Geometric Crossover for Supervised Motif Discovery

Rolv Seehuus*
Norwegian University of Science and Technology
Department of Computer and Information Science
7–9 Sem Selandsv
7042 Trondheim
NORWAY
rolv.seehuus@idi.ntnu.no

Alberto Moraglio
University of Essex
Department of Computer Science
Wivenhoe Park
Colshester, CO43SQ
UK
amoragn@essex.ac.uk

ABSTRACT

Motif discovery is a general and important problem in bioinformatics, as motifs often are used to infer biologically important sites in bio-molecular sequences. Many problems in bioinformatics are naturally cast in terms of sequences, and distance measures for sequences derived from edit distance is fundamental in bioinformatics.

Geometric Crossover is a representation-independent definition of crossover based on a distance on the solution space. Using a distance measure that is tailored to the problem at hand allows the design of crossovers that embed problem knowledge in the search.

In this paper we apply this theoretically motivated operator to motif discovery in protein sequences and report encouraging experimental results.

Categories and Subject Descriptors

I.2.8 [**Problem Solving, Control Methods, and Search**]: Miscellaneous

General Terms

Geometric Crossover, Motif Discovery, Biopolymers

1. INTRODUCTION

Motifs, often modeled as short regular expressions, are a way to describe homologous relationships between proteins or functional sites in proteins, DNA and RNA. Prosite is a hand curated database of such motifs foundin proteins, with cross-references to the Swiss-Prot annotated database of proteins [6, 2]. Often such a motif can highlight important characteristic regions of proteins with similar function and common ancestral background. Also, motifs are easy to interpret – a biologist may immediately see the important aspects of a region in protein, RNA or DNA sequences.

The automatic discovery of conserved motifs under different pattern models has received considerable attention during the last two decades. Rigoutsos et al. [14] give a survey of motif discovery algorithms in relation with their own works. Although new motifs for several biological problems are now often presented as statistical models with much resemblance to Hidden Markov Models, the automatic generation of biological motifs remains an active research area. A new motif that can replace the need of the statistical profile is always welcome.

The problem of motif discovery has been studied with more traditional techniques in many years, and there exist good algorithms for data mining, for example Splash [3]. These algorithms work by primarily finding motifs expressed within a positive set by applying measures like support and information content, and setting some fixed parameters on the shape and expected number of non-wildcards in the patterns. For these algorithms search-time is depending on chosen parameters defining the looks of the pattern. They also tend to become expensive when the number of sequences are high – resulting in an increased number of potential patterns to explore.

The mentioned algorithms work well for pattern discovery, but applying Evolutionary Algorithms (EA) potentially allows us to search the pattern space without making any prior assumptions on the compositions of the patterns (other than those implied by the EA operators). Also, an EA or other heuristics might be used to improve motifs found by fast approximate heuristics simply by seeding the population with motifs from such algorithms.

EAs have been applied to motif discovery and related problems at a few occasions. However most such applications do not pay much attention to the underlying structure of the problem. Recently Heddad et al. did some work on protein targeting (which is related to protein motif discovery), using a very general and elaborate grammatical GP system [5]. Other researchers have usually performed some kind of pre-shaping of the training sets. Some strategies for pre-shaping include pre-calculated alignments of positive samples [15], selection of particularly difficult sequences as negative examples [7] or relatively small numbers of randomly generated sequences as negative samples [15].

An inconclusive attempt to compare the performance of lists and trees as representations for motif discovery have been performed by Seehuus et al. [17], where the authors indicated that their experiments seemed to favor a linear genome. Also, a linear genome have been used to discover motifs with better classification performance than those found in the manually curated Prosite database for a couple of families [16, 6].

Geometric Crossovers (GC) are representation- and problem independent recombinative operators for EAs that are defined using a distance measure relating to the solution representation [8]. Informally, a geometric crossover generates offspring that lies on a

*Corresponding author.

GECCO'06, July 8–12, 2006, Seattle, Washington, USA.

shortest path between two parents. The formal definition of geometric crossover can be used to design new crossover operators specific to non-standard representations, by using distance measures meaningful for the specific representation as a guide [10]. Geometric crossover generalizes many existing search operators for the major representations used in EAs, such as binary strings and real vectors [8], permutations [10], syntactic trees [9] and sequences [11]. In particular, for sequences the geometric definition of crossover applied to edit distance becomes a homologous crossover that requires alignment of the contents before exchange of genetic material [13].

