Algorithmic Techniques Employed in the Isolation of Codon Usage Biases in Prokaryotic Genomes

Douglas W. Raiford III
Wright State University

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

Part of the Computer Engineering Commons, and the Computer Sciences Commons

Repository Citation
https://corescholar.libraries.wright.edu/etd_all/826

This Dissertation is brought to you for free and open access by the Theses and Dissertations at CORE Scholar. It has been accepted for inclusion in Browse all Theses and Dissertations by an authorized administrator of CORE Scholar. For more information, please contact library-corescholar@wright.edu.
Algorithmic Techniques Employed in the Isolation of Codon Usage Biases in Prokaryotic Genomes

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy

By

DOUGLAS W. RAIFORD III
M.S.C.S., Wright State University, 2005

2008
Wright State University

________________________________________________________________________
Michael L. Raymer, Ph.D.
Dissertation Director

________________________________________________________________________
Thomas A. Sudkamp, Ph.D.
Director, Computer Science Ph.D. Program

________________________________________________________________________
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

Committee on Final Examination

________________________________________________________________________
Michael L. Raymer, Ph.D.

________________________________________________________________________
Travis E. Doom, Ph.D.

________________________________________________________________________
Dan E. Krane, Ph.D.

________________________________________________________________________
Thomas Wischgoll, Ph.D.

________________________________________________________________________
Dale E. Courte, Ph.D.
ABSTRACT


While genomic sequencing projects are an abundant source of information for biological studies ranging from the molecular to the ecological in scale, much of the information present may yet be hidden from casual analysis. One such information domain, trends in codon usage, can provide a wealth of information about an organism’s genes and their expression. Degeneracy in the genetic code allows more than one triplet codon to code for the same amino acid, and usage of these codons is often biased such that one or more of these synonymous codons is preferred. Isolation of translational efficiency bias can have important applications in gene expression prediction, heterologous protein production, prediction of organismal lifestyle, and identification of candidates for horizontal gene transfer. Methods for identifying codon usage bias in genomic data that rely solely on genomic sequence data can be confounded by the presence of factors simultaneously influencing codon selection. Presented here are new techniques (deterministic and stochastic) for removing the effects of one of the more common confounding factors, GC(AT)-content, and of analyzing/visualizing the search-space for codon usage bias through the use of a solution landscape. These techniques successfully isolate expressivity-related codon usage trends, using only genomic sequence information, where other techniques fail due to the presence of GC(AT)-content confounding influences. While the disambiguation techniques presented here are for genomes confounded by GC(AT)-content usage trends, these methods should be equally applicable to any other well-characterized confounding bias.
## Contents

1 Background and Significance ........................................... 1
   1.1 Introduction .................................................. 1
       1.1.1 Problem Description ..................................... 2
   1.2 Significance .................................................... 4
   1.3 Background ..................................................... 5
       1.3.1 Protein Production ....................................... 5
       1.3.1.1 DNA, Chromosomes, and the Genome .................. 5
       1.3.1.2 Base-Pairs and Strand Orientation ................. 8
       1.3.1.3 The Gene ............................................. 8
       1.3.1.4 Codons and Degeneration ............................ 9
       1.3.1.5 Central Dogma ...................................... 9
       1.3.2 Prediction of Expressivity and Preferred Codons ......... 10
       1.3.3 More on Transfer RNAs .................................. 13
       1.3.4 On Causes of Codon Usage Bias ......................... 15
       1.3.5 Sequence and Expression Databases ..................... 17
       1.3.5.1 Sequence Searches .................................. 17
       1.3.5.2 Expression Data .................................... 18
       1.3.6 Organization ............................................ 19

2 Literature Review .................................................... 24
   2.1 Bias Discovery Techniques ........................................ 24
       2.1.1 Frequency of preferred codons (FOP) ....................... 24
       2.1.2 Indexes, Statistics, and Clustering ....................... 26
       2.1.2.1 Bennetzen and Hall’s Codon Bias Index (CBI) ........... 26
       2.1.2.2 Correspondence Analysis ............................. 26
       2.1.2.3 P1 and P2 Index .................................... 27
       2.1.3 Codon Preference Bias ..................................... 27
       2.1.4 Clustering ................................................ 28
       2.1.5 Codon Adaptation Index (CAI) ........................... 29
       2.1.6 Scaled $\chi^2$ ........................................... 30
       2.1.7 Effective Number of Codons ............................... 31
       2.1.8 Morton’s Codon Bias Index (CBI) ......................... 33
       2.1.9 Intrinsic Codon Deviation Index (ICDI) .................. 33
       2.1.10 Principal Components Analysis ......................... 34
# CONTENTS

May 27, 2008

2.1.11 CAI Revisited ........................................ 37
2.1.12 Summary ........................................ 38

2.2 Multiple Biases ........................................ 38
   2.2.1 Content Criterion .................................. 40
   2.2.2 Ribosomal Criterion ................................ 40
   2.2.3 Strand Criterion .................................... 41
   2.2.4 Strength Criterion ................................... 41
   2.2.5 Candidates for Horizontal Gene Transfer .......... 42

2.3 Confounding Effects of Other Biases ....................... 45

2.4 Alternative Expression Prediction ......................... 50

2.5 Visualization – Principal Components Analysis .......... 50

2.6 Evolutionary Computation (EC) .......................... 53
   2.6.1 Taxonomy of EC ..................................... 53
   2.6.2 Overview ........................................... 54
   2.6.3 An Example ........................................ 55
   2.6.4 Multi-Objective Problems .......................... 57
      2.6.4.1 Dominance and Pareto Fronts ................. 57
      2.6.4.2 Niching ....................................... 59
   2.6.5 Why EC Works ...................................... 60
   2.6.6 Fitness Landscapes ................................. 60
      2.6.6.1 Relevance to the Search for Translational Efficiency Bias 62
      2.6.6.2 Analysis ....................................... 62
      2.6.6.3 Impact of Representation ...................... 63
      2.6.6.4 Overcoming Adjacency and High-Dimensionality Issues 64

3 Analysis of the Problem Domain: Fitness Landscapes and Novel Visualization Techniques 66
   3.1 Visualizing the Fitness (RSQ) Landscape ................. 67
      3.1.1 Reducing Bit-String High-Dimensionality .......... 67
      3.1.2 Implementing the Solution ......................... 68
         3.1.2.1 Quality (Fitness) Scores ....................... 69
         3.1.2.2 Constructing the Surface ..................... 70
      3.1.3 RSQ Landscape for a High AT-Content Organism .... 70
      3.1.4 A Translational Efficiency Dominated RSQ Landscape 71
      3.1.5 A Strand Bias Dominated RSQ Landscape .......... 76
   3.2 Fitness Landscape Analysis ................................ 82
      3.2.1 Random Walk – Region Near Perfect Solution ........ 82
      3.2.2 Random Walk – Random Starting Location .......... 82
      3.2.3 Ruggedness, Smoothness, Neutrality ................ 84
      3.2.4 What This Tells Us ................................ 86
         3.2.4.1 Mutation Operator and Domain Knowledge ...... 87
   3.3 Genomic Adherence to Specified Bias ..................... 93
      3.3.1 Polar Bias View .................................... 95
   3.4 Conclusion ........................................... 96
# Deterministic Search Methods for Translational Efficiency Bias

## 4 Distinct Biases – Local Search Methods

### 4.1 Localized Descending Plane

#### 4.1.1 Local Search Methods

#### 4.1.2 Hill Climber

### 4.2 Modified SCCI

#### 4.2.1 High-Level Description of Approach

#### 4.2.2 Description of Modified SCCI Algorithm

- **4.2.2.1** Using a High Expression Database to Determine $\alpha$: HEDB Criterion

### 4.3 Microarray Expression Data

### 4.4 Results

#### 4.4.1 Modified SCCI Results

#### 4.4.2 HEDB Genes as Reference Set

#### 4.4.3 Comparison to Microarray Expression Data

#### 4.4.4 Reference Set Quality

#### 4.4.5 HEDB Criteria vs. Ribosomal Criteria

#### 4.4.6 Distance Between Biases and $\alpha$

#### 4.4.7 RSQ Landscape After Adjusting for GC-content Bias

#### 4.4.8 Characteristics of Confounded Genomes

- **4.4.8.1** Original Ribosomal Criterion
- **4.4.8.2** Self-Consistency Level of Ribosomal Protein Coding Genes
- **4.4.8.3** GC-content and Genome Size

### 4.5 Discussion

# Stochastic Search Methods for Translational Efficiency Bias

## 5 Finding a Better Reference Set

### 5.2 Genetic Algorithm

#### 5.2.1 Genetic Algorithm

- **5.2.1.1** Representation, Operators, & Settings
- **5.2.1.2** Rank-Fitness
- **5.2.1.3** Pareto Ranking
- **5.2.1.4** Niching

#### 5.2.2 Microarray Data

#### 5.2.3 Sequence Data

#### 5.2.4 Implementation and Hardware

- **5.2.4.1** Object Oriented
- **5.2.4.2** Distributed/Parallelized Processing
- **5.2.4.3** Other Design Considerations

### 5.2.5 Results

### 5.2.6 Discussion

## 5.3 Finding a Better Set of Weights

### 5.3.1 Genetic Algorithm

- **5.3.1.1** Representation
- **5.3.1.2** Mutation Operator
- **5.3.1.3** Crossover Operator

#### 5.3.2 Repair Operator

#### 5.3.3 Fitness Function

#### 5.3.4 Initializing the Population

### 5.3.5 Results
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.3.6 Discussion</td>
<td>169</td>
</tr>
<tr>
<td>5.4 Self-Consistency Proof</td>
<td>171</td>
</tr>
<tr>
<td>5.5 Summary</td>
<td>175</td>
</tr>
<tr>
<td>6 Conclusions</td>
<td>176</td>
</tr>
<tr>
<td>6.1 Summary of Contributions</td>
<td>176</td>
</tr>
<tr>
<td>6.2 Future Research</td>
<td>178</td>
</tr>
<tr>
<td>6.2.1 Make the Approach Available Online</td>
<td>178</td>
</tr>
<tr>
<td>6.2.2 Extend the Study to Additional Organisms</td>
<td>178</td>
</tr>
<tr>
<td>6.2.3 A Hidden Markov Model (HMM) Approach</td>
<td>178</td>
</tr>
<tr>
<td>6.2.4 Biological Questions</td>
<td>180</td>
</tr>
<tr>
<td>6.2.4.1 Verify Theory of GA Performance</td>
<td>181</td>
</tr>
<tr>
<td>6.2.4.2 Possible Causes of a Distinct Ridge Configuration</td>
<td>181</td>
</tr>
<tr>
<td>6.2.4.3 Investigate <em>Escherichia coli</em> HGT Event</td>
<td>181</td>
</tr>
<tr>
<td>6.2.4.4 Eukaryotes</td>
<td>182</td>
</tr>
<tr>
<td>6.2.5 Algorithmic Research</td>
<td>183</td>
</tr>
<tr>
<td>6.2.5.1 Removal of Multiple Biases, Simultaneously</td>
<td>183</td>
</tr>
<tr>
<td>6.2.5.2 Better Fitness Function</td>
<td>185</td>
</tr>
<tr>
<td>6.3 Bringing it all Together: Flow Chart</td>
<td>185</td>
</tr>
</tbody>
</table>

**Bibliography**

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>A Appendix A: Degeneracy Table</td>
<td>204</td>
</tr>
<tr>
<td>B Appendix B: Description of PCA</td>
<td>205</td>
</tr>
<tr>
<td>B.1 Covariance Matrix</td>
<td>206</td>
</tr>
<tr>
<td>B.2 Eigenvector</td>
<td>206</td>
</tr>
<tr>
<td>B.3 Determinants</td>
<td>207</td>
</tr>
<tr>
<td>B.4 Characteristic Equation and Polynomial</td>
<td>207</td>
</tr>
<tr>
<td>C Appendix : C++ GA Software Design</td>
<td>209</td>
</tr>
<tr>
<td>Figure</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>1.1</td>
<td>Circular DNA</td>
</tr>
<tr>
<td>1.2</td>
<td>DNA and RNA</td>
</tr>
<tr>
<td>1.3</td>
<td>Base-pairing of Nucleotides in Complementary DNA Strands</td>
</tr>
<tr>
<td>1.4</td>
<td>1’ through 5’ Carbon Atoms</td>
</tr>
<tr>
<td>1.5</td>
<td>The Central Dogma of Molecular Biology</td>
</tr>
<tr>
<td>1.6</td>
<td>tRNA Molecule with Amino Acid and Anticodon</td>
</tr>
<tr>
<td>1.7</td>
<td>Typical Annotation for Protein Expressing Gene</td>
</tr>
<tr>
<td>2.1</td>
<td>RSCU Matrix</td>
</tr>
<tr>
<td>2.2</td>
<td>Factor Loading Determination</td>
</tr>
<tr>
<td>2.3</td>
<td>$N_c$ vs. $GC_3$</td>
</tr>
<tr>
<td>2.4</td>
<td>$N_c$-plot for Yeast and Simulated <em>E. coli</em> K12 genes</td>
</tr>
<tr>
<td>2.5</td>
<td>Graphical representation of $X \cdot b_i = Z_i'$</td>
</tr>
<tr>
<td>2.6</td>
<td>Example of Reference Sets Located in Distant Regions of the Codon Usage Space</td>
</tr>
<tr>
<td>2.7</td>
<td><em>Nostoc sp. PCC 7120</em> GC-content</td>
</tr>
<tr>
<td>2.8</td>
<td>Pseudo Code Describing Typical EC Algorithm</td>
</tr>
<tr>
<td>2.9</td>
<td>Example of Crossover and Mutation of a Bit-String</td>
</tr>
<tr>
<td>2.10</td>
<td>Example of a Pareto Front for a Two-Objective Problem</td>
</tr>
<tr>
<td>2.11</td>
<td>Example of a Simple Fitness Landscape</td>
</tr>
<tr>
<td>3.1</td>
<td>Example of Reference Sets Located in Distant Regions of the Codon Usage Space</td>
</tr>
<tr>
<td>3.2</td>
<td><em>Nostoc sp. PCC 7120</em> RSQ Landscape</td>
</tr>
<tr>
<td>3.3</td>
<td>RSQ Landscape for <em>Escherichia coli</em></td>
</tr>
<tr>
<td>3.4</td>
<td>PCA Comparison of Highly Expressed Genes and Horizontally Transferred Genes in <em>E. coli</em></td>
</tr>
<tr>
<td>3.5</td>
<td>RSQ Landscape for <em>Streptomyces coelicolor</em> A3(2): Exhibits Dominant GC-content Bias</td>
</tr>
<tr>
<td>3.6</td>
<td>RSQ Landscape for <em>Neisseria meningitidis</em> MC58</td>
</tr>
<tr>
<td>3.7</td>
<td>RSQ Landscape for <em>Borrelia burgdorferi</em> B31</td>
</tr>
<tr>
<td>3.8</td>
<td>PCA for <em>B. burgdorferi</em> Genome</td>
</tr>
<tr>
<td>3.9</td>
<td>Random Walk in Codon Usage Search Space for <em>E. coli</em> (Started at Summit)</td>
</tr>
<tr>
<td>3.10</td>
<td>Random Walk in Codon Usage Search Space for <em>E. coli</em> (Started at Random Location)</td>
</tr>
<tr>
<td>3.11</td>
<td>Autocorrelation Plot for Random Walk in Codon Usage Search Space for <em>E. coli</em></td>
</tr>
<tr>
<td>3.12</td>
<td>Histogram of Rank-Fitness Values: Random Mutation</td>
</tr>
<tr>
<td>3.13</td>
<td>Average Eigenvalue for Reference Set Eigenvectors</td>
</tr>
<tr>
<td>3.14</td>
<td>Calculation of Distance to Axis</td>
</tr>
<tr>
<td>FIGURE NUMBER</td>
<td>FIGURE DESCRIPTION</td>
</tr>
<tr>
<td>--------------</td>
<td>------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3.15</td>
<td>Axis of Greatest Variance for Reference Set</td>
</tr>
<tr>
<td>3.16</td>
<td>Histogram of Rank-Fitness Values: Local Mutation</td>
</tr>
<tr>
<td>3.17</td>
<td><em>Nostoc</em> sp. <em>PCC 7120</em> SCCI values</td>
</tr>
<tr>
<td>3.18</td>
<td><em>Nostoc</em> sp. <em>PCC 7120</em> Distribution of SCCI values</td>
</tr>
<tr>
<td>3.19</td>
<td>Geometric Representation of Codon Usage Polar View</td>
</tr>
<tr>
<td>3.20</td>
<td>Example of Polar Bias View of Translational and GC bias for <em>Nostoc</em> sp. <em>PCC 7120</em></td>
</tr>
<tr>
<td>3.21</td>
<td><em>Streptomyces coelicolor</em> A3(2) Polar Bias View</td>
</tr>
<tr>
<td>4.1</td>
<td>Localized Descending Plane</td>
</tr>
<tr>
<td>4.2</td>
<td>RSQ Landscape for <em>Streptomyces coelicolor</em> A3(2): Exhibits Dominant GC-content Bias</td>
</tr>
<tr>
<td>4.3</td>
<td>RSQ Landscape for <em>Pseudomonas aeruginosa</em> PAO1</td>
</tr>
<tr>
<td>4.4</td>
<td><em>Chlorobium tepidum</em> TLS HEDB Criteria at Various Alpha Values</td>
</tr>
<tr>
<td>4.5</td>
<td>Association Between HEDB Criteria (mSCCI and Traditional CAI)</td>
</tr>
<tr>
<td>4.6</td>
<td>Distribution of Quality Scores (mSCCI and Traditional CAI)</td>
</tr>
<tr>
<td>4.7</td>
<td>Relationship Between HEDB Criteria and Ribosomal Criteria</td>
</tr>
<tr>
<td>4.8</td>
<td>Relationship Between $\alpha$ and Distance Between Biases</td>
</tr>
<tr>
<td>4.9</td>
<td>Side-by-side Comparisons of RSQ Landscapes for Confounded Genomes</td>
</tr>
<tr>
<td>4.10</td>
<td>Ribosomal protein coding genes in <em>Aeropyrum pernix</em> K1</td>
</tr>
<tr>
<td>4.11</td>
<td>Histogram of RSQ landscape elevation values for <em>Aeropyrum pernix</em> K1</td>
</tr>
<tr>
<td>4.12</td>
<td>RPCGs Residing in Low Elevations vs. Ribosomal Criteria</td>
</tr>
<tr>
<td>4.13</td>
<td>Large Genome-sizes Corresponds to Ease of Isolation</td>
</tr>
<tr>
<td>4.14</td>
<td>Number of Genes vs. GC-content</td>
</tr>
<tr>
<td>5.1</td>
<td>Visualization of Reference Sets</td>
</tr>
<tr>
<td>5.2</td>
<td>Object Oriented Model Utilized in the Genetic Algorithm</td>
</tr>
<tr>
<td>5.3</td>
<td>Distributed Nature of Genetic Algorithm</td>
</tr>
<tr>
<td>5.4</td>
<td>Population on Pareto Front and Entire Population for <em>P. aeruginosa</em></td>
</tr>
<tr>
<td>5.5</td>
<td>Population on Pareto Front and Entire Population for <em>S. coelicolor</em></td>
</tr>
<tr>
<td>5.6</td>
<td>$r_S$ vs. Ribosomal Criterion of Population Members on the Pareto Front</td>
</tr>
<tr>
<td>5.7</td>
<td>Fitness History</td>
</tr>
<tr>
<td>5.8</td>
<td>Fitness History</td>
</tr>
<tr>
<td>5.9</td>
<td>Simulated Annealing</td>
</tr>
<tr>
<td>5.10</td>
<td>Fitness History</td>
</tr>
<tr>
<td>5.11</td>
<td>$r_S$ History</td>
</tr>
<tr>
<td>5.12</td>
<td>Comparison of strength and ribosomal criteria (EDHE)</td>
</tr>
<tr>
<td>5.13</td>
<td>Comparison of strength and ribosomal criteria (HEDB)</td>
</tr>
<tr>
<td>5.14</td>
<td>PCA Summary Performance: $GA_{EDHE}$ and $CAI_{EDHE}$</td>
</tr>
<tr>
<td>5.15</td>
<td>PCA Summary Performance: $GA_{HEDB}$ and $CAI_{HEDB}$</td>
</tr>
<tr>
<td>6.1</td>
<td>Three Tiered Architecture</td>
</tr>
<tr>
<td>6.2</td>
<td>Various Rates Associated with Expressivity</td>
</tr>
<tr>
<td>6.3</td>
<td>Flow Chart</td>
</tr>
<tr>
<td>B.1</td>
<td>RSCU Matrix</td>
</tr>
<tr>
<td>B.2</td>
<td>Graphical representation of $X \cdot b_1 = Z_1'$</td>
</tr>
<tr>
<td>C.1</td>
<td>Unified Modeling Language (UML) Chart for C++ GA Code</td>
</tr>
</tbody>
</table>
List of Tables

1.1 RNA Triplet Codons to Amino Acid Translation ........................................... 10
1.2 Pairing at the Third or “Wobble” Position of the Codon .............................. 14
2.1 Summary of All Described Bias Measures .................................................... 39
4.1 Criteria for CAI, SCCI, and modified SCCI Runs on GC-Confounded Organisms ........ 111
4.2 Criteria for CAI, SCCI, and modified SCCI Runs on AT-Confounded Organisms .................. 112
4.3 Criteria for CAI, SCCI, and Modified SCCI Runs on Un-Confounded Organisms .......... 113
4.4 Correlations Between Expression Data and Bias Measures (Confounded Organisms) ........... 116
4.5 Correlations Between Expression Data and Bias Measures (Non-Confounded Organisms) .. 116
5.1 Correlation, Fitness, and Criterion Data ...................................................... 144
5.2 Performance Measurements for Genetic Algorithms (Un-Confounded Organisms) ............ 159
5.3 Performance Measurements for Genetic Algorithms (Confounded Organisms) .................. 160
5.4 Performance Measurements for Deterministic Algorithms (Un-Confounded Organisms) ...... 161
5.5 Performance Measurements for Deterministic Algorithms (Confounded Organisms) ............. 162
5.6 p-values for t-Tests Between Criteria of GA and Traditional CAI ........................... 169
5.7 Four Genes With No Perfect Solution ....................................................... 173
A.1 Degeneracy Table .......................................................................... 204
Acknowledgements

Throughout this endeavor I have been assisted, guided, and encouraged by many individuals and institutions. First, I would like to acknowledge and express my gratitude to Drs. Michael Raymer, Travis Doom, and Dan Krane, for their insights, direction, and support during my stay at Wright State University. I also would like to thank the Wright State University Research Council (research incentive program for 2007), the Dayton Area Graduate Studies Institute (DAGSI), and the School of Graduate Studies for their generous financial assistance. Finally, my sincere thanks go to the members of Wright State’s Bioinformatics Research Group (Paul Anderson, Gina Cooper, Ryan Flynn, Jason Gilder, Amanda Hanes, Akash Jain, Benjamin Kelly, David Paoletti, Michael Peterson, Sridhar Ramachandran, and Deacon Sweeney) for their encouragement and unselfish assistance.
Dedicated to the pursuit of truth.
The secrets of the universe await us.
Background and Significance

1.1 Introduction

Analysis of sequence data can provide valuable insight into the molecular makeup and evolutionary history of a species. Genomic databases have proven a fertile research ground for finding new genes (Zhang, 2002), identifying the ancestral origins of genes and other sequences, and sorting out the phylogenetic relationships among species (Swofford, 2000; Sokal and Michener, 1958). Recent research has suggested that one potentially valuable source for information regarding coding regions lies in the inherent degeneracy of the genetic code. Since most amino acids are represented by more than one codon (triplet of nucleotides, see Appendix A for a cross reference chart of codons to amino acids), preferential usage of specific codons within genes can provide insight into the origins of genes and even their role within the cell. It is well established that organisms preferentially utilize one or more of the available synonymous codons for each amino acid (Grantham et al., 1980, 1981; Ikemura, 1981a; Sharp et al., 1988). This codon usage bias has been used to help identify horizontally transferred and other recently acquired genes within species (Garcia-Vallvé et al., 2000; Chou and Zhang, 1995, 1994), and has been shown to correlate with the degree to which the gene is expressed for many prokaryotic species (Gouy and Gautier, 1982; Ikemura, 1981b; Sharp and Li, 1986) and some low-order eukaryotes (Sharp et al., 1988; Carbone et al., 2003). Expressivity in turn has been shown to be related to a variety of characteristics including specific amino acid selection (Akashi and Gojobori, 2002; Heizer Jr. et al., 2006).
Selective pressure to enhance translational efficiency is thought to be one of the underlying causes of this bias, and it is this translational efficiency bias that is related to gene expressivity (Ikemura, 1981b; Sharp and Li, 1986). Since the transfer RNAs associated with each codon vary in their relative abundance, efficiencies in translation can be realized when the tRNA of highest abundance is employed. This is known as the tRNA adaptation theory (Garel et al., 1970; Chavancy and Garel, 1981). An additional benefit realized when preferred codons are used is an increase in the accuracy (as much as ten fold) of translation (Precup and Parker, 1987).

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 (genes introduced into a host organism’s genome for the purpose of producing a given protein). 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.

1.1.1 Problem Description

Biases associated with translational efficiency are not the only biases found in prokaryotic and small eukaryotic genomes. Codon usage can also be affected by tendencies toward high or low GC-content as well as strand bias (Carbone et al., 2003, 2005). Strand bias is brought on during the processes of transcription and replication. The transcription-induced bias is thought to be caused by increased deamination of cytosine leading to C→T transitions on the non-transcribed strand (Francino et al., 1996; Francino and Ochman, 1997; Beletskii and Bhagwat, 1998). The replication-induced bias is thought to be the result of the discontinuous process for replicating the lagging strand where Okazaki fragments are synthesized and then joined. This process results in the leading strand being richer in G+T than the lagging strand (Lafay et al., 2000, 1999; Rocha et al., 1999). In some cases these biases (content and strand) can coexist with translational efficiency bias (Grocock and Sharp, 2002; Carbone et al., 2005). When this occurs the translation-driven codon usage bias
can be obscured, making gene expression levels difficult to predict.

Several approaches have been employed to identify and measure codon usage biases including: frequency of preferred codons (FOP) (Ikemura, 1981a), codon bias index (CBI) (Bennetzen and Hall, 1982), P1 and P2 indices (Gouy and Gautier, 1982), codon preference bias (McLachlan et al., 1984), scaled $\chi^2$ (Shields and Sharp, 1987; Shields et al., 1988), effective number of codons (Wright, 1990), intrinsic codon deviation index (ICDI) (Freire-Picos et al., 1994), major codon usage (MCU) (Kanaya et al., 1996, 1999) and codon adaptation index (CAI) (Sharp and Li, 1987; Carbone et al., 2003). Some methods, such as codon adaptation index (Sharp and Li, 1987) and frequency of preferred codons (Ikemura, 1981a), require prior knowledge of a set of genes known to be highly expressed. Others, such as the updated frequency of preferred codons determination method now termed major codon usage (Kanaya et al., 1996, 1999), and the automated codon adaptation index algorithm (CAI algorithm) (Carbone et al., 2003) attempt to identify the bias using coding sequence information only. Algorithms that take the latter approach (using sequence information only) tend to find the dominant bias in the genome. If the intent is to isolate translational efficiency bias then this search can be confounded by other biases that exist within the target genome (e.g. GC- or strand bias) (Carbone et al., 2003, 2005; McHardy et al., 2004). Identified biases may not be those associated with translational efficiency.

This is an important shortcoming of automated approaches to identifying translational efficiency bias. They are susceptible to the confounding influences of GC- and strand bias. It follows that when an algorithmic approach to isolating translational efficiency bias is employed, the solution must be checked to ensure that the correct bias has been identified. This is usually accomplished by sorting all of the genes in the genome according to their codon usage relative to the identified bias and then identifying where in the sorted order “known highly expressed” genes reside. If the isolation method does not place these genes high in the listing then the gene ranking cannot be trusted, and a different approach must be employed. This is usually in the form of using these same “known highly expressed” genes along with one of the above methods that can use these genes to directly calculate codon usage bias.
A well-formed solution to the issue of a confounded search for translational efficiency bias will be one that moves the state of the art to a condition where, given sequence information alone, translational efficiency bias can be isolated and gene expressivity accurately predicted, even in the presence of confounding factors. For the purposes of this research the attention is focused on the removal of the confounding effects of GC(AT)-content. This is due to the fact that GC(AT)-content appears to be the most common example of a confounding bias. 40 organisms studied in Carbone et al. (2005) had a dominant bias that was something other than translational efficiency. 35 of these were dominant for GC(AT)-content.

1.2 Significance

Various studies rely upon the detection of codon usage bias to predict translational efficiency and thereby expressivity. They include such analyses as whether metabolic efficiency is a source of selection in prokaryotic organisms (Akashi and Gojobori, 2002; Heizer Jr. et al., 2006; Raiford III, 2005; Heizer Jr., 2005; Raiford III et al., 2007). These studies compared amino acid biosynthetic cost with protein production rates and established that natural selection favors substitutions that result in the utilization of less energetically costly amino acids where possible in highly expressed genes. Another use of translational efficiency bias prediction is the improvement of heterologous protein production (Grote et al., 2005; Hannig and Makrides, 1998). 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 (Carrondo and Fussenegger, 2005). Additionally, it has been shown that deviation from use of optimal codons can regulate ribosomal translation rate (Lawrence and Hartl, 1991). In particular, regions of a gene’s sequence occurring between domains tend to deviate from optimal codon usage in a way that causes a ribosomal-mediated translational pause (Thanaraj and Argos, 1996). It is hypothesized that this allows each domain of the protein time to fold before translation continues.
1.3. BACKGROUND

Knowledge of the need for this inter-domain decrease in optimal codon usage must be taken into account when modifying the sequence of an introduced gene for the purposes of heterologous protein production.

When translational efficiency bias cannot be isolated due to confounding factors the above research becomes impacted. Erroneous correlations can be identified in the metabolic efficiency studies (Rocha et al., 1999), and heterologous protein production can be adversely affected.

1.3  Background

1.3.1  Protein Production

In order to examine the confounding factors in the translational efficiency bias discovery process it is necessary to have a clear understanding of the protein synthesis process. Proteins are tied in a very specific way to an organism’s genome. Production of each protein is directed by one of the organism’s genes. The following sections will describe this process in enough detail to enable understanding of the subsequent analysis and computations.

1.3.1.1  DNA, Chromosomes, and the Genome

The term genome refers to a complete set of chromosomes from a single species with its associated genes. In microbial organisms there is usually a single, circular chromosome (Figure 1.1), and in some cases, additional, smaller circular strands of DNA called plasmids. Depending upon the application, plasmids are sometimes included in what is deemed a prokaryote’s genome. In this analysis, plasmids are not included as part of the organism’s genome. An organism’s chromosomes are, essentially, long DNA molecules. DNA, or deoxyribonucleic acid, is the now well-known double-helix molecule (Figure 1.2) that encodes an organism’s heritable traits (Watson and Crick, 1953).
1.3. BACKGROUND

Figure 1.1: Circular DNA. In prokaryotes chromosomes are composed of a single, circular strand of DNA. Images reproduced from
and
http://gsbs.utmb.edu/microbook/images/fig5_8.jpg
respectively

While a single molecule of DNA is associated with a prokaryotic chromosome, DNA itself is comprised of constituent building blocks known as nucleotides (Figure 1.3). The four nucleotides that make-up a DNA molecule are each composed of a phosphate group, a ribose sugar, and a nitrogenous base. The nucleotides are differentiated from each other by the type of base that each contains. They are guanine, adenine, thymine, and cytosine (Figure 1.2). The abbreviations G, A, T, and C are often used to describe the nucleotides (Miescher, 1897; Nirenberg and Matthaei, 1961; Nirenberg et al., 1963; Fischer, 1897, 1898, 1899).
Figure 1.2: Overview of composition and structure of DNA and RNA. Reproduced from (Access Excellence, 2005). Image resides at URL: http://www.accessexcellence.org/RC/VL/GG/rna2.html
1.3. BACKGROUND

1.3.1.2 Base-Pairs and Strand Orientation

Cellular DNA usually exists as a two-stranded molecule, where the strands are held together by weak molecular attraction (hydrogen bonds) between pairs of nucleotides. Specifically, a guanine on one strand will always pair with a cytosine on the opposite strand. Likewise, thymine is always paired with adenine (Figure 1.3). For this reason the strands are known as complements. These complementary nucleotides (or bases) are known as base-pairs (Chargaff, 1950; Watson and Crick, 1953; Franklin and Gosling, 1953).

