A Comparison of Evolved Finite State Classifiers and Interpolated Markov Models for Improving PCR Primer Design

Daniel A. Ashlock[1] Scott J. Emrich[2] Kenneth M. Bryden[3] Steve M. Corns[3] Tsui-Jung Wen[4] Patrick S. Schnable[4]

Abstract-**This presents results on training both finite state classifiers and interpolated Markov models as classifiers for polymerase chain reaction primers. The goal of the study is to find techniques to decrease the number of primers that fail to amplify correctly within a large genomics project. Standard primer design packages already select primers in a manner consistent with current knowledge of the biophysics of DNA. The classifiers trained in this effort are used to capture lab and organism specific features of primer data and are used to post-process the output of standard primer design packages. The finite state classifiers in this study are trained with a novel evolutionary algorithm that uses an incremental fitness reward system and multi-population hybridization. This hybridization is akin to population seeding, not the more usual hybridization of evolutionary computation with other techniques. The interpolated Markov model is a form of Markov model that adapts to data rich and data sparse portions of the training set by using a variable order in its modeling. The interpolated Markov models exhibited slightly superior performance and trains with far higher speed. The finite state classifiers provide a substantially different classification, however, and require less training data.**

I. Introduction

The design of polymerase chain reaction (PCR) primers is a well-studied problem when considered in terms of the biophysics of DNA. Software for selecting primers that amplify known DNA sequences is already available. This study uses the Primer 3 package provided in NCSA's biology workbench. Designing primers, however, that not only have good biophysical properties but also take into account idiosyncratic features of particular organisms, labs, and technicians is a more difficult problem.

Machine learning can be used to improve future overall efficiency of these biochemical reactions in a long-term project in which hundreds or thousands of PCR primers are designed for a single organism. Although the training data for this work were drawn from an ongoing maize (*Zea mays* L., commonly called corn in the U.S.) genetic mapping project, we note that it can be of equal importance in genome finishing, and annotation within mammalian [19] and "gene-enriched" plant [6], [17] genome projects using reverse transcription-polymerase chain reactions (RT-PCR). An inexpensive alternative to multiplexed technology like microarrays, RT-PCR provides a quick method for experimentally validating subsets of *ab initio* gene predictions—especially those with no homology to other known sequences—found within complete and partial genomic sequences.

This study presents and compares two different methods for creating project-specific post-processing tools for primer selection. We demonstrate that these can partially compensate for organism and/or lab specific factors affecting the success of PCR primer amplification. Effective use of these tools requires that sufficient primers have been experimentally validated on a desired organism in order to provide the necessary training data. With such a set of primers available, an evolutionary algorithm uses the success and failure data to train finite state classifiers (FSCs) that predict if a given primer pair will or will not succeed in amplification. Likewise, training data may be used to estimate transition probabilities for an interpolated Markov model.

The training data available for an effort of this kind are noisy for several reasons. Perfectly good primers are sometimes scored as bad primers because of errors made by the technician or reagent suppliers. Since primers work in pairs, a good primer may end up being scored as a bad primer because of flaws in its partner. Finally, the process of scoring primers into the categories "worked" and "did not work" is itself at least slightly subjective. The reader is assumed to be familiar with the process of PCR amplification, which is explained in some detail in [11].

A. The Problem

If PCR primer design is performed with state-of-the-art general purpose primer-selection software then what remains to be done? A key point is that a selected primer must not bind to more than one location. If it does, then unintended DNA may be amplified. Since most PCR attempts start with organismal samples, which include the entire genome of that organism, many potential binding locations exist for each primer sequence. The lengths of PCR primers are chosen such that it would be a shocking statistical anomaly if it occurred more than once at random. Evolution, however, functions in many cases by duplicating genetic material and then modifying

[1]Mathematics Department, Iowa State University, Ames, Iowa 50011
[2]Bioinformatics Program, Iowa State University, Ames, Iowa 50011
[3]Mechanical Engineering, Iowa State University, Ames, Iowa, 50011
[4]Agronomy Department,Iowa State University, Ames, Iowa, 50011

0-7803-8728-7/04/$20.00 ©2004 IEEE

one copy. Because of this phenomenon, many sequences of arbitrary lengths will be found multiple times in a genome. Unless a primer design program has access to the entire genome sequence of an organism it cannot compensate for this source of bad PCR reactions. Maize does not yet have a fully sequenced genome and it also contains a large number of transposable elements. Sometimes called "jumping genes", these DNA sequences function as a kind of genetic parasite by copying themselves throughout the genome, and make the problem of long, repeated sequences a serious concern. With over 27k primer pairs available, machine learning can be used to detect such confounding sequence redundancy along with and other maize or lab specific factors that prevent selected primers from functioning nominally.

