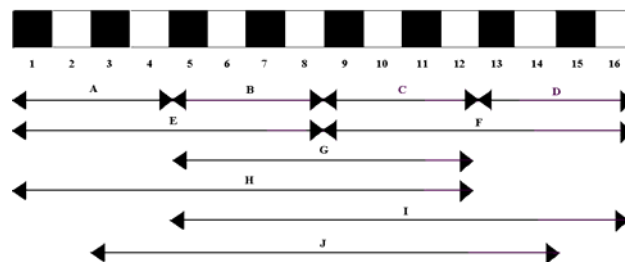

Figure 1: The 10 descriptor regions (A-J) for a theoretical protein sequence of 16 amino acids. Adapted from Zhang *et al.* (unpublished). The regions A-J are determined by firstly dividing the entire sequence into four equal regions (A-D) and then two equal regions (E-F). G represents the central 50% of the sequence, while H the first 75%, I the final 75% and J the central 75%.

2 METHODS

2.1 GPCR Classification

In order to develop an effective algorithm for GPCR sequence classification, it was necessary to build a large and comprehensive dataset of GPCR sequences with which to train and test the classifier. Protein sequences were identified using the Entrez search and retrieval system. The system searches protein databases such as SwissProt, PIR, PRF, PDB, as well as translations from annotated coding regions in DNA databases, such as GenBank and RefSeq. Text-based searching identified all sequences within each sub-subfamily of the hierarchy. These composite groups were then used to build each GPCR sub-family and Class level dataset. Sequences shorter than 280 amino acids were excluded to eliminate incomplete protein sequences, and all identical sequences within the dataset were removed to avoid redundancy. This left 8354 protein sequences in 5 classes at the family level (A-E). Class F was not considered as it contains too few sequences from which to develop an accurate classification algorithm.

2.2 Local Descriptors

In developing their local-descriptors technique, Cui et al**.** (2007) divided the amino acids into three functional groups: hydrophobic (CVLIMFW), neutral (GASTPHY), and polar (RKEDQN), as suggested by Chothia and Finkelstein *et al.* (1990). The variation of these groups within a sequence is the basis of the three local descriptors: composition (C), transition (T), and distribution (D). C is the proportion of amino acids with a particular property (such as hydrophobicity). T is the frequency with which amino acids with one property are followed by amino acids with a different property. D measures the chain length within which the first, 25%, 50%, 75% and 100% of the amino acids of a particular property are located. Given that the amino acids are divided into three groups in this instance, the calculation of the C, T and D descriptors generates 21 attributes in total (3 for C, 3 for T and 15 for D). While this technique would be valid if applied over the whole amino acid sequence, Zhang *et al.* (2005) split the amino acid sequences into 10 overlapping regions in order to better capture epitope binding patterns (see Figure 1). For sequences A-D and E-F there may be cases where the sequence cannot be divided exactly, in which case each sub-

sequence may be extended by one residue. Each descriptor - C, T, and D - is calculated over the 10 subsequences, resulting in 210 features describing the protein. For such a representation, we need not define a specialised data mining (classification) algorithm, as the protein can be represented by 210 numerical attributes. Thus, predictions can be made using any of the many suitable, well-documented classification algorithms with little or no modification.

2.3   Optimiser

The opt-aiNet algorithm (de Castro and Von Zuben 2001; Andrews and Timmis 2005; Timmis and Edmonds 2004) was used to optimise groupings. Opt-aiNet belongs to a class of algorithms known as Artificial Immune Systems (AIS) (de Castro and Timmis 2002a; de Castro and Timmis 2002b). The artificial immune system that has been used (opt-aiNET) has previously been benchmarked against other evolutionary algorithms, such as genetic algorithms, and has been found to be very competitive. Such immune algorithms are either population-based (where every individual in the population encode potential solution) or network-based (where individuals again encode potential solution but interact via some form of simulation and/or suppression). The algorithm is evolutionary in nature and uses a selective pressure applied to the whole population of candidate solutions to the groupings. This has the effect, over many generations, of improving the average quality of the population. The algorithm uses a combination of the clonal selection principle and idiotypic network theory to drive the optimisation process. A population of individuals (artificial immune cells) is generated where each member encodes a grouping scheme for the 20 amino acids.