Since nucleotides are oriented in opposite directions from one strand to the other, one strand is said to be the reverse complement of the other. Organic chemists use the orientation of the carbon atoms in the ribose sugars to differentiate the strands. The carbon atoms are designated 1’ through 5’ (pronounced one prime through five prime) (Figure 1.4). Nucleotides in one strand of the DNA molecule are aligned such that the 5’ carbon of one nucleotide is connected by a phosphodiester bond with the 3’ carbon of the next. If one strand of the DNA molecule is oriented from its 3’ end to its 5’ end, the adjacent strand will be oriented in the opposite direction (Watson and Crick, 1953).

Most biological reactions take place in the 5’ to 3’ direction. It is for this reason that the convention for describing a molecule of DNA is to list the nucleotides on one strand in the 5’ to 3’ direction. With this knowledge a molecule of DNA can be exactly described by a string of letters, each an abbreviation representing one nucleotide (Garrett and Grisham, 1995, pp. 191-193) (e.g. GATATTAT...).

1.3.1.3 The Gene

For the purpose of this investigation a gene is considered to be a sequence of nucleotides within a strand of DNA that contains the information needed for the synthesis of a protein. Bacteria, or prokaryotic organisms, generally have a single chromosome with only a few thousand genes.
Eukaryotes are much more complex. They are distinguished from prokaryotes by the existence of a cellular membrane separating the nucleus from the rest of the cell (Woese and Fox, 1977). The research in this dissertation pertains to the genomes found in microbial, prokaryotic organisms. For this reason the discussion will be limited to the genomic composition of these organisms.

### 1.3.1.4 Codons and Degeneration

As stated previously, proteins are chains of amino acids. There are twenty different amino acids that can combine to make up these chains. It has been noted that proteins are synthesized by following instructions encoded in genes. How does a gene, with its four constituent nucleotides, encode a protein with its twenty component amino acids? Each amino acid is coded by a triplet of consecutive nucleotides, called a codon (Table 1.1 and Appendix A). There are sixty-four different triplet combinations formed by the four possible nucleotides ($4^3 = 64$). This means that some codons code for the same amino acids as other codons. This is known as degeneracy in the genetic code (Nirenberg and Matthaei, 1961; Martin et al., 1962; Matthaei et al., 1962; Jones and Nirenberg, 1966; Nirenberg et al., 1963).

In prokaryotes the beginning and end of the contiguous set of nucleotides that form the coding sequence of a gene is easily recognized. The coding region generally begins with the triplet codon: adenine, thymine, and guanine (ATG). The tail-end of a coding region is identified by the combinations TAA, TAG, or TGA, also known as stop codons (Jones and Nirenberg, 1966). Genes are always long enough that the absence of intervening stop codons is statistically unusual.

### 1.3.1.5 Central Dogma

Protein production is a multi-step process. There are two major steps: transcription and translation. Transcription is the process of synthesizing a complementary copy of the DNA in the form of
1.3. BACKGROUND

The process of generating a protein through transcription and translation (Figure 1.5) is stated by the central dogma of molecular biology (Crick, 1958).

1.3.2 Prediction of Expressivity and Preferred Codons

Because organisms require different concentrations of different proteins, not all genes are transcribed and translated (expressed) at the same rate. The expressivity of a gene is a measure of that

<table>
<thead>
<tr>
<th>Leucine</th>
<th>Serine</th>
<th>Arginine</th>
<th>Valine</th>
<th>Alanine</th>
<th>Glycine</th>
<th>Proline</th>
<th>Threonine</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUA</td>
<td>UCU</td>
<td>CGU</td>
<td>GUU</td>
<td>GCU</td>
<td>GGU</td>
<td>CCU</td>
<td>ACU</td>
</tr>
<tr>
<td>UUG</td>
<td>UCC</td>
<td>CGC</td>
<td>GUC</td>
<td>GCC</td>
<td>GCC</td>
<td>CCC</td>
<td>ACC</td>
</tr>
<tr>
<td>CUU</td>
<td>UCA</td>
<td>CGA</td>
<td>GUA</td>
<td>GCA</td>
<td>GGA</td>
<td>CCA</td>
<td>ACA</td>
</tr>
<tr>
<td>CUC</td>
<td>UCG</td>
<td>CGG</td>
<td>GUG</td>
<td>GCC</td>
<td>GGC</td>
<td>CCC</td>
<td>ACG</td>
</tr>
<tr>
<td>CUA</td>
<td>AGU</td>
<td>AGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUG</td>
<td>AGC</td>
<td>AGG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Isoleucine</th>
<th>Stop</th>
<th>Phenylalanine</th>
<th>Aspartate</th>
<th>Histidine</th>
<th>Glutamine</th>
<th>Glutamate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUU</td>
<td>UGA</td>
<td>UUU</td>
<td>GAU</td>
<td>CAU</td>
<td>CAA</td>
<td>GAA</td>
</tr>
<tr>
<td>AUC</td>
<td>UAA</td>
<td>UUC</td>
<td>GAC</td>
<td>CAC</td>
<td>CAG</td>
<td>GAG</td>
</tr>
<tr>
<td>AUA</td>
<td>UAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Asparagine</th>
<th>Lysine</th>
<th>Cysteine</th>
<th>Tyrosine</th>
<th>Tryptophan</th>
<th>Methionine</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAU</td>
<td>AAA</td>
<td>UGU</td>
<td>UAU</td>
<td>UGG</td>
<td>AUG</td>
</tr>
<tr>
<td>AAC</td>
<td>AAG</td>
<td>UGC</td>
<td>UAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 1.1: RNA Triplet Codons to Amino Acid Translation

<table>
<thead>
<tr>
<th>Leucine</th>
<th>Serine</th>
<th>Arginine</th>
<th>Valine</th>
<th>Alanine</th>
<th>Glycine</th>
<th>Proline</th>
<th>Threonine</th>
</tr>
</thead>
<tbody>
<tr>
<td>UUA</td>
<td>UCU</td>
<td>CGU</td>
<td>GUU</td>
<td>GCU</td>
<td>GGU</td>
<td>CCU</td>
<td>ACU</td>
</tr>
<tr>
<td>UUG</td>
<td>UCC</td>
<td>CGC</td>
<td>GUC</td>
<td>GCC</td>
<td>GCC</td>
<td>CCC</td>
<td>ACC</td>
</tr>
<tr>
<td>CUU</td>
<td>UCA</td>
<td>CGA</td>
<td>GUA</td>
<td>GCA</td>
<td>GGA</td>
<td>CCA</td>
<td>ACA</td>
</tr>
<tr>
<td>CUC</td>
<td>UCG</td>
<td>CGG</td>
<td>GUG</td>
<td>GCC</td>
<td>GGC</td>
<td>CCC</td>
<td>ACG</td>
</tr>
<tr>
<td>CUA</td>
<td>AGU</td>
<td>AGA</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CUG</td>
<td>AGC</td>
<td>AGG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Isoleucine</th>
<th>Stop</th>
<th>Phenylalanine</th>
<th>Aspartate</th>
<th>Histidine</th>
<th>Glutamine</th>
<th>Glutamate</th>
</tr>
</thead>
<tbody>
<tr>
<td>AUU</td>
<td>UGA</td>
<td>UUU</td>
<td>GAU</td>
<td>CAU</td>
<td>CAA</td>
<td>GAA</td>
</tr>
<tr>
<td>AUC</td>
<td>UAA</td>
<td>UUC</td>
<td>GAC</td>
<td>CAC</td>
<td>CAG</td>
<td>GAG</td>
</tr>
<tr>
<td>AUA</td>
<td>UAG</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Asparagine</th>
<th>Lysine</th>
<th>Cysteine</th>
<th>Tyrosine</th>
<th>Tryptophan</th>
<th>Methionine</th>
</tr>
</thead>
<tbody>
<tr>
<td>AAU</td>
<td>AAA</td>
<td>UGU</td>
<td>UAU</td>
<td>UGG</td>
<td>AUG</td>
</tr>
<tr>
<td>AAC</td>
<td>AAG</td>
<td>UGC</td>
<td>UAC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

a ribonucleic acid molecule (termed messenger RNA or mRNA). The second major step in the process, translation, uses the mRNA as a template to create the amino acid chain that ultimately folds into the resultant protein. This is accomplished by a ribosome that attaches to the mRNA and builds a chain of amino acids based upon the codons (triplets of nucleotides) in the mRNA chain. The ribosome does not synthesize the amino acids but rather uses amino acids that it encounters in its environment. This is facilitated by another form of RNA known as transfer RNA (tRNA) (Hoagland et al., 1958, 1957; Nirenberg and Matthaei, 1961). The process of generating a protein through transcription then translation (Figure 1.5) is stated by the central dogma of molecular biology (Crick, 1958).
gene’s protein production rate. Direct measurement of this rate is non-trivial though there are several techniques for estimating it (Munoz et al., 2004). These include such techniques as microarray technology (Conway and Schoolnik, 2003; Pease et al., 1994; Brown and Botstein, 1999; Duggan et al., 1999; Nagpal et al., 2004; Romualdi et al., 2003; Asyali et al., 2004), sequential analysis of gene expression (SAGE) (Velesculescu et al., 1995), various spinoffs of SAGE (Datson et al., 1999; Gowda et al., 2004; Vilain et al., 2003), enzymatic fragmentation fingerprints (Shimkets et al., 1999), polymerase chain reaction (PCR) amplification (Uematsu et al., 2001), RNAi library analysis (Shirane et al., 2004), and EST abundance (Gitton et al., 2002; Skrabanek and Campagne, 2001; Mu et al., 2001; Sorek and Safer, 2003). These methods are relatively expensive in terms of time and reagent cost. Additionally, they estimate the expression rate by determining mRNA or protein abundance. Protein and mRNA abundances are impacted by many factors including degradation and transport rates, therefore, they do not directly measure the underlying rates of production. Due to the difficulty in directly measuring expressivity, scientists have long sought to determine alternate methods of predicting protein production rates. One such method that has gained recent attention is the examination of gene sequence data for indicators of production rates.

Early research noted that genes with high expression rates tend to exhibit a bias in their choice of codons (Gouy and Gautier, 1982). Most amino acids have several codons that code for them (for example, each of the codons GUU, GUC, GUA, and GUG code for the amino acid Valine) (Table 1.1 and Appendix A). In the absence of any bias one would expect each of the codons to be utilized in equal numbers. In highly expressed genes there tends to be a strong bias in codon usage in prokaryotes. Those codons that tend to be used preferentially are known as optimal, preferred, or major codons (Ikemura et al., 1980; Ikemura, 1981a,b, 1985).

Identification of preferred codons through sequence analysis allows for the prediction of a gene’s expressivity by examining that gene’s degree of usage of these codons (Sharp and Li, 1987). The more a gene adheres to this usage the more likely it is to be highly expressed. There have been many methods proposed for estimating a gene’s adherence to codon usage bias (Shields et al., 1988; Wright, 1990; Morton, 1993; Freire-Picos et al., 1994; Sharp and Li, 1987; Ikemura, 1981b;
1.3. BACKGROUND

Gouy and Gautier, 1982; Carbone et al., 2003). For the purposes of this dissertation codon adaption index (Sharp and Li, 1987) has been chosen as a starting point for the research contained herein and as a baseline against which to compare any results. It has several advantages with respect to the goals of this research. It has been shown to be one of the better predictors of expressivity, due in part to its use of a scaled measure (Friberg et al., 2004) for codon adaptiveness. Prior to CAI the more popular measurements were essentially binary – either the codon in question was optimal or it was not. There was no gradation. CAI assigns a value of between 0 and 1 as an indication of the codon’s relative adaptiveness. Additionally, there is an existing method of determining CAI that is purely algorithmic (Carbone et al., 2003). This automated approach works by first searching for a reference set (small set of genes that define a bias to which the reference set itself adheres more strongly than all other genes in the genome), and then using that reference set in the same way as traditional CAI did with its set of known highly expressed genes. This will be discussed in detail in chapter 2.

The traditional Sharp and Li (1987) method requires prior knowledge of a set of known highly expressed genes in order to compute the gene CAI scores. While the newer Carbone et al. (2003) automated technique removes this requirement, it is designed to locate the dominant bias for the organism, which is not necessarily translational efficiency bias. When the intent is to locate translational efficiency bias, a check must be performed where CAI scores of known highly expressed genes are evaluated to ensure that they fall in the upper regions of the distribution of all CAI scores (for the entire genome). So, once again, a set of known highly expressed genes is required to complete the bias isolation. But, at its core, the Carbone et al. (2003) algorithm is a search routine for a reference set that adheres to certain mathematically defined characteristics. This makes it amenable to improvement through the application of alternative search techniques, optimization approaches, and altered search-criteria – all of which are opportunities that are beneficial with respect to this research.

A major contribution of this work is the removal of the need for prior knowledge of a set of highly expressed genes for automated identification of translational efficiency bias. Before
1.3. BACKGROUND

May 27, 2008

describing how this is accomplished, a treatment of existing methods and other research in this area is presented in the following literature review (Chapter 2).

1.3.3 More on Transfer RNAs

Earlier it was suggested that efficiencies in translation can be realized when the tRNA of highest abundance is utilized. As with many biological systems, this idea is more complicated than it at first seems. First, not all tRNA’s are created equal – some tRNAs are inherently more efficient when being incorporated into the ribosomal complex (Thomas et al., 1988). Second, one tRNA molecule can serve the needs of more than one codon (Crick, 1966). The inherent efficiency of some tRNA’s is a direct result of this multiple-codon servicing phenomenon. This can begin to be understood by examining the anatomy of a tRNA molecule (Figure 1.6). Transfer RNA has at least two important regions within its sequence. The first is the 3’ aminoacyl acceptor. It is in this region that a bond is formed between the tRNA and an amino acid. Each tRNA has an affinity for a specific amino acid and it is in this way that tRNA transports the amino acid to the ribosome for translation. The other region of importance is the anticodon loop. When the ribosome is processing a given codon within the messenger RNA it will bring in a tRNA with a complementary anticodon (Figure 1.6). This determines the next link in the molecular chain of amino acids that is the protein.

In 1966 Crick (1966) noted that, while there were 61 possible amino acid coding triplets, there were fewer than 61 species of tRNA represented in any given organism’s translational apparatus. He postulated that “in the base-pairing of the third base of the codon there is a certain amount of play, or wobble, such that more than one position of pairing is possible. In other words, a single aminoacyl-tRNA (a charged tRNA, or a tRNA that is carrying its associated amino acid) could serve as the amino acid carrier for more than one codon. Crick went on to characterize the bonding energy of all pairwise combinations of nucleotides in the third, or “wobble” position, and determined which groupings were possible, and even probable (Table 1.2).
Table 1.2: Pairing at the Third or “Wobble” Position of the Codon. Inosine (I) is the nucleotide of the base hypoxanthine, a deaminated adenine. Deamination is the removal of an amine group (nitrogen) from a molecule. Often it is the only tRNA for the Ile codon family, which has three codons (ATA, ATC, and ATU).

Given these pairings, an organism’s genome could have as few as 32 tRNA species, and still service all 61 amino acid coding triplets. As an example of this reduced representation, *E. coli* has two tRNA’s that carry the alanine amino acid, one with the anticodon GGC and the other with the anticodon UGC (reverse complements of GCC and GCA, respectively) (Lowe and Eddy, 1997). There are, however, four codons that code for alanine (GCA, C, G, and U). The implication (according to Table 1.2) is that the tRNA with anticodon GGC serves both the GCC and GCU codons, and the UGC tRNA serves both the GCA and GCG codons.

Given the existence of only two species of tRNA serving the needs of four alanine codons (in *E. coli*), and that only one of the aminoacyl-tRNAs can be more abundant than the other, the question arises, which of the codons would be preferred? Might two codons be equally preferred? Thomas et al. (1988) determined that the ribosome programmed with the codon that represents the perfect Watson-Crick pairing (versus the wobble-pairing) tends to react faster with the associated tRNA. Additionally, these pairings have been shown to exhibit up to 10-fold differences in accuracy of translation (Precup and Parker, 1987). Thus, choice of the appropriate Watson-Crick pairing codon can provide translational efficiency through access to the more abundant tRNA and through the naturally occurring increased tRNA-to-ribosomal incorporation speed and increased accuracy (and therefore less need for proofreading and repair). Counter to the intuition that increased speed might cause more errors, selection for a preferred codon can provide both increased speed and
1.3. BACKGROUND

May 27, 2008

1.3.4 On Causes of Codon Usage Bias

Related to the tRNA adaptation theory is the homogeneity hypothesis. The tendency for an organism to show evidence of translational efficiency bias is greater in fast growing organisms than in those that exhibit slow growth (Carbone et al., 2003; Lafay et al., 2000). Competition during periods of exponential growth is thought to contribute to selection for codon usage associated with enhanced translational efficiency, while an absence of such growth periods could explain a lack, or lessening, of such bias (Lafay et al., 2000; Bulmer, 1987; G. and G., 1997).

More recently, the co-evolution of codon usage bias and the propensity for certain codon associated tRNA's to exhibit higher abundances than their synonymous codon counterparts has evolved into a discussion of population size and drift in a theory known as selection-mutation-drift (Bulmer, 1991; Rocha, 2004). The expression regulation theory (Grosjean and Fiers, 1982; Konigsberg and Godson, 1983; Walker et al., 1984; Hinds and Blake, 1985) has been proffered as an explanation for why rare codons are used in weakly expressed genes. The theory is that they act as a mechanism for keeping the gene’s expression low. Additionally, it has been found that requirements for elongation speed and the fact that different codons have different translation rates plays into this evolutionary process (Varenne et al., 1984; Srensen et al., 1989). This is based, in part, on the neutral theory of molecular evolution (Kimura, 1979, 1983; King and Jukes, 1969).

The neutral theory of molecular evolution “holds that at the molecular level most evolutionary change and most of the variability within species are not caused by Darwinian selection but by random drift of mutant alleles that are selectively neutral or nearly neutral (Kimura, 1983, pg. 34).” Some have taken this to be a direct attack upon evolutionary principles put forward by Darwin (possibly due to the inflammatory title “Non-Darwinian Evolution” of the King and Jukes (1969) article). The neutral theory does not, however, assert that natural selection is non-existent; simply that most of the variability within a population is from random, neutral mutations. Kimura cites as
1.3. BACKGROUND

May 27, 2008

evidence of this the fact that while highly conserved proteins (due to natural selection) show a very low mutation rate, if this same rate is assumed to exist throughout the genome (a very conservative assumption) then a typical mammalian population’s genome will exhibit, conservatively, one nucleotide change every two years from neutral (or nearly neutral) substitutions. This is a much higher rate of mutation than previously thought to exist (Haldane, 1957). Were these substitutions to be non-neutral, then no mammalian species could tolerate such a substitutional load, hence, they must be mostly neutral substitutions. Many of the precepts of neutral theory have become incorporated into the consensus understanding of evolutionary principals, though there is still debate over semantics and degree.

The large number of neutral substitutions leads to a phenomenon known as genetic drift. This can be compared to the concept of entropy in physics. In the absence of selective pressure the genomic content tends to drift randomly away from the fixed content at a specific point in time. In this way, drift and natural selection can be thought of as competing forces. The more important the genomic content is to survival, the stronger the selective force exerted on that region of the genome, and the weaker the exhibition of drift, while regions where substitutions can be tolerated will exhibit more drift and less selective conservation. This tendency is employed to explain the variation in codon usage bias throughout a genome. Genes that are highly expressed benefit from adherence to the translational efficiency bias and will come under selective pressure to maintain this bias. The less highly expressed a gene is, the less it will experience this pressure, leaving open the possibility of genetic drift away from adherence to the bias, giving rise to the term selection-mutation-drift (Bulmer, 1991; Rocha, 2004). It is even thought that genetic drift could bring a gene into partial adherence to a beneficial bias such that the accumulated substitutions could provide enough selective advantage for natural selection to begin to take effect.
1.3.5 Sequence and Expression Databases

1.3.5.1 Sequence Searches

As genomes (and their respective proteomes) are sequenced, they are placed in publicly accessible repositories so that researchers can easily retrieve and analyze the information. One such repository is the National Center for Biotechnology Information (NCBI, 2005). To date, 679 microbial genomes have been sequenced and listed on NCBI’s web site. These postings are in the form of annotated text files with a listing of the organism’s entire genome along with individual descriptions of genes and where they reside (Figure 1.7).

Simply publishing the annotated files to a web site does not support the needs of the research community. Often there is a need for the ability to search for the existence of a particular type of gene or protein, and individually posted text files do not support this requirement. An investigator may know the sequence of protein in one organism, and wish to determine whether another organism produces a similar protein (known as a homolog and described as being homologous). The proteome could be searched for an exact match for the protein but it is unlikely that such a search would be successful – even very closely related organisms have differences in the sequences of related proteins. Over time genes accumulate mutations so that one must be able find sequences that are “similar” (as opposed to an exact match) if one is to be able to search for homologous proteins in other organisms. The problem is compounded by the fact that there could be insertions and deletions in addition to the substitutions, making the ability to align sequences a very important part of any sequence search. A search of a database of protein sequences that yields the best alignment will almost certainly be the best match. A good alignment is one where gaps are strategically placed in the sequences so that the query and target sequences look the most similar at the aligned positions. Needleman and Wunsch (1970) devised a dynamic programming solution to the problem. The Needleman-Wunsch approach is described as a global alignment. The entire query sequence is aligned to the full target sequence. If, instead of individual genes or proteins, the database is comprised of the complete chromosomal sequence, including the regions between
genes, an alternative approach must be used. In this case one would wish to identify regions of good local alignment. A local alignment finds the best matching subsequence. An example of a local alignment algorithm is the Smith-Waterman 1981 algorithm.

The problem with both the Needleman-Wunsch or Smith-Waterman approaches is their computational complexity. The dynamic programming technique employed has a complexity of $O(mn)$ where $m$ is the length of the query and $n$ is the length of the target sequence. While this is polynomial time, it is still computationally expensive when even a small prokaryotic genome is the target (millions of base-pairs in length), and often you might wish to search across many genomes (or proteomes). A more efficient search is required. The basic local alignment search tool (or BLAST) implements such an algorithm (Altschul et al., 1990). BLAST finds areas of interest [using a deterministic finite automaton approach (Mealy, 1955)], and then performs alignments in these areas. Generally, searches for proteins yield better results. This is due in part to the larger alphabet (20 amino acids verses 4 nucleotides). A match of a given length in a protein sequence is more rare than the same length match in genomic data. Also, the various chemical properties of the amino acids can be used to characterize some substitutions as being more likely than others.

### 1.3.5.2 Expression Data

In much the same way that sequence data is published, expression data is also publicly available through NCBI’s gene expression omnibus (GEO) (Barrett et al., 2007). GEO allows the uploading of experimental data from most of the high-throughput and parallel molecular abundance-measuring technologies. These include microarray-based and non-array-based techniques [such as serial analysis of gene expression (SAGE), massively parallel signature sequencing (MPSS), and some peptide profiling techniques such as tandem mass spectrometry (MS/MS)]. Generally the data are in the form of comparisons between treated and untreated samples at various time-points. It is important to remember that these techniques estimate expression rates by determining mRNA or protein abundance. They do not directly measure the underlying rates of production.
1.3. **BACKGROUND**

May 27, 2008

1.3.6 **Organization**

This dissertation is organized into six chapters. The first chapter (this one) introduces the problem and significance of isolating translational efficiency bias, and provides some background so that a reader who is conversant with typical computer science subject matter will be able to follow the rest of the document. The second chapter is the literature review, and it explains what others have done in investigating this problem. The last chapter summarizes and discusses the findings of the dissertation, leaving three, main chapters on the methodologies employed in solving the problem of isolating translational efficiency bias in the presence of a confounding GC(AT)-content bias. The first is an analysis of the codon usage search space. This includes visualizing the search space’s fitness landscapes, statistical analyses, and interpretations of the results. The second discusses a deterministic approach to discovering the translational efficiency bias that is a modification to an existing technique. The third describes an evolutionary computational approach to the discovery process.
1.3. BACKGROUND

May 27, 2008

Figure 1.3: Base-pairing of Nucleotides in Complementary DNA Strands

Adapted from (Access Excellence, 2005). Image resides at URL:
http://www.accessexcellence.org/RC/VL/GG/basePair2.html
Figure 1.4: 1’ through 5’ Carbon Atoms. Nucleotides are added to growing DNA and RNA molecules at their 3’ end.

Figure 1.5: The Central Dogma of Molecular Biology
Figure 1.6: tRNA Molecule with Amino Acid and Anticodon
1.3. BACKGROUND

May 27, 2008

Gene 167484..169727
/gene="fhuA"
/locus_tag="b0150"
/note="synonyms: tonA, T5rec"
CDS 167484..169727
/gene="fhuA"
/locus_tag="b0150"
/function="outer membrane protein receptor for ferrichrome, colicin M, and phages T1, T5, and phi80"
/codon_start=1
/transl_table=11
/product="FhuA"
/protein_id="NP_414692.1"
/db_xref="GI:16128143"
/translation="MARSKTAQPKHSLRKIAVVVATAVSGMSVYAQAAVEFKEDTITV TAAAPQESAWGPAAATIAARQSATGKTDTPQKVPQISVVTAEEMALHCPKSVKEA LSYTPGVSVOQTGASNYTDIIIRGFAEREGQSQNNYLGLKQGFNYDAMITFYMLE RAEIMRGPVSVLGYKSSPGGLLMWVKRPTEPPLKEVQFKAGTDLSITQFDFTSDLD DDGNYLRLTGLARASANNQKQGSEEQRYAIAAPFTRWPDDKTNFTFLSYPQNEPETGY YGKLKPEGTVELPNSKRLPTDFNEAGAQNNTYSRNKMGYSDFHDFNDTFTVQNLR FAENKTSQNSYGYGYSDDPANAYSQKCALAPADKSHYHARAYTDKDNQFSTVD QLGSKFATGDIDHTLTTGDFMRMRNDINAWFGYDSVP1LNLINPVNNTDFDNAKDP ANSYPYIRILNQKQGQTVYQDDQWQKVLTVLGGYDWAQESLSNRAVGTDDPRDKQ FTKRGVQNLDFKVTPYYSIESFESFFTSQLGDKGNNIFAPSQKQIEVQKYYVEDRP IVTVAGYLTITKINLAMDFEGSFFVSGSEIEAIRGVEIAEALASASAVVYGGYTT DAEYTDTTYYTQDTPAQVAKMAMSLADYTFDDGPLSGLTGLGTHYGTGSGYDPANS FKVGSTVYDDALVRVGMAGSNVALHVNMLFDERYVASCFTNGCFWAGERQVV ATATRF"

Figure 1.7: Typical Annotation for Protein Expressing Gene

Text extracted from the annotated genome file for *Escherichia coli K12* found in NCBI’s complete microbial genomes database NCBI (2005)
Literature Review

2.1 Bias Discovery Techniques

As stated before, there are many bias discovery techniques from which to choose. Following is a description of those that have gained widespread acceptance. They are presented in the order in which they were published to give the reader a sense of the way in which the state of the science has evolved.

2.1.1 Frequency of preferred codons (FOP)

The relationship between codon usage bias and expressivity was first documented in 1981 (Ikemura, 1981a). At that time there were only a few dozen genes sequenced for *E. coli*. Now there are over 4,000 (NC_000913.2) sequenced protein coding genes in *E. coli*. While research had already noted that a bias existed (Fiers et al., 1975; Air et al., 1976; Efstratiadis et al., 1976) it was Ikemura (1981a) that identified the underlying association with expressivity. His research began by studying the correlation of codon usage bias and tRNA abundance because it was becoming clear that the bias was “mostly attributable to the availability of transfer RNA within a cell” (Post et al., 1979; Post and Nomura, 1980; Nichols and Yanofsky, 1979; Nakamura et al., 1980; Yokota et al., 1980; Ikemura, 1981a; Ikemura et al., 1980). Ikemura (1981a) found that codon usage bias did, indeed, adhere to the tRNA adaptation theories but he identified another interesting trend. He wrote a second paper that same year proposing that synonymous codon usage could be used as a
predictor of expression rates (Ikemura, 1981b). The synonymous codons that are revealed most often in highly expressed genes were termed optimal or preferred codons. Ikemura (1981a) found that there was a “tendency that the genes encoding abundant protein species selectively use the codons of major isoacceptors...” Further, he found that this choice is strictly constrained by tRNA availability.

This precipitated the identification of four rules that predict the choice of major codons. His earlier research (Ikemura, 1981a) yielded the first: thiolation of uridine in the wobble position (the third and most highly variable nucleotide of a codon) of an anticodon produces a preference for using an A-terminated codon over a G-terminated codon. Other research (Grosjean et al., 1978) provided a second rule: codons of type (A or U)-(A or U)-(pyrimidine) would support an optimal interaction strength between a codon and an anticodon when the third nucleotide is C. To these Ikemura added two new constraints. The first was: “the introduction of inosine (a nucleoside formed by the deamination of adenosine. Important because the nucleotide fails to form specific pair bonds with the other bases) at the wobble position may produce a possible preference for U- and C- terminated codons over the A-terminated codon, which must lead to purine-purine wobble pairing.” The second was: “synonymous codon usage is governed by the most highly available tRNA.” These rules and subsequent trends led to the concept of frequency of use of optimal codons.

\[
FOP = \frac{\text{number of optimal codons}}{\text{number of optimal codons} + \text{number of non-optimal codons}}
\]  

(2.1)

This frequency was found to be highly correlated with protein abundance. All the rules except tRNA availability were the same from species to species and the translational efficiency attained through tRNA abundance was presumed to be the driving force behind the correlation.
2.1.2 Indexes, Statistics, and Clustering

2.1.2.1 Bennetzen and Hall’s Codon Bias Index (CBI)

Very soon after the publication of Ikemura’s *Frequency of use of optimal codons* paper, Bennetzen and Hall (1982) published results of their work on a *Codon Bias Index*. The research was carried out in parallel with Ikemura’s and drew many of the same conclusions. They characterized bias as a ratio whose numerator contained the number of preferred codons in the gene less the number expected if codon usage is random. The denominator was the total number of codons. In this research preferred codons were identified by examining highly biased genes. Any codon with usage of 85% or greater in highly expressed genes was identified as preferred. There were only eight genes examined in this research, two of which were characterized as highly expressed (alcohol dehydrogenase I and glyceraldehyde-3-phosphate dehydrogenase). From the behavior of these codons four empirical rules were formulated to help identify preferred codons. These rules were similar to Ikemura’s in that they were related to wobble position nucleotides and tRNA abundance.

2.1.2.2 Correspondence Analysis

In 1981 Grantham et al. (1980, 1981, 1985) employed correspondence analysis to compare codon usage with expressivity. The method extracted two dimensions from third base frequencies using correspondence analysis (Benzecri, 1973; Hill, 1974) [similar to principal components analysis (Hotelling, 1933; Jolliffe, 1986), also see Appendix B, but the data must be positive and is typically normalized] and projected the gene frequency data into the two-dimensional space defined by these axes. The genes were then manually labelled as highly or weakly expressed. Two highly distinct groups of genes emerged from this analysis.

In 1986 additional work was done employing similar methods, though various additional parameters and relationships were examined (Holm, 1986).
2.1. BIAS DISCOVERY TECHNIQUES

2.1.2.3 P1 and P2 Index

In October of 1982, research confirmed that “bias in codon usage has two main components: Correlation with tRNA level in the cell and non-random choices between pyrimidine ending codons” (Gouy and Gautier, 1982). Gouy and Gautier went on to quantify the relationships by creating two simple indexes based “on the differential usage of iso-tRNA species during gene translation, the other on choice between Cytosine and Uracile for [the] third base.” The first index was the average number of tRNA discriminations per elongation cycle (P1 index) and the second was the frequency of “right choices between the pyrimidines among codons beginning with AA, AU, UA, UU, CC, CG, GC or GG” (P2 index). Once again a large P1 index was strongly correlated to gene expressivity.

