A Multi-Objective Genetic Algorithm that Employs a Hybrid Approach for Isolating Codon Usage Bias Indicative of Translational Efficiency

Douglas W. Raiford
Wright State University - Main Campus

Dan E. Krane
Wright State University - Main Campus, dan.krane@wright.edu

Travis E. Doom
Wright State University - Main Campus, travis.doom@wright.edu

Michael L. Raymer
Wright State University - Main Campus, michael.raymer@wright.edu

Follow this and additional works at: https://corescholar.libraries.wright.edu/knoesis

Part of the Bioinformatics Commons, Communication Technology and New Media Commons, Databases and Information Systems Commons, OS and Networks Commons, and the Science and Technology Studies Commons

Repository Citation

This Conference Proceeding is brought to you for free and open access by the The Ohio Center of Excellence in Knowledge-Enabled Computing (Kno.e.sis) at CORE Scholar. It has been accepted for inclusion in Kno.e.sis Publications by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
A Multi-Objective Genetic Algorithm that Employs a Hybrid Approach for Isolating Codon Usage Bias Indicative of Translational Efficiency

Douglas W. Raiford\(^1\), Dan E. Krane\(^2\), Travis E. Doom\(^1\), and Michael L. Raymer \(^*\)

\(^1\)Department of Computer Science and Engineering
\(^2\)Department of Biological Sciences
Wright State University
3640 Colonel Glenn Highway
Dayton, OH, 45435, United States

Abstract

Isolation of translational efficiency bias can have important applications in gene expression prediction and heterologous protein production. In some genomes the presence of a high GC\(\text{AT}\)-content bias can confound the isolation of translational efficiency bias. In other organisms translational efficiency bias is weak making it difficult to isolate. Described here is a multi-objective genetic algorithm that improves the isolation of translational efficiency bias in Streptomyces coelicolor A3(2) and Pseudomonas aeruginosa PAO1, two organisms shown to have high GC-content and weak translational efficiency bias.

1 Introduction

The genetic code is degenerate in that multiple codons can code for the same amino acid. It has been shown that some organisms preferentially utilize one or more of the synonymous codons that code for a given amino acid \([13, 14, 18]\). There are several causes for this biased usage including tendencies toward high or low GC-content, strand bias \([5, 6]\), and translational efficiency bias. Strand bias is brought on by transcription induced C\(\rightarrow\)T transitions on the leading strand thought to be caused by increased deamination of cytosine \([3, 10]\). Translational efficiency bias is thought to be the result of selective pressure to enhance the rate of translation particularly in highly expressed genes \([19, 21]\). The transfer-RNAs associated with each codon vary in their relative abundance presenting opportunities for gains in the efficiency of translation when the most available tRNA is utilized.

The research described here is concerned with improving the isolation of translational efficiency bias in organisms where this bias has been shown to be weak and that have as a dominant bias either high or low GC-content. This can be beneficial if there is a need to predict expressivity (highly expressed genes tend to adhere to translational efficiency bias the strongest) \([19, 21]\) or to improve heterologous protein production \([15]\). When genes are introduced into host organisms for the purpose of producing a given protein, inefficiencies can occur due to a mismatch between optimal codons in the two genomes. By modifying the introduced gene to conform to the host organism’s codon usage bias an increase in protein product can be realized.

Searches for a codon usage bias that is reflective of translational efficiency fall into two categories: one that requires prior knowledge of a set of highly expressed genes, and another that attempts to determine the dominant bias using a purely algorithmic method. This analysis investigates whether a hybrid approach that combines aspects of these two approaches might yield better results. Expression level is related at least in part to transcript abundance and one method for determining whether an improved methodology is achieved is to compare the resultant ranking of genes according to their adherence to the identified codon usage bias to an experimentally derived ranking representative of transcript abundance. Experimental data used in this analysis is derived from transcript abundance results from microarray experimentation. Of particular interest are organisms that exhibit weak translational efficiency bias and a more dominant GC- or AT-content bias. Purely algorithmic approaches will find the GC- or AT-content bias in these organisms. It may also be the case that an interfering GC\(\text{AT}\)-content bias obstructs the empirical approach’s (prior knowledge of highly expressed genes) ability to accurately capture the translational efficiency bias. The organisms chosen to test the methodology presented here are Streptomyces coelicolor A3(2) and Pseudomonas aeruginosa PAO1. Both are characterized by high GC-content and

\(*\)Corresponding author: michael.raymer@wright.edu
weak translational efficiency biases.