5 amino acids are assigned to 3 groups. Each position in a cell's string represents an amino acid; the value at that position represents the group ID to which the amino acid is assigned. During initialisation of the algorithm, each member of the population is initialised by placing random values in each position in the artificial immune cell, thus generating random groupings of amino acids. The quality of each cell is assessed, each cell is then cloned and mutated with a rate inversely proportional to their parent's (and therefore their) quality. The better the solution that the cell encodes, the fewer positions that are mutated. When all the cells in the population have been cloned and mutated, a small number of poorly performing cells are discarded through a process of suppression and interaction between the cells, which replaces them in the population with an equal number of randomly generated cells. The injection of randomly configured cells discourages premature convergence on a local optimum.

2.4   Fitness function

Several procedures are required to assess the representation as encoded by the cell. The groupings defined by a cell must be translated from that cell's representation. The groups are then used as described above (Local Descriptors section) to create numerical attributes for every protein within the dataset. A dataset was produced consisting of 70n predictor attributes (where n is the number of groups defined by the cell). This dataset (the training data) was then split into two further sets, sub-training and validation sets, in the ratio 80%/20%. A classification algorithm was trained on the sub-training data and tested using the validation data. The quality of the cell is the percentage predictive accuracy output by the classifier on the validation data. Since each cell encodes a different set of groups, creating a new training set from the encoded groupings and then training and testing the classifier must be repeated for fitness evaluation.

2.5   Protocol

A Naïve Bayes classification algorithm from the WEKA data mining toolkit (Witten and Frank 2005) provided the fitness function, along with several auxiliary functions regarding data manipulation. Naïve Bayes was chosen as the classifier for the evaluation function of the optimiser mainly because it is computationally fast, which is an important consideration given the very time-consuming nature of the optimisation process.The optimiser was run 10 times and the output recorded. Each run was one single fold of a 10-fold cross validation test over the entire dataset. To reduce the probability of overfitting and reduce computing time, for each fold the number of training items was reduced randomly to half its size. A balance must be struck between optimising the representation using the training data rather than optimising for the training data. In the original opt-aiNet, the algorithm terminates when there is no improvement beyond a population threshold between successive iterations. As the present problem is more complex, several iterations could pass without improvement, and so the system was terminated after a specified number of iterations. The opt-aiNet optimiser is run for a total of 50 generations, using a population size of 20 individuals (artificial cells). The parameters of the algorithm are shown in Table 1.

Number of initial cells in the network	20
Number of clones generated for each network cell	20
Maximum number of algorithm iterations	50
Suppression threshold for network cell affinities	0.5
Max number of groups	16

Table 1: Defined parameters for the opt-aiNet optimiser

While the algorithm could form groups using any combination of amino acids, a total of 16 groups was enforced: this allowed fair comparison with the seeded groupings, as defined below. Moreover, enforcing such a maximum is a compromise between the time needed for fitness evaluation as the number of groups and predictor attributes increases and not constraining the system so that it produces sub-optimal groupings. Preliminary tests showed that groupings that performed well rarely contained more than 12 groups, thus 16 was a safe threshold.

Groupings	Groups No.	Reference
CMFILVWY AGTSNQDEHRKP	2	Li *et al.*
CMFILVWY AGTSP NQDEHRK	3	Li *et al.*
CMFWY ILV AGTS NQDEHRKP	4	Li *et al.*
FWYH MILV CATSP G NQDERK	5	Li *et al.*
FWYH MILV CATS P G NQDERK	6	Li *et al.*
CFYWMLIV GPATSNHQEDRK	2	Li *et al.*
CFYWMLIV GPATS NHQEDRK	3	Li *et al.*
CFYW MLIV GPATS NHQEDRK	4	Li *et al.*
CFYW MLIV G PATS NHQEDRK	5	Li *et al.*
CFYW MLIV G P ATS NHQEDRK	6	Li *et al.*
ARNDCQEGHKPST ILMFWVY	2	Cannata *et al.*
ARNDQEGHKPST C ILMFWVY	3	Cannata *et al.*
ARNDQEGHKPST C ILMFVY W	4	Cannata *et al.*
AGPST RNDQEHK C ILMFVY	4	Cannata *et al.*
AGPST RNDQEK H C ILMFVY W	6	Cannata *et al.*
A RK N D C Q E G H IV LM F P ST W Y	16	Cannata *et al.*
AS RK N D C Q E G H IV LM F P T W Y	16	Cannata *et al.*
A R K N D C Q E G H IV LM FY P ST W	16	Cannata *et al.*
AST R K N D C Q E G H IV LM F P W Y	16	Cannata *et al.*
A R K N D C QE G H IV LM F P ST W Y	16	Cannata *et al.*

Table 2: The set of optimised amino acid groupings from Li *et al* 2003 and Cannata *et al* 2002 that were used to initiate the Seeded grouping simulations.