It is important to note that the approach of synthesizing multiple primers for each target so as to ensure success is not an optimal approach for some projects. The project to improve the genetic map of corn that formed to context of this research had far more potential targets than there was resources for primer synthesis. The number of target sequences mapped and service to the biological community are maximized by enhancing the number of correct primers in the first pass of primer design.

II. Finite State Classifiers

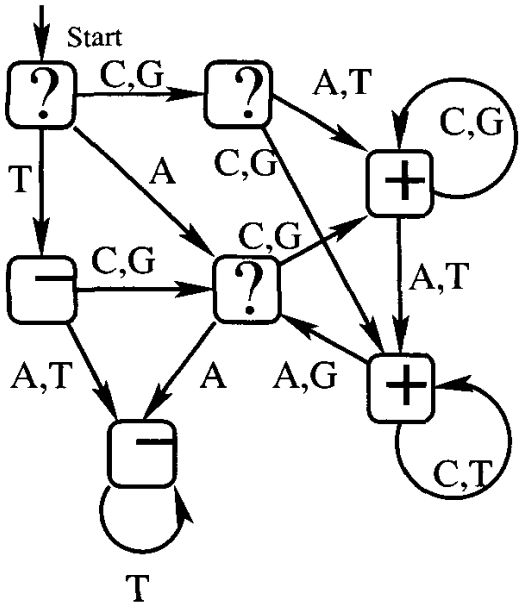


Fig. 1. **Part of a finite state classifier of the type used in this study to learn patterns in DNA. The classifier shown above has its transitions driven by nucleotides and has the following states: ? (don't know), + (good primer), and - (bad primer).**

Two key features of any evolutionary-computation-based machine learning system are the data structure holding the *putative problem solutions* to be evolved and the fitness function. Our data structure is a finite state classifier with 32, 64, or 128 states. Transitions in these classifiers are driven by DNA bases, or nucleotides. Finite state machines are a standard representation for diverse tasks in evolutionary computation [4], [7], [8], [13], [15], and Figure 1 shows a portion of such a machine. The states of the FSC have three possible types: ? (don't know), + (good primer), and - (bad primer). These state labels permit the finite state machine to function as a classifier, though not in the usual manner, as we will see when the fitness function is specified below.

For training a population of FSCs we used a collection of 27408 PCR primer pairs, 17224 of which amplified correctly and 10184 of which either failed to amplify at all or apparently amplified multiple targets. Subsets of 500 good and bad primers (1000 total) are reserved for cross validation of the classifiers. The fitness of an FSC on a training set of primers is computed as follows. Two thousand each of good and bad primers are selected at random from the available primer training data outside of the cross validation set. This selection is performed again in each generation of the evolutionary algorithm. Each member of this set of primers is then run through the FSC. As the classifier passes through each state values from Table I are summed. These numbers represent complete neutrality to all factors except "it works" and "it doesn't work." Fitness for an FSC is then summed over the 2000 good and 2000 bad primers being used for fitness evaluation in a given generation.

TABLE I
Incremental payoffs per state for FSCs.

Primer is	State +	-	?
Good	1	-1	0
Bad	-1	1	0

This fitness function rewards the FSC incrementally after each state transition. Scoring FSCs on their accuracy of their final state and no other was attempted in a preliminary study and worked badly. Classifiers that refused to classify most primers (in which states labeled with ? formed a majority) were a common outcome. Our hypothesis is that the incremental reward acts to smooth the fitness landscape, in essence simplifying the evolutionary search problem. In addition, the incremental fitness permits the FSCs to be indecisive. Large positive scores indicate a primer that the FSC classifies as quite likely to work correctly, while large negative scores are votes of no confidence in a primer. Scores near zero, however, indicate either ignorance (the primer is of a type not encountered before) or confusion (primers with good and bad sequence features). For selecting new primers, we chose the highest scoring pair that both returned positive scores.

III. Interpolated Markov Models

Markov models are useful statistical sequence models typically used in the field of computational biology for gene prediction[5], although equivalent constructs have been used for protein and sequence classification. In any fixed-order Markov model, the probability of the next character is dependent only on a constant number of preceding characters. For example, in a 1st order model, a specific sequence x of length n has probability:

$$P(x) = P(x_1)\prod_{i=2}^{n}(x_i|x_{i-1})$$

Thus, given any arbitrary Markov model and a set of initial probabilities, we can easily calculate the probability of observing an any sequence using a Markov chain as shown above. The power of such an approach is that we can also construct a null-hypothesis model to assess significance using a log-likelihood ratio that follows a χ^2 distribution.