The hybrid approach proposed here uses prior knowledge of a set of genes generally known to be highly expressed (ribosomal protein coding genes) and an evolutionary algorithmic approach that optimizes both where the highly expressed genes fall, and where a proposed reference set of genes fall, in the resultant gene ranking according to adherence to the identified bias. Section 2 briefly describes the two previously published approaches upon which the hybrid methodology is based. Following this are descriptions of data acquisition techniques, the genetic algorithm that hybridizes these two approaches, and an account of the various measures employed in the analysis.

2 Methods

One commonly employed measure of a gene’s adherence to codon usage bias is its codon adaptation index (CAI). There are two approaches for deriving CAI scores for the genes in a target genome. The original method, put forward by Sharp and Li [22], uses as a starting point a known set of highly expressed genes. Carbone et al. [6] employ an iterative search routine that first algorithmically identifies a reference set before calculating CAI scores. Certain aspects of each of these approaches are employed in our hybrid approach to identifying CAI scores for the genes in a target genome.

2.0.1 Sharp and Li CAI

The Sharp and Li codon adaptation index measurement assigns a value known as the relative adaptiveness (or weight) to each codon. Prior to this the prevailing methods assigned essentially binary values to codon usage. Either the codon in question was optimal or it was not. There was no gradation. The weight for each codon is derived from the set of genes that is determined experimentally to be the most highly expressed genes of the genome. This is known as the reference set. From the sequences of these genes codon usage values are determined. These frequencies are normalized by the average usage for the set of synonymous codons related to the codon in question. This normalized frequency is known as relative synonymous codon usage (RSCU) (1). RSCU and the weight of a codon \( w_{ij} \) is:

\[
RSCU_{ij} = \frac{X_{ij}}{\frac{1}{n_i} \sum_{j=1}^{n_j} X_{ij}} \quad (1)
\]

\[
w_{ij} = \frac{RSCU_{ij}}{\text{RSCU}_{i_{\text{max}}}} = \frac{X_{ij}}{X_{i_{\text{max}}}} \quad (2)
\]

\( X_{ij} \) is the count of the codon (within the subset of genes known as the reference set), and \( X_{i_{\text{max}}} \) is the count of its sibling with highest count (the maximal sibling will have a weight of one). \( w_{ij} \) is the weight \( w \) of the \( i^{th} \) codon for the \( j^{th} \) amino acid. A geometric mean is taken of these normalized RSCU values to calculate CAI. This reduces to:

\[
CAI(g) = \frac{L}{\prod_{i=1}^{L} w_i} \quad (3)
\]

\( L \) is the length of the gene (number of codons in the gene). The CAI value for a gene is a geometric mean of the weights assigned to each of the codons in that gene. The analysis performed herein uses as a reference set the most highly expressed genes as determined from experimental microarray expression data (section 2.2).

2.0.2 Carbone et al. CAI

Carbone et al. removed the need for prior knowledge of gene expressivity from the CAI calculation process [6]. Theirs is a greedy algorithm that works first to identify the proper reference set of genes and then calculates the CAI score for each gene based upon this reference set. Identification of the reference set is performed by assigning a precise mathematical definition to reference set membership and then searching for the genes that match this definition. Carbone et al. define a reference set as a small set of genes (1% of genome) characterizing a bias to which its (the reference set’s) adherence is stronger than all other genes in the genome. The algorithm is iterative in nature. Identification of the reference set starts by considering the set of all genes as the reference set. It assigns a weight to each codon based upon the codon usage in this all-inclusive reference set. The weight for a given codon is calculated as before (2) based upon the identified reference set. CAI scores are calculated for all genes as (3), the list of genes is sorted by this CAI score, and then the genes in the top half of the list are kept as the new reference set. New \( w \) values are calculated, followed by new CAI values for the genes. This half-splitting technique is repeated until the reference set represents one percent of the original number of genes.

The hybrid approach taken in this work investigates whether combining both empirical (prior knowledge of highly expressed genes) and theoretical (mathematical definition of a reference set) approaches can produce an improved ranking of genes with respect to experimentally derived expression data. A genetic algorithm (GA) is employed that attempts to maximize these two aspects of a solution.