Two sets of tests were run. The first used previously determined groupings from Li *et al.* (2003) and Cannata *et al.* (2002), which reduced the amino acid alphabet from 20 to a range of 2-16 allowing a wide range of initial pre-defined groupings to be represented. This population began as biologically-grounded grouping schemes rather than random groupings; however, the algorithm was free to change these groupings in a data-driven manner. These are the "seeded" groupings. The second used a randomly initialised population as is usual in AIS; these are the "random" groupings. For the seeded population, the initial groupings are displayed in Table 2: each row represents the seeded grouping of one of the 20 artificial cells in the population. The original opt-aiNet algorithm injected randomly configured cells at each step to maintain population diversity. This was removed here, as they were incompatible with the notion of seeding. This has the added advantage that the final population will contain cells descended from an initial cell. As such, it is possible to interrogate the final population to determine how the initial groupings changed during optimisation. The experimental protocol and algorithm parameters were kept constant between the two sets of tests.

3 RESULTS

The overall accuracies of the simulation are shown in Figure 2 and tended to vary between 87-90% accuracy at the GPCR Class level. The accuracy from the "seeded" experiment is show to be slightly superior to that of the random grouping and this is maintained throughtout subsequent iterations. Previous work using amino acid composition at the basis of local descriptors had shown an accuracy of 56% at the class level, proving the local descriptors provide a significantly stronger basis for the representation of protein sequences.
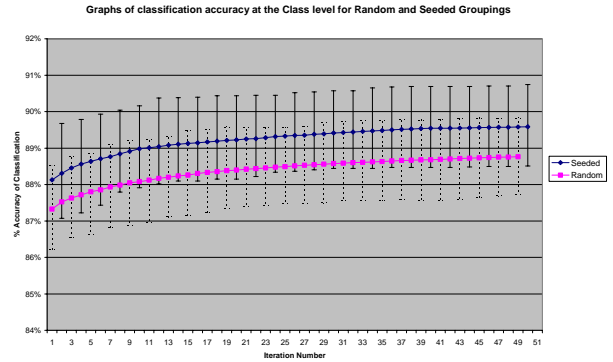

Figure 2. Graphs of classification accuracy at the Class level of the course of grouping optimization for the Seeded and Random populations

For the seeded experiment, the 20 suggested groupings were assessed before mutation occurred so that the initial favoured grouping was always the same grouping of 16, which pairs glutamine and glutamtic acid (QE), isoleucine and valine (IV), leucine and methionine (LM) and serine and threonine (ST). These groupings represent a relatively minor reduction of the alphabet. Subsequent iterations generated final groupings containing 6 to 11 individual groups. This represents a more substantial alphabet reduction (see Table 3). The initial population contained between 2 and 16 but individuals representing fewer than 5 or more than 14 groups are quickly lost, suggesting that 7 to 11 groups is optimal. Variation in the mean group size during optimisation is shown in Figure 2a-b. On average, the number of groups per cell is slightly higher for the random simulation, but this may result from the initial random groupings vary from 8 to 14, so that weak groupings are eliminated quickly. The number of groups and the quality of the cells has a tendency to stabilise during the final stages of optimisation. However, the random set has a significant tendency to produce higher numbers of groups throughout the simulation (see Figure 3).