2.1.3 Codon Preference Bias

In an attempt to remove the a priori knowledge of tRNA activity requirement from the determination of codon usage bias, McLachlan et al. (1984) developed the Codon Preference Bias approach. It is a statistical method of measuring the codon preference but it is “defined strictly relative to a given observed amino acid composition.” In other words, given an organism’s base composition, how probable is any given gene’s observed codon frequency? The codon frequency bias is large any time the codon usage pattern is “intrinsically improbable.” The approach taken is to observe that if a sequence is completely random then the expected frequency of a codon $f_c$ with a fractional base composition $b_i$ for base $i$ can be determined by the following relationship:

$$f_c = b_i b_j b_k$$  \hspace{1cm} (2.2)

A multinomial approach is employed to determine the probability of deviation from this expected frequency. Their results were compared to those achieved by Ikemura and Bennetzen & Hall and were “well correlated.”
2.1. BIAS DISCOVERY TECHNIQUES

2.1.4 Clustering

In 1986 Sharp et al. (1986) used cluster analysis to predict expressivity. The genes in yeast formed two clusters that could, by inspection, be classified as highly expressed and not highly expressed. Their method used Relative Synonymous Codon Usage (RSCU, Equation 2.3) values as entries in a 64-dimensional vector. Each dimension is associated with a codon while each vector is representative of a gene.

\[ RSCU_{ij} = \frac{X_{ij}}{\frac{1}{n_i} \sum_{j=1}^{n_i} X_{ij}} \]  

(2.3)

\( X_{ij} \) is the number of occurrences of the \( j^{th} \) codon for the \( i^{th} \) amino acid and \( n_i \) is the number (from one to six) of alternative codons. These gene data points (vectors) were then clustered using Ward's clustering algorithm (Ward, 1963) where the two most similar genes are found and replaced by the gene that is on the midpoint between them. This process is repeated until all genes have been replaced. The resulting dendrogram, built during the clustering process, indicated the presence of two clusters that were subsequently characterized as highly and not highly expressed genes. This characterization was performed by inspection. A chi-squared statistic (Equation 2.4) was calculated and used to determine the bias levels of the genes.

\[ \chi^2 = \sum_{i=1}^{64} \frac{(CU_i - \overline{CU}_i)^2}{\sigma_i^2} \]  

(2.4)

\( CU_i \) is the codon usage for codon \( i \) (number of codon \( i \) used in the gene) and \( \overline{CU}_i \) is the average codon usage for codon \( i \) across the entire genome. \( \sigma^2 \) is the square of the standard deviation for codon \( i \)'s average. The resultant \( \chi^2 \) was then scaled by two times the number of codons in the gene.
2.1. BIAS DISCOVERY TECHNIQUES

2.1.5 Codon Adaptation Index (CAI)

In 1986 Sharp was again involved in the development of a measure of synonymous codon usage bias. It was called the Codon Adaptation Index (CAI) (Sharp and Li, 1987). The measure was created to address several perceived weaknesses in the existing measures. Prior to CAI the more popular measurements were essentially binary – either the codon in question was optimal or it was not. There was no gradation. Also, it was not possible to determine whether a codon was optimal in every case. Sometimes codons had to be excluded because their status was unclear. Finally, Sharp and Li observe that no between-species comparisons could be performed because the “proportional division of the codon table into the two categories [differed from species to species].”

An already existing measure known as a codon preference statistic addressed the first two issues (Gribskov et al., 1984). This statistic is calculated as the probability of finding a particular codon in a highly expressed gene compared to the probability of finding it in a random sequence made up of the same nucleotides. Unfortunately the codon preference statistic can produce two very different results for genes with different amino acid compositions even if both used only optimal codons. The codon adaptation index corrected this deficiency by including normalization. This makes interspecies comparisons possible and convenient. The process of calculating CAI requires a priori knowledge of expression rates for an organism’s genes. The gene set that is most highly expressed is known as the reference set. From the sequences of these genes a table of codon usage values is built. Once again Relative Synonymous Codon Usage (RSCU) values are used (Equation 2.3).

The relative adaptiveness (or weight) of a codon $w_{ij}$ is:

$$w_{ij} = \frac{RSCU_{ij}}{RSCU_{i_{max}}} = \frac{X_{ij}}{X_{i_{max}}}$$

(2.5)

Where $w_{ij}$ is the weight of the $i^{th}$ codon for the $j^{th}$ amino acid. The $x$ in the numerator is the count for that codon and the denominator ($y$) is the count of the maximal sibling for the amino
2.1. BIAS DISCOVERY TECHNIQUES

May 27, 2008

acid in question. Next, a geometric mean is taken of RSCU values to calculate CAI.

\[
CAI = \frac{CAI_{obs}}{CAI_{max}} \tag{2.6}
\]

\[
CAI_{obs} = \sqrt[L]{\prod_{k=1}^{L} RSCU_k} \tag{2.7}
\]

\[
CAI_{max} = \sqrt[L]{\prod_{k=1}^{L} RSCU_{k_{max}}} \tag{2.8}
\]

While these formulas make clear (and support) the notion of codon adaptiveness, the calculations can be simplified by reducing equation 2.6 to:

\[
CAI = \sqrt[L]{\prod_{i=1}^{L} w_i} \tag{2.9}
\]

2.1.6 Scaled \(\chi^2\)

In 1988 a Scaled \(\chi^2\) measure was introduced as a measure of codon bias (Shields et al., 1988; Shields and Sharp, 1987). Sharp was involved [from the clustering and CAI methods (Sharp and Li, 1987; Sharp et al., 1986)] so there were similarities in the methods employed (e.g. RSCU was used along with the \(\chi^2\) metric, though this time it was scaled). *Drosophila melanogaster* (fruit flies) was the target genome. *D. melanogaster* is eukaryotic (vs. the single-celled prokaryotic organisms commonly studied) and is, therefore, much more complex. In this case, clustering was inappropriate since the within-species variation was continuous rather than discrete.

A silent site was defined as a synonymously variable position within a codon and their research uncovered evidence of bias in selecting nucleotides at these positions. A \(\chi^2\) calculation was performed that examined deviation of codon usage from expected values. “Since these values are
generally highly correlated with gene length, they were then scaled with division by the number of codons in the gene (excluding Trp and Met codons, which do not contribute to chi).”

\[ \chi^2 = \sum_{i=1}^{64} \frac{(CU_i - \bar{CU})^2}{\sigma_i^2} \]  

(2.10)

\[ \chi^2_{scaled} = \frac{\chi^2}{N_{codons}} \]  

(2.11)

### 2.1.7 Effective Number of Codons

The **effective number of codons** for a gene is a measure of how biased a gene is in favor of a subset of codons (Wright, 1990). It was developed in 1990 as a means of determining codon usage bias with sequence information only. No *a priori* knowledge of tRNA concentrations or expressivity was required. There are 61 codons that can code for the 20 amino acids. The index is designed such that uniform usage of codons yields an effective number of codons of 61. If some codons are used more than others the number of effective codons begins to decline. If, for each amino acid, a single codon is used to the exclusion of its synonymous codons, an effective codon number of 20 can be attained. A set of synonymous codons is analogous to a set of alleles (e.g. an amino acid with four synonymous codons is analogous to a locus with four alleles). The analogy to alleles allows for the use of existing techniques (Kimura and Crow, 1964) in the calculation and use of homozygosity \( (\hat{F}) \). Homozygosity describes how similar each of the alleles are. If all the codons are utilized equally the codon would be described as homozygous. With actual alleles this would be an indication that both genes were dominant or both were recessive.

\[ \hat{F}_{aa_j} = \frac{n_j \left( \sum_{i=1}^{k} \mu_i^2 \right) - 1}{n_j - 1} \]  

(2.12)
2.1. BIAS DISCOVERY TECHNIQUES

In the above equation $n_j$ is the number of occurrences of codons that code for amino acid $j$ and $p_i$ is the probability of occurrence associated with the codon $i$. If $n_i$ represents the number of occurrences of codon $i$ then the frequency ($p_i$) of that codon is $n_i/(n_1 + n_2 + \ldots + n_k)$ where $k$ is the number of synonymous codons that code for the associated amino acid. As an example, for an amino acid with four synonymous codons that exhibit even usage; the homozygosity for each of the four codons would be close to 0.25 \((0.242) = \left(\frac{n_j \times .25 - 1}{n_j - 1}\right)\), and would approach 0.25 as $n_j$ grows large (note that $\sum_{1}^{4} .25^2 = .25$. This is because .25 is really 1/4 so .25$^2$ is tantamount to saying $1/4 \times .25$, i.e. you are summing 4 instances of 1/4 of .25). If, however, only one codon were used to the exclusion of the other three, the homozygosity of that codon would be 1 \([\left(\frac{n - 1}{n - 1}\right)]\). From this statistic the number of effective codons for this amino acid can be calculated.

$$\hat{S}_{aa_j} = \frac{1}{\hat{F}_{aa_j}} \quad (2.13)$$

$\hat{S}_{aa_j}$ is the number of effective codons for amino acid $j$. In the above example of four synonymous codons, balanced usage would yield an effective number of codons very close to four. Exclusive use would cause an effective number of one. The number of effective codons ($N_c$) across all amino acids is the sum of $\hat{S}_{aa_j}$ for the 20 amino acids. This reduces to the following (where $\bar{F}_n$ is the average of the $\hat{F}_{aa_j}$'s that have $n$ codons in amino acid $j$’s family).

$$N_c = \sum_{j=1}^{20} \hat{S}_{aa_j} = 2 + \frac{3}{\bar{F}_6} + \frac{5}{\bar{F}_4} + \frac{9}{\bar{F}_2} + \frac{1}{\bar{F}_3} \quad (2.14)$$

This can be stated more generally in terms of redundancy classes (Equation 2.15). A redundancy class is a set of amino acids with the same number of synonymous codons. $ARC$ is the set of all redundancy classes, and $n_r$ is the number of amino acids in the redundancy class.
2.1. BIAS DISCOVERY TECHNIQUES

May 27, 2008

\[ N_c = \sum_{r \in ARC} \frac{n_r}{F_r} \]  

(2.15)

2.1.8 Morton’s Codon Bias Index (CBI)

Another Codon Bias Index (CBI) was developed in 1992 by Brian Morton. He created this index to facilitate his studies of the chloroplast genome (Morton, 1993). Morton was clearly influenced by the work of Sharp and Li and even borrowed their \( w_{ij} \) term from the codon adaptation index formula (Equation 2.5). This was clearly stated and cited in his paper. He called this term \( R_{ij} \) (Equation 2.16). Of the two CBI’s, Bennetzen and Hall’s is the better known.

\[ R_{ij} = \frac{n_{ij}}{n_{i_{\text{max}}}} \]  

(2.16)

\( n_{ij} \) is the count of the \( j^{th} \) synonymous codon for amino acid \( i \) and \( n_{i_{\text{max}}} \) is the count of the maximal sibling for that amino acid. Morton used this term to calculate his codon bias index (CBI) as follows:

\[ CBI = \sum_{i=1}^{18} \frac{n_i}{n_{\text{tot}}} \left( \sum_{j=1}^{S_i} \frac{(1 - R_{ij})^2}{S_i - 1} \right) \]  

(2.17)

\( S_i \) is the number of siblings for the \( i^{th} \) amino acid and \( n_i \) is the count for the \( i^{th} \) amino acid. The \( n_{\text{tot}} \) term is the total number of residues excluding methionine (Met) and tryptophan (Trp) since they have only one codon each. Using this formula a gene exhibiting no bias receives a CBI of 0. A gene that is extremely biased gets a score approaching 1.

2.1.9 Intrinsic Codon Deviation Index (ICDI)

In 1994 Freire-Picos et al. fashioned an index to address perceived weaknesses with the previous codon bias indices. The new index was known as the intrinsic codon deviation index (ICDI)
and required no a priori knowledge of tRNA levels or expression rates (unlike CBI) but has the advantage that “the theoretical range of values is the same [as that of] CBI,” which the effective number of codons \((N_c)\) (Wright, 1990) did not. ICDI is calculated as follows:

\[
ICDI = \frac{\sum S_2 + S_3 + \sum S_4 + \sum S_6}{18}
\]  

(2.18)

Where \(S_k\) is calculated as:

\[
S_k = \sum_{i=1}^{k} \frac{(n_i - 1)^2}{k(k-1)}
\]  

(2.19)

In the above formula \(n_i\) is the RSCU value (Equation 2.3) of the \(i^{th}\) codon and \(k\) is the corresponding value of degeneracy, 2, 3, 4, or 6. Once again 0 is indicative of an absence of bias while 1 is strongly biased.

### 2.1.10 Principal Components Analysis

In 1996 Kanaya et al. used principal components analysis to identify major codons (Kanaya et al., 1996, 1999). A codon frequency, or RSCU (Equation 2.3), matrix is generated whose rows represent genes and whose columns represent the codon frequencies in each gene (Figure 2.1). Principal components analysis (Hotelling, 1933; Jolliffe, 1986) (Appendix B) is performed on this matrix to derive the axis of greatest variance within this data (first principle component). The RSCU matrix will be represented as \(X\).

The intuition behind this approach is as follows: assume first that the primary force explaining the variance in a genome’s codon usage is translational efficiency bias (i.e. highly expressed genes show a high preference for major codons while weakly expressed genes do not). It follows that highly expressed genes should be found at one end of the axis of greatest variance, and weakly expressed genes should fall at the other. To determine where each gene is to be projected upon
the first principal component, a dot product is performed (principal components are Eigenvectors generated from $X$’s covariance matrix making them unit vectors). The resultant projection is $Z'_1$. Projections are often normalized. The prime ($'$) in this case indicates that this is the non-normalized version.

\[
X \cdot b_1 = Z'_1 \tag{2.20}
\]

In order to determine which codons are “major,” or preferred, it is a simple matter of determining whether or not the codon contributes to the overall ordering of genes on $Z'_1$. This is established by computing the correlation between the usage of each codon in each gene and each gene’s location on $Z'_1$. A positive correlation indicates that the codon contributes positively to the overall ordering of $Z'_1$ and can be considered preferred or major. More formally, this is accomplished by measuring the correlation between each column of the frequency (RSCU) matrix and $Z'_1$ (Figure 2.2). The resultant correlations are known as factor loadings. Kanaya et al. (1996) compared the factor loadings to the preferred codons derived using the Ikemura 4-rule method (Ikemura, 1981a) in order to validate their findings. No \textit{a priori} knowledge of tRNA level is required to determine which are major. Note that Ikemura is a coauthor on this work, so it can be seen as removing the \textit{a priori} requirement from the determination of FOP (Ikemura, 1981a).
2.1. BIAS DISCOVERY TECHNIQUES

Figure 2.2: Factor Loading Determination. The vertical column in the frequency matrix headed $c_i$ is a vector representation of the $i^{th}$ codon’s frequency across all genes. $Z'$ also has a dimensionality equal to the number of genes. It was derived by projecting each gene vector (the horizontal representation for each gene in the frequency matrix) onto the first Eigenvector.

$$r(Z'_1, X_i) = \frac{Cov[Z'_1, X_i]}{\sqrt{Var[Z'_1] Var[X_i]}}$$
2.1.11 CAI Revisited

In 2003 Carbone et al. removed the need for *a priori* knowledge of gene expressivity from the CAI calculation process (Carbone et al., 2003). Theirs is a two step approach that works first to identify the proper reference set of genes (using a greedy, hill-climbing algorithm) 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 search algorithm (for the reference set) is iterative in nature. Identification of the reference set begins by considering the entire genome as a 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 (Equation 2.5) based upon the identified reference set. CAI scores are calculated for all genes (Equation 2.9), 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.

To prevent confusion, when the term CAI is used it will be in reference to values derived using the traditional Sharp and Li approach. When the Carbone et al. (2003) method is utilized this text will employ the terminology “self-consistent codon index (SCCI),” as described in Carbone (2006). Self-consistency refers to the definitional condition that the identified reference set adhere more strongly to the bias (which the reference set itself defines, hence self-consistent) than all other genes in the genome. Equation 2.21 replaces equation 2.9 in calculating a gene’s adherence to the identified bias.

$$SCCI(g) = \sqrt[\frac{L}{\prod_{i=1}^{L} w_i}]$$

(2.21)
2.2. MULTIPLE BIASES

The SCCI algorithm described above is designed to identify the dominant bias within a genome. The dominant bias is not necessarily driven by the advantages associated with translational efficiency. A major contribution of this dissertation is to provide a technique for resolving this issue while preserving the algorithm’s independence from the need for prior knowledge of a set of highly expressed genes.

2.1.12 Summary

Table 2.1 is an abridged description of the bias measures covered in this literature survey. The methods described are sorted by submission date as was the coverage in the literature survey. This was done to facilitate an understanding of the way in which codon usage measurement techniques have evolved since first employed in 1981.

2.2 Multiple Biases

It has been shown that multiple biases can coexist in an organism’s genome and the presence of these biases can cause misleading results when it is assumed that either a priori methods [such as the original Sharp and Li CAI method (Sharp and Li, 1987)] or algorithmic [such as the updated CAI algorithm, SCCI (Carbone et al., 2003)] are isolating translational efficiency bias (Grocock and Sharp, 2002; Carbone et al., 2005; Raiford III et al., 2007a, 2006b). Carbone et al. (2003) went on to quantify some of the biases known to exist in genomic data. These biases are those associated with GC-content, strand, and translational efficiency (Carbone et al., 2003). Since their algorithm identifies the dominant bias, it was important for them to develop measures to indicate which bias was identified by their algorithm, and to determine the strength of these biases (Carbone et al., 2005). Following is a description of several of those measures.
### Table 2.1: Summary of All Described Bias Measures

<table>
<thead>
<tr>
<th>Measure</th>
<th>Date</th>
<th>a priori</th>
<th>Description</th>
<th>Authors</th>
</tr>
</thead>
<tbody>
<tr>
<td>FOP (Frequency of Preferred Codons)</td>
<td>1981</td>
<td>Yes</td>
<td>FOP = number of optimal codons/total number of codons in gene. Optimal codons determined by 4 rules.</td>
<td>Ikemura</td>
</tr>
<tr>
<td>CBI (Codon Bias Index)</td>
<td>1981</td>
<td>Yes</td>
<td>(preferred codons less expected if random)/ total number of codons Preferred status determined by 4 rules.</td>
<td>Bennetzen and Hall</td>
</tr>
<tr>
<td>Correspondence Analysis</td>
<td>1981</td>
<td>No</td>
<td>Project frequency data on first two principal components. Manually label as highly or weakly expressed. Similar to clustering.</td>
<td>Grantham et al.</td>
</tr>
<tr>
<td>P1 and P2 Index</td>
<td>1982</td>
<td>Yes</td>
<td>P1: average number of tRNA discriminations per elongation cycle P2: frequency of right choices between the pyrimidines</td>
<td>Gouy and Gautier</td>
</tr>
<tr>
<td>Codon Preference Bias</td>
<td>1984</td>
<td>No</td>
<td>Multinomial statistical method of measuring the codon preference. Answered question &quot;how probable was any given gene's observed codon frequency?&quot;</td>
<td>McLachlan et al</td>
</tr>
<tr>
<td>Clustering</td>
<td>1986</td>
<td>No</td>
<td>64 dim vector of RSCU scores. Ward's clustering algorithm followed by dendogram examination.</td>
<td>Sharp et al</td>
</tr>
<tr>
<td>CAI (Codon Adaptation Index)</td>
<td>1986</td>
<td>Yes</td>
<td>Reference set used to identify RSCU values. CAI score then generated forSharp and Li each gene using a geometric mean of RSCU values.</td>
<td>Sharp and Li</td>
</tr>
<tr>
<td>Scaled $\chi^2$</td>
<td>1988</td>
<td>No</td>
<td>Chi squared statistic for codon usage scaled by gene length.</td>
<td>Shields et al</td>
</tr>
<tr>
<td>Nc (Effective Number of Codons)</td>
<td>1990</td>
<td>No</td>
<td>64 effective codons implies balanced usage, only 20 effective codons implies extreme bias.</td>
<td>Wright</td>
</tr>
<tr>
<td>CBI (Codon Bias Index)</td>
<td>1992</td>
<td>No</td>
<td>Used CATIs w term to calculate. Took frequency of an amino acid and scaled it by its deviation from base composition.</td>
<td>Morton</td>
</tr>
<tr>
<td>ICDI (Intrinsic Codon Deviation Index)</td>
<td>1993</td>
<td>No</td>
<td>Deviation from equal use of all codons (as apposed to base composition)</td>
<td>Freire-Picos et al.</td>
</tr>
<tr>
<td>MCU (Major Codon Usage)</td>
<td>1999</td>
<td>No</td>
<td>PCA to reduce codon frequency data to one dimension. Correlation of each codon to this distribution as indicator of contribution to bias.</td>
<td>Kanaya et al.</td>
</tr>
<tr>
<td>CAI (Codon Adaptation Index)</td>
<td>2003</td>
<td>No</td>
<td>Treat whole genome as reference set. Calculate CAI of genes. Throw out lower half and repeat. Once reference set gets to 1% of original gene set stop and give final CAI scores to genes.</td>
<td>Carbone et al.</td>
</tr>
</tbody>
</table>
2.2. MULTIPLE BIASES

2.2.1 Content Criterion

One of the more common biases encountered in the study of codon usage bias is GC(AT)-content bias. To determine whether a bias described by a given reference set is influenced by GC-content a *content criterion* was developed (Carbone et al., 2005). Detection is accomplished by measuring the correlation between the GC3-content (percent of nucleotides in the third codon position that are G or C) of each gene with its CAI/SCCI value. Correlations > 0.7 (<−0.7 for AT-content) are an indication of a genome characterized by GC(AT) bias.

2.2.2 Ribosomal Criterion

A measure of how well a trend identified by CAI/SCCI captures the translational efficiency bias for a particular organism is known as the *ribosomal criterion*. This criterion is based on the degree to which the ribosomal protein coding genes, which are known to be highly expressed in many prokaryotic organisms, are found in the upper region of a sorted list of genes (by CAI/SCCI value). As employed by Carbone et al. (2005), it can be concluded that the SCCI algorithm has identified the translational efficiency bias when the average SCCI value for the organism’s ribosomal protein coding genes (RPCGs) is greater than one standard deviation above the mean SCCI value for the organism’s genome.

\[
z = \frac{(SCCI(r) - \bar{SCCI})}{\sigma_{SCCI}} \tag{2.22}
\]

The average of the z-scores for RPCGs (Equation 2.22), \(\bar{z}\), is used to define the ribosomal criterion. A genome characterized by translational efficiency bias will have a high \(\bar{z}\). Carbone et al. (2005) have determined \(\bar{z}\) values greater than 1 are indicative of organisms characterized by translational efficiency bias.
2.2. MULTIPLE BIASES

2.2.3 Strand Criterion

Strand bias is the result of the asymmetric processes of transcription and replication. Transcription induces $C \rightarrow T$ transitions on the nontranscribed strand (Francino et al., 1996; Francino and Ochman, 1997; Beletskii and Bhagwat, 1998) thought to arise from increased deamination of cytosine due, in part, to the non-template strand being left exposed to free radicals and the template strand being under the control and manipulation of various transcription related proteins. Replication induced bias is due to increased $G+T$ concentrations on the leading strand thought to be the result of the discontinuous process for replicating the lagging strand where Okazaki fragments are synthesized and then joined (Lafay et al., 2000, 1999; Rocha et al., 1999). While correcting for this type of confounding factor is beyond the scope of this work, following is the technique employed for identifying the existence of such a bias. Rather than attempting to identify the specific cause (replication or transcription) of the bias, Carbone et al., once again, rely on a statistical definition to assist in characterizing the bias. They begin by assuming that strand bias results in the “most biased genes of a (circular or linear with bidirectional replication) genome [to be] preferentially distributed in precisely one of its strands (typically the leading strand) (Carbone et al., 2005).” The presence of strand bias will cause the distributions of CAI/SCCI scores on the leading and lagging strands to drift apart. When the absolute value of the $t$-value generated from a $t$-test between the two distributions is $> 0.25$ the genome is deemed to exhibit strand bias.

2.2.4 Strength Criterion

A characteristic of a genome with a dominant bias of translational efficiency is that the most highly expressed genes (the reference set) tend to be much more highly biased than the rest of the genome. A strongly-biased reference set can be further evidence that a detected bias is that of translational efficiency bias. The CAI weights associated with the reference set are an indication of the strength of this bias. The maximal sibling will always have a weight assignment of one. The strongest possible bias would yield weights of zero for the rest of the siblings. Weaker biases would have
non-maximal sibling weights closer to one. Thus, one measure of the strength of the bias is indicated by how far the weight vector is from balanced usage. Another, perhaps more useful, measure is how far the weight vector is from the overall bias of the organism. Carbone et al. (2005) developed such a criterion for measuring bias strength (2.23).

\[
d(w^G, w^S) = \frac{\sum_{i=1}^{64} |w_i(G) - w_i(S)|}{2}
\]  

(2.23)

The overall bias of the organism is obtained by extracting the weight vector generated using the entire genome as the reference set \(w^G\). This is compared to the weight vector of the smaller reference set containing only highly expressed genes \(w^S\). The result can be thought of as measuring the number of codon families where the preferred codon is different in two biases.

Consider a case of a binary weight vector (each family has a single weight of one and the rest zero) where the major codons change for all families between \(G\) and \(S\). This yields a summation of the differences (Manhattan distance) of 40. Dividing by two brings the number back to the number of changed major codons (20). In other words, Equation 2.23 is equal to one-half the Manhattan distance.

A genome characterized by translational efficiency bias should have a relatively large strength measure. Carbone et al. (2005) set the criterion for an organism’s bias to be considered strong at \(d(w^G, w^S) > 8\), but they note that not all organisms characterized by translational efficiency bias meet this criterion. Instead, they use it in conjunction with the ribosomal criterion to differentiate weak and strong translational efficiency bias.

### 2.2.5 Candidates for Horizontal Gene Transfer

Yet another source of codon usage bias is horizontal gene transfer. Research in this area has primarily focused on the identification of candidate genes for horizontal gene transfer. Since the assumptions about expressivity are based on each gene’s adherence to an established bias within
the genome, any genes that have entered the genome recently will not necessarily have had time to evolve in such a way as to adhere to the bias. Thus, genes that deviate from normal organismal genomic patterns are more likely to have been introduced in recent evolutionary time (Koonin et al., 2001). Identifying horizontally transferred genes is performed in a number of steps. The first is by comparing the genes to other known phage related genes. Genes known to be phage related are typically annotated in the sequence files provided by the National Center for Biotechnology Information (NCBI) (NCBI, 2005).

Another indicator of possible HGT candidacy can be found through the examination of GC-content (Kaplan and Fine, 1998). Given a random choice in nucleotides an organism’s genome should contain about as many guanines and cytosines (Gs and Cs) as it has adenines and thymines (As and Ts). Recall that Gs and Cs are complementary pairs and are always found opposite each other on the DNA strands. This equal distribution is seldom the case. Over time organisms tend to develop and adhere to a bias in GC-content, particularly in the first and third codon positions. If a gene’s GC\textsubscript{1} and GC\textsubscript{3} content (Gs and Cs found in the first and third codon locations) are significantly different than those of the rest of the organism’s genes it can be reasonably concluded that the gene has been recently introduced into the genome, and thus has not had time to evolve into adherence with the rest of the genome (known as the amelioration process).

Research in this area (Garcia-Vallvé et al., 2000) selects candidates for HGT by examining a gene’s GC\textsubscript{1} and GC\textsubscript{3} deviation from the organism’s norm. If both deviate from the norm in the same direction and at least one is greater than 1.5 standard deviations from the mean (of the genome’s GC\textsubscript{1} and GC\textsubscript{3} content) it is considered a candidate. Using a sliding window of eleven genes, any window with five or more extraneous genes is an indication of an alien genomic strip. This is an area in the genome where multiple genes were incorporated in a single event.

Another criteria used to identify candidates for horizontal gene transfer is by detecting regions of unusual composition in the organism’s genome (Kaplan and Fine, 1998; Chou and Zhang, 1995; Eisen, 2000; Ragan, 2001; Waack et al., 2006). This is based upon the genome hypothe-
sis which assumes that codon usage and GC-content are distinct, global features of a prokaryotic genome (Grantham et al., 1980). One measure of such deviation from normal composition is the Mahalanobis distance (Mahalanobis, 1936) from a genome’s average codon usage. Mahalanobis distance can be thought of as the distance between two points in a multidimensional space normalized by the variance of the overall data in each dimension. A frequency matrix similar to the one used in the MCU calculations is generated. Each row represents a gene and each column a codon. The associated gene/codon cell is the frequency with which that codon occurs within the gene. Genomic averages are calculated for each codon frequency. The Mahalanobis distance is calculated between each gene and the mean of the organism (Chou and Zhang, 1994). The distances do not follow a normal distribution and so a Monte Carlo procedure (Gillespie, 1977) is employed that generates a random sample of sequences based upon the mean and standard deviations of the actual codon usage for the genome. Any that differ by more than two standard deviations from the mean are considered extraneous.

Computation of Mahalanobis distance is facilitated by the use of a covariance matrix. The following formula represents the covariance matrix $S$. $X_{k,i}$ is the frequency of the $i^{th}$ codon in the $k^{th}$ gene, $\overline{X}_i$ is the average frequency for the $i^{th}$ codon, and $N$ is the total number of genes.

$$S_{ij} = \sum_{k=1}^{N} [X_{k,i} - \overline{X}_i][X_{k,j} - \overline{X}_j]$$ (2.24)

The Mahalanobis distance for gene $X$ is calculated using the following formula:

$$d^M(X, \overline{X}) = (X - \overline{X})^T S^{-1} (X - \overline{X})$$ (2.25)

A series of horizontally transferred genes located near each other on the DNA strand is evidence that an entire strip of genes was transferred in one operation. An 11-gene sliding window is used to identify possible alien gene strips. Any window with five or more HGT genes is considered an alien gene strip. Finally, these strips must be filtered to disregard short isolated segments and
to include genes that were not initially considered extraneous but that have a deviation of their GC content of the same sign as the deviation of the strip to which they belong. Organisms for which this analysis has been performed have HGT genes listed in the Horizontal Gene Transfer Database (HGT-DB) (Garcia-Vallvé et al., 2003).

2.3 Confounding Effects of Other Biases

There have been no comprehensive studies comparing the various bias measures in terms of their ability to predict expressivity. One study (Comeron and Aguadé, 1998) did compare CAI, scaled $\chi^2$, $N_c$, CBI, and ICDI, in terms of how well they handled open reading frames of differing lengths. They note that $N_c$ and CAI perform best in this regard. Of these two, for the express purpose of determining adherence to translational efficiency bias, CAI is a logical choice. It specifically measures adherence to the bias defined by highly expressed genes (and therefore genes that have a high translational efficiency bias), whereas $N_c$ simply measures the degree of bias, regardless of the underlying cause. As an example, if a highly biased gene were introduced into a genome through horizontal gene transfer, and yet its bias was very different from that of the target organism’s translational efficiency bias, it would still rank high in terms of degree of bias (low $N_c$ number). In fact, of the methods described, only FOP, CBI, P1 & P2, and CAI are expressly designed to identify translational efficiency bias (not simply measure degree of bias). Of these methods, only CAI assigns a measure that captures the degree to which a codon is preferred. The others treat each codon in a binary fashion; either it is preferred or it is not. MCU and SCCI are purely algorithmic versions for determining FOP and CAI, respectively. One could use the factor-loadings in MCU to achieve this varying degree of major-ness, however, since MCU relies on determining the axis of greatest variance and then determining the degree to which each codon contributes to the genome’s arrangement along this axis, it is highly susceptible to interference from other biases, such as GC-content. SCCI is less susceptible to these effects because it specifically measures the adherence to the bias defined by a set of highly-biased, self-consistent genes. It is not until the alternative bias
(e.g. GC-content or strand) is the dominant bias that SCCI becomes susceptible to interference (this is considered interference only if the intent is the isolation of translational efficiency bias, not the dominant bias). A major objective for this work is the removal of this vulnerability while maintaining independence from the requirement for a priori knowledge of a set of highly expressed genes.

There is little research into techniques for the computational removal of confounding effects of GC-content (or any other factors) from the search for translational efficiency bias. An example of one of the few studies is of Grocock and Sharp (2002) where an extensive analysis of the genome for Pseudomonas aeruginosa PAO1 was performed. They showed that the genome is dominant for GC-content bias by performing principal components analysis upon the RSCU data and noting that the placement of the data on the first principal component was strongly correlated with each gene’s GC-content. They also noted that the data placement on the second principal component appeared to be driven by translational efficiency bias. This was accomplished by noting that known highly expressed genes (ribosomal protein coding genes) reside at one extreme of the second principal component, and that when the preferred codons of the 50 genes at the top of axis 2 (second principal component) were used to calculate FOP (Equation 2.1) for each gene, the resultant scores were highly correlated with placement on the second principal component. Once again, techniques such as this require knowledge of a set of highly expressed genes to enable a choice from among principal components. This makes them, qualitatively, no different from existing methods (such as FOP and CAI).