2.1 Genetic Algorithm

A genetic algorithm is an algorithm modeled after nature’s approach to optimizing a species’ fitness for survival.
[11, 16]. The algorithm simulates natural selection by maintaining a population of individual members, each of which is a proposed solution to the problem at hand. The search for the best solution proceeds by selecting members from the population to produce offspring that combine traits from the parents. A fitness score is assigned to each proposed solution and poor solutions are removed from the population in a process that mimics natural selection. Offspring in nature vary from their parents through recombination of the parents' DNA (known as crossover). Variation is also occasionally introduced into population s through the occurrence of mutation. In order to imitate this behavior the GA must have a representation of the solution that is amenable to these operations (crossover and mutation).

2.1.1 Representation, Operators, & Settings

The problem being solved in this study is one of a search for a subset of genes in the overall genome of an organism (the reference set). A natural representation for a solution to this problem is a bit string where each bit represents a gene in the genome. If the value of that bit is a one then the gene is a member of the reference set. If it is a zero then it is not. The bit string in the GA is known as a chromosome, and corresponds directly to the chromosome of the organism. Genetic algorithms have several operators and settings, the choice of which are representation and problem dependent. Parameters and operators were chosen that seemed to make the most sense in light of the nature of the problem being investigated.

Identification of the parents for the purposes of recombination is performed using tournament selection [12]. Candidates are chosen at random until a tournament size of three is achieved. The member with the best fitness (see section 2.1.4) within this group of three is the tournament winner and is selected as the parent. Once two parents are identified, uniform crossover is employed to generate a child chromosome. Each gene (bit in the bit string) is set by randomly choosing one or the other of the parents as the source. In the GA implemented here both parents are equally probable as source candidates for each bit in the bit string. Since bits are set randomly from both parents, the number of bits turned-on in the child bit string may or may not contain the correct number of ones (a reference set should be of size 1% of genome size). In the extreme cases there could be as many as twice the acceptable number of bits turned-on in the child chromosome (2%), or as little as zero bits turned-on. To correct for this a simple repair mechanism is utilized that randomly turns off (or on) bits until the target size is achieved.

To introduce additional variation into the population a mutation operator is employed that randomly turns off a bit that is currently on and randomly turns on a bit that is currently off. The number of mutations is set such that each bit in the chromosome has a probability of 1/N (where N is the number of genes in its genome) that it will be involved. In this way an individual will usually incur a single mutation per generation, though the stochastic nature of the process will sometimes increase (or decrease) this number. Changing approximately one bit in the reference set for any given generation lends gradualism to the search process. In like fashion, uniform crossover becomes more gradual as the population begins converging on a good solution space.

The GA population size is set at 100 with the number of offspring also set at 100. Population management follows that described by [1] (μ + λ) where λ offspring are generated from μ parents and combined into a single population in which both parents and children must compete for survival. A degree of elitism is introduced as in [8] by maintaining a certain number of the best members of the combined population in the next generation. We set this elitism parameter to 60% (60% of the next generation is made up of 60 of the best members in the previous combined population of 200) in order to gain insights into the nature of the problem space (Figs. 1(a) and 2(a)). The remaining 40 slots in the next generation's population are selected using fitness proportional (roulette) selection technique [2] where the more fit the individuals in the combined population are, the more likely they are to be drawn into the following generation.

The fitness calculation is computationally intensive, so to prevent recalculation of solutions previously visited a hash of all bit strings already computed is maintained in memory. This is a simple form of a tabu search [4]. Two fitness measures are required to capture the hybrid nature of the approach – one that captures the mathematically rigorous definition of a reference set, and one that measures an empirically derived characteristic of the proposed solution.

2.1.2 Rank-Fitness

The mathematically based measure proposed here is known as rank-fitness, and is based upon the reference set definition put forward by Carbone et al. [6]. Rank-fitness measures the degree to which a reference set adheres to the bias that the reference set itself defines. If the reference set adheres to this bias stronger than all other genes, then the reference set will be the first genes in a listing of all genes when sorted by CAI score. All non-reference set genes will fall below the lowest reference set gene in the listing. To measure the degree to which this occurs the genes are sorted by CAI score and an index is assigned to all genes – the gene with the smallest CAI value will be assigned an index of 0 while the gene with greatest CAI value will be assigned an index of N - 1 where N is the number of genes in the genome. The indexes of the reference set genes are summed (IDX's in (4)) and this number is normalized.
by the maximum attainable rank-fitness value (5) yielding a value between 0 and 1 representing the degree to which the reference set rises to the top of a sorted (by CAI score) list of genes (6).

\[ f(RS) = \sum_{i=1}^{[RS]} ID X_i \]  \hspace{1cm} (4)  
\[ f_{\text{max}}([RS]) = \sum_{i=N-|RS|}^{N-1} i \]  \hspace{1cm} (5)  
\[ f_{\text{norm}}(RS) = \frac{f(RS)}{f_{\text{max}}([RS])} \]  \hspace{1cm} (6)

While rank-fitness measures the adherence to a theoretical definition of fitness, the empirically derived prior knowledge of a set of highly expressed genes is measured by observing where in the distribution of CAI scores a set of known highly expressed genes falls. Ribosomal protein coding genes are used for this purpose as they are known to be highly expressed. Carbone et al. have developed a measurement known as ribosomal criterion that can be used for these purposes [5].