Preliminary results using fixed order Markov chains for primer validation were marginally successful using 5th, 6th, and even 7th order models on limited training sets used to estimate the conditional probabilities. Higher order Markov models, however, are better suited for sequence discrimination but require extensive training data since model parameters grow exponentially. Interpolated Markov Models (IMMs)—successfully used in bacterial gene prediction [5]—effectively remove this limitation by utilizing multiple fixed-order models via linear combination in order to take full advantage all oligonucleotide information available. This "interpolation" makes IMMs theoretically more powerful than fixed-order models since they are able to utilize higher order models when sufficient training exists. Such an approach should therefore be able to detect specific features without over-fitting other parameters of the model.

IV. Experimental Design

A population of 600 FSCs evolved for 1000 generations were used when training classifiers. The FSCs are initialized uniformly at random, filling in both transitions and state labels at random. The model of evolution is single tournament selection with tournament size four. The population is randomly shuffled into groups of four and the two most fit FSCs reproduce and replace the two least fit in each group. Reproduction treats the string of states in an FSC as a linear chromosome. The two FSCs reproducing are copied, the copies undergo two-point crossover, and then each copy is subjected to one mutation. The mutation modifies the initial state of the FSC 10% of the time, randomly picks a new destination for one of the transitions 30% of the time, and modifies the label $\{+,-,?\}$ on a state 60% of the time. During crossover the initial state of the FSC moves with the first state. One hundred simulations with distinct starting populations were performed, saving the best FSC from each simulation.

The best-of-simulation FSCs are used to initialize additional sets of simulations which we term *genetic hybridizations*. For other instances of this type of hybridization in the context of evolutionary computation see [1], [2]. These simulations are identical to the first set except that 100 members of the initial random population are replaced with the best-of-run FSCs from the first 100 simulations. The other members of these initial populations are still generated uniformly at random. Genetic hybridization, the cross breeding of elite genes from multiple populations, is distinct from the hybridization of evolutionary computation with other techniques. This latter practice is also called hybridization and we diffidently suggest that it be named *algorithmic hybridization* to distinguish it from the form of radical population seeding defined here.

Our alternative statistical method for assessing primers is based on the following idea: if there are subtle differences in nucleotide composition between good and bad primers, we should be able to score these differences using a likelihood ratio calculated from the results of separate IMMs. Training currently uses the *build-icm* program of the GLIMMER2 package to build statistical models of each primer class. The associated IMM scoring functions are then used to obtain the log-likelihood ratio score for a specific primer, given these two previously trained models. This slightly modified version of the GLIMMER2 core has been dubbed prIMMer within our primer selection pipeline.

Recently funded maize genome sequencing projects may also offer a glimpse of the genomic features being modeled using these machine learning approaches. Since available Bacterial Artifical Chromosome (BAC) end sequences represent a relatively uniform sample [12], these sequences can be used to estimate the number of occurrences of these primers within the 60-80% repetitive maize genome. Potential binding locations of each primer were calculated using a standard sequence alignment algorithm, and matches that contained at most one difference were saved for each member of the entire set of over 500k BAC end sequences available. A similar analysis was performed against known repeats and overrepresented sequences within maize [6] in order to further interpret these results.

V. Results

In order to get a classification of a primer as good or bad with a finite state classifier we count the number of $+$, $-$, or $?$ states encountered as the FSC is driven by the primer. The classification of the primer by the FSC is the type of state encountered the most. The classifications on the training and cross validation primer data sets for the most fit classifier in each of the three sets of initial evolutionary runs is given in Table II. The results for the hybridization runs are given in Table III.

The distribution of final best-of-run fitnesses for the initial sets of runs is given in Figure 2. The analogous data for the hybridization runs are given in Figure 3. Notice that the initial runs for 32 and 64 state classifiers contain clear high-end outliers while the 128 state runs at least have an upward tail. These high end outliers provide the motivation for performing

TABLE II

Predictions versus truth results for the most fit FSC located during the first set of simulations.

32 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	4299	11596	829	245	221	34
Bad	3277	5873	534	311	167	22

64 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	4565	11292	867	246	228	26
Bad	3610	5531	543	296	172	32

128 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	5671	10048	1005	234	224	42
Bad	4454	4592	638	283	176	41

TABLE III

Predictions versus truth results for the most fit FSC located during the hybridization simulations.