While there are few techniques that attempt to remove the confounding effects of GC-contnet, there are studies that attempt to quantify the effect of various confounding factors. The criteria proposed by Carbone et al. (2005) has already been described. Another such quantification was performed by Wright (1990) where the consequence of GC-content on the effective number of codons was determined mathematically and shown to hold empirically. The effect that GC₃ will have on $N_c$ can be predicted through the use of the following equation (Wright, 1990):
2.3. CONFOUNDING EFFECTS OF OTHER BIASES

\[ N_c = 2 + s + \frac{29}{s^2 + (1 - s^2)} \]  \hspace{1cm} (2.26)

Where \( s \) represents the fraction of GC3-content (from 0 to 1). Figure 2.3 shows the trend demonstrated in the yeast genome (the 111 genes from chromosome three). The degree to which the genes deviate from the prediction line is an indication of some force other than GC-content at play in the codon usage bias (generally thought to be translational selection, particularly when highly expressed genes reside furthest from the prediction line).

![Figure 2.3: \( N_c \) vs. GC\(_3\) for 111 Genes from Yeast Chromosome Three. Line represents expected bias (\( N_c \)). Degree to which genes deviate from this line is an indication of some bias, other than GC-content, at play in the genome.](http://codonw.sourceforge.net/ENcVsGC3s.gif)

There is no description in (Wright, 1990) of the method by which he derived Equation 2.3. dos Reis et al. (2004) reveal this omission, and go on to derive their own equation (Equation 2.27) where \( x_g \) represents silent GC\(_3\)-content of a gene (percent of nucleotides in the third codon position...
that are G or C) and $a$, $b$, and $c$ are constants that must be determined for a given species. They determined these constants through regression on a randomly generated set of genes that adhere to the target genome’s base composition. The constants were shown to be very similar between the genomes of *E.coli* and *H.sapiens* allowing them to recommend the values $-6.0$, $34.0$, $1.025$ for $a$, $b$, and $c$, respectively. Figure 2.4 reveals compelling evidence that this is a valid approach to the derivation.

$$N_{c_g} = f(x_g) = a + x_g + \frac{b}{x_g + (c-x_g)^2}$$  \hspace{1cm} (2.27)

Wan et al. (2004) performed a similar analysis on their own codon bias measure, synonymous codon usage order (SCUO), which they had developed previously (Wan et al., 2003). Their statistic is a measure that relies on Shannon information theory and the maximum entropy metric to determine deviation from order in codon usage. They too showed a bell shaped prediction curve with respect to GC-content.

Other methods of adjusting for base composition take an approach similar to that of the scaled $\chi^2$ measure (Shields et al., 1988; Shields and Sharp, 1987). These techniques modify the actual measure to reflect how much it deviates from background content, verses simply predicting how a gene exhibiting balanced usage would behave given unbalanced GC-content. Two such approaches are the maximum-likelihood codon bias statistic (MCB) (Urrutia and Hurst, 2001) and $B_g(a)$ from Karlin and Mrzek (1996). These were created to address perceived weaknesses in existing techniques (e.g. requirement for a reference set, inability to test a null hypothesis to determine whether a level of bias is significant, sensitivity to biased amino acid usage, etc.). Similarly, Novembre (2002) proposed an adjustment to the calculation of $N_c$ that not only corrects for the underlying base composition, but addresses a criticism from Comerón and Aguadé (1998) of $N_c$, and other measures (such as MCB and $B_g(a)$) that they are affected strongly by the sequence lengths being studied. Short genes are problematic in that their average bias levels differ from longer genes even when their pseudo-random codon sequences are drawn from the same distributions.
Figure 2.4: $N_c$-plot for yeast and simulated *E. coli* K12 genes. Grey points, simulated *E. coli* K12 genes; red points, actual yeast genes; dashed line, Wright’s proposed function (Equation 2.26); bold line, the function proposed by dos Reis et al. (2004) (Equation 2.27) with optimized parameters.

Reproduced from URL:
http://www.pubmedcentral.nih.gov/picrender.fcgi?artid=521650&blobname=gkh834f2.gif
2.4 Alternative Techniques for Predicting Expression Rates

There are other, non codon usage bias methods that have been attempted in predicting protein production rates. One such method is the use of hidden Markov models (Goutsias, 2006) for studying transcriptional regulation. The underlying methodology is to simulate the biological system through the use of a stochastic model and employ the hidden Markov model to determine the “hidden” number of occurrences of the underlying reactions. This is similar to other stochastic models of protein behaviors, such as that employed by Raser and O’Shea (2004) and Anderson et al. (2005), except that in those the reaction rates are known and fragment concentration is predicted while the hidden Markov model has a known concentration and attempts to predict the reaction rate.

2.5 Visualization – Principal Components Analysis

This research employs several data visualization techniques for exploratory data analysis. Those that are novel are covered in Chapter 3. An existing technique is principal components analysis (PCA) (Hotelling, 1933; Jolliffe, 1986) (Appendix B). PCA is a commonly employed method for dimensionality reduction that has been applied to visualization, and has frequently been applied to codon usage analysis (Kanaya et al., 1996; Carbone et al., 2003; Grantham et al., 1981). It is an integral part of some codon usage determination techniques (Kanaya et al., 1996, 1999; Grantham et al., 1980, 1981, 1985), though some shortcomings have been noted, such as a loss of information when relative frequencies are used, leading to difficulties in interspecies comparison (Perrire and Thioulouse, 2002).

PCA was described earlier in the section on calculating major codon usage (MCU) (Section 2.1.10) (Kanaya et al., 1996). In the case of MCU it was used to determine the placement of the genes on the first principal component (axis of greatest variance). For visualization purposes, the placement of the genes on the first two principal components can be determined (Equations 2.28...
and 2.29; Figure 2.5) and those values graphed yielding a two-dimensional view of the codon usage space.

\[
X \cdot b_1 = Z'_1 \\
X \cdot b_2 = Z'_2
\]  

(2.28)  
(2.29)

Figure 2.5: Graphical representation of \( X \cdot b_i = Z'_i \)

PCA is useful in identifying discrepancies in the locations of sets of genes. Figure 2.6 illustrates how PCA can be used to depict the possible presence of confounding biases. Figure 2.6(a) reveals the location of the SCCI derived reference set (small set of genes that adhere strongest to the bias they, themselves, define) for \( Nostoc \ sp. \ PCC \ 7120 \). Figure 2.6(b) depicts the location of ribosomal protein coding genes which are known to be highly expressed. It is noteworthy that they are not in the same region of the genome as the reference set representative of the dominant bias. Shading can be used to explore possible origins of this condition. Figure 2.7 displays genes shaded according to their GC-content. Darker genes have higher GC-content. The reference set for the dominant bias (Figure 2.6(a)) is more likely to be the set of genes identified due to a high AT-content bias. Indeed, using content criteria (Section 2.2.1), it can be shown that \( Nostoc \ sp. \ PCC \ 7120 \) has AT-content as its dominant bias (Table 4.2).
Figure 2.6: *Nostoc sp. PCC 7120*. 2.6(a) Reference Set. Small set (1% of genome) of genes identified by CAI algorithm as being highly expressed. Each point represents a gene. The dark genes comprise the reference set. 2.6(b) Ribosomal protein coding genes. Genes known to be, generally, highly expressed. Each point represents a gene. The dark genes are ribosomal protein coding genes. Ribosomal protein coding genes are distant from region identified by the reference set. This indicates that the bias (strong and consistent use of specific codons) identified by the CAI algorithm is confounded and is not representative of translational efficiency bias.

Figure 2.7: *Nostoc sp. PCC 7120* GC-content. Dark genes have higher GC-content. Lighter genes have low GC-content (high AT-content). The high AT-content genes are in the same region identified by the CAI algorithm as representative of the reference set. This indicates that the CAI algorithm identified AT-content bias, not translational efficiency bias.
As stated before, a major contribution of this work is the creation of an automated approach to isolating translational efficiency bias that is resistant to the confounding influence of GC(AT)-content. I will present two techniques for achieving this objective: one that is a modification to the existing SCCI algorithm, and another that employs a soft computing technique known as a Genetic Algorithm (GA). Thus, one solution (modified SCCI) is deterministic (greedy hill-climber) while the other is more stochastic in nature.

An evolutionary computational technique is one that is modeled after nature’s approach to optimizing a species’ fitness for survival (Friedman, 1956; Holland, 1975a,b; Goldberg, 1989). The algorithm simulates natural selection by maintaining a population of individual members, each of which is a proposed solution to a problem. The search for the best solution proceeds by selecting members from the population to produce offspring that are similar to the parents, but that have slight differences thereby injecting genetic diversity into the population. A fitness score is assigned to each proposed solution and poor solutions are removed from the population in a process that mimics natural selection (hence the description: evolutionary computation).

### 2.6.1 Taxonomy of EC

The taxonomy of evolutionary computational (EC) techniques has grown in diversity and complexity over time. Some approaches attempt to combine the traits of multiple parents through a process known as crossover (inspired by eukaryotic recombination). These approaches tend to fall into a category known as genetic algorithms (Holland, 1962), and traditionally represent solutions as bit-strings on which the crossover and mutation operations are performed (though this is not usual anymore). Other methods traditionally represent solutions as real valued vectors (vs. bit-strings) and rely on mutation only (vs. crossover and mutation). These are known evolutionary programming techniques (Fogel, 1962). Others traditionally employ real-valued vectors with crossover and
Randomly generate a set of proposed solutions (members)
Assign fitness scores to initial population
For N generations (or until an appropriately good solution is found)
   Draw parents from the population
   Generate offspring from the subpopulation of parents
   Select members that will constitute the next generation

Figure 2.8: Pseudo Code Describing Typical EC Algorithm

mutation. These techniques are in the family known as evolution strategies (Bienert, 1967).

2.6.2 Overview

A detailed treatment of the various evolutionary computational techniques is beyond the scope of this dissertation. Instead, an overview of general procedures will be given to provide the reader with a practical working knowledge of evolutionary computing techniques. Most EC approaches have a common algorithmic structure (Figure 2.8). They begin with a randomly generated set of proposed solutions (the population). Fitness scores are assigned to each member of this population according to how well the respective solution performs (e.g. in a search for a reference set one could assign fitness based upon how self-consistent the solution is). Parents are drawn (selected) from this population and offspring generated for the next generation. Finally, the members that will constitute the next generation are selected, and the process is repeated until a suitable solution is discovered. The various disciplines within the broader category of evolutionary computation differ only in the implementation of these basic steps. They might have different representations or different ways in which to perform the selection of parents and/or members of the next generation or different stopping criteria. For instance, some approaches draw the next generation from just the children while other techniques draw from both the parent and child populations, forcing them to compete against each other.
2.6.3 An Example

As an example, consider knapsack problem. It is an optimization problem where items must be chosen for inclusion in a knapsack that maximizes the value of the items included without exceeding a preset threshold-weight. Each item is assigned a value and a weight. The problem becomes a search for the set of items whose combined weight is less than the threshold but represents the maximum value attainable given the weight constraint. There are many variations on this problem, such as using volume instead of weight, etc. The problem could be represented by a bit-string where each item is represented by one of the bits. A one would then represent the inclusion of the associated item in the knapsack and a zero could represent excluding it. The fitness might simply be the summed cost of the items included in the knapsack (and zero if the weight threshold is exceeded). An initial population could be generated by randomly turning on bits in a series of bit-strings (population of search members). The fitnesses of the initial population would be calculated and then parents chosen in order to create offspring. Figure 2.9 depicts crossover being performed on two parents in order to generate a child member (where the first three bits of parent one and the last four of parent two are used). Mutation might be performed (commonly a mutation rate of approximately one bit over the entire bit-string is flipped). In this case, the fifth bit of the child bit-string is flipped from 1 to 0. The child now represents an entirely new solution for which a fitness score can be computed and a determination made as to whether it should be kept for the next generation.

The crossover operation described in Figure 2.9 represents one-point crossover. Variations on this approach include two, three, etc., -point crossover. Uniform crossover is another alternative, where a probability is assigned to determine the likelihood that each bit will come from a given parent. Mutation rates have an effect on the distance from the original solution a child solution is. Some approaches utilize mutation rates that start out high but decreases later in the search, as population members begin to converge on the solution.
Figure 2.9: Example of Crossover and Mutation of a Bit-String
Multi-Objective Problems

Many problems are not easily mapped to a fitness value represented by a single number. Often there are multiple characteristics of a solution to consider. For instance, in the knapsack problem, what if an optimal solution was one that filled the knapsack with as little combined weight as possible and with the greatest overall monetary value. It is no longer a matter of simply keeping the weight below a given threshold and maximizing value. Now, if two solutions have the same value but one is lighter it is a better solution. But what if one is heavier with a greater value, and the other is lighter with a lesser value? Which is the better solution? Often, multiple objectives of this nature are combined into a single measure, such as “cost per unit weight.” But it is not always easy to establish a relationship between multiple fitness measures that allow this type of transformation. This is certainly the case in the problem of identifying reference sets associated with translational efficiency. One might wish to measure how self-consistent a solution is, but another measure may need to be incorporated that describes how likely the solution is to be one associated with translational efficiency (verses, say, GC-content). There has been a great deal of research involving multi-objective evolutionary algorithms and the problem of determining a single fitness measure when there are multiple objectives.

Dominance and Pareto Fronts

One of the better-known techniques is Pareto ranking (Fonseca and Fleming, 1993) which is based upon the concept of dominance. In the case of a two-objective problem 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. However, 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 optimized, 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 (Pareto, 1897) (Figure 2.10). 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 generating a count of the other members that dominate the member in question. One member, whose objective measures can be represented by a $k$-dimensional vector ($\vec{u}$) is said to dominate ($\vec{u} \succeq \vec{v}$) another ($\vec{v}$) 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:

$$\forall i \in \{1, ..., k\}, u_i \geq v_i \land \exists i \in \{1, ..., k\} : u_i > v_i$$

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. This count can be used as measure of fitness that incorporates both objective measures. Members on the Pareto front constitute the Pareto optimal set ($P^*$) which is defined as the set of all $x \in \Omega$ [$\Omega$ is the set of proposed solutions (inputs to the multi-objective fitness function) that meet any imposed restrictions or constraints] such that there exists no other solutions that are better (that dominate $x$) (Equation 2.31). The multiple objectives being measured are returned from the function $\vec{f}$ which takes as input the proposed solution $x$. This allows the Pareto front ($PF^*$) to be defined in terms of the Pareto optimal set (Equation 2.32).

$$P^* := \{ x \in \Omega | \exists x' \in \Omega \; \vec{f}(x') \succeq \vec{f}(x) \}$$

$$PF^* := \{ \vec{f}(x) | x \in \Omega \land x \in P^* \}$$

Over time the number of solutions that an evolutionary algorithm generates that are on the Pareto front (during a given generation) may exceed the allowed population size for the next generation. When this occurs a method must be devised to identify which members of the population
to keep, and which to discard. A commonly employed technique for making this determination is niching.

### 2.6.4.2 Niching

Niching techniques often 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 is injected into the process under the assumption that there is something inherently noteworthy about the region of the search space described by the cluster.
**2.6.5 Why EC Works**

The motivation behind developing EC strategies is the discovery of solutions to problems that previously appeared intractable. Evolution in nature has solved very difficult problems for species in order that they might survive and flourish. Proponents of EC approaches believe that bringing this same powerful methodology to bear on complex problems in computer science might yield solutions that would otherwise be unobtainable (Rechenberg, 1965, 1994; Schwefel, 1965; Schwefel et al., 1995). Theorists have put forward various explanations, from a computer science perspective, as to why evolutionary computation performs so well on these difficult problems. Most point to the implicit parallelism of EC techniques (Wright et al., 2003) and the way information is shared between population members (regarding the solution space in their neighborhood) through operations such as crossover (Sywerda, 1989; Holland, 1975a; Goldberg, 1989). It is generally accepted, however, that no single approach is best for all problem domains. This is known as the “no free lunch” theorem (Wolpert and Macready, 1995). This leads to the conclusion that successful search strategies are often those that inject a good balance of domain knowledge into the search process.

**2.6.6 Fitness Landscapes**

There is a great deal of research into the analysis of search spaces encountered in optimization problems. The more that is known about the nature of the search space for a given problem, the better able one will be to design an appropriate search methodology. Search spaces are often described as fitness landscapes where the elevation of the current location in the space represents the fitness of the represented solution (Figure 2.11). One of the major objectives of this research is the analysis and visualization of the fitness landscape for the problem of locating a reference set indicative of genomic codon usage bias [Section 3.1 and (Raiford III et al., 2006b)]. This work builds upon a broad body of work previously performed in the area of biological fitness landscapes (Wright, 1932) as well as techniques for analysis of fitness landscapes in evolutionary computation (Ghosh et al., 2003).
Global Maximum

Local Maximum

Simple One-Dimensional Search Space

Figure 2.11: Example of a Simple Fitness Landscape
2.6. EVOLUTIONARY COMPUTATION (EC)  

2.6.6.1 Relevance to the Search for Translational Efficiency Bias

Fitness landscapes are relevant to the problem of identifying codon usage bias because, in the case of SCCI, the problem can be thought of as a search for a subset of genes (the reference set) that maximizes a particular criterion. The optimal reference set is a set of genes that define a bias and that adhere more strongly to that bias than any other genes in the genome. The degree to which a proposed solution (proposed reference set) adheres to this definitional requirement represents the quality, or fitness, of the solution. This concept is mathematically codified in Section 3.1; however, for now it is sufficient to have a conceptual understanding. The span of possible solutions that must be searched can be thought of as a landscape, and the quality or fitness of each proposed solution determines whether the visited solutions are good solutions to the search problem.

2.6.6.2 Analysis

Fitness landscapes can be characterized in terms of their ruggedness, smoothness (opposite of ruggedness), and neutrality. If a landscape is not rugged then it is either smooth or neutral. A landscape is often thought-of as a view of the fitness of solutions given all possible paths through the solution space utilizing the smallest possible incremental moves. Neutrality introduces the idea that a fitness landscape must, instead, be thought of in terms of the moves enacted as a result of mutation or crossover. A given operation may move an individual in the population a large distance when compared to single point mutations or small, incremental changes in a real valued space. When this occurs, and when similar fitness values are encountered often after such jumps, the landscape is characterized as neutral.

There are various methods of statistically characterizing fitness landscapes through the use of random walks, such as the degree to which the landscapes are autocorrelated (Vassilev et al., 2003; Weinberger, 1990; Stadler, 1996). A fitness landscape is autocorrelated if the fitness values accumulated during a random walk correlate with themselves. Clearly, data will correlate perfectly (a correlation coefficient of 1.0) when compared to itself. An autocorrelation performs a series of
these measurements shifting one set of the data by a predetermined amount for each measurement. Such a series of measurements will necessarily begin with a correlation value of 1.0, and will get progressively worse as the starting point is advanced until there is no correlation between the two random walks.

Measurements taken during an actual search can yield insights into the landscape, as well. An example is the search velocity of the population. If a population gets “stuck” on a local optimum, the rate at which the population moves about the search space will decrease until some new, more fit solution is introduced into the population. An indication of this phenomenon can be found in the centroid-to-centroid distance for populations that are some number of generations apart. This is known as a lagged diffusion coefficient, and it can provide insights into the nature of the search space and how well the parameters being employed (mutation rates, amount of crossover, representation, etc.) perform in the given problem domain.

2.6.6.3 Impact of Representation

Representation of possible solutions has a large impact on fitness landscapes. In the search for a reference set of genes the objective is to find a fixed-size subset equal to about 1% of the number of genes in the genome. An obvious choice of representation is a bit-vector with ones indicating membership in the reference set and zeroes indicating nonmembership. With respect to visualizing fitness landscapes, the difficulty resides in deciding what represents a neighboring point in the landscape. To begin with there are thousands of dimensions (there are thousands of genes in most prokaryotic genomes) and these dimensions contain binary data. In order to visualize a fitness landscape for this problem domain one must first devise a method of arranging such a complex set of solutions in one or two dimensions in order to display the fitness in the space above the associated solutions is a difficult proposition. Add to this the fact that the search space is incredibly large (all possible sets of 1% of the genes in a genome) and you can begin to see the difficulties involved in visualization. As an example, the organism *Nostoc sp. PCC 7120*
(a genome studied extensively in this work) has 5,366 genes. The number of possible solutions is \( \binom{5,366}{54} \), or \( 8.35 \times 10^{129} \) alternatives. It would be impossible to exhaustively solve all of these possible solutions, let alone visualize them. Even if one could calculate the fitness of one of these solutions in one nano-second it would still take \( 2.65 \times 10^{116} \) years to complete this exercise. With such a large solution space it becomes very important to select an appropriate sampling of possible solutions when displaying the topology of the search space.

### 2.6.6.4 Adjacent Solutions and Mapping a High-Dimensional Search Space Into a Reduced-Dimensionality Visualization Space

It is also important to be able to identify near neighbors in the high-dimensional search space. To illustrate this point, consider the smallest movement through the search space: the point mutation. Clearly, any bit that gets flipped must be accompanied by some other bit being flipped in the opposite direction (if a 1 is set to a 0, then an accompanying 0 must be set to 1) in order to maintain the same number of ones in the bit-vector. The problem becomes one of determining what it means to be neighbors in this space (and how to map this concept into a lower-dimensional visualization space). There has been some research in this area in the form of analysis of digital necklaces (Wang and Savage, 1996). Digital necklaces are an equivalence class of binary strings (under rotation) in which each instance has exactly the same number of ones and that are gray code in nature. Ordinarily adjacent gray codes differ by a single bit, but in this case they must differ by two in order to maintain the same number of bits. The term necklace comes from the fact that a given configuration of ones is listed only once. If the vector is thought of as a necklace where the start and ending points are connected, then two configurations are considered equivalent if they can be made identical by rotating the strings about the necklace until all the ones are aligned.

Unfortunately, having a representation that identifies neighbors and that captures the gradual nature of di-point mutations is only half of the battle. If the landscape is to be visualized, a method must be devised to map the enormous number of near neighbors into a two-dimensional grid for
visualization (the third dimension to be used to depict level of fitness). A small genome of 3,000 genes yields a reference set of size 30. Any given proposed solution would have, for each 1 that is flipped to a 0, approximately 3,000 nearest neighbors ($30 \times 3,000$) yielding approximately 90,000 neighbors. A contribution of this work is the formulation of a method for selecting a small number of appropriate neighboring solutions and arranging them in an easy to visualize two-dimensional grid with the fitness of these solutions depicted as a surface suspended above the grid (Section 3.1).
Analysis of the Problem Domain: Fitness Landscapes and Novel Visualization Techniques

Before setting about automating the search for translational efficiency bias, a careful inspection of the problems associated with the codon usage bias search space is in order. Examination of fitness landscapes can be useful in determining search strategies and predicting overall problem difficulty. In particular, visualization can help in the exploratory data analysis phase of research. While the PCA derived two-dimensional space shows the relationship among genes in terms of their codon usage, this two-dimensional projection does not facilitate recognition of strong trends or biases. A more direct visualization of competing biases can be realized by extending this space into a third dimension that is based on the quality of reference sets in the specified region of codon usage. The resulting three dimensional space is termed a fitness landscape in evolutionary computation, but will be identified as a *reference set quality landscape*, or RSQ landscape, to avoid confusion with the fitness landscapes described in molecular evolution research.
3.1 Visualizing the Fitness (RSQ) Landscape

Some of the difficulties associated with visualizing the fitness landscapes associated with searching for translational efficiency bias were first visited in Section 2.6.6. The solution space of all possible subsets of genes (of size one percent) in a typical genome is incredibly vast (one organism's genome was shown to have $8.35 \times 10^{129}$ possible solutions). Furthermore, the most obvious representations have more dimensions than can be visualized (that same organism had 5,366 dimensions in its bit-string representation). A method must be devised for reducing the number of solutions that must be computed to a number that is manageable, and another method must be developed for arranging these representative solutions in one or two dimensions to facilitate visualization of the fitness associated with each solution. These are difficult issues. How does one reduce the dimensionality of a bit-string? Which solutions in this vast space are worthy of depiction?

3.1.1 A Proposed Method for Reducing High-Dimensionality of the Bit-String Representation and Selection of Solutions

The method proposed here takes advantage of characteristics that all reference sets (here it is meant a small set of highly expressed genes) share. Most reference sets consist of genes that are in close proximity to one another in the codon usage space (Figures 2.6(a), 2.6(b), and 2.7). This property is exploited in order to choose a manageable number of solutions for depiction. A solution is unlikely to be a high-quality solution if its member-genes are in disparate regions of the codon usage space. This allows us to reduce the number of candidate solutions to those whose constituent genes are in close proximity to one another. Depending on what criteria one chooses to make this determination, the number of solutions could still be too large to compute in a reasonable amount of time. If the criteria is made as strict as possible (i.e. the reference set is only considered if it represents a set of genes with no other genes within the boundaries established by the outer members of the reference set) then the number of solutions being considered begins to approach something reasonable.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

It is unlikely that a solution with the best reference set quality score would exhibit such tight compaction; however, there would exist such a candidate reference set somewhere within its boundaries. If it is assumed that such a subsumed reference set would have a bias similar to that of the less-tightly-packed, though higher quality, encompassing reference set then it can be safely (also) assumed that it will have a similar (though slightly lower) fitness score. Also, it is likely that other reference sets drawn from this region will have similar fitness scores yielding a generally raised region in a visualized fitness landscape. So, while the best solution is unlikely to be included in the displayed set of solutions (unless it happens to be a very tightly packed set of genes) solutions near it will be included, and it will be easy to determine which regions will yield high-quality reference sets.

3.1.2 Implementing the Solution

To implement and test these theories my procedure selects sample solutions and arranges them for viewing in the following way. A reference set is created for each gene in the codon usage space (row in the RSCU matrix; Figure 2.1). It is built from the nearest neighbors to the gene in question. The second issue, that of how to reduce the dimensionality, is now easily solvable since there is a reference set associated with each gene in the genome and researchers in this field are accustomed to viewing the RSCU data in a two-dimensional representation using principal components analysis. The quality of each gene-associated reference set is determined and the quality score is used as the value in the third, orthogonal, dimension. This creates a surface suspended above the PCA derived two-dimensional codon usage space.

The specific procedures for building the RSQ landscape are described in (Raiford III et al., 2007a, 2006b). In summary, proposed reference sets representative of biases in different regions of the PCA derived two-dimensional codon usage plots are generated by gathering the nearest neighbors to each gene into discrete reference sets. This is accomplished by first determining the Euclidean distance between the RSCU vector of each gene and that of every other gene. Once these
3.1. **VISUALIZING THE FITNESS (RSQ) LANDSCAPE**

May 27, 2008

Distances are computed, a set of local neighbors is constructed for each gene. This set consists of the one percent of genes nearest to the gene in question based on the RSCU distance. The RSQ landscape is assembled by determining a quality score for each gene’s neighboring set and using this as a value in the third dimension, orthogonal to the two-dimensional PCA derived plane. A surface is then constructed using these quality scores.

### 3.1.2.1 Quality (Fitness) Scores

To calculate the quality score for gene $i$, SCCI scores are first assigned to all genes using formula 2.21 where the weights are defined by the reference set comprised of the nearest neighbor genes (1% of genes in genome) to gene $i$. The genome is then sorted by this SCCI value, and the degree to which gene $i$’s reference set rises to the top of the sorted list is assessed (Equations 3.1 through 3.3). A reference set is defined as a small set of genes (1% of genome) characterizing a bias to which its adherence is stronger than all other genes in the genome. Intuitively, the closer a proposed reference set’s behavior is to the definition of a reference set, the higher its quality can be said to be. In Equation 3.1 $RS$ is the proposed reference set, $|RS|$ is the size of the proposed reference set, $N$ is the number of genes within the genome, and $IDX$ is the index of the proposed reference set gene $i$ in the sorted list of all genes. An ideal reference set (one that matches, exactly, the mathematical definition of a self-consistent reference set) will rise to the top of this list and is represented by Equation 3.2. This measure will assign a score (from 0 to 1) to a proposed reference set. To put this in context, the Carbone et al. algorithm is designed to locate reference sets with perfect quality scores. The score is used in the construction of the RSQ landscape, but it can also be viewed as a measure of the quality of a proposed reference set. This measure of quality is known as a fitness score in evolutionary computation techniques, and is used in the construction of fitness landscapes for the purpose of search-space analysis (Ghosh et al., 2003; Vassilev et al., 2003; Weinberger, 1990). This approach is, in turn, based upon the work of biologists in the field of genetics and evolutionary theory (Wright, 1932).
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

\[ f(RS) = \sum_{i=1}^{\mid RS \mid} IDX_i \]

(3.1)

\[ f_{max}(\mid RS \mid) = \sum_{i=N-\mid RS \mid}^{N-1} i \]

(3.2)

\[ f_{norm}(RS) = \frac{f(RS)}{f_{max}(\mid RS \mid)} \]

(3.3)

3.1.2.2 Constructing the Surface

The surface of the RSQ landscape is constructed by creating a regularly spaced grid of points within the PCA determined two-dimensional plot space. A triangle-based linear interpolation method (based on a Delaunay triangulation using the quickhull algorithm for convex hulls) is used to determine an associated value at each grid point (Barber et al., 1996). This value is an aggregation of the nearby gene quality scores. The surface is then rendered orthogonal to the two-dimensional plot of RSCU data projected onto the first two principal components.

3.1.3 RSQ Landscape for an Organism Dominated by AT-Content Bias with Weaker Translational Efficiency Bias

A good place to begin the study of RSQ landscapes is with Nostoc sp. PCC 7120. It has one of the more fascinating RSQ Landscapes in that it portrays a genome that exhibits two distinct biases in two entirely different regions of the codon usage space. This was shown using PCA in Section 2.5, and is presented here again (Figure 3.1) for convenience. The SCCI algorithm discovers a reference set in a different region of the codon usage space than where ribosomal protein coding genes are found. The SCCI algorithm is designed to find the dominant codon usage bias for a genome and, in this case, the dominant bias is in a region where high AT-content genes are found (Figure 3.1(c)). As supporting evidence that this is, indeed, the case, the reference set discovered by SCCI results
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

May 27, 2008

in a content criterion of $-0.915$ for the genome (Table 4.2; Section 2.2.1). Content criterion is the correlation between SCCI values and GC$_3$-content. The threshold for concluding that content bias is dominant is a correlation $> 0.7$ for GC-content or $< -0.7$ for AT-content (Carbone et al., 2005). Clearly, SCCI has discovered a dominant bias of AT-content for this organism.

Although AT-content is the dominant bias, the ribosomal protein coding genes are very tightly grouped (Figure 3.1(b)) indicating that they exhibit similar (to one another) codon usage. Their proximity to the outer edge of the PCA plot suggests that they are highly biased (PCA is plotted along the axes of greatest variance so genes at the extremities are those that deviate most from average usage). Thus, if they are tightly grouped and exhibit strong bias there is a good possibility that they will be self-consistent and form a high-quality reference set. In this situation the RPCGs should adhere more strongly than most other genes to the bias that they themselves define (self-consistent). If this is the case then an RSQ landscape should depict two regions where high-quality reference sets can be drawn – one in the region where high AT-content genes are located, and the other where genes that adhere to translational efficiency bias reside. And that is precisely the result. Figure 3.2 depicts a landscape dominated by two regions of the codon usage space where high-quality reference sets can be found. Their corresponding proximity to the locations of the SCCI reference set (representative of the bias induced by high AT-content) and the ribosomal protein coding genes is a strong indication that one ridge corresponds to high-quality AT-bias solutions and the other to high-quality translational efficiency bias solutions.