### 2.1.3 Ribosomal Criterion

Ribosomal criterion is the average CAI score for ribosomal protein coding genes when converted to standard normal form. The CAI score for a gene is converted to standard normal form as in (7).

\[ \bar{z} = \frac{CAI(r) - C\bar{A}I}{\sigma_{CAI}} \]  \hspace{1cm} (7)

The average of the z-scores for ribosomal protein coding genes (RPCGs), \( \bar{z} \), is the ribosomal criterion. A genome characterized by translational bias will have a high \( \bar{z} \). Carbone et al. have determined \( \bar{z} \) values greater than 1 (i.e. the average CAI values for RPCGs > 1 standard deviation above the average for the entire genome) are indicative of organisms characterized by translational efficiency bias [5].

By searching for a reference set that optimizes both ribosomal criterion and rank-fitness, it is hoped that a set of weights can be found that will better characterize the translational bias resident within the organism’s genome. Determination of whether this is accomplished will be measured by the correlation of the resultant CAI scores with experimentally derived transcript abundance data. This correlation will then be compared with the correlation achieved when the traditional Sharp and Li method of deriving CAI values is used [22]. This approach should be successful because there is a great deal of room for improvement in both the ribosomal criterion and rank-fitness when the traditional Sharp and Li technique is utilized (e.g. the Sharp and Li determined CAI scores for *P. aeruginosa* exhibit a rank-fitness of 0.730 (Table 1) and a ribosomal criterion of .762). Optimizing both ribosomal criterion and rank-fitness makes this a multi-objective problem. In order to assign a single fitness value to a solution a standard multi-objective technique known as Pareto ranking is utilized [9].

#### 2.1.4 Pareto Ranking

When a solution to a problem requires optimizing more than one measure of fitness, in some cases it becomes difficult to determine whether one solution is better than another. In the case of a two-objective problem (as in this work), when both measures are better in one member than in another, it is easy to conclude that the first member represents a better solution. In this case the first member is said to dominate the other. But, when each member is better than the other in one of its objectives then neither dominates the other and a system must be devised to determine which, if any, is the better solution. Often when two objectives are being sought, an improvement in one comes at the expense of the other. The best solutions, those not dominated by any other proposed solutions, will align themselves on a non-dominated frontier known as a Pareto front [20] (Figs. 1(a) and 2(a)). Each member on this front may have a slightly lower fitness in one of its objective measures than its neighbors, but it will have a slightly better fitness in the other.

A Pareto ranking for a given search member is determined by counting the number of other members that dominate the member in question. One member, whose objective measures can be represented by a \( k \)-dimensional vector \( (\bar{u}) \) is said to dominate \( (\bar{u} \succeq \bar{v}) \) another \( (\bar{v}) \) if all of its objective measures are greater than or equal to (and at least one of them is greater than) the other’s objective measures. Mathematically, this can be described:

\[ \forall i \in \{1,\ldots,k\}, u_i \geq v_i \land \exists i \in \{1,\ldots,k\} : u_i > v_i \]  \hspace{1cm} (8)

Members on the Pareto front will have 0 as the count of other members that dominate them, while a poor solution may be dominated by all other members (a count of 99). This count is used as measure of fitness that incorporates both objective measures. To maintain consistency the GA maximizes the Pareto ranking so each member’s count is subtracted from the maximum count in order to give the largest values to those with the greatest fitness.

Over time the GA described here generates more members on the Pareto front (during a given generation) than are kept for the next generation. The number kept for the next generation is set by an elitism parameter. The technique employed to determine which members on the front to keep is niching.
2.1.5 Niching

When there are more members on the Pareto front than can be kept for the purposes of elitism members with the best ribosomal criterion and the best rank-fitness are kept. Members are also drawn from the remaining portion of the Pareto front that reside between these two extremes in such a way as to maximize the spread (members that are evenly distributed along the Pareto front are kept for the next generation).