32 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	5031	11109	584	202	278	20
Bad	3873	5436	375	264	227	9

64 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	7317	8503	904	267	206	27
Bad	5286	3833	565	280	190	30

128 States

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	5608	10104	1012	199	267	34
Bad	4445	4626	613	246	227	27

hybridization runs. Hybridization clearly improves fitness but also creates substantial potential for over-training.

Table IV shows the IMM results using log-likelihood ratios as the scoring criteria. Questionable primers are defined to have a score that is at most 0.1 in absolute value. As shown, prIMMer does much better learning features of the good primer training set (72%) than those within the bad primer set (56%). Notice, however, that these numbers are better than those obtained from any of the best FSC results and the initial number of false positives are much higher (62%) than the best FSCs. Further investigation shows that the average score of a bad primer in the cross validation set is in the questionable range, while the positive cross validation primers recieve a score of well over one on average. As expected, statistical analysis shows that specificity is indeed moderately correlated with score and as a result using the highest scoring primers

TABLE IV

Predictions versus truth results for prIMMer predictions using the same datasets as the FSCs

	Training data Prediction			*Crossvalidation Set* Prediction		
	-	+	?	-	+	?
Good	3947	12108	669	146	329	25
Bad	5426	3759	499	168	310	22

can reduce the false negative rate as low as 40%, which is comparable to that obtained using evolved FSCs.

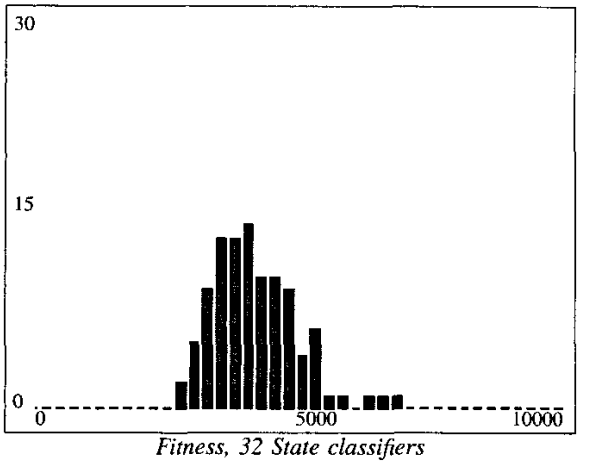
The same number of good and bad primers exactly matched at least one BAC end, although when approximate matches with one difference were allowed the bad primers had four times as many hits. Additional comparisons against a database of known repeats and overrepresented sequences within BAC ends [6] showed a similar uniform match rate, although only two of the matches overlapped between the two primer classes indicating distinct features within these sets.

VI. Discussion

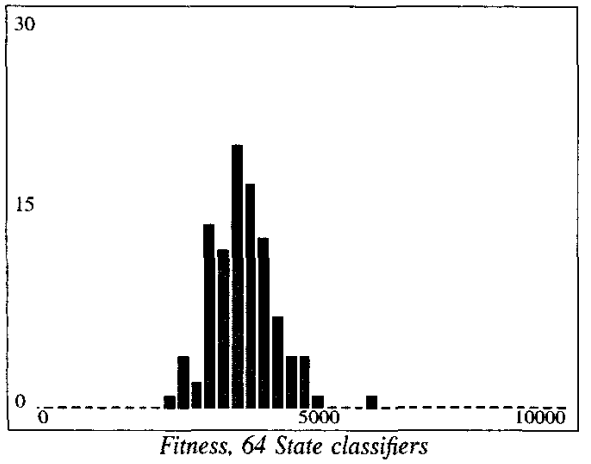
PCR-based techniques are involved in important biological experiments including the mapping of genes, and can be used in finishing, annotation and validation of whole genome assemblies. Machine learning approaches offer biological projects that rely on the success of PCR primers to in essence learn from their mistakes. In our current collaborative work on maize genome assembly we intend to use this information to not only help validate the assembly but also to test the expression of thousands of novel cereal genes predicted within our latest "gene-enriched" assembly. Any improvement in primer design translates into more efficient annotation of these genes and therefore allows quick dissemination of the results to the plant biological community.

The training data used in this particular study, however, were solely drawn from the ongoing genetic mapping project in the Schnable Lab at Iowa State University. Primers are used in pairs to amplify parts of the maize genome that exhibit size polymorphisms as a means of tracing the co-inheritance of genes. Failures can result from flaws in one of the primers, in both, or in a mismatch between the two primers. This means that some of the bad primers in the training data are in fact primers that might work just fine with a different partner. In addition, PCR reactions sometimes go awry for reasons unrelated to the primers themselves. Equipment failure, operator failure, or incorrect or expired reagents can all invalidate a reaction run with what might otherwise be good primers. This means that our positive examples are almost all correct while our negative examples are partly wrong.