3.1.4 A Translational Efficiency Dominated RSQ Landscape

Having presented an unusual genome exhibiting distinct multiple-biases, the analysis turns now to a more traditional genome, *E. coli*. *E. coli* is one of the most extensively studied genomes in all of biology. It is often considered a model organism for the study of prokaryotic genomes, and conclusions drawn from its analysis are frequently generalized to that of other prokaryotic organisms. But even this prototypical organism presents remarkable results when studying its RSQ landscape.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

Figure 3.1: Nostoc sp. PCC 7120. 3.1(a) Reference Set. Small set (1% of genome) of genes identified by SCCI algorithm as being representative of the dominant bias. Each point represents a gene. The dark genes comprise the reference set. 3.1(b) Ribosomal protein coding genes. Genes known to be, generally, highly expressed. Each point represents a gene. The dark genes are ribosomal protein coding genes. Ribosomal protein coding genes are distant from the region identified by the SCCI reference set. This indicates that the bias (strong and consistent use of specific codons) identified by the SCCI algorithm is confounded and is not representative of translational efficiency bias. 3.1(c) Dark genes have higher GC-content. Lighter genes have low GC-content (high AT-content). The high AT-content genes are in the same region identified by the SCCI algorithm as representative of the reference set. This indicates that the SCCI algorithm identified AT-content bias, not translational efficiency bias.
Figure 3.2: *Nostoc sp. PCC 7120* RSQ Landscape. The surface represents quality scores of reference sets in close proximity to the genes in a PCA-derived two-dimensional space ($Z_1$ and $Z_2$) depicted along an orthogonal axis (vertical axis). Elevated regions (elevated regions are red in color, low-lying regions are in yellow) of the reference set quality (RSQ) landscape can be thought of as regions from which high-quality reference sets can be drawn.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

Figure 3.3 reveals the RSQ landscape for \textit{E. coli}. Once again, the landscape is dominated by two regions from which high-quality reference sets can be drawn. The larger, dominant one is associated with translational efficiency bias. This can be confirmed by examining the location of highly expressed genes [Figure 3.4(b); Expression data taken from NCBI’s Gene Expression Omnibus (Barrett et al., 2007; Liu et al., 2005)] within the PCA-derived two-dimensional space. The RSQ landscape is suspended above this plot meaning that the ridge directly above the location of the highly expressed genes is likely to represent genes exhibiting strong translational efficiency bias. Also, the ribosomal criterion achieved when these genes are used as a reference set is 2.390 indicating the bias described by these genes is likely that of translational efficiency (the threshold is one standard deviation above the average CAI score for the genome).

The second peak appears to be associated with genes that have been horizontally transferred into \textit{E. coli}’s genome. Figure 3.4(a) depicts the location of genes thought to be horizontally transferred. Once again, the ridge is found above the same region where high concentrations of these genes can be found. The implication is that a large-scale HGT event (a relatively large number of genes) took place where the genes were all of similarly biased composition and where the bias was relatively strong. The fact that they are of similar composition is evidence that they are all from the same (or similar) organism(s). Interestingly, Carbone et al. (2003) indicated that when they employed a random search and they began the search in the area indicated by this peak their algorithm found a reference set indicative of the predominant HGT bias, but made no mention that this implied a large-scale HGT event.

\textit{Nostoc} and \textit{E. coli} are organisms with two biases in distinct regions of their codon usage space (translational efficiency and AT-content for \textit{Nostoc} and translational efficiency and HGT-introduced bias for \textit{E. coli}). This is not always the case. Some biases act in tandem on similar sets of genes, and when this occurs the ridges in the RSQ landscape can intersect, overlap, or merge. \textit{Streptomyces coelicolor A3(2)} is an organism that exhibits a merged bias. It can be seen that the ribosomal protein coding genes are on the ascending slope of the dominant solution’s ridge (Figure 3.5(c)).
Figure 3.3: RSQ Landscape for *Escherichia coli*. High-quality reference sets can be expected to be drawn from elevated regions (elevated regions are red in color, low-lying regions are in yellow and orange). The dominant ridge is associated with translational efficiency bias. The secondary peak is associated with a region believed to be occupied by horizontally transferred genes.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

Neisseria meningitidis MC58 presents an interesting RSQ landscape (Figure 3.6). There are two obvious ridges; one that is associated with high GC-content (\( \sim 77\% \)) and one associated with high AT-content (\( \sim 82\% \)). Chapter 4 will show that there is actually a third bias, translational efficiency, that resides between these two dominant biases [the SCCI algorithm isolates the GC-content bias (content criterion 0.750) making it the more dominant of the two biases]. One could make an argument that there are three relatively strong biases at play in this organism’s genome.

3.1.5 A Strand Bias Dominated RSQ Landscape

The three biases for which Carbone et al. (2005) developed measurement criteria are translational efficiency bias, GC-content bias, and strand bias. The previous sections depicted reference set quality landscapes of organisms whose genomes are dominated by translational efficiency bias and GC(AT)-content bias. Figure 3.7 is the RSQ landscape for Borrelia burgdorferi B31 which has been shown to be dominant for strand bias (Carbone et al., 2005). Note the presence of two ridges.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

Figure 3.5: 3.5(a) *Streptomyces coelicolor A3(2)* is an example of a genome where GC-content constitutes the dominant bias. The translational efficiency bias ridge is merged or subsumed into and by the GC-content ridge. To help understand the surface of the RSQ landscape, a two-dimensional slice that is horizontal with $Z1$ (slice made at -3 on $Z2$) is provided in 3.5(b). Evidence of this can be seen in 3.5(c) and 3.5(d) where ribosomal protein coding genes (white points) are shown to be on the ascending slope of the dominant ridge, below the identified reference set. The reference set is as defined by the SCCI algorithm (GC-bias).
Figure 3.6: RSQ Landscape for *Neisseria meningitidis* MC58. High-quality reference sets can be expected to be drawn from the elevated regions (elevated regions are red in color, low-lying regions are in green and yellow). There are two dominant ridges. One is associated with high GC-content (≈ 77%) while the other is associated with high AT-content (≈ 82%). Chapter 4 will show that there is actually a third bias, translational efficiency hiding between these two.
at opposite ends of the codon usage space.

Overall, the dominant bias in *B. burgdorferi* is neither translational efficiency nor content bias (SCCI yields a ribosomal criterion of 0.11 and a content criterion of 0.27; the traditional Sharp and Li (1987) approach using RPCGs as a reference set yields a ribosomal criterion of 0.33). This is to be expected since it is known that it is dominant for strand bias. The leading strand is very close to being dominated by AT-bias (when SCCI is calculated on just leading strand genes, their content criterion is −0.66; if it were < −0.7 one would identify it as being dominant for AT-bias). A possible explanation may be found in the fact that one of the causes for strand bias is C→T transitions on the non-template strand. The leading strand has more genes (543 vs. 268), so the non-template strand, when expressing these genes, would build up more T’s (and the template strand their associated A’s). The leading and lagging strand genes generate a ribosomal criterion of −0.40 and −8.56, respectively. While one cannot say that either strand exhibits translational efficiency bias, the dramatically less negative ribosomal criterion for the leading strand may be evidence that the leading strand is a better search space for such translational efficiency bias. This is a good topic for future research.

In order to verify that the two ridges are indeed representative of the biases resident on the two DNA strands, SCCI scores were calculated for the genes resident on each strand. Figure 3.8 depicts the genes in PCA space color-coded according to their adherence to the biases inherent to the two strands. The blue genes adhere strongly to the lagging strand bias, while the redish/orange genes adhere more strongly to the leading strand bias. Since the two distinct regions coincide with the location of the ridges, this is strong evidence that the two ridges are, in fact, associated with biases resident on the two strands.
3.1. VISUALIZING THE FITNESS (RSQ) LANDSCAPE

May 27, 2008

Figure 3.7: RSQ Landscape for *Borrelia burgdorferi* B31. Note the presence of two, distinct, ridges on either side of the codon usage space (elevated regions are red in color, low-lying regions are in blue and green). Strand bias is characterized by distinct biases on each strand due to transcription induced C→T transitions on the nontranscribed strand (Francino et al., 1996; Francino and Ochman, 1997; Beletskii and Bhagwat, 1998) or replication induced increased G+T concentrations on the leading strand (Lafay et al., 2000, 1999; Rocha et al., 1999). It is worth investigating whether these two ridges are representative of the two separate biases resident on the two strands.
Figure 3.8: PCA for *B. burgdorferi* Genome. SCCI scores were calculated for the genes resident on each strand. Figure 3.8 depicts the genes in PCA space. They have been color-coded according to their adherence to the biases derived from each of the two strands. The blue genes adhere strongly to the lagging strand bias, while the redish/orange genes adhere more strongly to the leading strand bias. This is strong evidence that the two ridges in the RSQ landscape are, in fact, associated with biases resident on the two strands.
3.2 Fitness Landscape Analysis

3.2.1 Random Walk – Region Near Perfect Solution

Visualizing the fitness landscapes associated with the codon usage bias search space, while useful and informative, does not provide quantitative measures useful in devising promising search strategies. For this, the investigation turns to sampling and statistical analysis. A random walk through the search space can provide insights into the nature of the problem domain. For the purposes of this work, the moves generated during the random walk are those performed by a genetic algorithm whose chromosomal representation is a bit-string with each bit representing a gene. To achieve as detailed an analysis as possible, the smallest incremental move in this search space (given the bit-string representation of the chromosome) will be employed. A “1” at a location in the chromosome indicates that the proposed solution (a proposed reference set) contains the indicated gene. Since a reference set has a fixed size of 1% of the genome size, the smallest step in the search space involves randomly turning-off one of the bits that are on (has a value of 1) and randomly turning-on one of the other bits. This mechanism (that modifies the chromosome) is termed a random di-bit mutation operator. Figure 3.9 is a representation of fitness levels for the various steps (produced by the random di-bit mutation operator) taken in a random walk that begins at the summit in the RSQ landscape (the prototypical organism, *E. coli* was chosen for this analysis) where the ideal translational efficiency solution is found (fitness of 1.0). Note that there is a rapid fall-off as the “walker” moves away from the summit. Also note that it quickly levels off to a range of between 0.4 and 0.6. This analysis was performed many times and for walks of length 10,000 random di-bit moves with the same result (data not shown).

3.2.2 Random Walk – Random Starting Location

To determine whether this relatively flat landscape is characteristic throughout the solution space, and not simply in the region visited when starting from the summit, experiments were run where
3.2. FITNESS LANDSCAPE ANALYSIS

Figure 3.9: Random Walk in Codon Usage Search Space for *Escherichia coli*. The random walk begins at the summit. The moves are the result of random di-bit mutations.
3.2. **FITNESS LANDSCAPE ANALYSIS**

the starting solution was randomly chosen (a random set of 1% of the genes in the chromosome). Figure 3.10 depicts a typical such random walk. A simple visualization of the values encountered during a random walk can give us a fairly good idea of the nature of the search space; however, it does not give us any meaningful quantitative measurements describing the space.

![Random Walk in Codon Usage Search Space for *Escherichia coli*.](image)

**Figure 3.10**: Random Walk in Codon Usage Search Space for *Escherichia coli*. The random walk begins at random location. The moves are the result of random di-bit mutations.

### 3.2.3 Ruggedness, Smoothness, Neutrality

The smoothness (and its opposite, ruggedness) of a landscape describes the likelihood that neighboring points in the search space will have similar fitness values (Kauffman, 1989; Palmer, 1991). It characterizes whether established trends (like fitness values being on the rise) are likely to continue for a given path through the solution space. Related to these is neutrality (Reidys and Stadler, 2001). Seldom do stochastic search techniques move through a solution space a single step at a time, particularly with operations such as crossover. For this reason fitness landscapes can be
thought of as operator dependent. The landscape changes with the parameters selected for the search operators. If, for a given set of parameters, it is likely that a move will take the search to a region of the solution space that has similar fitness values then the space is said to exhibit neutrality. The set of reachable neutral regions from a starting location is said to form a neutral network.

A useful tool for measuring ruggedness/smoothness is an autocorrelation on the fitness values generated by a random walk. An autocorrelation is the correlation of a set of data with itself. Clearly, if such a correlation is performed with no modification to the data, it will yield a perfect correlation coefficient ($r = 1.0$). If the copy of the data (that is, the data to which the correlation with the original data will be measured) is shifted by one step, a slightly lower value will result. If this shifting is continued the correlations will eventually “settle” at some steady-state value determined by the nature of the search space. Weinberger (1990) used this form of analysis to characterize the ruggedness of the search space. If the steady-state value is positive then the fitness landscape is deemed smooth. A negative value indicates a rugged landscape. Figure 3.11 portrays the average autocorrelation plot for 100 random walks of length 1,000 steps for \textit{E. coli}’s search space.

The autocorrelation graph settles, briefly, at a slightly negative range of correlations before becoming uncorrelated (Figure 3.10). This is an indication that the landscape is somewhat rugged. Another trait of the landscape that can be ascertained from the plot is correlation distance. This is the point at which the plot crosses the x-axis. Intuitively, it can be thought of as a distance horizon. At the x-intercept point there is no correlation between a random starting point and moves $i$ steps away ($i$ being the point at which the autocorrelation plot crosses the x-axis, or more precisely, the point at which y-axis values become non-significant). This makes prediction of fitness $i$ moves away difficult, if not impossible. Said another way, beyond the correlation length the random walk has no memory. There is disagreement as to the best way to measure correlation length (Hordijk, 1996); however, Equation 3.4 is often employed as an estimator (Weinberger, 1990) where $\rho(1)$ the autocorrelation value for lag-1 ($i = 1$) and $\tau$ is the correlation length. Correlation length ($\tau$)
is often used as a summary statistic for autocorrelations with smaller \( \tau \)'s being indicative of more rugged landscapes. Using this approach the correlation length for *E. coli*’s fitness landscape is 64.54.

\[
\tau = -\frac{1}{\log(\rho(1))}
\]  

(3.4)

---

**Figure 3.11: Autocorrelation Plot for Random Walk in Codon Usage Search Space for *E. coli***

---

### 3.2.4 What This Tells Us

The investigations of fitness landscapes for codon usage bias searches provides several interesting results. For one, it can be seen that the search space is large with little deviation in fitness except in the comparatively tiny regions where good solutions reside (Figures 3.9 & 3.10). Good solutions are comprised of genes that are in close proximity to one another (Figures 3.4(b) & 2.6), and moves that involve genes of any meaningful distance from this tightly packed reference set lead to a dramatic and rapid fall-off in scores (Figure 3.10). Thus, an effective search strategy will incorporate
tactics that look for proximal solutions (solutions comprised of genes in close proximity to one another), and once found, moves should be made that do not stray very far from this region.

### 3.2.4.1 Mutation Operator and Domain Knowledge

To demonstrate the concept of improved performance when reference sets are constrained to genes that are in close proximity to one another, the landscape terrain near a good translational efficiency solution can be examined using techniques described by Grefenstette (1995). First, a histogram of fitness values for 10,000 random di-bit mutation moves (all using the same starting point) is depicted (Figure 3.12). The genome for *Neisseria meningitidis Z2491* was chosen for this analysis due to its especially good fitness score for the translational efficiency solution [0.996 using the quality of reference set equation (3.3) used in the construction of the QRS landscape]. This solution was attained through the use of the modified SCCI algorithm that will be described in Chapter 4.

![Figure 3.12: Histogram of Rank-Fitness Values Using Random Mutation](image)

The histogram shows that a single random di-bit mutation move, on average, moves to a location with a fitness of 0.974 ($\sigma = 0.017$), a decrease of 0.022 from the fitness value of the
starting location. A single move can result in as low of a fitness value as 0.903, and only 887 moves (out of the 10,000) resulted in improved fitness scores.

Knowing that good solutions are generally comprised of genes that are in close proximity to one another allows us to inject some domain knowledge into the search process, and eliminate vast regions of the search space from consideration. In order to get a better idea of the shape of a typical reference set cloud, the Eigenvalues associated with each reference set of organisms known to be dominant for translational efficiency bias (ribosomal criteria greater than one) is gathered and analyzed. In this way it can be determined if the shape of the cloud is a hyper-sphere or some other shape that will aid in selecting an appropriate mutation operator. Figure 3.13 shows that, on average, one reference set principal component dimension captures the majority of the codon usage variance.

![Figure 3.13: Average Eigenvalue for Reference Set Eigenvectors. Reference sets were determined using SCCI algorithm. Principal components analysis was performed on each reference set’s RSCU values. The Eigenvalues of each reference set were sorted and an average for each dimension was calculated. Only the first ten dimensions (ten dimensions with largest Eigenvalues) are depicted.](image-url)
Given the shape of a typical reference set, it stands to reason that if the axis of greatest variance were identified for the reference set, then good alternative candidates for membership in the reference set would be in close proximity to the axis. A simple geometric approach can be employed to determine the physical location of this axis and the distance of all genes in the genome from the axis (Figure 3.14). The first principal component characterizes the direction of the axis. To use this vector in the calculation, the center-point of the reference set is first calculated (average RSCU vector of the reference set). Any gene for which it is desired to calculate the distance to the axis can be transformed into a new point ($p'$) such that it is the same distance from the axis identified by the first principal component (whose origin is located at the origin of the RSCU vector space) as it is from the true axis of greatest variance for the reference set (which is not a vector and, therefore, has no origin, and is positioned within the reference set cloud). To accomplish this the center-point’s RSCU vector is subtracted from the gene for which the distance is being calculated ($p' = p - c$, where $c$ is the center-point). Now, the magnitude of the projection of this point ($p'$ which represents a vector) onto the first principal component is computed by taking the dot product of the two vectors ($p' \cdot PC_1$). To generate a vector with this magnitude the ratio of this value to the magnitude of $PC_1$ is calculated and used to scale $PC_1$ (Equation 3.5). This projection vector will have a magnitude such that a line connecting $p$ to the point represented by the projection vector will form a right angle. This makes calculating this distance a simple matter of calculating the Euclidean distance between $p'$ and the point that represents the projection vector (Equation 3.6 and Figure 3.14).

\[ p_{proj} = \frac{p' \cdot PC_1}{PC_1 \cdot PC_1} \ast PC_1 \]  

(3.5)

\[ eucDist(p, axis) = eucDist(p', p_{proj}) \]  

(3.6)

To validate this procedure the axis of greatest variance for the reference set representative of translational efficiency bias for Nostoc was computed and projected into the PCA derived two-
3.2. FITNESS LANDSCAPE ANALYSIS

Figure 3.14: Two-Dimensional Example for Calculating the Distance Between a Gene and a Reference Set’s Axis of Greatest Variance.
3.2. FITNESS LANDSCAPE ANALYSIS

The axis appears to be centered in the region where the ribosomal protein coding genes reside (Figure 2.6(b)) and it points along the longest axis of this set of genes.

Figure 3.15: Axis of Greatest Variance for the Reference Set Representative of Translational Efficiency Bias for Nostoc. Genes are shaded according to adherence to the translational efficiency bias.

The inter-gene distances were already calculated for each genome during the generation of the RSQ landscapes and so the gene closest to the center-point (called the centroid henceforth) was used in the distance-from-axis calculations. To determine the effects of injecting domain knowledge into the search process, the histogram of fitness values generated by 10,000 di-bit mutations
was rerun, this time incorporating this knowledge. The mutation operator was altered to increase the chances of selecting a particular gene for inclusion to the reference set according to how close that gene is to the axis of greatest variance for reference set in question. When this strategy is employed the di-bit mutations tend to generate a distribution with higher fitness scores (RSQ scores) than when random di-bit mutations are employed (Figure 3.16).

Figure 3.16: Histogram of Rank-Fitness Values Using a Mutation Operator That Chooses Genes in Close Proximity to the Axis of Greatest Variance of the Reference Set

The new histogram shows that a single “domain-knowledge-incorporated” di-bit mutation, on average, moves to a location with a fitness of 0.985 (vs. 0.974 with a random di-bit mutation) ($\sigma = 0.011$ vs. 0.017 with a random di-bit mutation). This represents a decrease of 0.011 (vs. 0.022 with a random di-bit mutation) from the fitness value of the starting location. A single move can result in as low of a fitness value as 0.930 (vs. 0.903 with a random di-bit mutation), and 1,764 (vs. only 887 with random di-bit mutation) moves (out of the 10,000) resulted in improved fitness scores.
3.3 Genomic Adherence to Specified Bias

Once a trend in codon usage has been isolated it is useful to determine how strongly the genome adheres to that bias. This can be determined by identifying and aggregating the bias adherence of the individual genes. The SCCI/CAI score for each gene is one measure of how closely the gene follows a trend in its codon usage. A gene that displays perfect adherence achieves a SCCI/CAI value of 1.0. SCCI/CAI values depicted graphically (Figure 3.17) form a characteristic curve. The area under this curve (a summation of the discrete SCCI/CAI values for all genes) is representative of the genome’s adherence to the bias. If all genes adhere perfectly to the bias their sum will equal the number of genes in the organism’s genome ($N$). This allows for the use of $N$ as a normalizing value.

Figure 3.17: *Nostoc sp. PCC 7120* SCCI values. The $x$ axis is a listing of genes arranged by ascending SCCI score. The $y$ axis is the SCCI score of each corresponding gene. The area under this characteristic curve represents the genomic adherence to a specified bias.
3.3. GENOMIC ADHERENCE TO SPECIFIED BIAS

\[ GASB = \sum_{i=1}^{N} SCCI_i \quad (3.7) \]

\[ GASB_{max} = \sum_{i=1}^{N} SCCI_i = \sum_{i=1}^{N} 1 = N \quad (3.8) \]

\[ GASB_{norm} = \frac{GA}{GA_{max}} \quad (3.9) \]

Because genomic adherence to a specified bias (GASB) is normalized by \( N \), the adherence metric also describes the organism’s average SCCI/CAI score. This makes other related quantities, such as variance and standard deviation, available and useful. This is especially true since organismal SCCI/CAI scores generally adhere to a normal distribution (Figure 3.18). This also implies that a \( t \)-test can be employed to verify whether one genomic adherence score is significantly greater than another (say, between a translational efficiency bias and a GC bias).

![Figure 3.18: Nostoc sp. PCC 7120 Distribution of SCCI values. Gene SCCI/CAI scores generally adhere to a normal distribution making such measures as standard deviation and \( t \)-tests meaningful.](image)
3.3. GENOMIC ADHERENCE TO SPECIFIED BIAS

3.3.1 Polar Bias View

In Chapter 4 techniques for isolating translational efficiency bias in the presence of confounding GC(AT)-content bias will be examined. In those situations there will be at least two biases present in the codon usage data [GC(AT)-content and translational efficiency]. A technique useful for visualizing this phenomenon is the polar bias view (Raiford III et al., 2006a). It is a technique used to visualize genomic adherence to two different biases. Given a genome’s SCCI/CAI scores for two different biases, visualizing the genome’s adherence to the two biases is a matter of trigonometric calculation of the distance of each gene from the associated biases. It can be useful for gaining insights into how the biases interrelate. The distance metric for gene \( g \) from bias \( b \) is simply 

\[
1 - SCCI_b(g)
\]

To understand why this is so it is useful to examine the range of attainable SCCI scores. The maximum SCCI score is 1 so \( 1 - SCCI(g) \) will be high for genes that are far from the pole (low SCCI scores) and low for genes that are close (high SCCI scores). Figure 3.20 is an example of this data-view given a genome with two biases in distant regions of the codon usage space.

The procedure for generating the polar bias view is to first determine the location of the poles (the two end points of \( b \) in Figure 3.19). The magnitude of \( b \) is determined by finding the gene with the smallest distance from both poles; that is, the gene whose summed distances to the poles \((1 - SCCI(g)_1) + (1 - SCCI(g)_2)\) is smaller than that for any other gene. This can be accomplished by storing both SCCI values in a listing of genes, calculating the result of the equation for each gene, and then sorting the gene list by that value. Trigonometric techniques are employed to evaluate \( x_1 \) and \( y \) values (Equations 3.10 and 3.11). The law of cosines is used to determine \( \theta_1 \) and \( \theta_2 \).

\[
x_1 = \left(1 - CAI(g)_1\right) \cos(\theta_1) \tag{3.10}
\]

\[
y = \left(1 - CAI(g)_1\right) \sin(\theta_1) \tag{3.11}
\]
3.4. Conclusion

May 27, 2008

3.4. CONCLUSION

Figure 3.19: Geometric Representation of Codon Usage Polar View. Bias poles located at opposite ends of base b. Gene located at apex of triangle (top). Gene location relative to two biases defined by \([x_1, y]\) coordinates. To build the polar bias view for an organism, \(x\) and \(y\) are calculated for each gene.

An example of a polar bias view for an organism whose biases are very close can be seen in Figure 3.21. In these depictions the degree to which a different ordering of genes occurs, when sorted by SCCI/CAI, is indicated by the horizontal spread of the gene cloud. The wider the spread the more dissimilar the ordering.

3.4 Conclusion

The visualization techniques described here, as well as the conclusions drawn from the analysis of the fitness landscapes, will be utilized throughout the rest of this dissertation. They will be employed to describe the nature of the search spaces and the repercussions of adjustments made in search strategies, and they will be drawn upon to explain the behavior of existing and developed algorithms, as well as to justify design decisions in the proposed methodologies.

A key distinction to be drawn between the random-walk-generated statistics and the fitness landscape visualization is that the random walk, by its very nature, is random, and therefore not proximal. The solutions examined are of reference sets comprised of genes that are randomly
Figure 3.20: Example of Polar Bias View of Translational and GC bias for *Nostoc sp. PCC 7120*. Each point represents a gene and the distance from that point to a bias pole is $1 - SCCI(g)$ for that gene as defined by the reference set associated with that bias. Even though AT-content confounds the SCCI algorithm, once isolated, translational efficiency bias exhibits stronger genomic adherence (i.e. on average the the genes are closer to the translational efficiency bias pole than to the content pole).
Figure 3.21: *Streptomyces coelicolor* A3(2) Polar Bias View. This representation is indicative of biases that are in close proximity. Each point is a gene and the distance from the gene to either pole (bias) is $1 - SCCI(g)$. Previous polar view (Figure 3.20) was of biases in disparate regions of the codon usage space.
drawn from the entire genome. They are not necessarily in close proximity to one another. The majority of possible solutions exhibit this characteristic, and the landscape of such a space is a vast wasteland of fitness values that range between .4 and .6 with occasional spikes where high-quality solutions reside. The solutions depicted in the fitness landscapes, on the other hand, are all proximal in nature. When only proximal solutions are examined then the search space appears relatively smooth with ridges that are more easily locatable and that dominate the landscape.
Deterministic Methods for Removing the Confounding Effects of GC(AT)-Content from the Search for Translational Efficiency Bias

Automated methods of isolating codon usage bias tend to find a genome’s dominant bias. When the intent is the discovery of translational efficiency bias, this can be problematic. One of the more common confounding-biases is GC(AT)-content [40 organisms studied in (Carbone et al., 2005) had a dominant bias that was something other than translational efficiency. 35 of these were dominant for GC(AT)-content]. Following is a description of ways in which to direct the search for a reference set that is indicative of translational efficiency bias away from those genes that exhibit unbalanced GC-content. These techniques have been published (or are in submission) in (Raiford III et al., 2006b, 2007a).

4.1 Distinct Biases – Local Search Methods

When an RSQ landscape exhibits distinct elevated ridges, this is indicative of independent codon usage trends that are comparatively easy to disambiguate. By limiting the search to a specific
4.1. DISTINCT BIASES – LOCAL SEARCH METHODS

region of the RSCU space (that is, a limited set of similarly-biased genes) the bias associated with translational efficiency (and thus, highly-expressed genes) can be isolated. Carbone et al. employed a random search methodology seeded with genes in a localized area of interest (Carbone et al., 2003) (though not specifically for the purposes of isolating translational efficiency bias).

As an aside, the two-distinct-ridge condition, where one is that of GC-content, and the other is related to translational efficiency, often can be recognized by observing the content and ribosomal criteria of an organism. If an organism is characterized by GC (or AT) bias (content criterion > 0.7 or < −0.7) and negative values for ribosomal criteria, this is an indication of possible distinct ridges. The state of the various criteria for an organism is known as a signature (Carbone et al., 2005). Negative values for ribosomal criteria are indicative of a translational efficiency bias ridge in a region distant from the trend identified by CAI in the codon usage space.

4.1.1 Localized Descending Plane

The localized descending plane method is a straightforward procedure for implementing a limited search. A reference set indicative of translational efficiency bias is obtained by lowering a plane (RSQ threshold) in the region near the translational efficiency bias ridge until the number of genes above the threshold equals one percent of the genes in the entire genome (Figure 4.1). In the indicated example the translational efficiency bias ridge was identified by its proximity to ribosomal protein coding genes. The genes above the plane constitute a reference set, and in the case of Nostoc sp. PCC 7120 this reference set represents a bias with a ribosomal criterion of 1.187 and a quality score of 0.982 (Equation 3.3).

4.1.2 Hill Climber

One can easily improve upon the fitness (quality score) obtained when using the localized descending plane method by continuing the “climb” up the fitness hill. This is accomplished by iteratively
Figure 4.1: *Nostoc sp. PCC 7120* RSQ Landscape with Plane (Threshold) at 1% Reference Set Level. The localized descending plane is a technique for identifying translational efficiency bias when there is clear separation between it and a confounding bias. The plane is descended in the region thought to contain the translational efficiency bias reference set. When the number of genes above the plane reaches 1% those genes above the plane are identified as the translational efficiency bias reference set.
4.2. MODIFIED SCCI

May 27, 2008

sorting the genes by SCCI score, setting the upper 1% of genes as the new reference set, recalculating SCCI, and repeating, until a fitness of 1.0 is achieved. When this is performed for Nostoc a fitness of 1.0 is, indeed, achieved. The reference set acquired by this method attains a ribosomal criterion of 1.312.

The disadvantage of this technique is that it works only for organisms that exhibit distinct RSQ ridges. When the translational efficiency bias ridge and the confounding GC(AT)-content ridge intersect or are subsumed, the hill-climber will behave just as in the final stages of the SCCI algorithm does – it will climb the largest nearby hill, and thus, isolate the dominant bias. An example of an organism that exhibits a subsumed translational efficiency bias ridge is S. coelicolor (Figure 4.2). Another organism’s genome that reflects a translational efficiency ridge that intersects a confounding GC(AT)-content ridge is that of Pseudomonas aeruginosa PAO1 (Figure 4.3). The translational efficiency bias ridge for P. aeruginosa runs perpendicular to the GC-content ridge discovered by the SCCI algorithm and it is more rugged than that of the dominant ridge.

4.2 Modified SCCI: Both Distinct and Merged Biases

4.2.1 High-Level Description of Approach

In cases where there are merged confounding and translational efficiency biases (where the hill climber is ineffective) it is necessary to direct the search for a reference set away from genes exhibiting unbalanced GC-content. Recall that the SCCI algorithm locates the reference set associated with the dominant bias by starting with the entire genome as the reference set, and iteratively reducing its size until the SCCI algorithm converges on the dominant bias reference set. During each iteration, the new, smaller reference set is selected by sorting the genes according to the current reference-set-defined SCCI scores. It is during this phase that the SCCI algorithm can be steered away from genes with unbalanced GC-content. It can be accomplished by forcing the high GC(AT)-content genes toward the bottom of the sorted list. This will prevent them from being
Figure 4.2: 4.2(a) *Streptomyces coelicolor A3(2)* is an example of a genome where GC-content constitutes the dominant bias. The translational efficiency bias ridge is merged or subsumed into and by the GC-content ridge. To help understand the surface of the RSQ landscape, a two-dimensional slice that is horizontal with Z1 (slice made at -3 on Z2) is provided in 4.2(b). Evidence of this can be seen in 4.2(c) and 4.2(d) where ribosomal protein coding genes (white points) are shown to be on the ascending slope of the dominant ridge, below the identified reference set. The reference set is as defined by the SCCI algorithm (GC-bias).
Figure 4.3: *Pseudomonas aeruginosa* PAO1 RSQ Landscape. The translational efficiency bias ridge runs perpendicular to the GC-content ridge discovered by the SCCI algorithm. The translational efficiency bias region is also more rugged than that of the dominant ridge.
4.2. MODIFIED SCCI

May 27, 2008

included in the next iteration’s reference set, thereby steering the search away from these regions of the codon usage search space. This technique for disambiguating the search for translational efficiency bias is described in (Raiford III et al., 2006b, 2007a).

4.2.2 Description of Modified SCCI Algorithm

By making a minor modification to the SCCI algorithm one can be assured that the bias identified is more likely to be that of translational efficiency. The modification of the SCCI algorithm presented here came about as a direct result of the exploratory data analysis enabled by the bias visualization techniques described previously.

The SCCI algorithm is designed to find the dominant bias. If the dominant bias is GC-content then the algorithm can be modified to give lower SCCI scores to genes whose GC$_3$-content deviates from balanced usage. To avoid confusion, these scores will be described as modified SCCI, or mSCCI, scores. The modification allows the discovery of the presence of translational efficiency secondary bias (and does not inhibit the search when translational efficiency is the primary bias). Recall that the CAI/SCCI score for a gene is calculated as the geometric mean of the weights associated with each codon used in that gene (Equation 2.9 & 2.21). The modified SCCI algorithm multiplies each codon-associated weight by a factor, $\beta$ (Equation 4.2), that is inversely proportional to the gene’s deviation from balanced GC$_3$-use. The result is a reduction in the mSCCI score for genes that do not exhibit balanced GC$_3$-content.