The high degree of elitism (60%) was chosen in order that there would be enough members on the front to clearly define its boundaries. A true niching technique would cluster members in the population according to some similarity metric and choose representatives of each cluster to keep. Additionally, pressure to choose parents from the same clusters would be injected into the process under the assumption that there is something inherently noteworthy about the region of the search space described by the cluster. These techniques were not employed here due to the simple goals of the investigation, but future versions of the GA should include them.

2.2 Microarray Data

Experimentally derived expression data are utilized to compare the performances of the traditional Sharp and Li method and the multi-objective genetic algorithm. The expression data used are taken from the Gene Expression Omnibus (GEO) found on the National Center for Biotechnology Information (NCBI) web site (http://www.ncbi.nlm.nih.gov/). The specific study from which the expression data for \emph{P. aeruginosa} are drawn is [7]. That analysis investigated the effect on gene expression in response to the introduction of hydrogen peroxide and utilized Affymetrix \emph{P. aeruginosa} GeneChip arrays. For the purposes of this study the “no treatment” data sets are used as a measure of transcript abundance for the organism under normal conditions. There are four such sets and the values from the four sets are averaged for each gene. Those genes listed as “absent” are removed from consideration. Similarly, the expression data for \emph{S. coelicolor} is from the genomic expression omnibus. The data were extracted from the work of [17] who performed a global analysis of growth phase responsive gene expression and regulation of antibiotic biosynthetic pathways. Again, control data were extracted from their nine experiments and averaged to determine no-treatment expression rates.

2.3 Sequence Data

The sequence information required for calculating weights and the corresponding CAI values are derived from annotated whole genome files located in the complete microbial genomes database on the NCBI web site (http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi).

2.4 Implementation and Hardware

Genetic algorithms lend themselves to a high degree of parallelism. A simple technique is to manage the population on a central node (parent selection, crossover, mutation, etc.) and to compute fitness concurrently on satellite systems. The GA manager and fitness calculator components described here are written in C++. The jobs are presented to the remote hosts in the form of bit strings sent over a socket connection while the results are returned over the same socket connections in the form of “long float” data types.

3 Results

Spearman rank correlations between CAI scores and transcript abundance data were computed for CAI scores generated by the traditional Sharp and Li method [22] and the multi-objective genetic algorithm (Table 1). The GA was able to achieve a Spearman rank correlation between CAI and transcript abundance of 0.370 \((P < 0.05)\) for \emph{P. aeruginosa} and 0.155 \((P < 0.05)\) for \emph{S. coelicolor}. These were more strongly correlated than those achieved by the Sharp and Li technique. They also were stronger than those achieved by the Carbone \emph{et al.} approach \((r_s = 0.131\) and 0.204 for \emph{S. coelicolor} and \emph{P. aeruginosa}\) respectively, with \(P < 0.05\) for both), but this is to be expected as that approach is designed to identify the dominant bias (GC-content in the case of these two organisms).

The final population of the GA had approximately 60 members of its population that were non-dominated (on the Pareto front) (Figs. 1(a) and 2(a)). When these population members were examined for their correlation with transcript abundance, the non-dominated population members with the higher ribosomal criteria tended to be the ones with better correlations with transcript abundance (Fig. 3). For \emph{S. coelicolor} there is a gap in the Pareto front between those with the very best ribosomal criteria and the rest of the members on the front (Fig. 2(a)). The expectation is that a better niching strategy will eliminate such gaps, and improve search efficacy of the GA overall.

The fitness history (Figs. 4(a) and 5(a)) shows that dramatic improvements in rank-fitness occur very quickly in the search process while improvements in ribosomal criterion require somewhat longer (Figs. 4(b) and 5(b)). All members, even those with the highest ribosomal criteria (and, therefore, lowest rank-fitness) have greater rank-fitnesses than those described by the traditional Sharp and Li approach (Table 1 and Figs. 1 and 2).
Table 1. Correlation, Fitness, and Criterion Data

<table>
<thead>
<tr>
<th></th>
<th>S. coelicolor</th>
<th>P. aeruginosa</th>
</tr>
</thead>
<tbody>
<tr>
<td>r_s with Expression Data</td>
<td>0.133</td>
<td>0.155</td>
</tr>
<tr>
<td>Rank-Fitness</td>
<td>0.717</td>
<td>0.971</td>
</tr>
<tr>
<td>Ribosomal Criterion</td>
<td>0.505</td>
<td>1.092</td>
</tr>
<tr>
<td>Strength</td>
<td>0.967</td>
<td>2.056</td>
</tr>
<tr>
<td>Average CAI Score</td>
<td>0.678</td>
<td>0.532</td>
</tr>
<tr>
<td>Standard Deviation for CAIs</td>
<td>0.108</td>
<td>0.128</td>
</tr>
</tbody>
</table>

GA data pertain to the population member found on the Pareto front with the largest ribosomal criterion. Note: for *Streptomyces coelicolor* A3(2) there is one population member with a better correlation than the member identified in this table (Fig. 3(b)).