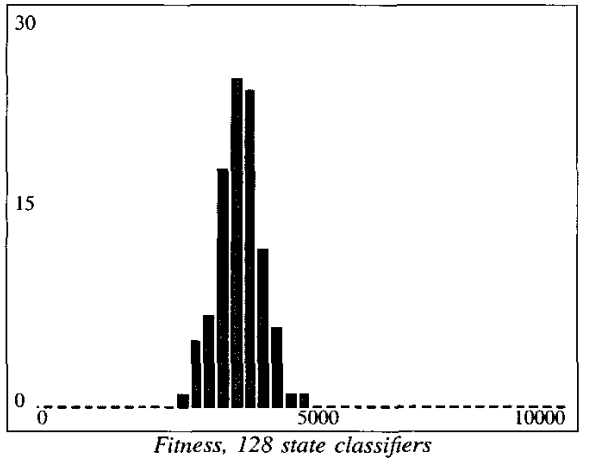
The difference between fitness and performance for the FSCs can be seen by comparing the fitness histograms, Figures 2 and 3 with Tables II and III. The best fitness appeared in the 64 state hybridized classifiers but they were outperformed on the cross validation set by the hybridized 32 state classifiers. This difference may well be due to over-training and shows that cross validation is absolutely necessary.



Fitness, 32 State classifiers

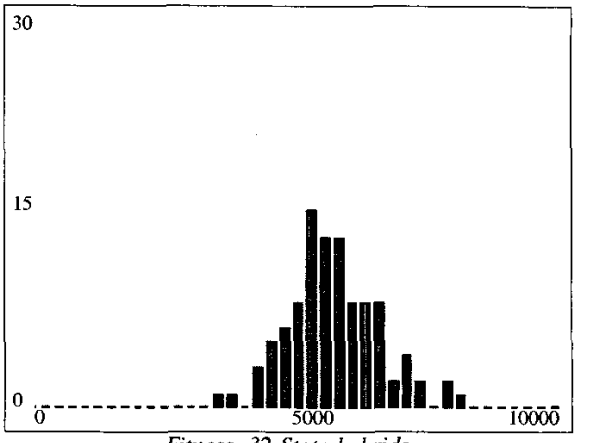


Fitness, 64 State classifiers

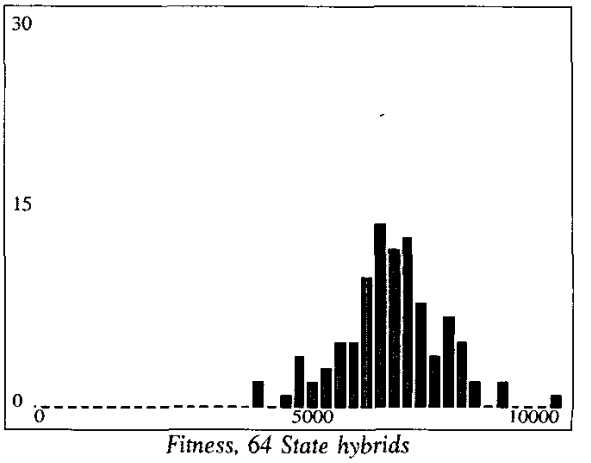


Fitness, 128 state classifiers

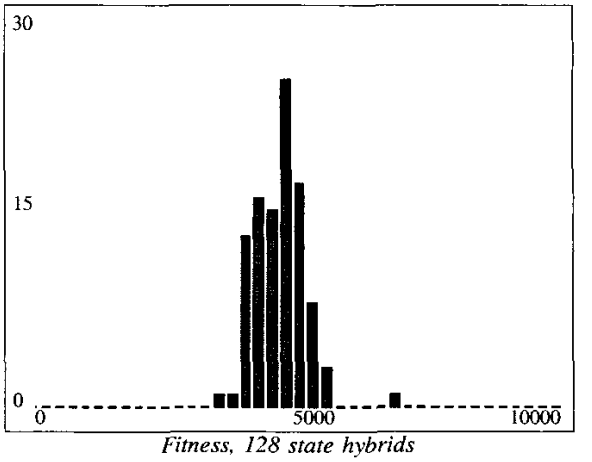
Fig. 2. Histograms of the distribution of fitnesses for initial populations with 32, 64, and 128 states.



Fitness, 32 State hybrids



Fitness, 64 State hybrids



Fitness, 128 state hybrids

Fig. 3. Histograms of the distribution of fitnesses for hybridized populations with 32, 64, and 128 states.