The degree to which high/low GC-content genes should be penalized depends upon the interplay of competing biases in codon usage for an organism. The scaling constant, $\alpha$ (Equation 4.1), is introduced to regulate the amount of penalty imposed by the $\beta$ factor, and thus on the mSCCI scores of high GC(AT)-content genes. The scaling constant is organism specific. In biological terms $\alpha$ can be thought of as representing the degree to which the biases (translational efficiency bias and the confounding bias) work in concert ($\alpha = 0$), or are at odds with one another ($\alpha = 2$). Biases in close proximity work in tandem on a gene’s codon usage. The ranking of genes when
4.2. **MODIFIED SCCI**

May 27, 2008

sorted according to their adherence to the two biases will be similar. These organisms will tend to require smaller values for $\alpha$. Biases that are far apart in the codon usage space are in opposition to each other. The gene orderings (when sorted by mSCCI and SCCI) will be opposite (or nearly so) for the two biases. These organisms will tend to require larger values for $\alpha$.

$$\beta = 1 - \alpha |GC_3(g) - .5|$$  \hspace{1cm} (4.1)

$$w'_i = \beta \times w_i$$ \hspace{1cm} (4.2)

The weight modification factor ($\beta$) has a range from 1 downward to a lower limit set by the scaling constant, $\alpha$. The scaling constant can range from 0 (no adjustment or unmodified) to 2. In the case of $\alpha = 2$, $\beta$ goes to zero when a gene with 100% GC(AT)$_3$-bias is encountered. In these situations the weights are set to 0.01, which is the same value used in the SCCI algorithm when a particular codon exhibits no codon usage in a reference set. A different $\beta$ value is generated for each gene while $\alpha$ remains constant for the entire genome.

### 4.2.2.1 Using a High Expression Database to Determine $\alpha$: HEDB Criterion

The organism’s $\alpha$ is determined iteratively by running the modified SCCI algorithm with various $\alpha$’s and employing a golden section search algorithm (Press et al., 1999) to determine the $\alpha$ associated with having the highly expressed genes fall as high as possible in the overall gene ranking (by SCCI). Ribosomal protein coding genes would work for this purpose (ribosomal criterion). This is problematic because a design requirement of the modified SCCI algorithm is that it be able to isolate translational efficiency bias using sequence information only. Since knowing which genes are ribosomal protein coding genes violates the spirit of this design, I employ a database of proteins associated with genes known to be highly expressed (high expression database or HEDB) to help identify a set of genes that are likely to be highly expressed in the target genome. The
proteins chosen for the database are ribosomal proteins, elongation factors, and RNA polymerase subunits (excluding the sigma subunit). These were chosen because they are known generally to be highly expressed, and they tend to exhibit high overall average CAI/SCCI scores in well characterized genomes (extensively studied genomes whose dominant bias is known to be translational efficiency) (data not shown).

For each gene in the target genome, an unfiltered blast search is performed (Altschul et al., 1990) using the corresponding protein as a query against the HEDB. The query protein is considered to be a homolog of a protein in the database if they exhibit $\geq 60\%$ identity. The database contains proteins from 66 organisms, none of which are used as target genomes in this study. In this way independence between those genomes being analyzed and the HEDB is maintained. The organisms used to build the HEDB are drawn from 25 different bacterial taxonomical subclasses, or groups, in order to achieve a class-wide representative sampling.

Any genes considered homologous to a database-identified highly expressed gene are placed in a list of HEDB genes for the target genome. Similar to Carbone et al.’s ribosomal criterion, a $\bar{z}$ statistic is calculated on the CAI/SCCI scores for these genes and is referred to as the HEDB criterion. Due to the strong correlation that exists between HEDB and ribosomal criterion (Figure 4.7), the same threshold is employed ($\bar{z} > 1$, or the average CAI/SCCI score for HEDB genes being one standard deviation above the genomes average).

To identify the appropriate $\alpha$ (Equation 4.1) for an organism, the modified SCCI algorithm is run iteratively with various $\alpha$’s and employing a golden section search algorithm (Press et al., 1999) to determine the $\alpha$ resulting in the highest HEDB criterion (Figure 4.4). The $\alpha$’s that generate reference sets with quality scores (Equation 3.3) of less than 0.900 are discarded regardless of achieved HEDB criterion.

The modified weights are only used during the phase of the algorithm that searches for the reference set. Using the adjusted weights to calculate the final mSCCI scores for the genome may introduce unnecessary bias in the gene ranking. After locating the reference set using modified
weights the final gene ranking is produced using unadjusted mSCCI scores in the traditional manner. In the case of *Streptomyces coelicolor A3(2)* the HEDB criterion ($\bar{z}$) goes from 0.594 (SCCI algorithm) to 1.183 (modified SCCI algorithm). The method of using HEDB genes as a reference set achieves a HEDB criterion of 1.089 while the traditional Sharp and Li technique (reference set of actual most highly expressed genes) is 0.713.

The modified SCCI algorithm also works for organisms characterized by separate GC(AT) and translational efficiency bias ridges. When the modified SCCI method is utilized for *Nostoc sp. PCC 7120* an HEDB criterion ($\bar{z}$) of 1.418 is attained. The HEDB criterion ($\bar{z}$) is -0.692 using the SCCI algorithm. Additionally, the content criterion (correlation of SCCI with GC-content) is -0.915, indicating that the bias identified by the algorithm is that of high AT-content. The content criterion drops in magnitude to 0.095 using the modified SCCI algorithm indicating that the identified bias is no longer representative of AT-content.

The implementation utilized to calculate all versions of the SCCI algorithm is written in PERL. I chose not to utilize available tools [such as CAIJava (Zinovyev and Carbone, 2002)] due to the need for control over weight values on a gene by gene basis.
As indicated before, prior to this methodology, the accepted technique for isolating translational efficiency bias was to use either an automated approach and verify that known highly expressed genes were identified as such (as highly expressed), or to simply use the known highly expressed genes to define the bias and measure the rest of the genes’ adherence to that bias. With the advent of the high-expression database (HEDB), either of these two approaches could be taken, calling into question the need for the more complex modified SCCI algorithm. The advantage to the mSCCI technique becomes evident when comparing the predictive capabilities of the methods. In order to evaluate CAI/SCCI/mSCCI calculation methods, their generated CAI/SCCI/mSCCI values are compared to experimentally derived expression quantities. Protein expression data is not widely available; however, the widespread use of oligonucleotide microarrays has made mRNA abundance data easily accessible. With the exception of *Nostoc*, all expression data were retrieved from the NCBI Gene Expression Omnibus (Barrett et al., 2007; Qiu et al., 2006; Maeurer et al., 2007; Liu et al., 2005; Huang, 2005; Den Hengst et al., 2005; Chang et al., 2005a; Biondi and Laub, 2007; Baliga et al., 2006; Allenby et al., 2005) (accession numbers GSE4896, GDS1469, GSE2983, GSE3876, GSE2667, GSE4617, GSE7070, GDS1099, and GSE2823). For *Nostoc* expression data was retrieved from Wünschiers’ Hydrogen Database (HyDaBa) which focuses on gene-expression data from the filamentous nitrogen fixing cyanobacterium *Nostoc PCC 7120* (Wünschiers and Eckes, 2005). In dual channel experiments the results from the reference channel are utilized (no treatment or pre). For *E. coli*, those trials using glucose as the carbon supply were used. When raw data was provided, background was subtracted from signal and user-determined flags and thresholds were accepted. For pre-processed data, genes listed as absent are removed from consideration.
4.4 Results

4.4.1 Modified SCCI Results

The Sharp and Li (using HEDB genes as a reference set), Carbone et al., and modified SCCI techniques were applied to many of the same organisms studied in (Carbone et al., 2005). The organisms are grouped into three categories; 13 organisms confounded by GC-content (Table 4.1), 13 organisms confounded by AT-content (Table 4.2), and another 45 organisms believed to have no confounding biases (Table 4.3). All sequence data were obtained from the National Center for Biotechnology Information (NCBI, 2005). The modified SCCI algorithm is able to isolate translational efficiency bias in six of the 13 GC-confounded organisms. Translational efficiency bias in one of the AT-confounded organisms can be isolated yielding a total of seven organisms that the modified SCCI algorithm can disambiguate.

Table 4.1: Criteria for CAI, SCCI, and modified SCCI Runs on GC-Confounded Organisms

<table>
<thead>
<tr>
<th>Organism Name</th>
<th>CAI&lt;sub&gt;HEDB&lt;/sub&gt;</th>
<th>SCCI</th>
<th>mSCCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HEDB</td>
<td>Content</td>
<td>HEDB</td>
</tr>
<tr>
<td>Aeropyrum pernix K1</td>
<td>0.865</td>
<td>0.810</td>
<td>0.730</td>
</tr>
<tr>
<td>Caulobacter crescentus CB15</td>
<td>1.381</td>
<td>0.825</td>
<td>0.915</td>
</tr>
<tr>
<td>Chlorobium tepidum TLS</td>
<td>0.478</td>
<td>0.896</td>
<td>-0.006</td>
</tr>
<tr>
<td>Halobacterium sp. NRC-1</td>
<td>0.141</td>
<td>0.937</td>
<td>0.215</td>
</tr>
<tr>
<td>Methanopyrus kandleri AV19</td>
<td>0.953</td>
<td>0.930</td>
<td>0.372</td>
</tr>
<tr>
<td>Neisseria meningitidis MC58</td>
<td>1.489</td>
<td>0.508</td>
<td>-0.185</td>
</tr>
<tr>
<td>Neisseria meningitidis Z2491</td>
<td>1.498</td>
<td>0.481</td>
<td>-0.314</td>
</tr>
<tr>
<td>Pseudomonas aeruginosa PAO1</td>
<td>0.523</td>
<td>0.771</td>
<td>-0.899</td>
</tr>
<tr>
<td>Pyrobaculum aerophilum str. IM2</td>
<td>0.111</td>
<td>0.501</td>
<td>0.070</td>
</tr>
<tr>
<td>Ralstonia solanacearum GM1000</td>
<td>0.667</td>
<td>0.851</td>
<td>-0.199</td>
</tr>
<tr>
<td>Streptomyces coelicolor A3(2)</td>
<td>1.089</td>
<td>0.857</td>
<td>0.466</td>
</tr>
<tr>
<td>Thermoplasma acidophilum</td>
<td>0.600</td>
<td>0.717</td>
<td>-0.005</td>
</tr>
<tr>
<td>Xanthomonas campestris str. ATCC 33913</td>
<td>1.015</td>
<td>0.858</td>
<td>0.461</td>
</tr>
</tbody>
</table>

HEDB Criterion: values > 1 indicate that the organism is characterized by translational efficiency bias. Content Criterion: values > 0.7 indicate that the organism is characterized by GC-content bias (<−0.7 for AT-content). CAI<sub>HEDB</sub> indicates CAI values are determined using Sharp and Li approach with HEDB genes as a reference set. SCCI indicates SCCI values are determined using Carbone et al. SCCI. mSCCI indicates SCCI values are determined using modified SCCI.
Table 4.2: Criteria for CAI, SCCI, and modified SCCI Runs on AT-Confounded Organisms

<table>
<thead>
<tr>
<th>Organism Name</th>
<th>CAI&lt;sub&gt;HEDB&lt;/sub&gt; HEDB</th>
<th>Content</th>
<th>SCCI HEDB</th>
<th>Content</th>
<th>mSCCI HEDB</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>Buchnera aphidicola str. Sg</td>
<td>-0.013</td>
<td>-0.756</td>
<td>-0.554</td>
<td>-0.762</td>
<td>-0.182</td>
<td>-0.517</td>
</tr>
<tr>
<td>Campylobacter jejuni NCTC 11168</td>
<td>0.838</td>
<td>-0.714</td>
<td>-0.094</td>
<td>-0.786</td>
<td>-0.066</td>
<td>-0.778</td>
</tr>
<tr>
<td>Fusobacterium nucleatum ATCC 25586</td>
<td>0.933</td>
<td>-0.739</td>
<td>0.500</td>
<td>-0.781</td>
<td>0.500</td>
<td>-0.781</td>
</tr>
<tr>
<td>Leptospira interrogans L1-130 chromosome I</td>
<td>0.627</td>
<td>-0.302</td>
<td>-0.941</td>
<td>-0.877</td>
<td>0.937</td>
<td>-0.073</td>
</tr>
<tr>
<td>Mycoplasma genitalium G-37</td>
<td>-0.596</td>
<td>-0.808</td>
<td>-1.056</td>
<td>-0.937</td>
<td>0.475</td>
<td>0.011</td>
</tr>
<tr>
<td>Mycoplasma pneumoniae M129</td>
<td>0.895</td>
<td>0.135</td>
<td>-0.246</td>
<td>-0.887</td>
<td>0.760</td>
<td>0.325</td>
</tr>
<tr>
<td>Mycoplasma pulmonis UAB CTIP</td>
<td>0.939</td>
<td>-0.650</td>
<td>0.573</td>
<td>-0.778</td>
<td>0.573</td>
<td>-0.778</td>
</tr>
<tr>
<td>Nostoc sp. PCC 7120</td>
<td>0.713</td>
<td>-0.232</td>
<td>-0.692</td>
<td>-0.915</td>
<td>1.418</td>
<td>0.095</td>
</tr>
<tr>
<td>Rickettsia conorii str. Malish 7</td>
<td>0.829</td>
<td>-0.874</td>
<td>0.612</td>
<td>-0.866</td>
<td>0.948</td>
<td>-0.787</td>
</tr>
<tr>
<td>Sulfobus solfatarius P2</td>
<td>0.971</td>
<td>-0.872</td>
<td>0.917</td>
<td>-0.877</td>
<td>0.972</td>
<td>-0.734</td>
</tr>
<tr>
<td>Sulfobus tokodaii str. 7</td>
<td>0.858</td>
<td>-0.914</td>
<td>0.938</td>
<td>-0.883</td>
<td>0.948</td>
<td>-0.882</td>
</tr>
<tr>
<td>Thermoanaerobacter tengcongensis MB4</td>
<td>-0.107</td>
<td>-0.417</td>
<td>-0.562</td>
<td>-0.853</td>
<td>-0.004</td>
<td>-0.001</td>
</tr>
<tr>
<td>Thermoplasma volcanium GSS1</td>
<td>1.042</td>
<td>-0.070</td>
<td>0.472</td>
<td>-0.796</td>
<td>0.472</td>
<td>-0.796</td>
</tr>
</tbody>
</table>

HEDB Criterion: values > 1 indicate that the organism is characterized by translational efficiency bias. Content Criterion: values > 0.7 indicate that the organism is characterized by GC-content bias (< −0.7 for AT-content). CAI<sub>HEDB</sub> indicates CAI values are determined using Sharp and Li approach with HEDB genes as a reference set. SCCI indicates SCCI values are determined using Carbone et al. SCCI. mSCCI indicates SCCI values are determined using modified SCCI.

### 4.4.2 HEDB Genes as Reference Set

When the performance of modified SCCI is compared to that of the Sharp and Li method (using HEDB genes as the reference set) very similar results are obtained (at least in terms of attained HEDB criteria) (Figure 4.5). To determine whether which approach yields the best results, predicted expression orderings are compared to expression data from microarray experiments.

### 4.4.3 Comparison to Microarray Expression Data

Of particular interest are organisms whose dominant bias is GC(AT)-content that mSCCI was able to demonstrate are also characterized by translational efficiency bias, since it is for these genomes that the algorithm was specifically designed. I was able to locate data for four such organisms, plus one additional organism (Halobacterium sp. NRC-I) that showed some translational efficiency bias, though not the requisite HEDB criterion (\( \bar{z} \)) of one standard deviation above the average
### 4.4. RESULTS

May 27, 2008

Table 4.3: Criteria for CAI, SCCI, and Modified SCCI Runs on Un-Confounded Organisms

<table>
<thead>
<tr>
<th>Organism Name</th>
<th>CAIHEDB</th>
<th>SCCI</th>
<th>mSCCI</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Agrobacterium tumefaciens str. C58</em></td>
<td>1.545</td>
<td>1.650</td>
<td>1.879</td>
</tr>
<tr>
<td><em>Archeoglobus fulgidus DSM 4304</em></td>
<td>1.059</td>
<td>1.266</td>
<td>1.232</td>
</tr>
<tr>
<td><em>Aquilix aerolicus VF5</em></td>
<td>1.513</td>
<td>1.526</td>
<td>1.540</td>
</tr>
<tr>
<td><em>Bacillus halodurans C-125</em></td>
<td>2.630</td>
<td>2.915</td>
<td>2.972</td>
</tr>
<tr>
<td><em>Bacillus subtilis subsp. subtilis str. 168</em></td>
<td>2.992</td>
<td>3.298</td>
<td>3.304</td>
</tr>
<tr>
<td><em>Bifidobacterium longum NCC2705</em></td>
<td>1.347</td>
<td>1.415</td>
<td>1.606</td>
</tr>
<tr>
<td><em>Brucella melitensis 16M chromosome I</em></td>
<td>1.597</td>
<td>1.592</td>
<td>1.746</td>
</tr>
<tr>
<td><em>Brucella suis 1330 chromosome I</em></td>
<td>1.691</td>
<td>1.764</td>
<td>1.843</td>
</tr>
<tr>
<td><em>Chlamydia muridarum Nigg</em></td>
<td>1.069</td>
<td>1.078</td>
<td>1.339</td>
</tr>
<tr>
<td><em>Chlamyphilia pneumoniae AR39</em></td>
<td>1.320</td>
<td>1.392</td>
<td>1.392</td>
</tr>
<tr>
<td><em>Chlamydophila pneumoniae J138</em></td>
<td>1.498</td>
<td>1.661</td>
<td>1.695</td>
</tr>
<tr>
<td><em>Clostridium acetobutylicum ATCC 824</em></td>
<td>1.453</td>
<td>1.715</td>
<td>1.699</td>
</tr>
<tr>
<td><em>Clostridium perfringens str. 13</em></td>
<td>1.356</td>
<td>1.706</td>
<td>2.050</td>
</tr>
<tr>
<td><em>Corynebacterium glutamicum ATCC 13032</em></td>
<td>2.462</td>
<td>2.646</td>
<td>2.653</td>
</tr>
<tr>
<td><em>Deinococcus radiodurans R1 chromosome 1</em></td>
<td>1.386</td>
<td>1.590</td>
<td>1.676</td>
</tr>
<tr>
<td><em>Escherichia coli K12</em></td>
<td>1.974</td>
<td>2.469</td>
<td>2.498</td>
</tr>
<tr>
<td><em>Escherichia coli O157:H7</em></td>
<td>2.046</td>
<td>2.518</td>
<td>2.597</td>
</tr>
<tr>
<td><em>Escherichia coli O157:H7 EDL933</em></td>
<td>2.004</td>
<td>2.493</td>
<td>2.586</td>
</tr>
<tr>
<td><em>Haemophilus influenzae 86-028NP</em></td>
<td>1.718</td>
<td>2.051</td>
<td>2.124</td>
</tr>
<tr>
<td><em>Lactococcus lactis II403</em></td>
<td>2.389</td>
<td>2.845</td>
<td>2.939</td>
</tr>
<tr>
<td><em>Listeria innocua Clp11262</em></td>
<td>1.824</td>
<td>1.839</td>
<td>1.935</td>
</tr>
<tr>
<td><em>Listeria monocytogenes EGD-e</em></td>
<td>1.886</td>
<td>1.929</td>
<td>1.964</td>
</tr>
<tr>
<td><em>Mesorhizobium loti MAFF303099</em></td>
<td>1.542</td>
<td>1.317</td>
<td>1.500</td>
</tr>
<tr>
<td><em>Mycobacterium leprae str. TN</em></td>
<td>1.294</td>
<td>1.279</td>
<td>1.372</td>
</tr>
<tr>
<td><em>Oceanobacillus iheyensis HTE831</em></td>
<td>1.602</td>
<td>1.462</td>
<td>1.831</td>
</tr>
<tr>
<td><em>Pasteurellula multocida str. Pm70</em></td>
<td>1.884</td>
<td>2.351</td>
<td>2.351</td>
</tr>
<tr>
<td><em>Pyrococcus abyssi GE5</em></td>
<td>1.164</td>
<td>1.446</td>
<td>1.463</td>
</tr>
<tr>
<td><em>Rickettsia prowazekii str. Madrid E</em></td>
<td>1.138</td>
<td>1.007</td>
<td>1.035</td>
</tr>
<tr>
<td><em>Salmonella typhimurium LT2</em></td>
<td>1.902</td>
<td>2.465</td>
<td>2.629</td>
</tr>
<tr>
<td><em>Shewonella oneidensis MR-1</em></td>
<td>2.586</td>
<td>3.008</td>
<td>3.014</td>
</tr>
<tr>
<td><em>Shigella flexneri 2a str. 2457T</em></td>
<td>2.066</td>
<td>2.525</td>
<td>2.586</td>
</tr>
<tr>
<td><em>Staphylococcus aureus Mu50</em></td>
<td>2.112</td>
<td>2.389</td>
<td>2.425</td>
</tr>
<tr>
<td><em>Staphylococcus aureus MW2</em></td>
<td>2.212</td>
<td>2.454</td>
<td>2.489</td>
</tr>
<tr>
<td><em>Staphylococcus aureus N315</em></td>
<td>2.167</td>
<td>2.415</td>
<td>2.467</td>
</tr>
<tr>
<td><em>Streptococcus agalactiae 2603V/R</em></td>
<td>2.390</td>
<td>2.867</td>
<td>2.915</td>
</tr>
<tr>
<td><em>Streptococcus agalactiae NEM316</em></td>
<td>2.529</td>
<td>3.006</td>
<td>3.057</td>
</tr>
<tr>
<td><em>Streptococcus mutans UA159</em></td>
<td>1.903</td>
<td>2.281</td>
<td>2.301</td>
</tr>
<tr>
<td><em>Streptococcus pneumoniae R6</em></td>
<td>2.486</td>
<td>2.834</td>
<td>2.832</td>
</tr>
<tr>
<td><em>Streptococcus pneumoniae TIGR4</em></td>
<td>2.488</td>
<td>2.873</td>
<td>2.873</td>
</tr>
<tr>
<td><em>Streptococcus pyogenes MGAS315</em></td>
<td>2.565</td>
<td>2.983</td>
<td>2.983</td>
</tr>
<tr>
<td><em>Streptococcus pyogenes MGAS8232</em></td>
<td>2.620</td>
<td>2.983</td>
<td>2.987</td>
</tr>
<tr>
<td><em>Synecocystis sp. PCC 6803</em></td>
<td>0.960</td>
<td>1.363</td>
<td>1.385</td>
</tr>
<tr>
<td><em>Vibrio cholerae N16961 chromosome I</em></td>
<td>2.441</td>
<td>3.105</td>
<td>3.109</td>
</tr>
<tr>
<td><em>Yersinia pestis CO92</em></td>
<td>2.152</td>
<td>2.445</td>
<td>2.461</td>
</tr>
<tr>
<td><em>Yersinia pestis KIM</em></td>
<td>2.216</td>
<td>2.576</td>
<td>2.579</td>
</tr>
</tbody>
</table>

HEDB Criterion: values > 1 indicate that the organism is characterized by translational efficiency bias. Content Criterion: values > 0.7 indicate that the organism is characterized by GC-content bias (< -0.7 for AT-content). CAIHEDB indicates CAI values are determined using Sharp and Li approach with HEDB genes as a reference set. SCCI indicates SCCI values are determined using Carbone et al. SCCI. mSCCI indicates SCCI values are determined using modified SCCI.
Figure 4.5: Association Between HEDB Criteria When Using Modified SCCI and HEDB Genes as Reference Set. Note that most of the genomes that can be isolated using modified SCCI are also the genomes that can be isolated when using HEDB genes as a reference set.
SCCI for the organism (Table 4.4). *Halobacterium* has an HEDB Criterion of 0.430 (Table 4.1). Data for an equal number of organisms whose dominant bias is translational efficiency are included (Table 4.5) in order to show modified SCCI’s performance on non-confounded organisms.

In the confounded-organism analysis all instances of modified SCCI exhibit Spearman rank correlation coefficients ($r_S$) between mSCCI values and expression levels that are more positive than either of the Sharp and Li methods. Using a signs test, this is enough to infer that modified SCCI generates solutions that are more correlated with microarray data than either of these techniques (statistical significance of $p = 0.031$ for each method). Of the five confounded organisms, three exhibit (*Nostoc*, *Halobacterium*, and *P. aeruginosa*) a modified SCCI $r_S$ value that is significantly more positive ($p < 0.05$) than that attained using the Sharp and Li method (using HEDB genes as a reference set). Significance was determined using a two-tailed Fisher $z$-transform (Fisher, 1915) using 1.06 in the numerator of the variance calculation due to $r_S$ being non-parametric (Fieller et al., 1957). One of the organisms (*Nostoc*) was even significantly greater than the traditional Sharp and Li method (where the reference set is the set of genes whose actual expressivity is the greatest for the organism). As previously noted, the Carbone et al. method identifies the dominant bias. The dominant bias for these organisms is known to be that of GC(AT)-content, ensuring that all the Carbone et al. correlations will be weak or negative. None of the five organisms whose dominant bias is translational efficiency (Table 4.5) have statistically different $r_S$ values between any of the four tested methods for determining CAI/SCCI scores.

### 4.4.4 Reference Set Quality

The quality score (Equation 3.3) can give insights into the self-consistency of a putative reference set. Figure 4.6 shows that the quality scores attained when using modified SCCI on confounded organisms tend to be higher than that of the HEDB reference sets when using the traditional approach.
### Table 4.4: Spearman Rank Correlation Coefficients Between Microarray Expression Data and CAI$_{S&L}$/CAI$_{HEDB}$/SCCI/mSCCI Values as Determined by Various Methods: Confounded Organisms

<table>
<thead>
<tr>
<th>Organism</th>
<th>CAI$_{S&amp;L}$</th>
<th>CAI$_{HEDB}$</th>
<th>mSCCI</th>
<th>SCCI</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Halobacterium</em> sp. NRC-1</td>
<td>0.165</td>
<td>0.086</td>
<td>0.199†</td>
<td>0.151</td>
</tr>
<tr>
<td>Nostoc sp. PCC 7120</td>
<td>0.046</td>
<td>0.150</td>
<td>0.275†‡</td>
<td>-0.269</td>
</tr>
<tr>
<td><em>Pseudomonas aeruginosa</em> PAO1</td>
<td>0.347</td>
<td>0.303</td>
<td>0.381†</td>
<td>0.204</td>
</tr>
<tr>
<td><em>Streptomyces coelicolor</em> A3(2)</td>
<td>0.131</td>
<td>0.148</td>
<td>0.151</td>
<td>0.130</td>
</tr>
<tr>
<td><em>Shewanella oneidensis</em> MR-1</td>
<td>0.286</td>
<td>0.323</td>
<td>0.296</td>
<td>0.295</td>
</tr>
</tbody>
</table>

All Spearman rank correlation coefficients are significant ($p < 0.05$). All $r_S$ values for modified SCCI are more positive than the $r_S$ values for any of the other three techniques. CAI$_{HEDB}$ indicates CAI values are determined using HEDB genes as a reference set. CAI$_{S&L}$ indicates CAI values are determined using most highly expressed genes (using microarray expression data) as a reference set. mSCCI indicates SCCI values are determined using modified SCCI. SCCI indicates SCCI values are determined using Carbone et al. SCCI. † mSCCI $r_S$ value significantly more positive than that of CAI$_{HEDB}$. ‡ mSCCI $r_S$ value significantly more positive than that of CAI$_{S&L}$. Significance between $r_S$ values determined using a two-tailed Fisher z-transform (Fisher, 1915) with 1.06 in the numerator of the variance calculation due to $r_S$ being non-parametric (Fieller et al., 1957)

### Table 4.5: Spearman Rank Correlation Coefficients Between Microarray Expression Data and CAI$_{S&L}$/CAI$_{HEDB}$/SCCI/mSCCI Values as Determined by Various Methods: Non-confounded Organisms

<table>
<thead>
<tr>
<th>Organism</th>
<th>CAI$_{S&amp;L}$</th>
<th>CAI$_{HEDB}$</th>
<th>mSCCI</th>
<th>SCCI</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Bacillus subtilis</em> subsp. subtilis str. 168</td>
<td>0.159</td>
<td>0.121</td>
<td>0.135</td>
<td>0.131</td>
</tr>
<tr>
<td><em>Caulobacter crescentus</em> CB15</td>
<td>0.280</td>
<td>0.306</td>
<td>0.311</td>
<td>0.284</td>
</tr>
<tr>
<td><em>Chlamydia pneumoniae</em> AR39</td>
<td>0.230</td>
<td>0.228</td>
<td>0.213</td>
<td>0.213</td>
</tr>
<tr>
<td><em>Escherichia coli</em> K12</td>
<td>0.501</td>
<td>0.496</td>
<td>0.506</td>
<td>0.505</td>
</tr>
<tr>
<td><em>Lactococcus lactis</em> Il1403</td>
<td>0.403</td>
<td>0.417</td>
<td>0.417</td>
<td>0.414</td>
</tr>
</tbody>
</table>

All Spearman rank correlation coefficients are significant ($p < 0.05$). CAI$_{HEDB}$ indicates CAI values are determined using HEDB genes as a reference set. CAI$_{S&L}$ indicates CAI values are determined using most highly expressed genes (using microarray expression data) as a reference set. mSCCI indicates SCCI values are determined using modified SCCI. SCCI indicates SCCI values are determined using Carbone et al. SCCI.
Figure 4.6: Distribution of Quality Scores for the Confounded Organism Reference Sets when the Traditional (Sharp and Li) Approach is Utilized (with HEDB Genes as the Reference Set) and when the Modified SCCI Technique is Employed.
4.4. RESULTS

4.4.5 HEDB Criteria vs. Ribosomal Criteria

To determine the proper HEDB criteria threshold for determining whether an organism’s genome is dominant for translational efficiency, the HEDB criteria for all organisms is compared to their ribosomal criteria (Figure 4.7). The criteria are so closely related that the same threshold is used to determine whether organism is characterized by translational efficiency bias: HEDB criterion ($\bar{z}$) > 1.0.

\[ y = 0.922x + 0.085 \]
\[ R^2 = 0.982 \]

Figure 4.7: Relationship Between HEDB Criteria and Ribosomal Criteria. The criteria are so closely related that the same threshold is used to determine whether organism is characterized by translational efficiency bias: HEDB criterion ($\bar{z}$) > 1.0.
4.4. RESULTS

4.4.6 Distance Between Biases and $\alpha$

In Section 4.2.2, $\alpha$ (Equation 4.1) was described as a measure of the degree to which the two biases work in concert, or in opposition, to one another. A simple method for verifying this concept is to examine the relationship between $\alpha$ and the Euclidean distance between the reference sets representing the two biases. The distance between reference sets is calculated between the centerpoints of the two reference-set clouds, in the 59-dimension codon usage space (RSCU). One would expect biases in close proximity to work in concert and to exhibit low values of $\alpha$. Alternatively, those biases that are far apart should result in opposing gene orderings (when the genes are ranked according to adherence to each bias) and larger $\alpha$ values. Figure 4.8 illustrates that there is a positive correlation between $\alpha$ and bias distance ($r^2 = 0.48, p < 0.05$), supporting the given biological interpretation of $\alpha$. The biases being examined are those identified by the SCCI algorithm (the dominant bias) vs. the bias found by modified SCCI (translational efficiency bias).

4.4.7 RSQ Landscape After Adjusting for GC-content Bias

Figure 4.9 depicts several side-by-side comparisons of RSQ landscapes when both the SCCI algorithm and the modified SCCI algorithm are utilized in their generation. Note that the regions that previously dominated the landscapes have comparatively reduced elevations. This allows the discovery of the now dominant translational efficiency ridges by the modified SCCI algorithm.

4.4.8 Characteristics of Organisms Where Translational Efficiency Bias Cannot be Isolated

As previously described, not all organisms are amenable to having their translational efficiency bias (if it exists) isolated. It is useful to identify any common characteristics of such organisms in order to help predict when isolation will be successful.
Figure 4.8: Relationship Between $\alpha$ (Degree to which Two Biases Work in Concert) and Distance Between Biases ($r^2 = 0.48, p < 0.05$). Organisms that have confounding biases that are near the translational efficiency bias exhibit low $\alpha$’s while those that are far apart have large $\alpha$’s. Euclidean distance between center-points of reference sets is used to represent distance between biases. The biases being examined are those identified by the SCCI algorithm (the dominant bias) vs. the bias found by modified SCCI (translational efficiency bias).
Figure 4.9: Side-by-side Comparisons of Various RSQ Landscapes for Genomes Known to be Confounded. Unadjusted on Left, Adjusted for GC-content on Right. Note that the regions that correspond to high GC(AT)-content (that previously dominated the landscape) are depressed in the adjusted landscapes and no longer confound the search for translational efficiency bias.
4.4. RESULTS

4.4.8.1 Original Ribosomal Criterion

Translational efficiency bias may be more easily isolated in confounded organisms that start out with a negative ribosomal criterion. Five of the seven that are disambiguated started with a negative ribosomal criterion. This is an indication that the two biases (GC(AT) and translation) are in separate regions of the codon usage space. This might imply that these types of configurations are easier to isolate than merged biases, or it may simply be an artifact of the relatively small number of organisms amenable to translational efficiency bias isolation. More study is required before definitive conclusions can be drawn as there are 13 organisms whose ribosomal criteria begin negative, and yet only five of them were able to have their translational efficiency bias isolated.