To be effective, a crossover operator must embed problem knowledge in the search. This can be achieved by basing the crossover operator on a distance measure that is meaningful for the problem at hand. Moraglio et al. [9] suggested a rule of thumb: the distance chosen should make the resulting fitness landscape "smooth" in some statistical sense, so that low distance between two solutions are correlated with low difference in fitness. Naturally, this is only a general rule-of-thumb that needs to be validated on specific problems. In this paper we test this hypothesis on a motif discovery problem, by discovering motifs to discriminate between protein families. We model our motifs as fixed length sequences, and assume that a mutation based on the edit distance move previously used by Seehuus et al. [17] is a good mutation.

In section 2 we outline the geometric framework for crossover operators, and define a homologous crossover for edit distance. Section 3 describes the motif discovery problem and experimental setting, and section 4 presents results from experiments and a discussion of these. Finally we give some concluding remarks and outlines of future directions in section 5.

2. GEOMETRIC CROSSOVER FOR BIOINFORMATICS PROBLEMS

In this section, we report the basic ideas behind the geometric framework, then we describe the specific case of edit distances, sequences and homologous crossover.

2.1 Geometric Preliminaries

Formally, the term *metric* denotes any function that conforms to the axioms of identity, symmetry and triangular inequality. A simple connected graph is naturally associated to a metric space via its *path metric*: the distance between two nodes in the graph is the length of a shortest path between the nodes. Similarly, an edge-weighted graph with strictly positive weights is naturally associated to a metric space via a *weighted path metric*.

In a metric space (S, d), a *closed ball* is the set of the form $B(x; r) = \{y \in S \mid d(x, y) \leq r\}$ where $x \in S$ and r is a positive real number called the radius of the ball. A *line segment* (or closed interval) is the set of the form $[x; y] = \{z \in S \mid d(x, z)+d(z, y) = d(x, y)\}$ where $x, y \in S$ are called extremes of the segment. Metric ball and metric segment generalize the familiar notions of ball and segment in the Euclidean space to any metric space through distance redefinition. These generalized objects look quite different under different metrics. Notice that a metric segment does not coincide to a shortest path connecting its extremes (*geodesic*) as in an Euclidean space. In general, there may be more than one geodesic connecting two extremes; the metric segment is the union of all geodesics.

We assign a structure to the solution *set* by endowing it with a notion of distance d. $M = (S, d)$ is therefore a solution *space* and $L = (M, g)$ is the corresponding fitness landscape, where g is the fitness function over S. Notice that d is arbitrary and need not have any particular connection or affinity with the problem at hand.

2.2 Definition of geometric operators

The following definitions of mutation and crossover are *representation-independent* and are *functions of the metric* associated with the search space being based on the notions of metric ball and metric segment.

DEFINITION 1. *Geometric ϵ-mutation*
A unary operator is a geometric ϵ-mutation operator under the metric d if ϵ is the smallest radius of the ball centered in the parent so to include all offspring.

DEFINITION 2. *Geometric crossover*
A binary operator is a geometric crossover under the metric d if all offspring are in the segment between its parents.

DEFINITION 3. *Uniform geometric crossover*
The uniform geometric crossover UX under the metric d is the geometric crossover under d that uniformly picks offspring in the segments between the parents.

2.3 Geometric crossover, fitness landscape and problem knowledge

Geometric operators are defined in terms of the distance measure associated with the search space. However, the search space, e.g. the set of possible solutions to the problem at hand and the connectivity structure among them, does not come with the problem itself. The problem consists of the objective function to minimize (or maximize), and a description of how to evaluate candidate solutions. It does not give any structure on the solution set.

Creating a structure over the solution set is part of the design of the search algorithm, and it is a designer's choice. The designer could, in principle, pick any structure over the solution set, but choosing a structure randomly would decouple the problem from the structure of the search space. Consequently the search operators defined over such a structure would turn the search into something very close to random search.

In order to embed problem knowledge in the search, the connectivity structure of the fitness landscape has to be chosen well, and the resulting distance associated with the search space should be "natural" for the problem at hand. The designer can put problem knowledge in the landscape by studying the structure of the objective function of the problem and choose a neighborhood structure such that specific conditions that "couple" distance between solutions and their fitness values are satisfied. Once this is done problem knowledge can be exploited by search operators to perform better than random search, *even if the search operators are problem-independent*. The fitness landscape built in this manner can be viewed as a knowledge interface between the problem and a problem-independent search algorithm.

As previously mentioned, we expect geometric operators to perform well on landscapes with a "smooth trend". Landscape smoothness, in various forms, is a principle that has been discovered many times in different context and has been confirmed in many empirical studies with many neighborhood search meta-heuristics [12].