Although using fixed-order Markov models to learn oligonucleotide properties of effective primers was not an optimal approach, it did uncover an interesting trend within these data: ineffective primers have higher information content than effective primers. In other words, there are more unique oligomers in primers determined bad than in those found to be good. Any statistical machine learning approach, therefore, should ideally predict if a given primer will be good against the null hypothesis that it will not.

There is no noticeable difference in the occurrences within a set maize BAC ends that should be predominantly repetitive, but there is a clear a bias in terms of the sequences being matched. One possible hypothesis—marginally supported by annotation of some of these repeats—is that some ineffective primers may correspond to transposable element families supposed to have been active in the recent evolutionary history of maize [14]. If so, these sequences may still be intact and have sufficient similarity that would lead to non-specific PCR products. These results also indicate that experimental validation contains information that can not be currently detected using sequence comparison-based approaches alone.

A. Comparison of FSC and Interpolated Markov Results

The primary advantages of IMM-based primer validation are its speed and theoretical flexibility. Training an ICM using GLIMMER2 on 9000 primers takes 1.6 seconds on a 2.4 GHz Pentium machine and subsequent scoring of 17224 putative primers takes under 0.5 seconds. Both training and scoring scale linearly with input. Training and hybridization of 100 populations of FSCs, on the other hand, require between one and two days of compute time on a comparable machine.

Although the evolution of FSCs is more time-intensive, it should be able to better incorporate the low information content of experimentally-validated primers into a FSC-based classifier. As multiple FSCs do not agree, they can be used to isolate strongly-indicative patterns within good primers. As such patterns should recieve high ratio scores, which are statistically significant, this partially explains the convergance of these two methods when the best log-liklihood scores are considered. Average log-likelihood scores and the preponderance of ? states suggest that there is indeed more oligomers within bad primers and therefore a larger number of states can be used in the FSC approach.

B. Incremental Fitness

The fitness function reported here for the finite state classifiers incrementally rewards a given FSC, permitting the classifier to have multiple opinions in the course of processing a primer. This was done to compensate for poor performance in initial studies when the FSCs were scored only on a final call on each primer. It is conjectured that the effect of the incremental reward is to smooth the fitness surface. The shift of incremental fitness did have the effect of eliminating a large local optima in which the FSCs refused to classify many of the primers in the training data.

The improvement in performance forms a post-hoc justification of the incremental fitness function but it is the case that we are using the incremental fitness as a surrogate for our true target: correct classification. An earlier study [3] was performed that used incremental fitness for half of each of the simulation and then shifted to the ratio of correct to incorrect classifications. The results were intermediate in quality between the initial and hybridized runs reported in that study. Subsequent hybridization of those populations was not as useful, with final classification accuracy of hybridized populations being similar.

The numbers in Table I used to compute incremental fitness were chosen to match the neutral character of the training data with its equal number of good and bad examples. Since rejecting negative primers is more important for the system's potential economic impact, resetting the numbers to emphasize correct results on bad primers is a definite next step.

VII. Conclusions

The primary goal of this study was to improve primer design performance by using machine learning as post-processor to capture features of primer performance not related directly to the DNA biophysics already embedded in primer-selection packages. Both techniques tested yield improved performance. A study is currently underway to assess the impact of the FSCs on actual wet-lab performance and more definitive claims of success wait on these wet-lab results.

The novel features of this study, aside from its main goal, are the use of a per-state incremental fitness function, the use of hybridization and the introduction of interpolated Markov models for this problem.

Experimental results suggest that use of interpolation is able to avoid potential problems of insufficient primer training data that occur for higher order Markov models, while smoothing the fitness surface by utilizing as many equivalent FSC states as possible. In contrast, evolved FSCs seem to better condense the the unique features within these primers and provide a more specific classifier for strongly-indicitive patterns. We hypothesize that both methods can be combined in order to utilize the distinctive advantages of both approaches, although IMMs are clearly preferable given a choice between the two due to their speed, training flexibility, and the availability of a computable p-value based on the χ^2 distribution.

A very clear problem with this study is the quality of the input data. Calls of primers as bad may result from poor technique in performing the amplification reaction or error in primer synthesis. This, together with the entanglement of primers in pairs only one of which may in fact be bad, makes the classification of the training data very noisy indeed. This problem can only be addressed by either substantial improvements in lab technique or exceedingly expensive testing of bad primer pairs. To perform such testing alternative primer partners for each member of a bad pair would be picked and then amplifications performed. This latter procedure, while expensive and time consuming, would improve the quality of "bad" calls.