Although optimisation was driven by the accuracy of Naïve Bayes, it is noteworthy that the 1-Nearest Neighbour algorithm obtained a higher accuracy. One explanation for this is that Naïve Bayes assumes that predictor attributes are independent from each other and conditioned on the class to be predicted; in the present case this assumption is violated. Indeed there is considerable redundancy in the attributes derived from the local descriptors. For example, there is considerable overlap between the 10 different regions used to produce the local descriptors; see Figure 1.
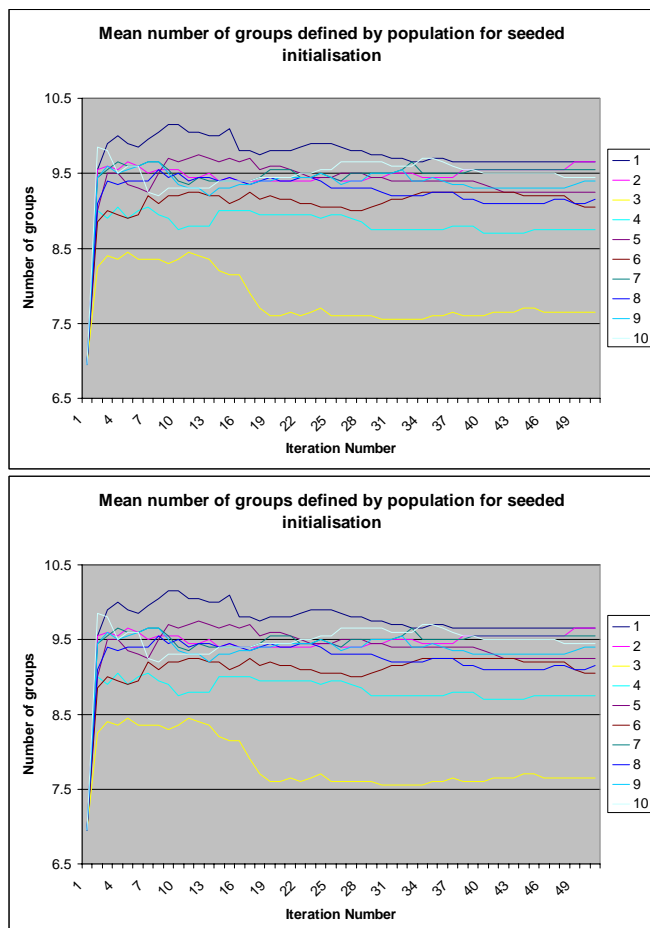
Mean number of groups defined by population for seeded initialisation

Mean number of groups defined by population for seeded initialisation

Figure 3a-b. Graphs of mean grouping population against time step of simulation for Seeded (top figure) and Random (bottom figure) Grouping

Final Groupings by Seeded Initialization	Cross-validation Fold	Groups No.
SQLGE DRP KT W HN C VIMAFY	1	7
SDKQGAE RN WF H C PLVT IM Y	2	8
SDPGATE RWHYN KQ C LVMF I	3	6
SQMAE DRKN WHG C PT LVF I Y	4	8
DKP SRGFNE WH C Q LVMAY I T	5	8
SG DRK WH C PQAE LVIMTN F Y	6	8
SGA DP RWHYN KE C Q LT IV MF	7	9
SPGAT DRKQFYE WHN C LVM I	8	6

	Cross-validation Fold	Groups No.
SWGA DRHQYNE KLVIF C PM T	9	6
SGE DP RWN KQ HLVIMFY C AT	10	7

Final Groupings by Random Initialization	Cross-validation Fold	Groups No.
S DP RKQ WHN C LGAE VM I F Y T	1	11
SG DRP KN WH C Q LYE VAT I M F	2	11
SWQGN DRKE HY C P LVMAF I T	3	8
SAE DL R KP WV HQ C GM IF YN T	4	11
SG DKA RN WHQE C PM LVT IF Y	5	9
S DRQ KPLYE WN H C V GI MAT F	6	10
SD RW KVA HGN CT P QFE L IM Y	7	10
SKA DRPQGE WN HF C L VY IM T	8	9
SQ DLV RPE KIA WGM H C F T YN	9	9
SG DVIA RQN KP WHY C LE MF T	10	9