A smooth landscape can be obtained indirectly, without having specific knowledge on the fitness landscape by using the following well-established operational practice: if one has a good mutation operator, and can build a crossover based on this, the resulting crossover is likely to be good [9]. This is because the mutation operator creates a neighborhood structure over the search space. If a particular mutation works well on a problem, the associated fitness landscape is likely to be smooth. Thus, if we build a crossover which uses the structure associated with such a mutation, we are

likely to obtain a good crossover because of the "smooth landscape" of the mutation operator. Since geometric crossovers can be constructed in such a way, the geometric crossover associated with a good mutation might be expected to be good.

2.4 Sequences, edit distance and alignments

The *edit distance* between two sequences is defined as the minimum number of edit moves (insertion, deletion and replacement) needed to transform one sequence in to the other. Edit distances are metrics [4]. The edit distance between two sequences is a measure of their syntactic dissimilarity.

An *alignment* of two sequences u and v is the pair of sequences u' and v' of the same length obtained by inserting the minimum number of place-holder characters "-" in u and v so as the place-holder display necessary insertions and deletions in the transformation under edit distance. If $u =$ "banana" and $v =$ "ananas" a possible alignment is the pair of sequences $u' =$ "banana-" and $v' =$ "-ananas". The *score* of an alignment is the number of aligned characters that are different in the two sequences. Given two sequences there may be more than one optimal alignment. It is well known that the score of an optimal alignment between two sequences equals their edit distance [4].

2.5 Edit-mutation and its homologous crossover

The edit distance mutation is defined as a random edit move, at a random position in the sequence. First a random position in the sequence is chosen uniformly, then the character at the selected position is either deleted, replaced by a random character from the alphabet, or a random character is inserted at this position of the string.

DEFINITION 4 (MASK-BASED HOMOLOGOUS CROSSOVER). *homologous crossover follows the following steps:*

1. *Let Q be the set of all optimal alignments of two sequences u and v under simple edit distance. Homologous crossover picks a random optimal alignment $q \in Q$ with a given probability distribution over Q. Let u' and v' be the two sequences aligned with gaps according to q.*
2. *Let l be the length of q and m be a mask drawn from $\{0, 1\}$ with a given probability distribution. m specifies for each position of q from which parent to copy the corresponding character to produce an aligned offspring w'*
3. *The actual offspring w is obtained by removing the dashes from w'.*

An example of a one point crossover is:

EXAMPLE 1.

m	`0000111`
u'	`banana-`
v'	`-ananas`

The given mask and alignment yields the child `bananas`, and for $\bar{m}$, `anana`.

An optimal alignment, and thus a homologous crossover, can be computed efficiently using a $O(|u||v|)$ dynamic programming algorithm [4].

THEOREM 1. *All mask-based homologous crossover operators are geometric crossovers under edit distance.*

THEOREM 2. *Every string w in the segment between two strings u and v under edit distance is reachable by mask-based homologous crossover applied to the parent sequences u and v.*

Theorem 1 states that homologous crossover is geometric crossover under edit distance. Theorem 2 states that any geometric crossover under edit distance takes the form of homologous crossover. Proofs for the theorems are given by Moraglio et al. [11].

3. MOTIF DISCOVERY PROBLEM AND EXPERIMENTAL SETTINGS

In this section we describe the motif discovery problem, selected datasets and experiments performed.

3.1 Correlation was used as fitness

We model the motif discovery problem as a supervised learning problem, with two classes of data – one positive (the in family sample) and one negative (the rest of the database). The goal is to find a motif that is a good discriminator between the positive and negative dataset.

When a candidate motif is found in a protein string this is counted as a true positive, if the protein sequence belongs to the current family. Otherwise it is counted as a false positive. We have p positive samples and n negative samples. Tp denotes the true positives and Fp the false positives. Then $Tn = n - Fp$ is the number of true negatives and $Fn = p - Tp$ is the false negatives.