4.4.8.2 Self-Consistency Level of Ribosomal Protein Coding Genes

Having shown that using RPCGs as a reference set can lead to difficulty in isolating translational efficiency bias, it is useful to examine the RPCGs more closely. An analysis of these genes (Figure 4.10) shows that many of them fall in codon-usage neighborhoods that exhibit very little self-consistency (i.e. lie in low elevation regions of the RSQ landscape).

A histogram of gene bias altitudes (taken from the RSQ landscape) shows a bimodal distribution of elevations in most organisms (Figure 4.11). Using a Gaussian kernel clustering technique (Dempster et al., 1977) to determine to which mode a ribosomal protein coding gene belongs allows the characterization of the degree to which an organism’s RPCGs reside in low RSQ regions of the codon usage space. For genomes that exhibit a unimodal distribution of elevations, a simple percentage of RPCGs that fall in the first quartile (least self-consistent) of elevations is used. There is a clear relationship between the organisms that exhibit above average residence of RPCGs in low elevations and the ability to isolate (or at least recognize when it has been isolated) translational efficiency bias for those organisms (Figure 4.12).

Although it may seem self evident that genomes exhibiting low ribosomal criteria have many ribosomal protein coding genes with little or no reference set quality (RSQ), I felt it important to
Figure 4.10: Ribosomal protein coding genes in *Aeropyrum pernix K1*. Typical example of a genome for which ribosomal protein coding genes (RPCGs) reside in low elevations. RPCGs are black circles. Note that they fall in the low-RSQ (low self-consistency) regions of the RSQ landscape (yellow and green regions).
Figure 4.11: Histogram of RSQ landscape elevation values for *Aeropyrum pernix K1*. First mode of this bimodal histogram represents low-elevation regions of the RSQ landscape. If a large percentage of the RPCGs fall in this region then it will probably be difficult to isolate translational efficiency bias.
Figure 4.12: Percentage of Ribosomal Protein Coding Genes that Reside in Low Elevations of RSQ Landscape vs. Ribosomal Criteria Achieved when Ribosomal Protein Coding Genes are used as the Reference Set. Only organisms confounded by GC(AT)-content are depicted. Presence in low elevations (first mode in bimodal distributions of altitude) determined by clustering as a mixture of Gaussians. In unimodal distributions presence in low elevations determined by percent in first quartile.
verify and document this characteristic as one of those features shared by organisms for which translational efficiency bias is difficult to isolate.

4.4.8.3 GC-content and Genome Size

Two additional characteristics of genomes in which isolation of translational efficiency bias is difficult are GC-content and genome size. Generally, ribosomal criteria above 1 are attainable for genomes with high GC-content and/or large genome size (Figure 4.13). These are treated together because of the known relationship between GC-content and genome size (Bentley and Parkhill, 2004; Heddi et al., 1998). Large soil-dweller genomes tend to exhibit higher GC-content (Figure 4.14) while smaller, less complex host obligates, tend toward high AT-content (Rocha and Danchin, 2002; Glass et al., 2000; Moran, 2002). Reasons for this are thought to include the relatively higher energy cost of G and C over A and T and the relative abundance of ATP because of its metabolic uses in the resource constrained environments of obligate parasites. Conversely, when energetic concerns are not an issue there may be a tendency toward usage of G and C over A and T to free ATP for use in a possibly less efficient metabolism (Rocha and Danchin, 2002). Also, small (reduced) genomes tend to shed DNA recombination and repair genes which allows for greater accumulation of point mutations. C to T and G to A mutations tend to occur more frequently and could explain the higher AT-content in the smaller genomes (Glass et al., 2000; Moran, 2002).

4.5 Discussion

Algorithms developed to determine gene bias levels (degree to which a gene selectively uses specific codons) using sequence information only tend to find the gene’s adherence to the dominating bias. This can be problematic when the intent is to find biased codon usage indicative of translational efficiency. Multiple biases can coexist, simultaneously, in a genome (Grocock and Sharp, 2002; Carbone et al., 2005). With the use of the RSQ landscape and modified SCCI techniques I
Figure 4.13: Genomes that exhibit high GC-content or that have large genome-sizes (number of genes in genome is large) are generally more amenable to translational efficiency bias isolation than high AT-content or small genomes.
Figure 4.14: Number of Genes vs. GC-content. Relationship between GC-content and genome size. An explanation for why both high GC-content and large genomes show similar trends in terms of ease of translational efficiency bias isolation.

have shown how to observe and, in many cases, remove the effects of GC(AT)-content bias from the translational efficiency bias discovery process, using sequence information alone. While the results demonstrate a disambiguation technique for genomes confounded by GC(AT)-content usage trends, these methods should be equally applicable to any other well characterized confounding bias. It would simply be a matter of determining an appropriate method for establishing a $\beta$ (Equation 4.1) that sets the amount of penalty to impose on genes that adhere more strongly to this confounding bias. As GC(AT)-content appears to be the most common example of a confounding bias [40 organisms studied in (Carbone et al., 2005) had a dominant bias that was something other than translational efficiency. 35 of these were dominant for GC(AT)-content] it made sense to begin with the removal of this GC(AT)-content as a confounding factor.

The modified SCCI algorithm is able to correct for a dominating GC- or AT-bias in several (7 of 26) tested organisms (Tables 4.1 and 4.2). The organisms for which this is true exhibit a dominant GC- or AT-content bias and an HEDB criterion less than one when the SCCI algorithm
is employed. For these organisms the modified SCCI algorithm identifies a reference set and subsequent gene ranking (by mSCCI) that is more indicative of a translational efficiency bias than either the traditional Sharp and Li technique or the purely algorithmic Carbone et al. approach (Table 4.4). When an organism’s dominant bias is that associated with translational efficiency, the modified SCCI algorithm achieves results that are comparable with [not significantly different from \((p > 0.05)\)] that of the Sharp and Li technique (Table 4.5) making the modified SCCI algorithm a reasonable choice in all situations. The confounded organisms that the traditional Sharp and Li approach are able to isolate tend to be the same as those that the modified SCCI algorithm can isolate (five organisms isolated by both; seven total organisms isolated by modified SCCI; six total organisms isolated by Sharp and Li technique)(Figure 4.5).

The modified SCCI algorithm works by repeatedly searching for reference sets using various values of \(\alpha\) until a reference set with the best HEDB criterion is achieved. A useful exercise is to consider why this approach can outperform (on confounded organisms) the more direct Sharp and Li method of simply utilizing these putative highly expressed genes as a reference set (under the assumption that they represent the genes that are composed of, predominantly, the most translationally efficient codons). One explanation could center around just how well the reference set captures the underlying translational efficiency bias. When CAI values are calculated in the traditional way (Sharp and Li using the most highly expressed genes as a reference set) genes are found with CAI values greater than the lowest reference set CAI score that are not part of the reference set. As an example, when the most highly expressed genes for \(E. coli\) are used as a reference set there are 1,073 non-reference set genes with better CAI scores than the lowest reference set CAI score. It may be that, in these confounded organisms, the modified SCCI algorithm allows for the discovery of a “better” reference set – a reference set that is more self-consistent and that better represents the expected high placement of the HEDB genes in the overall SCCI gene ranking. The quality score (Equation 3.3) of a reference set (degree to which the reference set rises to the top of a sorted list of genes) is a direct measurement of this characteristic, and Figure 4.6 shows that the reference sets identified by modified SCCI have generally higher quality scores than the Sharp and
4.5. DISCUSSION  

May 27, 2008  

Li approach when HEDB genes are used as the reference set (in organisms with GC(AT)-content as the dominant bias).

The topic of how to improve the quality score of the translational efficiency reference set even further is taken-up in Chapter 5. The modified SCCI algorithm brings the search into the general neighborhood of translational efficiency bias, but until a quality score of 1 is achieved the perfect reference set has not been discovered. Chapter 5 describes a stochastic approach to achieving an even better result.

My novel approach to visualizing the RSQ landscape is useful in gaining insights into the bias composition of an organism's genome. While it is computationally intensive to generate the topography, it can be constructive when interpreting results. The RSQ landscape results shown here tend to demonstrate the effectiveness of the modified SCCI algorithm. When a landscape is generated using the modified SCCI algorithm the ridges associated with the GC(AT)-content bias are diminished to a state where they no longer dominate the reference set quality landscape (Figure 4.9(b)). The modified SCCI algorithm isolates translational efficiency without requiring a priori knowledge of the set of most highly expressed genes. It can be used to isolate translational efficiency bias in all situations, regardless of which bias is dominant (GC(AT)-content or translational efficiency), but is particularly appropriate when GC(AT)-content is a confounding influence on the search for translational efficiency bias (Table 4.4).

Organisms for which translational efficiency bias is difficult to isolate tend to share several important characteristics. First, large genomes tend to exhibit higher GC-content (Figure 4.14) while smaller, less complex, genomes tend toward high AT-content (Rocha and Danchin, 2002; Glass et al., 2000; Moran, 2002; Bentley and Parkhill, 2004; Heddi et al., 1998). It is more difficult to isolate translational efficiency bias in smaller, AT-rich, genomes (Figure 4.13(b)). Reasons for the relationship between genome size and GC-content are thought to include the relatively higher biosynthetic cost of the GC-nucleotides and the relatively higher availability of the A-nucleotides (ATP - Adenosine-triphosphate) due to its role in energy metabolism. In resource-constrained
environments this might lead to greater use of AT-nucleotides and fewer GC-nucleotides (Rocha and Danchin, 2002).

Another shared characteristic (of genomes for which translational efficiency bias is difficult to isolate) is the bias level of the ribosomal protein coding genes (Figure 4.12). It has been observed that fast growing organisms exhibit translational efficiency bias more than slow growers (homogeneity hypothesis) (Carbone et al., 2003; Lafay et al., 2000). Competition during periods of exponential growth is thought to contribute to selection for codon usage associated with enhanced translational efficiency, while an absence of such growth periods could explain a lack, or lessening, of such bias (Lafay et al., 2000; Bulmer, 1987; G. and G., 1997).

A propensity for the ribosomal protein coding genes to form reference sets that exhibit low reference set quality scores might, in fact, be a characteristic of slow growing organisms since many of the genomes for which this is true are slow growers, while many for which it is not are fast growers (Carbone et al., 2003; Razin, 1985; Karlin et al., 2001). Indeed, exhibiting small genome size and high AT-content may also be characteristic of slow growth organisms since resource constrained environments could motivate a slow growth strategy.
5.1 Introduction

Chapter 4 left off with the observation that the modified SCCI algorithm directs the search to the general vicinity of translational efficiency bias, but that there is room for improvement since the quality scores are something less than perfect ($\leq 1$). This chapter undertakes the challenge of optimizing the solution to isolating translational efficiency bias, and it does so through the use of evolutionary computational techniques. The sheer enormity of the search space makes this a daunting task. *Nostoc sp. PCC 7120*, an organism whose genome was studied extensively in this work, has 5,366 genes. For a reference set comprising 1% of the genes in this genome, the number of possible solutions is $\binom{5366}{54}$, or $8.35 \times 10^{129}$ alternatives. This is greater than the estimated number of subatomic particles in the universe (Eddington, 1952; Kasner and Newman,
5.2. FINDING A BETTER REFERENCE SET

May 27, 2008

2001) [estimates range from $10^{79}$ to $10^{86}$ depending upon which assumptions are utilized and whether massless particles (such as photons, gravitons, neutrinos, etc.) are included].

The approaches taken in this research follow two, major directions of investigation. The first attempts to search for a better reference set (a reference set that more closely matches the mathematically defined characteristics of a reference set), and the second takes an entirely different approach and searches for a set of weights that better explain the placement of highly expressed genes in a listing sorted by SCCI score. More information on the first approach can be found in (Raiford III et al., 2007b).

5.2 Finding a Better Reference Set: a Hybrid Multi-Objective Approach

The two techniques that motivated this study are the CAI (Sharp and Li, 1987) and SCCI (Carbone et al., 2003) algorithms. The first requires prior knowledge of a set of highly expressed genes. The second is designed to find the dominant bias, which may not be associated with translational efficiency. The modified SCCI algorithm steers the reference set search away from genes with unbalanced GC-content, but it does so by maximizing the reference set’s HEDB criterion (in the search for the appropriate $\alpha$, Section 4.2.2.1). While the HEDB genes are determined algorithmically, requiring no prior knowledge of expression levels, this approach does, in the end, utilize a set of a set of assumed highly expressed genes, making the approach essentially a hybrid between CAI and SCCI. The hybrid approach proposed here 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. This study is limited to organisms that exhibit weak translational efficiency bias and a more dominant GC- or AT-content bias. Two organisms were chosen to
5.2. FINDING A BETTER REFERENCE SET

May 27, 2008

test the methodology presented here; *Streptomyces coelicolor A3(2)* and *Pseudomonas aeruginosa PAO1*. Both are characterized by high GC-content and weak translational efficiency biases. The conclusions drawn from this analysis lead to the second approach, where a set of weights is the target of the search. That methodology will be demonstrated on additional organisms.

5.2.1 Genetic Algorithm

A genetic algorithm is an optimization technique modeled after nature’s approach to maximizing a species’ fitness for survival and reproduction (Holland, 1975a; Goldberg, 1989). 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) and through the occurrence of genetic mutations. In order to imitate this behavior the GA must have a representation of the solution that is amenable to these operations (crossover and mutation).

5.2.1.1 Representation, Operators, & Settings

The problem being investigated here is the selection of an optimal 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. Genetic algorithms have several operators and settings, the choice of which are representation and problem dependent. Parameters and operators were chosen empirically.

Identification of the parents for the purposes of recombination is performed using tournament selection (Goldberg and Deb, 1991). Candidates are chosen at random until a tournament size of
5.2. FINDING A BETTER REFERENCE SET

three is achieved. The member with the best fitness (see section 5.2.1.3) within this group of three is the tournament winner and is selected as a parent. Once two parents are identified, uniform crossover 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 Bäck et al. (1991) ($\mu + \lambda$) where $\lambda$ offspring are generated from $\mu$ 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 De Jong (1975) by maintaining a certain number of the best members of the combined population in the next generation. This elitism parameter is set 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 (Figures 5.4(a) and 5.5(a)). The remaining 40 slots in the next generation’s population are selected using fitness proportional (roulette) selection technique (Baker, 1987) where the more
5.2. FINDING A BETTER REFERENCE SET

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 (Błaszewicz et al., 2000). 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.

5.2.1.2 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. (2003). 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/SCCI 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/SCCI score and an index is assigned to all genes. The gene with the smallest CAI value is assigned an index of 0 while the gene with greatest CAI value is 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 Equation 5.1) and this number is normalized by the maximum attainable rank-fitness value (Equation 5.2) yielding a value between 0 and 1 representing the degree to which the reference set rises to the top of a sorted (by CAI/SCCI score) list of genes (Equation 5.3). This is the same fitness (or quality of reference set) score used to construct the fitness landscape in Chapter 3. As a simple example, consider a genome of size ten with a reference set of size four (Figure 5.1). A perfect reference set would reside at indexes 6, 7, 8, and 9 in a sorted list of genes (Figure 5.1(a)). The sum of these indexes would result in a score of 30. Since this is the maximum attainable score, it would normalize to a score of 1. If, however, the reference set resided at indexes...
4, 5, 7, and 9 (a less than optimal reference set), it would have a fitness score of 25 (Figure 5.1(b)). The resultant normalized score would be 0.83 (25/30).

Figure 5.1: Visualization of reference sets. Black bars represent the index locations of reference set genes. Un-shaded slots represent genes that are not members of the reference set. Sub-figure 5.1(a) Perfect reference set residing at indexes 6, 7, 8, and 9 in a sorted list of genes yields a sum of 30 which normalizes to 1. Sub-figure 5.1(b) Non-perfect reference set at indexes 4, 5, 7, and 9 yields a fitness score of 25 which normalizes to 0.83 (25/30).

\[
f(RS) = \sum_{i=1}^{\lvert RS \rvert} IDX_i \quad (5.1)
\]

\[
f_{\text{max}}(\lvert RS \rvert) = \sum_{i=N-\lvert RS \rvert}^{N-1} i \quad (5.2)
\]

\[
f_{\text{norm}}(RS) = \frac{f(RS)}{f_{\text{max}}(\lvert RS \rvert)} \quad (5.3)
\]

Early experiments attempted to maximize this measure by itself and an interesting phenomenon was observed: solutions were found that were of high fitness, but that were diffuse, and were not proximal (genes in the proposed reference set were not in close proximity to one another in a two-dimensional PCA projection of the codon usage space). To combat this, a second fitness measure was introduced. 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/SCCI scores a set of known highly expressed genes falls. Ribosomal protein coding genes are used for this purpose as they are known, generally, to be highly expressed. The ribosomal criterion developed by Carbone et al. (2005) is used to capture this property.

### 5.2.1.3 Pareto Ranking

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 ($\vec{u}$) is said to dominate ($\vec{u} \succeq \vec{v}$) another ($\vec{v}$) 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:

$$\forall i \in \{1, \ldots, k\}, u_i \geq v_i \wedge \exists i \in \{1, \ldots, k\} : u_i > v_i$$

(5.4)

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. A niching technique is employed to determine which members on the front to keep.

### 5.2.1.4 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%) is chosen to ensure that there are 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 within 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 could investigate their utility.

5.2.2 Microarray Data

Recall that the overall objective of the GA is to identify a reference set that captures a bias in codon usage associated with translational efficiency. If found, such a bias can be expected to correlated well with the expressivity of the genes in the organism’s genome (Section 1.3.2). In order to measure the effectiveness of the GA in finding a high-quality reference set, an independent estimator of the expressivity of the genes is needed. With this estimate in hand, the GA can be compared with other methods in terms of its ability to identify a bias that correlates well with gene expressivity. Snapshots of mRNA expression levels, such as those obtained using microarray techniques, are one such estimator for the expressivity of each gene.

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.nih.gov/). The specific study from which the expression data for *P. aeruginosa* are drawn is (Chang et al., 2005b). That analysis investigated the effect on gene expression in response to the introduction of hydrogen peroxide and utilized Affymetrix *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 *S. coelicolor* is from the genomic expression omnibus. The data were extracted from the work of Huang et al. (2001) 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.

### 5.2.3 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).

### 5.2.4 Implementation and Hardware

#### 5.2.4.1 Object Oriented

The description of a genetic algorithm in terms of a population of members that exhibit varying degrees of fitness lends itself to programmatic abstraction. To this end, the GA described here is implemented in C++ in an object oriented fashion (Figure 5.2, also see Appendix C for a UML diagram). The main program instantiates a population object that is comprised of member objects each of which has access to a static vector of gene objects. Gene objects have sequence and name attributes while members have their own mutation and crossover operators (methods).

#### 5.2.4.2 Distributed/Parallelized Processing

Genetic algorithms lend themselves to a high degree of parallelism (Cantú-Paz and Goldberg, 2000). A simple technique is to manage the population on a central node (parent selection,
5.2. FINDING A BETTER REFERENCE SET

main()
• Instantiate population
• Generational loop

Population
• Vector of members

Member
• Static gene vector
• Mutation and crossover operators

Member
• Static gene vector
• Mutation and crossover operators

...

Member
• Static gene vector
• Mutation and crossover operators

Figure 5.2: Object Oriented Model Utilized in the Genetic Algorithm
5.2. FINDING A BETTER REFERENCE SET

May 27, 2008

crossover, mutation, etc.) and to compute the population fitness values concurrently on satellite systems. To accomplish this the optimizer is separated into manager and fitness calculator components. The manager is executed on a central node, while the fitness calculator components are run on satellite systems. 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 in the form of “long float” data types (fitness values) (Figure 5.3). Using this form of parallelization the algorithm was able to realize nearly linear speedup with the number of computing nodes employed.

![Diagram of distributed nature of genetic algorithm]

Figure 5.3: Distributed Nature of Genetic Algorithm

5.2.4.3 Other Design Considerations

As in any object oriented project there often are design decisions that must be made regarding the best placement of data and functionality. A case in point is the genetic sequence data. The member objects require access to the sequence information in order to compute fitness, but the population object is responsible for loading the sequence data into memory. Additionally, it would be wasteful of memory for each member object to carry-around its own copy of the entire genome. Where
practical, such issues are handled by traditional parameter-passing and static variable techniques. For example, when the sequence data is loaded, the population object reads in the gene information, stores it in a vector, and then passes this to a member object for static storage. Other situations require a more general approach. Information such as where the algorithm is in terms of generation count is needed at various levels in the object oriented hierarchy. One could pass this information along as a parameter as each object is accessed, however this leads to cumbersome code and difficulty in modification (if the information is needed at a low-level object, each function in the chain would need to be altered in order to pass this information down). In these situations a singleton design pattern is implemented where the information is implanted as a static variable in an object. Once initialized, it can be accessed anywhere within the hierarchy, and protections can be implemented so that control is maintained over who can modify the data. These situations were encountered only in management situations making global access (over the network) to the singleton unnecessary.

5.2.5 Results

Spearman rank correlations between bias adherence scores and transcript abundance data were computed for CAI/SCCI (Sharp and Li, 1987; Carbone et al., 2003) and the multi-objective genetic algorithm (Table 5.1). The GA was able to achieve a Spearman rank correlation between CAI and transcript abundance of $0.370 (P < 0.05)$ for \textit{P. aeruginosa} and $0.155 (P < 0.05)$ for \textit{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 et al. approach ($r_S = 0.131$ and $0.204$ for \textit{S. coelicolor} and \textit{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) (Figures 5.4(a) and 5.5(a)). When these population mem-
5.2. FINDING A BETTER REFERENCE SET

<table>
<thead>
<tr>
<th></th>
<th>S. coelicolor</th>
<th>P. aeruginosa</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sharp &amp; Li</td>
<td>GA</td>
</tr>
<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>0.957</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>

Table 5.1: Correlation, Fitness, and Criterion Data

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 (Figure 5.6(b)).

bers 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 (Figure 5.6). For *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 (Figure 5.5(a)). The expectation is that a better niching strategy will eliminate such gaps, and improve the search efficacy of the GA overall.

The fitness history (Figures 5.7(a) and 5.8(a)) shows that dramatic improvements in rank-fitness occur very quickly in the search process while improvements in ribosomal criterion require somewhat longer (Figures 5.7(b) and 5.8(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 5.1 and Figures 5.4 and 5.5).

5.2.6 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
5.2. FINDING A BETTER REFERENCE SET

Figure 5.4: Population on 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 (Equation 5.4(a)) and for the entire population (Equation 5.4(b)).
5.2. FINDING A BETTER REFERENCE SET

Figure 5.5: Population on 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 (Equation 5.5(a)) and for the entire population (Equation 5.5(b)).
Figure 5.6: $r_S$ 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.
5.2. FINDING A BETTER REFERENCE SET

Figure 5.7: Fitness History

_Pseudomonas aeruginosa PAO1_ fitness history: x axis is generation, y axis is the best (and average) rank-fitness or ribosomal criterion achieved during that generation.
5.2. FINDING A BETTER REFERENCE SET

Figure 5.8: Fitness History

*Streptomyces coelicolor A3(2)* fitness history: x axis is generation, y axis is the best (and average) rank-fitness or ribosomal criterion achieved during that generation.
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 (\textit{S. coelicolor} and \textit{P. aeruginosa}). They were chosen because of their high GC-content and weak translational efficiency biases.

The hybrid algorithm is able to achieve the desired outcome of generating CAI/SCCI scores that better correlate with experimentally derived expression data (Table 5.1). These improvements are modest but it is expected 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, \textit{E. coli} is characterized by a dominant translational efficiency bias (Carbone et al., 2005). When its rank-fitness is calculated on the Sharp and Li determined reference set, a rank-fitness score of 0.959 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 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 (Figure 5.6). 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 I set out to discover (and have succeeded in doing so) a reference set that is a “better” predictor of translational efficiency bias than a reference set already 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. The second half of this chapter will focus on a more direct method of searching for these optimal weights (vs. searching for a set of genes that indirectly produce the improved weights).

Earlier it was 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 (Equation 3.3).

5.3 Finding a Set of Weights that Better Explains the Relative Synonymous Codon Usage (RSCU) Composition of the Reference Set

In the previous section [and in (Raiford III et al., 2007b)] a multi-objective genetic algorithm was utilized to more accurately identify a gene’s adherence to translational efficiency codon usage bias within a genome with high GC(AT)-content. The results showed that, of the two objectives, optimizing the ribosomal criterion yielded the best results (the other objective was the optimization of the self-consistency characteristic, or rank-fitness). Another product of the previously described GA experiments was the suggestion that one could find a reference set that better captures trans-
lational efficiency bias than the true reference set (the experimentally determined set of genes that are the most highly expressed). Yet, such an improved reference set has no meaning other than its ability to provide a set of weights that more accurately captures translational efficiency bias. It follows that a direct computation of the weights from the codon usage in the reference set (counting the occurrences of each codon in a reference set and normalizing it by the count for the maximal sibling) is, perhaps, not the best approach. Thus, the GA described next no longer searches for a better reference set. Instead, it searches for a set of weights that better explains the codon usage bias found in the highly expressed genes. It does so by finding weights that yield high CAI/SCCI scores in the genes known to be highly expressed relative to the rest of the genes in the target genome. These genes can be RPCGs, HEDB genes, or genes identified experimentally using expression data. This material has been published (or submitted for publication) in (Raiford III et al., 2008a) and in (Raiford III et al., 2008b))

5.3.1 Genetic Algorithm

The same sequence and similar microarray data are utilized for this implementation of the GA (weight optimization) as in the previous version (reference set optimization). Because this GA has proven itself adept at disambiguating genomes that all tested deterministic methods cannot (CAI, SCCI, mSCCI, Table 5.1), I have removed Caulobacter crescentus CB15 from the confounded list (it did not confound CAI$_{HEDB}$ or mSCCI) and replaced it with Campylobacter jejuni NCTC 11168. It is a confounded genome for all of the listed deterministic methods (Table 5.5). Also, the same object oriented software design and distributed architecture are employed. There are, however, minor differences in the representation and operators. This is due to the need for maximizing highly-expressed gene placement in the sorted gene hierarchy through changes in the weight vector vs. searching for a set of genes to achieve such weights.
5.3. FINDING A BETTER SET OF WEIGHTS

5.3.1.1 Representation

Since the search is now for a set of weights, a bit-string is no longer an obvious choice for representation. Instead, each member object has a 64-dimensional vector data structure that contains the member’s real valued weights. It is these weights that will be optimized (i.e. the GA will search for weights that explain the high CAI/SCCI scores of high expression genes and lower scores for the rest of the genes in the genome).

5.3.1.2 Mutation Operator

Mutation is achieved by slightly perturbing each weight within the weight vector. The new weight is a randomly generated real number normally distributed about the current weight. Equation 5.6 demonstrates the generation of a mutated weight $w'_i$ from the original weight $w_i$. $N$ is a function that generates a random number that is normally distributed about a mean ($w$) with a given standard deviation ($\sigma$). $i$ is the index that identifies to which of the dimensions the weight is associated. The mutation is self-limiting to a value that is no greater than one. For the GA described here, the standard deviation of the random-normal probability distribution function is greater in the early generations and less during the late generations. This is related to the process of simulated annealing (Kirkpatrick et al., 1983; Cerny, 1985), and it allows for greater “jumps” early in the search process and smaller steps during the latter stages. For the purposes of this GA, the standard deviation ($\sigma$) for a given generation ($gen_{current}$) is determined as in Equation 5.7 where $gen_{final}$ is the number of generations that the GA will be allowed to run, $\sigma_0$ is the starting standard deviation (the standard deviation at generation zero), and $b$ is a parameter that changes the shape of the curve described by the function (Figure 5.9). The GA described here uses a $b$ parameter of one and an initial standard deviation of 0.3. The $b$ value of one is chosen to generate a simple linear decrease in the degree of mutation with respect to generation number. A beginning standard deviation of 0.3 is chosen to distribute the initial mutation-introduced variance normally with room for mutations of up to three standard deviations. The weight-scale is from zero to one; this allows three standard
5.3. FINDING A BETTER SET OF WEIGHTS

deviations of size 0.3 to fit within this range.

\[ w'_{\text{proposed}} = N(w, \sigma) \]  \hspace{1cm} (5.5)

\[ w' = \begin{cases} 
1 & \text{if } w'_{\text{proposed}} > 1 \\
 w'_{\text{proposed}} & \text{otherwise}
\end{cases} \]  \hspace{1cm} (5.6)

\[ \sigma = \sigma_0 \times \left(1 - \frac{\text{gen}_{\text{current}}}{\text{gen}_{\text{final}}} \right)^b \]  \hspace{1cm} (5.7)

5.3.1.3 Crossover Operator

Crossover between two population members whose chromosomes are represented as bit-strings is achieved by choosing the contents from one of the parents at each gene (bit) location, and inserting it into the appropriate location of the offspring’s bit-string. While this is a possible approach in real-valued vectors, there are other, often more effective, techniques in real-vector spaces (Michalewicz, 1998). Simply populating the offspring with entries from each parent can cause wild, unpredictable jumps in the search space. Instead, if a point is chosen that is somewhere between the two chromosomal-vectors, such as a mid-point or a weighted mid-point, then crossover can become a more meaningful search operation in many real-valued vector spaces.

The GA described here employs geometrical crossover as described in (Michalewicz et al., 1996) where the entry in the offspring’s real-valued vector is the geometric mean of the two parents. This method is chosen because the geometric mean favors the smaller weight (the conservative choice in terms of choosing most preferred codons). This is due, in part, to the “fundamental inequality” (the geometric mean is never greater than the arithmetic mean). Additionally, this is similar to the way in which CAI is determined: it is the geometric mean of the weights associated with the codons in each gene.
5.3. FINDING A BETTER SET OF WEIGHTS

Figure 5.9: Simulated Annealing Trajectories with Various Values for the $b$ Parameter.
5.3.2 Repair Operator

In order to ensure that the weights derived using the selected mutation and crossover operators are meaningful and comparable to those derived using traditional methods, a repair operator is employed. The maximal sibling must have a weight of one and all other siblings must have weights less than or equal to this. A geometric mean will never increase a weight above one, and the mutation operator is self limiting to no greater than one (Equation 5.6). While the crossover and mutation operators will never generate weights greater than one, they may cause a maximal sibling to have a value less than one. Therefore, a repair operator was developed that sets the maximum weight in each codon family to one, and it increases the weight for each family member an amount proportional to the increase of the maximal sibling. Equation 5.8 represents the repair performed to weight \( w \) at each dimension \( i \). The original weight for the maximal sibling is \( w_{\text{max,sib}} \). The maximal sibling weight is normalized to one. All other weights will be increased (or will remain the same) when divided by this value (as all weights are less than or equal to one).

\[
\frac{w_i'}{w_{\text{max,sib}}}
\]  

(5.8)

5.3.3 Fitness Function

The goal of the GA is the discovery of a set of weights that accounts for the high placement of the HEDB (or experimentally determined highly expressed) genes in a ranked listing of the genes (by CAI/SCCI score). There are two metrics that have been introduced that measure this: the standard normal form of the average CAI score for these genes (HEDB criterion in the case of HEDB genes; see Section 4.2.2.1) and the rank-fitness measure (Equation 3.3). Exploratory analysis indicates that rank-fitness outperforms HEDB criterion (data not shown). This is thought to be because the HEDB criterion is an aggregation and is, perhaps, not as fine a measure of gene placement as rank-fitness. A fitness function that is more severe in nature is, therefore, required.
5.3.4 Initializing the Population

The population weights begin with values that are defined using the mSCCI method (Section 4.2) and then mutated following a normal distribution with a 0.01 standard deviation (weight values fall between 0 and 1). This means that the search process begins in the neighborhood where mSCCI left-off.