The notion of *building blocks* [9], [10], sub-structures of members of an evolving population that can be mixed and matched by crossover to enhance evolutionary search, is a perennial topic in evolutionary computation research. It is intuitive that hybridization should function well when building blocks are available in a representation. It is less intuitive that hybridization should fail in the absence of building blocks, as crossover with structures evolved in a different population might serve as a useful macro-mutation. We suggest that hybridization studies may be able to help identify systems in which building blocks exist. This is an area for future study.

VIII. Next Steps

The clearest areas for additional work are improving the fitness function, fusing information from the Markov models and FSCs, and studying how to perform hybridization effectively.

A. Improving the Fitness Function

The opinion of a FSC after it has seen only a few nucleotides is probably uninformative, and many of our FSCs have numerous ? state labels during the initial processing of PCR primers. Following this notion, it might improve performance to weight incremental rewards gained later in the processing of a primer higher than those gained earlier. A preliminary study of this technique yields mixed results, but there are many possible schemes for increasing the weight of any reward as time goes by.

We also reported in Section VI that a preliminary study of shifting from incremental reward fitness to the ratio of correct to incorrect predications was promising, but, after hybridization, did not enhance final best performance. This preliminary study looked at only one way to shift from incremental reward to prediction accuracy. Since prediction accuracy is what we actually desire, schemes for involving prediction accuracy more directly in assessment of the FSCs merit additional study.

At present the incremental reward for states labeled with a ? is zero. A small study of the results of setting the reward for ? states to a small negative number substantially degraded performance. The logic behind that experiment was that a small negative score for ? states would encourage classifiers to make some guess about almost every primer. It did have this effect but the FSCs made more new bad guesses than new good guesses. Since some ? states are probably required early in the processing of a primer, it may be that such negative rewards should be phased in over the course of the examination of a primer.

B. Cross-technique Generalization

The FSCs used in this study do not agree with one another about which primers are good or bad. Likewise, the results from the interpolated Markov model do not agree with any particular FSC. This suggests that studying schemes, such as majority vote, that amalgamate these opinions may be worthwhile. Finding good relative weights of the opinions of different "voters" is likely to be a thorny problem. Something like stacked generalization from the neural net community may prove valuable. Stacked generalization proceeds by first training several neural net classifiers with different reserved cross validation sets and then training an additional neural net to decide which of them to listen to.

C. Understanding Genetic Hybridization

Genetic hybridization, the seeding of a set of simulations with the best-of-run structures from another collection of simulations, can improve performance. In this study we have verified that hybridization improves both fitness and prediction accuracy for the PCR primer FSCs. Reference [2] demonstrated that hybridization helped in code induction for the Tartarus [16] problem. In Reference [1], hybridization was critical in evolving simple optical character recognition systems. While we have examples that show hybridization is useful, we still lack the answers to at least two questions. What sort of problems will hybridization help with? What sorts of stopping and starting simulation schedules, which incorporate transfer of best-of-run structures to new starting populations, are good?

The first question is open ended and quite difficult to answer. We currently lack a good taxonomy of problems and therefore must pile up examples. It is clear, however, that problems where there is a unique best answer that is easy to locate will not benefit from hybridization. We conjecture that if a given representation of solutions for a problem exhibits a high level of epistasis then hybridization will not help.

The second question, that of finding effective hybridization schedules, is likely to have task-specific answers. It is clear that if multiple hybridizations are to be performed, some evolution must happen before re-hybridization takes place. Hybridization can be viewed as a more extreme version of the selection technique used in "Island Genetic Algorithms" [18], and it would not be difficult to test different hybridization schedules for particular problems. This is a part of the authors ongoing research.

D. Detecting Critical Patterns

If our machine learning techniques are detecting some systematic flaws in PCR primers that are not apparent to standard primer design packages, then those patterns are themselves of interest. We give two speculative procedures for locating and understanding these patterns.

- Systematic enumeration of bad patterns could be performed by tracing those paths that lead to large negative scores within multiple evolved FSCs or large negative log-likelihood scores using the Markov models. A potential problem with such an approach, however, is that many of these patterns might not be good primers according to standard primer design software and might not even appear in maize (or any organism).
- A modified systematic enumeration in which known maize DNA sequences are fed through the predictors in primer-sized chunks. The aggregate score from the predictors are then plotted along the length of the DNA

to see what features receive relatively high and low scores. It would not be surprising if known or unknown transposable elements were detected in this fashion.