It is important that our fitness measure is constructed in such a way that high fitness represents high classification quality. The failure-rate or accuracy of the classifier will not suffice, as the size of the positive set is a small fragment of the negative set (one fifth during training, and often less than a percent in the test-set.) Any hypothesis that don't match any of the negative samples will have a very high accuracy on this problem. The Matthews correlation coefficient is a measure that captures the classifiers ability to separate the datasets, and its use as a fitness measure was inspired by other works [7]. The fitness is then calculated according to equation 1

$$C = \frac{Tp * Tn - Fn * Fp}{\sqrt{(Tn + Fn)(Tn + Fp)(Tp + Fn)(Tp + Fp)}} \quad (1)$$

3.2 Simple regular expressions model patterns

We represented motifs as sequences. As alphabet, we used a simple pattern model, that models *identical* patterns from the set $(\Sigma \cup .)*$, where Σ is the twenty amino acid residues and . denotes a match with any character (a wildcard.) [1]

3.3 Selected problems was previously found to be hard

For our experiments, we selected six protein families from the Prosite database described in table 1. These were selected amongst a set of 44 protein families used in a previously published article, and we selected six families that seemed to be problematic for GP [16]. The six different problems all have more than 200 proteins in-family, and they also have imperfect models in Prosite[2] [6].

3.4 Experimental setup was a ten-fold cross validation on each family

For each of the six selected protein families, we labeled all proteins in the family as positive and all other proteins in the Swiss-Prot database as negative. For most families there existed proteins that were labeled as partial members of the families. These proteins were excluded from the study.

[1]These patterns are from the same pattern space as explored by the Teiresias algorithm, as discussed earlier in this paper.

[2]Values taken from Prosite Release 19.14.

Table 1: Protein families and the number of proteins in them

Accession Number	Name	In family
PS00097	Carbamoyl Transferase	406
PS50076	Dna J	510
PS51007	Cytochrome C	307
PS00198	4Fe-4S-Ferredoxin	462
PS50888	HelixLoopHelix	395
PS50222	EF Hand	919

For each of the families we ran a 10-fold cross-validation experiment for two different crossover operators and mutation only, totalling 180 experiments. The first crossover operator tested was a two-point geometric crossover based on the homologous crossover defined in section 2.5 (TPG), which is analogous to the standard two-point crossovers for binary GAs. The other was the previously used two-point crossover (TP) operator, which is computed by swapping random substrings within the genomes. This allows us to measure the quality of classifiers evolved by the two different operators on the entire Swiss-Prot database for these data.

We used a generational EA for the experiments, where the entire population was changed for each iteration. Except from changing the type of crossover (and turning it off entirely in the mutation-only experiments), we used the same settings through all experiments. We used tournament selection with 4 individuals in the tournament. 70% of the population was created using the crossover operator, and the rest was created by mutation. No mutation was performed on individuals produced from crossover. The mutation operator based on the edit-move was used with both crossovers. Each experiment was allowed to run for 50 generations, with no other stopping criteria.

The initial population was initialized by generating random patterns of length 5, where each position was selected from the alphabet including the wildcard symbol. We might speculate that we would have seen better performance if we had used random substrings or patterns derived from substrings that existed in the positive set, but we did not pursue this.

3.5 Suffix trees accellerated searching

To accelerate the evaluation of the population we used generalized suffix trees, and approximate search in these [4]. Some trial measurements showed that for random queries of length ten with the specified model, counting hits in suffix trees was faster than the boost regular expression library [3] when the number of queries was larger than two hundred (data not shown.)

Building a suffix tree of the entire set of negatives (consisting of close to 160000 different protein sequences) to test the motifs was not feasible. Instead, for each generation, we extracted a small sample of the negative data four times larger than the positive set, and built a suffix tree from these sequences (this is suggested in, for example, Banzhaf et al. [1])

4. RESULTS AND DISCUSSIONS

4.1 Homologous crossover seem better during training

Figure 1 shows the median of the maximum values found during training, for each generation. Figure 2 shows the population averages over all ten runs and all six families.

Some interesting differences in behavior is shown by figure 2. Although the maximum fitness always seem to be comparable in values, for most of the runs the population means shows different behavior. In figure 2 TPG always has much higher population average than TP.

Even though the medians of the maximum values on the training data are often higher for the runs using TPG, note that sometimes the minimum value for the TP runs is as good as or better than the minimum value for TPG. Also the quantiles are overlapping much, so no conclusions can be made on expected performance differences of the two operators.

4.2 TP found a biologically significant pattern for the Ferredoxin-family

When studying the final results in table 2, it is clear that we are unable to conclude if one operator is better than the other as the classification performance of the motifs found are similar over all six families over all ten runs.

Also, an investigation of the discovered motifs for the experiments on the ferredoxin family, reveals that the motifs with best generalizing performance was found by TP in six of the ten runs, with perfect classification on training and correlation 0.86 on the test set : `C..C..C...C` In comparison, notice the similarities with the 4Fe-4S-Ferredoxin motif from Prosite database: `C.{P}C..C{CP}..C[PEG]` In light of our simplified pattern model, this result could be considered optimal.