Table 3: Final amino acid groupings for the Seeded and Random Groupings

Table 4: Matrix of the incidence of paired amino acids within the same group

The average numbers of groups over all iterations and over all cells for the seeded and random groupings were 7.3 and 8.8 respectively. Despite the higher average group size for the random set, there is a clear tendency towards similar distributions. This is a hugely significant result: it suggests that the same factors drive the optimisation of groupings irrespective of the initial starting point. Most importantly, cysteine is put in its own group in all but one of the final groupings; see Figure 4. This may be because cysteine can form disulphide bonds, a unique property amongst residues and one which may be crucial for GPCR classification. Disulphide bonds stabilise GPCR structure and the formation of intermolecular bonds is believed to be crucial to receptor dimerisation and oligomerisation (Lee 2000). Moreover the GPCR Class B Secretin family has an N terminus of ~60–80 amino acids containing conserved disulphide bonds which bind to the receptor's large peptide hormone ligand (Fredriksson *et al.*, 2003). Cysteine constitutes only 1.51% of amino acids in an average protein, suggesting that it has a disproportionate influence on protein structure and stability. No other residue is placed

within a single group for more than 40% of final groupings. However, isoleucine and threonine do form a single group in 7 and 8 instances (out of 20). Although serine and threonine are small residues containing a hydroxyl group, there are only two incidences (out of 20) of them being paired. Isoleucine and leucine are isomeric and hence have very similar physiochemical properties, yet both show a greater propensity to pair with valine, another medium sized hydrophobic residue, than with each other.
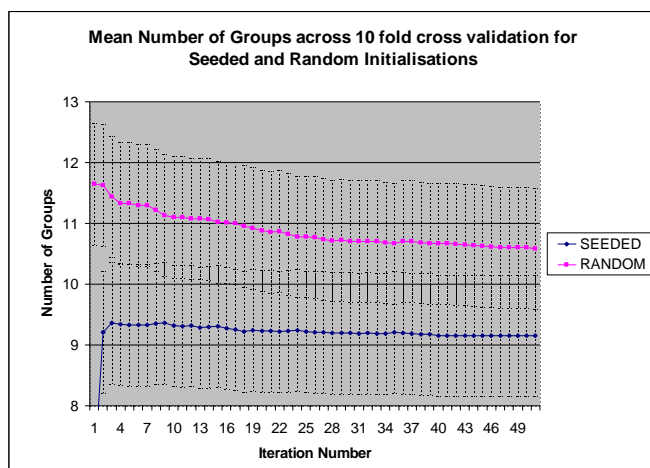

Figure 4: Mean number of groups (with error bars) across 10 fold cross validation for seeded and random initialisations.
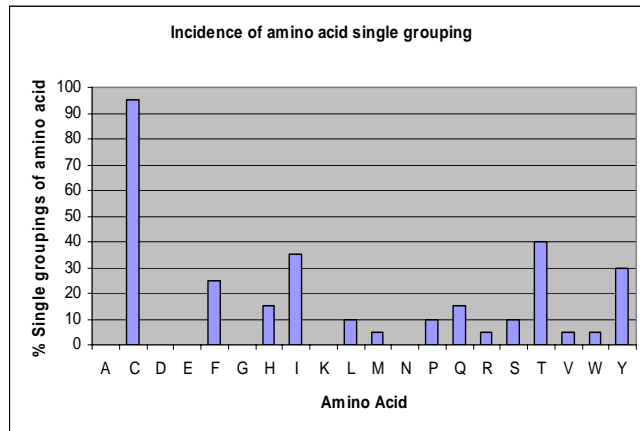

Figure 5: Incidence of amino acid single groupings. Cysteine is consistently grouped alone, suggesting its properties are more unique than the other side chains

The most frequent pairings of residues are Ser/Gly, His/Trp and Leu/Val. Serine and glycine are likely to be grouped as both have small side chains with molecular weights less than 110. The only other similarly sized amino acid, alanine (molecular weight of 85), is often grouped with both residues. Leucine and valine are medium-sized hydrophobic amino acids, although valine has a slightly shorter side chain. Tryptophan and histidine are a less obvious pairing; tryptophan is a large hydrophobic residue while histidine can move between the protonated and unprotonated forms due to its pKa value of ~6.0. Although this is a unique property amongst amino acids, histidine is not as

grouped singly as often as cysteine. What tryptophan and histidine do share is the presence of a nitrogen-containing aromatic ring. Tryptophan contains an indole ring, while histidine contains an imidazole ring. The other aromatics residues, phenylalanine and tyrosine, do not contain a nitrogen-bearing ring. It is possible that this ring is a property shared only by the paired residues. In all cases, it seems likely that the pairing of these particular residues causes no significant loss of information to the representation of the protein sequence and may therefore be useful reductions of the amino acid alphabet in the context of protein classification and analysis.