Figure 2. Pareto Front and Entire Population for *S. coelicolor*

Data points represent the ribosomal criterion (y axis) and rank-fitness (x axis) for members of the population on the Pareto front (2(a)) and for the entire population (2(b)).

Figure 1. Pareto Front and Entire Population for *P. aeruginosa*

Data points represent the ribosomal criterion (y axis) and rank-fitness (x axis) for members of the population on the Pareto front (1(a)) and for the entire population (1(b)).

Figure 3. r_s vs. Ribosomal Criterion of Population Members on the Pareto Front

r_s is the Spearman rank correlation between the CAI values as generated by the population member of the GA and the transcript abundance as determined experimentally using microarray analysis.
4 Discussion

This study set out to determine whether the search for translational efficiency bias in an organism that is not dominant for that bias can be enhanced through a hybrid approach. The two approaches being hybridized include the traditional empirical technique where highly expressed genes are used to determine the bias levels, and the theoretical method where the best solution matches the mathematically defined characteristics of a reference set. The two approaches are combined in a multi-objective genetic algorithm that optimizes these two measures. The hybrid approach is tested on two organisms (S. coelicolor and P. aeruginosa). They were chosen because of their high GC-content and weak translational efficiency biases.

The algorithm is able to achieve the desired outcome of generating CAI scores that better correlate with experimentally derived expression data (Table 1). These improvements are modest but it is thought that even better results can be obtained by adjusting the behavior of the GA. This version of the GA is exploratory in nature and is set for high elitism and to optimize multiple objectives in an attempt to determine what mix of these two objectives generates the best answers. Now that this has been answered, a more targeted search can be utilized that can find better solutions more efficiently. This approach may be useful in organisms that are not high or low in GC-content, and that exhibit strong translational efficiency bias (vs. the weak bias exhibited in the organisms studied here). As an example, Escherichia coli is characterized by a dominant translational efficiency bias [5]. When its rank-fitness is calculated on the Sharp and Li determined reference set, a rank-fitness score of 0.559 is attained, but even with this high score, there are 1,073 non-reference set genes with better CAI scores than the lowest reference set CAI score. This leaves open the possibility of significant improvement.

Of the best solutions, that is the solutions on the Pareto front, the ones that best correlate with the experimentally derived expression data tend to be located at the end of the Pareto front where the best ribosomal criterion scores are found (Fig. 3). Thus, a better search strategy would appear to be that of focusing solely on ribosomal criterion. Additionally, there is room for further research in optimal mutation, crossover, elitism, etc.

It is interesting to note that we have set out to discover (and have succeeded in doing so) a reference set that is "better" than a reference set already known to be the best (known to be the most highly expressed). But this calls into question what is meant by a better reference set. In the case of the GA it is simply a vehicle for discovering a better set of weights; a set that better captures the underlying bias and that better explains the ranking of genes that result from adherence to this bias. A rich area for future research will be
the determination of a more direct method of searching for these optimal weights (vs. searching for a set of genes that indirectly produce the improved weights).

Earlier we noted that there were non-reference set genes that adhered to the bias, as defined by the reference set, more strongly than the reference set itself; specifically, those genes with higher CAI scores than those of the reference set. A multi-objective genetic algorithm was utilized to find a reference set that more closely fit the mathematical/theoretical notion of a reference set (a set of genes that defines a bias to which they adhere more strongly than any other genes in the genome) with the intention of producing a better predictor of expressivity. The genetic algorithm successfully accomplished this task. The rankings produced by the GA were closer to the rankings as determined experimentally by microarray expression experiments, and its solution is closer to that of the mathematically defined reference set as measured by rank-fitness (6).

Having an improved method for determining codon adaptation index scores associated with translational efficiency bias will allow researchers to better predict expression rates and to apply this knowledge in such areas as heterologous protein production. Additionally, it will allow researchers to better understand the selective pressure associated with translational efficiency and the way in which this pressure affects genomic evolution.

References