The length of the motif and the fact that it is a good classifier is an indicator to the biological significance of the motif, and comparing it to the Prosite motif for the same family, show that biologists agree on the main features of the motif. In comparison the TPG was only able to discover fragments of this motif in the form of a suffix of the motif, `C...C` on nine out of ten runs, and the mutation operator discover similar motifs (although a longer suffix is discovered more often.) This might be attributed simply to that if we were choosing motifs `C..P` and `C..C` as parents, the space of possible offsprings from TP would contain e.g. `C..C..C` as a possible child and it is not very unlikely either. On the other hand, it is impossible to create such a solution from the two given parents with TPG.

It is noticeable that edit mutation alone have stable high performance, and often surpasses the results from the other two algorithms on the test set (see table 2).

4.3 Overfitting is accredited to unseen negative samples

When comparing the results on training and testing given in table 2, it seems that we experience substantial overfitting for all but the results for TP on the Ferredoxin-family.

Overfitting is often attributed to learning "noise" in the data, due to the complexity, or flexibility, of the model. Considering our very simple models, this might be surprising. To get a good explanation to how this might have happened in our experiments, we need to look at our selection of negative training samples.

During evolution we only looked at a fraction of the negative samples. For a protein family with 200 different protein sequences, 50 times 4 times 200 was extracted from Swiss-Prot, which yields maximum 40 000 proteins, or less than 25% of the negative data.

In this case it might be likely that motifs occurring in many positive samples also occur in the unseen data in the Swiss-Prot database - and therefore contributing to the observed overfitting.

5. CONCLUDING REMARKS

In this paper we used the geometric framework to motivate the general hypothesis that homologous crossover and edit mutation are likely to be good genetic operators for bioinformatical problems because such problems are naturally cast in terms of sequences and