IX. Acknowledgments

This research was supported by competitive grants from the National Science Foundation Plant Genome Program (award numbers: DBI-9975868 and DBI-0321711) to Patrick S. Schnable, Daniel Ashlock, and others. Scott Emrich was supported by a National Science Foundation Integrative Graduate Education and Research Traineeship (IGERT) fellowship (award number: DGE-9972653). We would like to thank the members of the Schnable Lab for their support of this work, Dr. Srinivas Aluru and the members of the Aluru lab for their support and help with computational aspects of the work, and the members of the Complex Adaptive Systems program at Iowa State for helpful discussions. We also thank the reviewer for enabling us to substantially clarify portion of the manuscript.

References

[1] Dan Ashlock. Data crawlers for optical character recognition. In *Proceedings of the 2000 Congress on Evolutionary Computation*, pages 706–713, 2000.

[2] Dan Ashlock and Mark Joenks. ISAc lists, a different representation for program induction. In *Genetic Programming 98, proceedings of the third annual genetic programming conference.*, pages 3–10. San Francisco, 1998. Morgan Kaufmann.

[3] Dan Ashlock, Andrew Wittrock, and Tsui-Jung Wen. Training finite state classifiers to improve pcr primer design. In *Proceedings of the 2002 Congress on Evolutionary Computation*, pages 13–18. IEEE Press, 2002.

[4] Kuma Chellapilla and David Czarnecki. A preliminary investigation into evolving modular finite state machines. In *Proceedings of the 1999 Congress on Evolutionary Computation*, pages 1349–1356, 1999.

[5] A. L. Delcher, S. Harmon, D. amd Kasif, O. White, and S. L. Salzberg. Improved microbial gene identification with GLIMMER. *Nucl. Acids Res.*, 27, 1999.

[6] S. J. Emrich, S. Aluru, Y. Fu, T. Wen, M. Narayanan, L. Guo, D. A. Ashlock, and P.S. Schnable. A strategy for assembling the maize (*Zea mays* L.) genome. *Bioinformatics*, 20:140–147, 2004.

[7] D.B. Fogel. Evolving behaviors in the iterated prisoners dilemma. *Evolutionary Computation*, 1(1), 1993.

[8] L. J. Fogel, A.J. Owens, and M.J. Walsh. Artificial intellegence through simulated evolution. In *Biophysics and Cybernetic Systems: Proceedings of the 2nd Cybernetic Sciences Symposium*, pages 131–155, 1965.

[9] David E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley Publishing Company, Inc., Reading, MA, 1989.

[10] John H Holland. *Adaption in Natural and Artificial Systems*. The MIT Press, Cambridge, MA, 1992.

[11] Benjamin Lewin. *Genes VI*. Oxford University Press, New York, 1997.

[12] B. C. Meyers, S. V. Tinge, and M. Morgante. Abundance, distribution, and transcriptional activity of repetitive elements in the maize genome. *Genome Res.*, 11:1660–1676, 2001.

[13] John H. Miller. The coevolution of automata in the repeated prisoner's dilemma. A Working Paper from the SFI Economics Research Program 89-003, Santa Fe Institute and Carnegie-Mellon University, Santa Fe, NM, July 1989.

[14] P. SanMiguel, B.S. Gaut, A. Tikhonov, Y. Nakajima, and J. L. Bennetzen. The paleontology of intergene retrotransposons of maize. *Nat. Genet.*, 20:43–45, 1998.

[15] E. Ann Stanley, Dan Ashlock, and Leigh Tesfatsion. Iterated prisoner's dilemma with choice and refusal. In Christopher Langton, editor, *Artificial Life III*, volume 17 of *Santa Fe Institute Studies in the Sciences of Complexity*, pages 131–176, Reading, 1994. Addison-Wesley.

[16] Astro Teller. The evolution of mental models. In Kenneth Kinnear, editor, *Advances in Genetic Programming*, chapter 9. The MIT Press, 1994.

[17] C. A. Whitelaw, W. B. Barbazuk, and G. Pertea *et al.* Enrichment of gene-coding sequences in maize by genome filtration. *Science*, 302(5653):2118–20, 2003.

[18] Darrell Whitley. The genitor algorithm and selection pressure: why rank based allocation of reproductive trials is best. In *Proceedings of the 3rd ICGA*, pages 116–121. Morgan Kaufmann, 1989.

[19] J. Q. Wu, D. Shteynberg, M. Arumugam, A. Gibbs, and M. R. Brent. Identification of rat genes by TWINSCAN gene prediction, RT-PCR, and direct sequencing. *Genome Res.*, 14:665–671, 2004.