4 CONCLUSION

Any rational grouping to form a reduced amino acid alphabet depends upon the relative importance given to each of their numerous physiochemical properties. It seems unlikely that a single universal grouping will be appropriate for all bioinformatics problems. Chothia and Finkelstein's three-way grouping is a somewhat simplistic basis for local descriptor generation and there is no evidence that it is the best representation. The optimisation algorithm proposed here suggests a larger number of groups would be necessary to fully represent amino acid diversity and that the optimal number of groups will lie in the 7-11 amino acid region.

Conversely, larger numbers of groups are also not favoured by the optimiser. This suggests that, within the context of automated sequence classification, twenty residues will not necessarily lead to optimal predictive accuracy. However, the prevalence of cysteine as a single grouping does suggest that certain residues display unique properties while others may be more readily paired. This is congruent with data suggesting that the 20 amino acid alphabet is redundant in a structural, if not in a functional, sense (Riddle *et al.*, 1997; Schafmeister *et al.*, 1997; Luthra *et al.*, 2007).

A key question is to what extent this result will hold for other protein data sets, involving very different protein proteins. It is clear that in trying to solve computationally expensive problems such as GPCR Classification there is considerable advantage in generating effective groupings of amino acids. In principle, our proposed optimisation methodology can optimise amino acid groupings for any protein grouping, allowing the customisation of groups so as to maximize predictive accuracy on the specific data being mined, rather than imposing a "one-size-fits-all" grouping of amino acids. It is important to stress that the process is essentially degenerate and that there are several equally effective groupings that could be applied to a specific problem. Equally, the optimised groupings are context dependent and a methodology derived for protein family will not provide the most appropriate groupings for another. We envisage that the nature of optimal groupings will vary from family to family, but to what extent higher order classification - membrane proteins versus globular versus disordered proteins, for example - will exhibit similar or different groupings remains to be seen.

the systems research group at the University of Kent for allowing the use of the pi-cluster of computers, EPSRC grant EP/C516966/1 *TUNA: Theory Underpinning Nanotech Assemblers (Feasibility Study).* An implementation of opt-aiNET in Java was kindly obtained from P. Andrews and modified as described previously.

REFERENCES

Andrews, P. (2005). opt-aiNet source code in Java.

Andrews,P.S. and Timmis,J. (2005). On Diversity and Artificial Immune Systems: Incorporating a Diversity Operator into aiNet. International Workshop on Natural and Artificial Immune Systems (NAIS), Vietri sul Mare, Salerno, Italy.

Attwood,T.K. *et al.* (2002) PRINTS and PRINTS-S shed light on protein ancestry. *Nucleic Acids Res,.* 30, 239-41.

Bissantz,C. (2003) Conformational changes of G protein-coupled receptors during their activation by agonist binding. *J Recept Signal Transduct Res.* 23, 123-153.

Cannata,N, *et al*. (2002) Simplifying amino acid alphabets by means of a branch and bound algorithm and substitution matrices. Bioinformatics. 18, 1102-8.

Chothia,C. and Finkelstein,A.V. *et al.* (1990). The classification and origins of protein folding patterns. *Annu. Rev. Biochem.* 59, 1007-1039.

Christopoulos,A. and Kenakin,T. (2002) G protein-coupled receptor allosterism and complexing. *Pharmacol Rev.* 54, 323-374.

Cui,J. *et al.* (2007) Computer prediction of allergen proteins from sequence-derived protein structural and physicochemical properties. *Mol Immunol.* 44, 514-20.

Davies,M.N., *et al.* (2007a) Proteomic applications of automated GPCR classification. *Proteomics.* 7, 2800-2814.

Davies,M.N., *et al.* (2007b) On the hierarchical classification of G Protein Coupled Receptors. *Bioinformatics.* 23, 3113-3118.