[3]Available from `http://www.boost.org/`

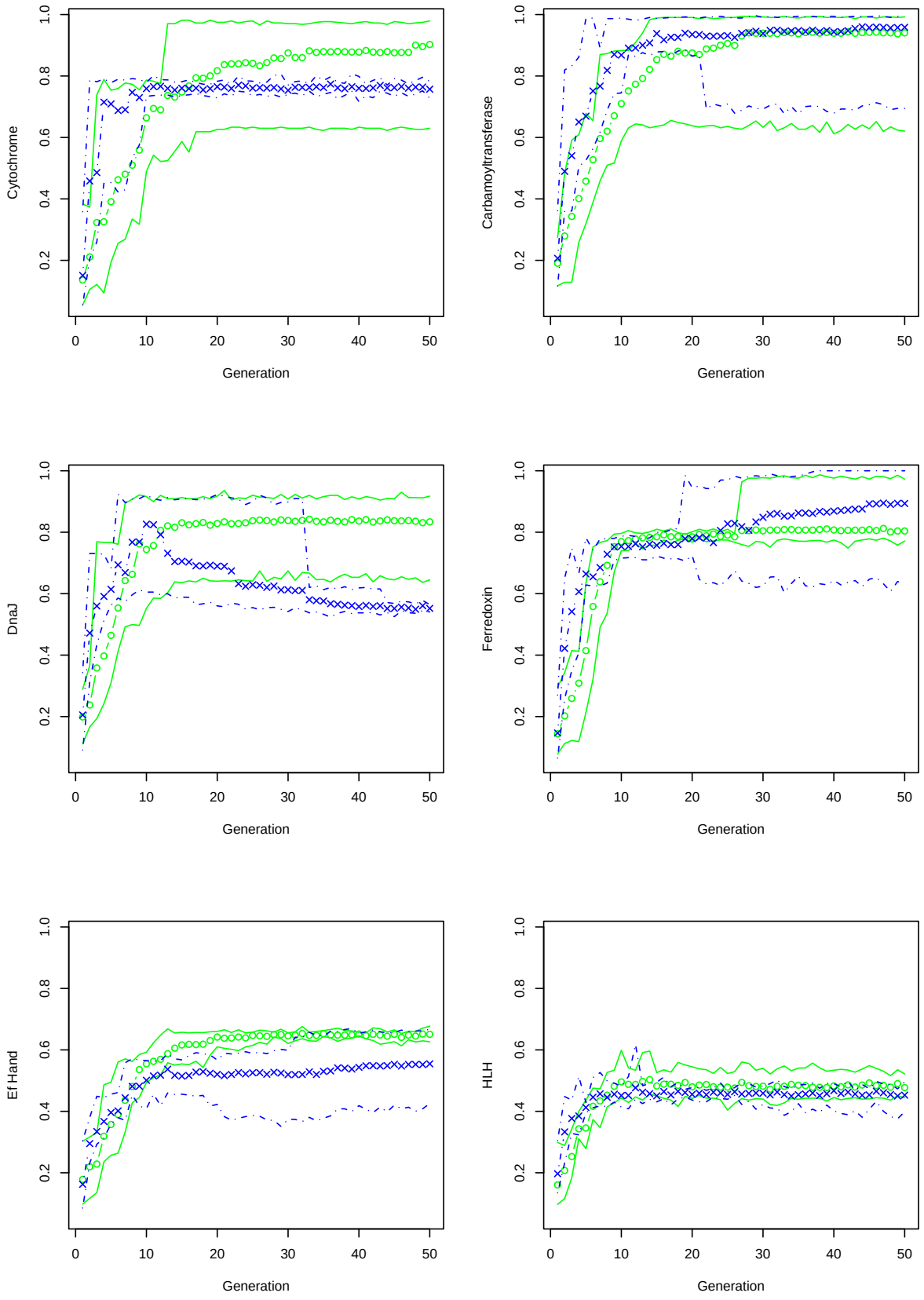

Figure 1: Maximum values. Medians plotted for 10 independent runs, with 0 and 100 percent sample quantiles. Circles are results for TPG, crosses are results for TP. Quantiles for TPG are plotted with a solid line, TPs with dashes.

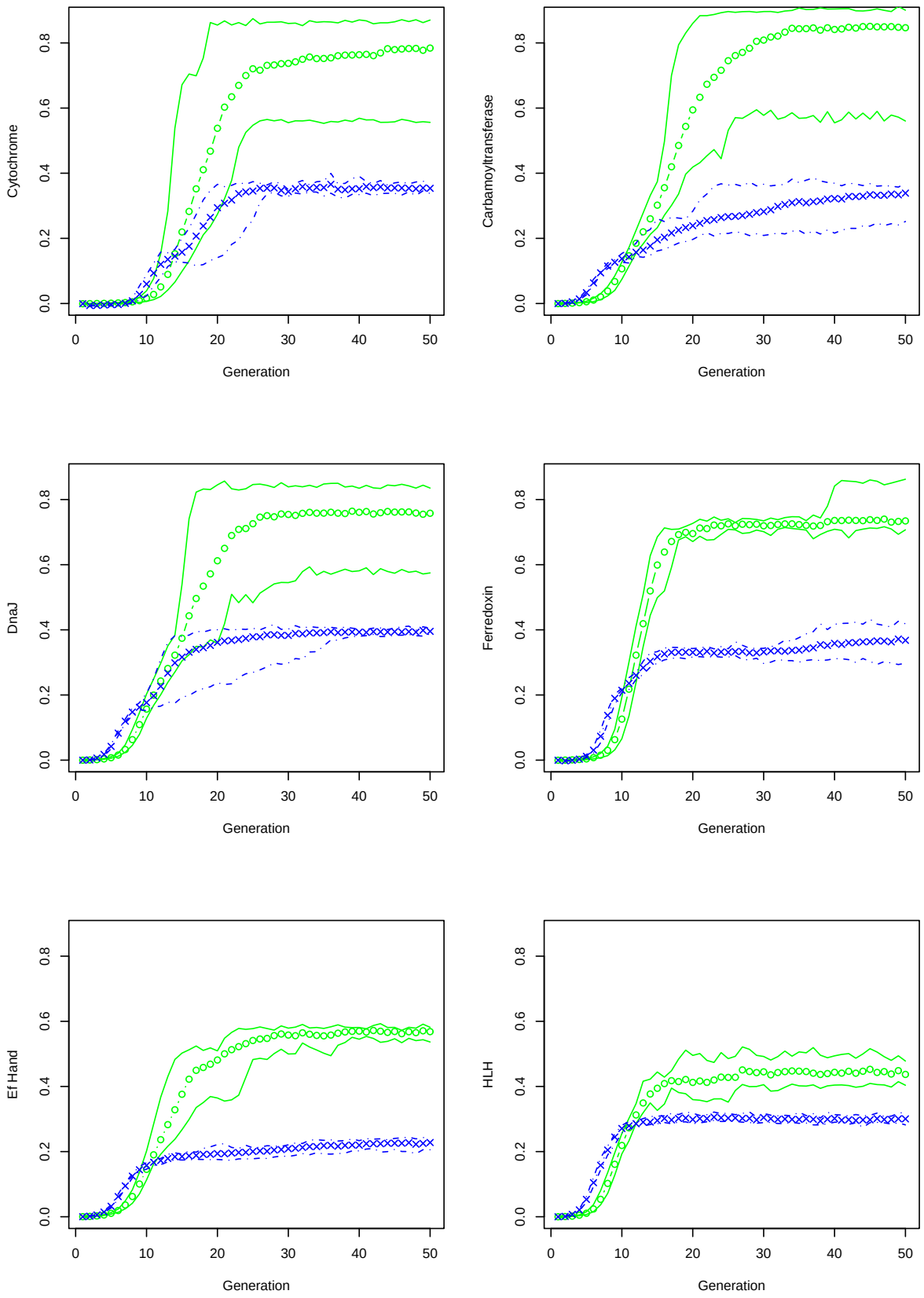

Figure 2: Average population fitness. Medians with 0 and 100 percent sample quantiles. Circles are results for TPG, crosses are results for TP. Quantiles for TPG are plotted with a solid line, TPs with dashes.

Table 2: Measured correlation coefficients on training sets and test sets for evolved classifiers found by TP, TPG and edit-mutation only.

		TP		TPG		Mut	
Family	Value	Training	Testing	Training	Testing	Training	Testing
Carbamoyl	Max	0.996	0.65	0.99	0.60	0.99	0.65
	Median	0.99	0.57	0.99	0.56	0.99	0.56
	Min	0.94	0.28	0.65	0.11	0.95	0.28
Dna J	Max	0.92	0.24	0.93	0.59	0.93	0.41
	Median	0.90	0.23	0.91	0.23	0.92	0.23
	Min	0.62	0.11	0.67	0.15	0.65	0.15
Cytochrome	Max	0.81	0.11	0.98	0.42	0.98	0.42
	Median	0.80	0.10	0.98	0.36	0.97	0.36
	Min	0.79	0.10	0.63	0.33	0.59	0.10
Ferredoxin	Max	1.0	0.91	0.99	0.42	0.99	0.44
	Median	0.998	0.87	0.81	0.13	0.96	0.39
	Min	0.74	0.11	0.79	0.13	0.80	0.13
HLH	Max	0.62	0.11	0.60	0.10	0.67	0.41
	Median	0.50	0.07	0.52	0.06	0.59	0.08
	Min	0.48	0.05	0.49	0.04	0.48	0.05
EF Hand	Max	0.67	0.19	0.68	0.19	0.67	0.28
	Median	0.62	0.15	0.67	0.19	0.66	0.19
	Min	0.48	0.13	0.65	0.15	0.48	0.13

homologous crossover and edit mutation are based on edit distance between sequences that is fundamental in bioinformatics.

We tested this hypothesis on the specific case of motif discovery problem. As a reference, we compared TPG, a homologous crossover that aligns the contents before segments are exchanged, with TP, a non-homologous crossover that randomly exchange segments. We found TPG to be slightly better than the non-homologous crossover during training, but no general conclusions can be made from the performance on the test sets. Interestingly, experiments show that TPG is a much less disruptive operator than TP, and this opens up the possibility to seed the initial population with good candidate patterns computed with some quick heuristics as raw material for the crossover.

Surprisingly we found that edit mutation seems to perform better alone than when coupled with crossovers, both homologous and non-homologous. This corroborates our initial assumption that the fitness landscape of motifs under edit distance has a "smooth" trend and that edit distance is a meaningful space for motif discovery. Second, it hints that our rule-of-thumb, that a "smooth" fitness landscape leads to a successful geometric crossover, somehow does not fully apply in this case. The homologous crossover associated with the same fitness landscape as edit mutation works well, but not as well as edit mutation alone. This does not directly contradict the "good mutation, good crossover" hypothesis, because homologous crossover still is good. However, it begs an explanation. We will in later work explore homologous crossover and edit mutation on other bio-informatical problems to see whether this is an isolated anomaly or a more general phenomenon.

Even though we note that our two operators find similar patterns and overfits, the geometric crossover have some properties that makes it a better choice, especially the indications of reduced destructiveness (consistently high population average) and the possibility to seed the population with candidate patterns. A selection scheme that discovers and highlight the difficult samples, might reduce the overfitting observed during training.

As we found a motif that might be seen as biologically meaningful for only one of the families, we might also speculate that our fitness measure might not be optimal for discovery of biologically relevant sites.