Dayhoff.M.O., *et al.* (1978a) In Dayhoff,M.O. (ed.). *Atlas of Protein Sequence and Structure.* National Biomedical Research Foundation, Washington, DC, 5, 345-352.

de Castro,L.N. and Timmis,J. (2002). Artificial Immune Systems: A New Computational Intelligence Approach, Springer-Verlag.

de Castro,L.N. and Timmis,J. (2002). An artificial immune network for multimodal optimisation. 2002 Congress on Evolutionary Computation (CEC 2002). Part of the 2002 IEEE World Congress on Computational Intelligence, Honolulu, Hawaii, USA, IEEE.

de Castro,L.N and Von Zuben,F. (2001). "Learning and Optimization Using the Clonal Selection Principle." IEEE Transactions on Evolutionary Computation, Special Issue on Artificial Immune Systems 6, 239-251.

de Castro,L.N and Von Zuben,F. (2002). aiNet: An Artificial Immune Network for Data Analysis. Data Mining: A Heuristic Approach. H. Abbass, R. Sarker and C. Newton, Idea Group: p231-259.

Flower,D.R. (1999) Modelling G-Protein-Coupled Receptors for Drug Design. *Biochim. Biophys. Act.,* 1422, 207-234

Flower,D.R. and Attwood,T.K. (2004) Integrative bioinformatics for functional genome annotation: trawling for G protein-coupled receptors. *Semin Cell Dev Biol.* 15,693-701.

Fredriksson, *et al.* (2003). The G protein-coupled receptors in the human genome form five main families. Phylogenetic analysis, paralogon groups, and fingerprints. *Mol. Pharmacol.* 63, 1256–1272.

Gangal,R. and Kumar,K.K. (2007) Reduced alphabet motif methodology for GPCR annotation. *J Biomol Struct Dyn.* 25, 299-310.

Gether,U. *et al.* (2002) Structural basis for activation of G-protein-coupled receptors. *Pharmacol Toxicol.* 91, 304-312.

Hulo,N. *et al.* (2006) The PROSITE database. *Nucleic Acids Res.* 34, D227-30.

Jing,L. and Wei,W. (2007) Grouping of amino acids and recognition of protein structurally conserved regions by reduced alphabets of amino acids. Science in China Series C: Life Sciences. 50, 392-402.

Lee,S.P. (2000). Oligomerization of dopamine and serotonin receptors. *Neuropsychopharmacology.* 23, S32-40.

Li,T. *et al.* (2003) Reduction of protein sequence complexity by residue grouping. *Protein Eng.* 16, 323-30.

López de la Osa,J. *et al.* (2007) Getting specificity from simplicity in putative proteins from the prebiotic earth. *Proc Natl Acad Sci U S A.,* 104,14941-6.

Luthra,A. *et al.* (2007) A method for computing the inter-residue interaction potentials for reduced amino acid alphabet. *J Biosci.,* 32, 883-9.

Matthews,C.N. and Moser,RE. (1967) Peptide synthesis from hydrogen cyanide and water. *Nature*, 215,1230-4.

Melo,F. and Marti-Renom,M.A. *et al.* (2006) Accuracy of sequence alignment and fold assessment using reduced amino acid alphabets. Proteins. 63, 986-95.

Riddle,D.S. *et al.* (1997) Functional rapidly folding proteins from simplified amino acid sequences. *Nat Struct Biol.,* 4, 805-9.

Schafmeister,C.E. *et al.* (1997) A designed four helix bundle protein with native-like structure. *Nat Struct Biol.,* 4,1039-46.

Taylor,W.R. (1986) The classification of amino acid conservation. *J Theor Biol*, 119, 205-218.

Timmis,J. and Edmonds,C. (2004). A Comment on opt-AINet: An Immune Network Algorithm for Optimisation. Genetic and Evolutionary Computation, Springer.

Wang,J., and Wang,W. (1999) A Computational Approach to Simplifying the Protein-Folding Alphabet. *Nature Structural Biology*, 6, 1033-1038.

Witten,I.H. and Frank,E. (2005). Data Mining: Practical Machine Learning Tools and Techniques. San Francisco, Morgan Kaufmann.

Zhang,Z.H. *et al.* (2007) AllerTool: a web server for predicting allergenicity and allergic cross-reactivity in proteins. *Bioinformatics.* 23, 504-6.

Zhang,Z.H. *et al.* (2005). Prediction of protein allergenicity using local description of amino acid sequence. *Bioinformatics.* 23, 504-50

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

Table 4: Matrix of the incidence of paired amino acids within the same group