6. REFERENCES

[1] Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, and Frank D. Francone. *Genetic Programming – An Introduction; On the Automatic Evolution of Computer Programs and its Applications*. Morgan Kaufmann, dpunkt.verlag, January 1998.

[2] Brigitte Boeckmann, Amos Bairoch, Rolf Apweiler, Marie-Claude Blatter, Anne Estreicher, Elisabeth Gasteiger, Maria J. Martin, Karine Michoud, Claire O'Donovan, Isabelle Phan, Sandrine Pilbout, and Michel Schneider. The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003. *Nucl. Acids Res.*, 31(1):365–370, 2003.

[3] Andrea Califano. Splash: structural pattern localization analysis by sequential histograms. *Bioinformatics*, 16(4):341–357, 2000.

[4] Dan Gusfield. *Algorithms on Strings, Trees and Sequences*. Cambridge University Press, 1997.

[5] Amine Heddad, Markus Brameier, and Robert M. MacCallum. Evolving regular expression-based sequence classifiers for protein nuclear localisation. In Guenther R. Raidl, Stefano Cagnoni, Jurgen Branke, David W. Corne, Rolf Drechsler, Yaochu Jin, Colin Johnson, Penousal Machado, Elena Marchiori, Franz Rothlauf, George D. Smith, and Giovanni Squillero, editors, *Applications of Evolutionary Computing, EvoWorkshops2004: EvoBIO, EvoCOMNET, EvoHOT, EvoIASP, EvoMUSART, EvoSTOC*, volume 3005 of *LNCS*, pages 31–40, Coimbra, Portugal, 5-7 April 2004. Springer Verlag.

[6] Nicolas Hulo, Christian J. A. Sigrist, Virginie Le Saux, Petra S. Langendijk-Genevaux, Lorenza Bordoli, Alexandre Gattiker, Edouard De Castro, Philipp Bucher, and Amos Bairoch. Recent improvements to the PROSITE database. *Nucl. Acids Res.*, 32(90001):D134–137, 2004.

[7] John R. Koza and David Andre. Automatic discovery using genetic programming of an unknown-sized detector of protein motifs containing repeatedly-used subexpressions. In

Justinian P. Rosca, editor, *Proceedings of the Workshop on Genetic Programming: From Theory to Real-World Applications*, pages 89–97, Tahoe City, California, USA, 9 July 1995.

[8] A. Moraglio and R. Poli. Topological interpretation of crossover. In *GECCO*, pages 1377–1388, 2004.

[9] A. Moraglio and R. Poli. Geometric crossover for the permutation representation. *Technical Report CSM-429*, 2005.

[10] A. Moraglio and R. Poli. Topological crossover for the permutation representation. In *GECCO 2005 Workshop on Theory of Representations*, 2005.

[11] A. Moraglio, R. Poli, and R. Seehuus. Geometric crossover for biological sequences. In *EUROGP*, 2006.

[12] M. G. C. Resende P. M. Pardalos, editor. *Handbook of Applied Optimization*. Oxford University Press, 2002.

[13] Michael Defoin Platel, Manuel Clergue, and Philippe Collard. Maximum homologous crossover for linear genetic programming. In *Genetic Programming: 6th European Conference*, Lecture Notes in Computer Science, pages 194–203. Springer-Verlag GmbH, 2003.

[14] I. Rigoutsos, A. Floratos, L. Parida, Y. Gao, and D. P latt. The emergence of pattern discovery techniques in computational biolog. *Metabolic Engineering*, 2:159–177, 2000.

[15] Brian J. Ross. The evolution of stochastic regular motifs for protein sequences. *New Generation Computing*, 20(2):187–213, February 2002.

[16] Rolv Seehuus. Protein motif discovery with linear genetic programming. In *KES (3)*, pages 770–776, 2005.

[17] Rolv Seehuus, Amund Tveit, and Ole Edsberg. Discovering biological motifs with genetic programming. In Hans-Georg Beyer, Una-May O'Reilly, Dirk V. Arnold, Wolfgang Banzhaf, Christian Blum, Eric W. Bonabeau, Erick Cantu-Paz, Dipankar Dasgupta, Kalyanmoy Deb, James A. Foster, Edwin D. de Jong, Hod Lipson, Xavier Llora, Spiros Mancoridis, Martin Pelikan, Guenther R. Raidl, Terence Soule, Andy M. Tyrrell, Jean-Paul Watson, and Eckart Zitzler, editors, *GECCO 2005: Proceedings of the 2005 conference on Genetic and evolutionary computation*, volume 1, pages 401–408, Washington DC, USA, 25-29 June 2005. ACM Press.