5.3.5 Results

Two versions of the GA were developed; one that maximizes the placement of experimentally determined highly expressed genes ($GA_{EDHE}$), and one that maximizes the placement of genes identified using the HEDB ($GA_{HEDB}$). The first requires a priori knowledge of gene expression levels. The second is fully automated, requiring sequence information only. For each of the GAs, two sets of results were generated. The first set of results include only organisms that exhibit a codon usage bias dominated by translational efficiency bias (un-confounded; ribosomal criterion $\geq 1$ when isolated using SCCI; Table 5.4) (GA results in Table 5.2). The second set of results include only organisms that have GC(AT)-content as their dominant bias (confounded; content criterion $\leq -0.7$ (dominant for AT-content) or $\geq +0.7$ (dominant for GC-content) when isolated using SCCI or CAI; Table 5.5] (GA results in Table 5.3). Using these confounded organisms ensures that the genome is not only dominated by GC(AT)-content (indicated by the SCCI algorithm), but also that cannot be disambiguated using the most easily employed method (CAI with known highly expressed genes).

Spearman rank correlations between bias adherence scores and transcript abundance data are computed for the GA generated CAI scores. The two GAs perform very similarly to one another. For confounded organisms the GAs were able to achieve Spearman rank correlations between CAI and transcript abundance that are significantly more positive than either of the Sharp & Li methods (using either experimentally determined highly expressed genes or HEDB genes) for four out of
5.3. FINDING A BETTER SET OF WEIGHTS

five organisms. The GA results are also more positive for the fifth organism, though not significantly so (Table 5.3). For un-confounded organisms the Spearman rank correlations generally exhibited no statistically significant difference. There were two exceptions; one where a Sharp & Li approach performed significantly better, and one where a GA approach performed significantly better. CAI_{HEDB} performed significantly better than either GA for \textit{L. lactis}, and GA_{EDHE} performed significantly better than CAI_{EDHE} for \textit{S. oneidensis} (Tables 5.2 and 5.4). Significance between \( r_S \) values determined using a two-tailed Fisher z-transform (Fisher, 1915) with 1.06 in the numerator of the variance calculation due to \( r_S \) being non-parametric (Fieller et al., 1957). When compared to mSCCI, the GAs performed significantly better for two confounded organisms; \textit{Halobacterium} and \textit{P. aeruginosa}. For un-confounded organisms mSCCI performed significantly better for two organisms; \textit{E. coli} and \textit{L. lactis}.

The performance of the GA on the genome for \textit{C. jejuni} is particularly noteworthy. For the other organisms the traditional method of calculating CAI, while confounded (ribosomal criteria \( \leq 1 \)), nevertheless exhibited a significantly positive correlation with experimentally determined expression data. \textit{C. jejuni} exhibits no such significant trend when traditional CAI is employed. The GA, however, is able to disambiguate the translational efficiency bias from the high AT-content bias and yield a ribosomal criteria \( > 1 \) and a significantly positive Spearman rank correlation with transcript abundance data (Table 5.3).

A typical fitness history (Figure 5.10) shows that dramatic improvements in rank-fitness occur very quickly in the search process. As the GA converges on the final solution the other population members begin approaching similar rank-fitness values to the best in the population (average, median, and worst population members begin approaching the best population member’s fitness).

A look at the \( r_S \) values (of GA_{EDHE} determined CAI values vs. experimentally determined expression levels) generated by the best population member at each generation (during a randomly chosen run for \textit{P. aeruginosa}) provides insight into GA performance during the search process. A well-designed GA should show improvement in predictive capability of a population member.
### Table 5.2: Performance Measurements for Genetic Algorithms (Un-Confounded Organisms)

<table>
<thead>
<tr>
<th>Organism</th>
<th>(r_S \mu)</th>
<th>(r_S \sigma)</th>
<th>(RF \mu)</th>
<th>(RF \sigma)</th>
<th>(S \mu)</th>
<th>(S \sigma)</th>
<th>(RC \mu)</th>
<th>(RC \sigma)</th>
<th>(r_S \mu)</th>
<th>(r_S \sigma)</th>
<th>(RF \mu)</th>
<th>(RF \sigma)</th>
<th>(S \mu)</th>
<th>(S \sigma)</th>
<th>(RC \mu)</th>
<th>(RC \sigma)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bacillus subtilis subsp. subtilis str. 168</td>
<td>0.146</td>
<td>0.044</td>
<td>0.978</td>
<td>0.001</td>
<td>14.112</td>
<td>0.692</td>
<td>4.344</td>
<td>0.185</td>
<td>0.149</td>
<td>0.004</td>
<td>0.978</td>
<td>0.001</td>
<td>13.728</td>
<td>0.685</td>
<td>4.244</td>
<td>0.163</td>
</tr>
<tr>
<td>Chlamydophila pneumoniae AR35</td>
<td>0.291</td>
<td>0.008</td>
<td>0.923</td>
<td>0.002</td>
<td>8.440</td>
<td>0.596</td>
<td>2.130</td>
<td>0.028</td>
<td>0.290</td>
<td>0.006</td>
<td>0.922</td>
<td>0.002</td>
<td>7.892</td>
<td>0.339</td>
<td>2.124</td>
<td>0.022</td>
</tr>
<tr>
<td>Escherichia coli K12</td>
<td>0.462</td>
<td>0.007</td>
<td>0.934</td>
<td>0.001</td>
<td>12.205</td>
<td>0.543</td>
<td>3.492</td>
<td>0.076</td>
<td>0.461</td>
<td>0.006</td>
<td>0.934</td>
<td>0.001</td>
<td>12.091</td>
<td>0.294</td>
<td>3.457</td>
<td>0.079</td>
</tr>
<tr>
<td>Lactococcus lactis Il1403</td>
<td>0.360</td>
<td>0.013</td>
<td>0.971</td>
<td>0.001</td>
<td>12.073</td>
<td>0.546</td>
<td>3.659</td>
<td>0.085</td>
<td>0.357</td>
<td>0.009</td>
<td>0.971</td>
<td>0.001</td>
<td>12.175</td>
<td>0.556</td>
<td>3.569</td>
<td>0.089</td>
</tr>
<tr>
<td>Shewanella oneidensis MR-1</td>
<td>0.331†</td>
<td>0.010</td>
<td>0.966</td>
<td>0.000</td>
<td>13.511</td>
<td>0.599</td>
<td>4.211</td>
<td>0.110</td>
<td>0.325</td>
<td>0.008</td>
<td>0.967</td>
<td>0.000</td>
<td>13.615</td>
<td>0.551</td>
<td>4.196</td>
<td>0.114</td>
</tr>
</tbody>
</table>

*GA\(_{EDHE}\)*: Results for the GA that maximizes the placement of experimentally determined highly expressed genes. *GA\(_{HEDB}\)*: GA that maximizes the placement of HEDB genes. \(r_S \mu\): Average Spearman rank correlation coefficient over ten runs with experimentally determined expression data. \(r_S \sigma\): Standard deviation for Spearman rank correlation coefficients. \(RF \mu\): Average rank-fitness values over ten runs (Equation 5.1). \(RF \sigma\): Standard deviation for rank-fitness values. \(S \mu\): Average strength (Equation 2.23) of bias identified by GA. \(S \sigma\): Standard deviation of strength values. \(RC \mu\): Average ribosomal criterion (Equation 2.22). \(RC \sigma\): Standard deviation for ribosomal criteria.

All Spearman rank correlation coefficients are significant \((p < 0.05)\).
### Table 5.3: Performance Measurements for Genetic Algorithms (Confounded Organisms)

<table>
<thead>
<tr>
<th>Organism</th>
<th>$r_S \mu$</th>
<th>$r_S \sigma$</th>
<th>$RF \mu$</th>
<th>$RF \sigma$</th>
<th>$S \mu$</th>
<th>$S \sigma$</th>
<th>$RC \mu$</th>
<th>$RC \sigma$</th>
<th>$r_S \mu$</th>
<th>$r_S \sigma$</th>
<th>$RF \mu$</th>
<th>$RF \sigma$</th>
<th>$S \mu$</th>
<th>$S \sigma$</th>
<th>$RC \mu$</th>
<th>$RC \sigma$</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Halobacterium sp. NRC-1</strong></td>
<td>0.388†‡*</td>
<td>0.017</td>
<td>0.861</td>
<td>0.007</td>
<td>9.492</td>
<td>0.834</td>
<td>1.802</td>
<td>0.092</td>
<td>0.393†‡*</td>
<td>0.010</td>
<td>0.868</td>
<td>0.004</td>
<td>9.305</td>
<td>0.498</td>
<td>1.845</td>
<td>0.080</td>
</tr>
<tr>
<td><strong>Nostoc sp. PCC 7120</strong></td>
<td>0.324†‡</td>
<td>0.030</td>
<td>0.968</td>
<td>0.003</td>
<td>12.690</td>
<td>0.883</td>
<td>2.774</td>
<td>0.169</td>
<td>0.324†‡</td>
<td>0.020</td>
<td>0.968</td>
<td>0.002</td>
<td>12.483</td>
<td>0.912</td>
<td>2.783</td>
<td>0.180</td>
</tr>
<tr>
<td><strong>Pseudomonas aeruginosa PAO1</strong></td>
<td>0.444†‡*</td>
<td>0.007</td>
<td>0.968</td>
<td>0.002</td>
<td>10.446</td>
<td>0.340</td>
<td>3.247</td>
<td>0.115</td>
<td>0.439†‡*</td>
<td>0.007</td>
<td>0.967</td>
<td>0.004</td>
<td>10.187</td>
<td>1.130</td>
<td>3.211</td>
<td>0.114</td>
</tr>
<tr>
<td><strong>Streptomyces coelicolor A3(2)</strong></td>
<td>0.149</td>
<td>0.006</td>
<td>0.743</td>
<td>0.003</td>
<td>8.846</td>
<td>1.086</td>
<td>3.218</td>
<td>0.203</td>
<td>0.146</td>
<td>0.004</td>
<td>0.747</td>
<td>0.002</td>
<td>8.076</td>
<td>0.437</td>
<td>2.956</td>
<td>0.306</td>
</tr>
<tr>
<td><strong>Campylobacter jejuni NCTC 11168</strong></td>
<td>0.093†‡*</td>
<td>0.017</td>
<td>0.970</td>
<td>0.003</td>
<td>10.621</td>
<td>2.023</td>
<td>2.972</td>
<td>0.206</td>
<td>0.088†‡*</td>
<td>0.014</td>
<td>0.964</td>
<td>0.008</td>
<td>9.553</td>
<td>1.687</td>
<td>2.668</td>
<td>0.333</td>
</tr>
</tbody>
</table>

**GA\text{EDHE}**: Results for the GA that maximizes the placement of experimentally determined highly expressed genes. **GA\text{HEDB}**: GA that maximizes the placement of HEDB genes. $r_S \mu$: Average Spearman rank correlation coefficient over ten runs with experimentally determined expression data. $r_S \sigma$: Standard deviation for Spearman rank correlation coefficients. $RF \mu$: Average rank-fitness values over ten runs (Equation 5.1). $RF \sigma$: Standard deviation for rank-fitness values. $S \mu$: Average strength (Equation 2.23) of bias identified by GA. $S \sigma$: Standard deviation of strength values. $RC \mu$: Average ribosomal criterion (Equation 2.22). $RC \sigma$: Standard deviation for ribosomal criteria.

All Spearman rank correlation coefficients are significant ($p < 0.05$). † $r_S$ value significantly more positive than that of CAI\text{HEDB}. ‡ $r_S$ value significantly more positive than that of CAI\text{EDHE}. * $r_S$ value significantly more positive than that of mSCCI (Table 5.5). Significance between $r_S$ values determined using a two-tailed Fisher z-transform (Fisher, 1915) with 1.06 in the numerator of the variance calculation due to $r_S$ being non-parametric (Fieller et al., 1957).
### Table 5.4: Performance Measurements for Deterministic Algorithms (Un-Confounded Organisms)

<table>
<thead>
<tr>
<th>Organism</th>
<th>CAI&lt;sub&gt;EDHE&lt;/sub&gt;</th>
<th>CAI&lt;sub&gt;HEDB&lt;/sub&gt;</th>
<th>mSCCI</th>
<th>SCI</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>r&lt;sub&gt;S&lt;/sub&gt;</td>
<td>RF</td>
<td>S</td>
<td>RC</td>
</tr>
<tr>
<td><em>Bacillus subtilis subsp. subtilis str. 168</em></td>
<td>0.159</td>
<td>0.730</td>
<td>3.065</td>
<td>1.674</td>
</tr>
<tr>
<td><em>Chlamydophila pneumoniae AR39</em></td>
<td>0.230</td>
<td>0.754</td>
<td>2.063</td>
<td>1.131</td>
</tr>
<tr>
<td><em>Escherichia coli K12</em></td>
<td>0.501</td>
<td>0.903</td>
<td>7.180</td>
<td>2.390</td>
</tr>
<tr>
<td><em>Lactococcus lactis Il1403</em></td>
<td>0.403</td>
<td>0.935</td>
<td>6.968</td>
<td>2.338</td>
</tr>
<tr>
<td><em>Shewanella oneidensis MR-1</em></td>
<td>0.286</td>
<td>0.919</td>
<td>7.543</td>
<td>2.930</td>
</tr>
</tbody>
</table>

CAI<sub>EDHE</sub>: CAI determined using the traditional Sharp & Li approach using experimentally determined highly expressed genes as a reference set. CAI<sub>HEDB</sub>: CAI determined using the traditional Sharp & Li approach using HEDB genes as a reference set. mSCCI: Modified SCCI from Section 4.2. SCCI: Carbone *et al.* method. r<sub>S</sub>: Spearman rank correlation coefficient with experimentally determined expression data. RF: Rank-fitness value (Equation 5.1). S: Strength (Equation 2.23). RC: Ribosomal criterion (Equation 2.22).

All Spearman rank correlation coefficients are significant (p < 0.05). †<sub>r</sub><sub>S</sub> value significantly more positive than that of GA<sub>EDHE</sub> (Table 5.2). ‡<sub>r</sub><sub>S</sub> value significantly more positive than that of GA<sub>HEDB</sub>. Significance between <sub>r</sub><sub>S</sub> values determined using a two-tailed Fisher z-transform (Fisher, 1915) with 1.06 in the numerator of the variance calculation due to <sub>r</sub><sub>S</sub> being non-parametric (Fieller et al., 1957)
### Table 5.5: Performance Measurements for Deterministic Algorithms (Confounded Organisms)

<table>
<thead>
<tr>
<th>Organism</th>
<th>$r_S$</th>
<th>RF</th>
<th>S</th>
<th>RC</th>
<th>$r_S$</th>
<th>RF</th>
<th>S</th>
<th>RC</th>
<th>$r_S$</th>
<th>RF</th>
<th>S</th>
<th>RC</th>
<th>$r_S$</th>
<th>RF</th>
<th>S</th>
<th>RC</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>Halobacterium sp. NRC-1</em></td>
<td>0.165</td>
<td>0.567</td>
<td>2.035</td>
<td>0.332</td>
<td>0.086</td>
<td>0.511</td>
<td>1.167</td>
<td>0.139</td>
<td>0.199†</td>
<td>0.969</td>
<td>3.433</td>
<td>0.493</td>
<td>0.151</td>
<td>1.000</td>
<td>2.99</td>
<td>0.215</td>
</tr>
<tr>
<td><em>Nostoc sp. PCC 7120</em></td>
<td>0.046</td>
<td>0.608</td>
<td>3.411</td>
<td>0.410</td>
<td>0.150</td>
<td>0.721</td>
<td>4.162</td>
<td>0.704</td>
<td>0.275‡†</td>
<td>0.996</td>
<td>8.606</td>
<td>1.467</td>
<td>-0.269</td>
<td>1.000</td>
<td>5.63</td>
<td>-0.606</td>
</tr>
<tr>
<td><em>Pseudomonas aeruginosa PAO1</em></td>
<td>0.347</td>
<td>0.730</td>
<td>3.114</td>
<td>0.762</td>
<td>0.303</td>
<td>0.658</td>
<td>2.315</td>
<td>0.283</td>
<td>0.381†</td>
<td>0.924</td>
<td>4.096</td>
<td>1.292</td>
<td>0.204</td>
<td>1.000</td>
<td>1.76</td>
<td>-0.899</td>
</tr>
<tr>
<td><em>Streptomyces coelicolor A3(2)</em></td>
<td>0.132</td>
<td>0.713</td>
<td>0.957</td>
<td>0.507</td>
<td>0.148</td>
<td>0.832</td>
<td>1.819</td>
<td>0.981</td>
<td>0.151</td>
<td>0.591</td>
<td>2.074</td>
<td>1.097</td>
<td>0.130</td>
<td>1.000</td>
<td>2.07</td>
<td>0.466</td>
</tr>
<tr>
<td><em>Campylobacter jejuni NCTC 11168</em></td>
<td>-0.023</td>
<td>0.589</td>
<td>1.456</td>
<td>0.145</td>
<td>-0.004</td>
<td>0.745</td>
<td>2.754</td>
<td>0.740</td>
<td>-0.012</td>
<td>1.000</td>
<td>2.297</td>
<td>-0.136</td>
<td>-0.014</td>
<td>1.000</td>
<td>2.21</td>
<td>-0.152</td>
</tr>
</tbody>
</table>

*CAI<sub>EDHE</sub>:* CAI determined using the traditional Sharp & Li approach using experimentally determined highly expressed genes as a reference set. *CAI<sub>HEDB</sub>:* CAI determined using the traditional Sharp & Li approach using HEDB genes as a reference set. *mSCCI*: Modified SCCI from Section 4.2. *SCCI*: Carbone et al. method. $r_S$: Spearman rank correlation coefficient with experimentally determined expression data. RF: Rank-fitness value (Equation 5.1). S: Strength (Equation 2.23). RC: Ribosomal criterion (Equation 2.22).

All Spearman rank correlation coefficients are significant ($p < 0.05$) with the exception of those for *C. jejuni*. † mSCCI $r_S$ value significantly more positive than that of *CAI<sub>HEDB</sub>* (Table 5.3). ‡ mSCCI $r_S$ value significantly more positive than that of *CAI<sub>EDHE</sub>*. Significance between $r_S$ values determined using a two-tailed Fisher z-transform (Fisher, 1915) with 1.06 in the numerator of the variance calculation due to $r_S$ being non-parametric (Fieller et al., 1957)
5.3. FINDING A BETTER SET OF WEIGHTS

Figure 5.10: Best, average, median, and worst fitness values for each generation of a typical GA_{EDHE} run drawn from one for *P. aeruginosa* (improved $r_S$ value) as its fitness improves. A typical $r_S$ history (Figure 5.11) shows that this is generally the case. There are, however, regions where seemingly better $r_S$ values are achieved (sporadically), and they appear to be associated with regions where jumps in rank-fitness occur (Figure 5.10). This leaves open the possibility of finding a better measurement for fitness. One could use $r_S$, itself, as a fitness measure, but this assumes that expression data is available, and the intent is to be able to predict relative expression levels with sequence information only. It is possible that this approach (of using $r_S$ with expression data as a measure of fitness) could be useful in a situation where the desire is to analyze codon adaptiveness for some reason other than as a predictor of expressivity.

As the GA works by maximizing rank-fitness for a given reference set, it stands to reason that the rank-fitness values will be greater for the GA than for the traditional method of determining CAI. Less obvious is the effect that this search strategy would have on ribosomal criterion and bias strength. Tables 5.2, 5.3, 5.4, and 5.5, and Figures 5.12 and 5.13 show that the GA identifies a bias that is both stronger and with higher ribosomal criterion than traditional methods. Comparisons
are made between techniques that use the same reference set genes, though other combinations yield similar results (data not shown).

To summarize the operation of the GA in the dimensions of strength, ribosomal criterion, rank fitness, and Spearman rank correlation coefficient, principal components analysis (Hotelling, 1933; Jolliffe, 1986) is performed, and then the original data is projected onto the first two principal components (Figures 5.14 and 5.15). There is clear separation between the GA and CAI data for confounded organisms.

A series of t-tests are employed to determine whether the GA criteria are statistically greater than those of traditional CAI (Table 5.6). All \( p \)-values are less than 0.05 except those associated with tests between \( r_S \) values for un-confounded organisms. The \( p \)-values for the \( r_S \) values are of particular interest. These tests indicate whether one should have statistical confidence that the GA is a better predictor of expressivity than the traditional tally-based methods. The \( p \)-values for the \( r_S \) values for confounded organisms are both significant (\( p = 0.035 \) and 0.050 for EDHE and HEDB reference sets respectively, Table 5.6). This is strong evidence that the GA is the preferred
Figure 5.12: Comparison of strength criterion (Equation 2.23) and ribosomal criterion (Equation 2.22) for confounded organism between GA that maximizes location of experimentally determined highly expressed genes and traditional CAI using those same genes as a reference set. Each point represents an organism. GA data are the means of ten runs. Error bars represent standard error of the means of the GA runs.
Figure 5.13: Comparison of strength criterion (Equation 2.23) and ribosomal criterion (Equation 2.22) for confounded organisms between GA that maximizes the location of HEDB genes and traditional CAI using those same genes as a reference set. Each point represents an organism. GA data are the means of ten runs. Error bars represent standard error of the means of the GA runs.
Figure 5.14: Principal components analysis performed on strength, ribosomal criterion, rank fitness, and Spearman rank correlation coefficient dimensions for confounded organisms. Reference sets were based upon experimentally determined high expression genes (EDHE). There is clear separation between the GA and CAI groups.
Figure 5.15: Principal components analysis performed on strength, ribosomal criterion, rank fitness, and Spearman rank correlation coefficient dimensions for confounded organisms. Reference sets were based upon genes identified through the use of the high expression database (HEDB). There is clear separation between the GA and CAI groups.
5.3. FINDING A BETTER SET OF WEIGHTS

EDHE: experimentally determined expression data used to identify reference set. HEDB: high expression database used to identify reference set. $r_S$: Spearman rank correlation coefficient between CAI data and expression data. $S$: strength criterion (Equation 2.23). RC: ribosomal criterion (Equation 2.22).

### 5.3.6 Discussion

Early methods for determining adherence to translational efficiency bias did so first by examining highly expressed genes to identify which codons were preferred, and then calculating the frequency of optimal codon usage (FOP) for the genes in the genome [Section 2.1.1, (Ikemura, 1981b)]. It was shown that FOP was highly correlated with protein expression levels in prokaryotes. This is a binary interpretation of codon preference. Either the codon is identified as optimal, or it is not. Later, the measurement of adherence to translational efficiency bias was refined by introducing gradation to the process. The concept of degree of preference, or the degree to which a codon has adapted, was introduced with the codon adaptation index (CAI) measurement [Section 2.1.5, (Sharp and Li, 1987)]. These types of methods (like FOP and CAI) have remained in use, largely unmodified, since their conception. They rely upon the uniform conformance of highly expressed genes to the underlying translational efficiency bias in order to identify the degree of preference for each codon. It turns out, though, that the adherence of highly expressed genes to translational efficiency bias is by no means uniform. As an example, only 58% of the the highly expressed genes for *Saccharomyces cerevisiae* prefer the codon that is preferred “overall.” Additionally, the most highly expressed genes (those among the top 1% of genes in terms of expressivity) often are ranked in the lower quartile of translational efficiency bias as determined by FOP or CAI.
5.3. FINDING A BETTER SET OF WEIGHTS  

There are many possible reasons for the non-uniform usage of preferred codons. It has been shown that deviation from use of optimal codons can regulate ribosomal translation rate (Lawrence and Hartl, 1991). In particular, regions of a gene’s sequence occurring between domains tend to deviate from optimal codon usage in a way that causes a ribosomal-mediated translational pause (Thanaraj and Argos, 1996). This allows each domain of the protein time to fold before translation continues. The requirements of each protein’s adherence are unique, and when aggregated may not precisely capture the underlying tRNA abundance driving the bias. This could be particularly true in genomes with other biases (such as GC-content) that drive codon usage. It follows that there may be a way to improve upon the methods for isolating the underlying translational efficiency bias.

To identify the underlying codon preferences in the face of this non-uniform codon preference, I have developed a genetic algorithm that performs a search for the codon preference values (weights). The GA generates solutions (sets of weights) that produce a ranking of genes (based upon adherence to the identified codon usage bias) that better explains the high placement of reference set genes in an experimentally determined listing of genes ranked by expression level (or, in the case of HEDB genes, expected placement).

The GA performs significantly better than traditional methods of determining CAI (Table 5.6) on confounded organisms ($p$-values of two-tailed, paired t-tests between GA $r_S$ values and traditional CAI $r_S$ values are 0.035 and 0.050 for confounded organisms when EDHE and HEDB genes are used as reference sets, respectively). For confounded organisms the GA performs similarly to traditional CAI method ($p$-value of two-tailed, paired t-test between GA $r_S$ values and traditional CAI $r_S$ values is 0.924 and 0.981 for EHDE and HEDB reference sets, respectively).

In addition to good performance in predicting expressivity, the GA provides solutions that exhibit traits generally known to be shared by genomes dominant for translational efficiency bias (Carbone et al., 2005). These traits include the strength of bias exhibited by the reference set, and the degree to which ribosomal protein coding genes adhere to the translational efficiency bias.
Both tend to be high in organisms dominant for translational efficiency bias and are measured by strength and ribosomal criteria, respectively [ribosomal criterion > 1 (Equation 2.22) and strength > 8 (Equation 2.23)]. The strength and ribosomal criteria values for the biases identified by the GA are significantly greater than those of the traditional methods for all organisms, confounded and un-confounded (p-values of two-tailed, paired t-tests between GA strength and ribosomal criterion values and traditional CAI strength and ribosomal criterion values are all less than 0.05, Table 5.6).

All of these factors taken together [better predictive capabilities (more positive $r_S$ values), stronger identified bias, and a gene ranking with higher ribosomal criteria] lend support for the conclusion that the GA provides a bias identification methodology that is more indicative of the underlying translational efficiency bias. This is true for confounded organisms (with performance equivalent to traditional methods for un-confounded organisms) whether experimentally determined high expression genes are used as a reference set, or genes identified through the use of the high expression database are used. While I have shown that the GA produces a gene ranking more consistent with that associated with a ranking by expression level, the underlying assumption is that the technique has identified the “true” translational efficiency bias – that is, the adherence to a bias influenced by the abundance of the associated tRNAs. Future research should work to corroborate this premise by examining the relationship between weight values and the associated tRNA abundances.

5.4 Proof: Is There Always a Perfectly Self-Consistent Reference Set?

The previous two sections found solutions that appear to be better predictors of expressivity than existing solutions. The derived solutions have good, but not perfect RSQ scores (Sections 5.2 and 5.3). This leaves open the question of whether it is worth searching for the perfect reference set (one with an RSQ score of 1). The greedy solution (the SCCI algorithm) always seems to find a
solution, and by definition, that solution will have a perfect RSQ score. This seems to hint at a conclusion that there is always a solution, but is it unique? And is the existence of a perfect solution guaranteed, or is there some force at work in nature that causes genomes to arrange themselves in such a way as to always have a perfect solution.

It has not yet been shown that a solution is guaranteed to exist (or, alternatively, that it might not exist). However some useful information was uncovered in the work described here that may shed some light on this rather difficult problem. First, the greedy algorithm (for determining SCCI) does not always find a perfect solution at the reference set size of 1%. The SCCI algorithm converges to a reference set of size 1% by progressively using smaller and smaller reference sets (starting at size 100% and iteratively halving the size of the reference set). Once a reference set of 1% is achieved, the SCCI scores are calculated, the genome resorted, and a check is performed to determine if the reference set is still at the top of the sorted list of genes (indicating a perfect RSQ score). If not, the top genes are set as the reference set, and the procedure is repeated until it stabilizes. In some cases, the search oscillates from one reference set, through several others, until the search returns to the original reference set. In this case the search will loop indefinitely. When this occurs, the gene with the lowest SCCI score is removed from the reference set, and a reference set of slightly smaller than 1% is tested. This is repeated as necessary until a stable reference set is identified. This means that some genomes may not have a solution at the 1% reference set size. Since all solutions cannot possibly be checked, this is not known for sure, but the possibility exists (or at least it has not been proven otherwise).

While it has not been shown that a 1% “reference set size” solution always exists at typical genome sizes, I have been able to prove this for a very small genome size (four genes) with a relatively large reference set size (50%). The genes depicted in Table 5.7 represent four genes that have no reference sets (of size two) with a perfect RSQ score (a score of “1” is a perfect RSQ score, see Equation 3.3).

Even this simple genome required the use of a GA to generate a set of four genes for which
Table 5.7: Four Genes for Which There is No Reference Set of Size Two With a perfect RSQ Score
there was not perfect solution. The solution was represented as four 59-dimensional vectors, each representing a gene (each dimension representing the count of a particular codon). A simple mutation operator was utilized that introduced variance in the number of a randomly chosen codon (a mutation rate that affected approximately one codon per gene). Fitness was measured in terms of the number of pairwise combinations that represented reference sets that had perfect RSQ scores (the lower the fitness score the closer the solution is to proving that there is a four-gene genome for which no perfect solution exists). Early trials generally always start with every solution (every pairwise combination) having a perfect RSQ score, so an additional measure was introduced into the fitness calculation. To illustrate why this is necessary, if a reference set represents a perfect solution, the original version gave it a score of 1, and the scores of all pairwise combinations were added together. This meant that early generations all had overall fitness scores equal to the number of pairwise combinations in the genome [six different pairwise combinations exist for four genes: (1,2)(1,3)(1,4)(2,3)(2,4)(3,4)]. To show a change in fitness in these early generations, the difference between the CAI score of the second and third gene was added to the score of any reference set with a perfect RSQ score. In this way, the closer a gene was to interceding in the reference set (thus making it a non-perfect reference set) the lower the score. The depicted gene-set (Table 5.7) was acquired after 73 generations with a one-hundred member population.

This method of proof does not scale well. It relies on testing all possible solutions of a designated reference set size to show non-existence. This is not tractable for typical genomes. Remember that the Nostoc genome has $8.35 \times 10^{129}$ possible solutions (Section 2.6.6.3). As the number of genes in the reference set approaches 1 or $n$ (where $n$ is the number of genes), the number of solutions that need to be tested (to test whether there are no solutions) goes down (the number of solutions is at its max at reference size $n/2$). For instance, with a reference set of size 75% in a four gene genome there are only two that need to be tested, verses the six in the 50% solutions. However, as the size of a genome goes up, the number of possible combinations of reference sets goes up. Using this technique, in order to say definitively that no perfectly self-consistent solution exists for a given genome one would need to test all of these possible solutions.
5.5 Summary

This dissertation chapter deals with stochastic approaches to isolating translational efficiency bias in prokaryotic genomes. Specifically, genetic algorithms are employed in the search. The first approach examined was a search for a reference set that is at once more self-consistent (rank-fitness) and that also generates a set of weights that brings known highly expressed genes to the upper regions of a gene listing when sorted by adherence to bias (ribosomal criteria). This is a multi-objective genetic algorithm and it combines the features of the traditional approach to determining codon adaptation index scores (Sharp and Li, 1987) (in the form of using a set of known highly expressed genes) with the purely algorithmic method of first identifying a self-consistent reference set (Carbone et al., 2003) (in the form of maximizing rank-fitness), and then using that reference set to calculate the codon adaptation indexes. But this engenders the question of what it means to be a reference set that better captures translational efficiency bias when, in fact, one begins with an experimentally determined set of genes shown to be the most highly expressed genes. One cannot improve on such a set of genes. In the case of the multi-objective 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 results from adherence to this bias. This led to the creation of the second GA covered in this chapter – one that explicitly searches for this “better” set of weights. This version of the GA isolated translational efficiency bias measurably better than traditional approaches in, essentially, all situations (for confounded and un-confounded organisms, using experimentally determined highly expressed genes or genes located using the high expression database). In the one category where it did not perform measurably better (in un-confounded organisms where HEDB genes are used as a reference set) it performed just as well as traditional methods, and so is a logical alternative regardless of choice of reference set genes or whether the organism is confounded. This is an entirely novel approach to identifying codon preference in prokaryotic genomes, and is perhaps the most important contribution of this work.
Conclusions

6.1 Summary of Contributions

The search space for a self-consistent set of genes, even though prohibitively large, exhibits certain characteristics that do allow for a computational solution. The solutions tend to be proximal, and the proximal solutions tend to become gradually worse as one explores the codon usage space that extends away from an optimal solution. This type of search space lends itself to traditional deterministic methods such as a greedy hill-climber approach taken by the Carbone et al. (2003) SCCI algorithm.

But purely algorithmic techniques tend to be easily confounded. GC(AT)-content can quickly dominate the codon usage bias of an organism, and any algorithm searching for the prevailing bias in such a genome will find this bias at the expense of any other. I have shown that, with a slight modification to an existing technique, the search can be directed away from genes with unbalanced GC(AT)-content, and toward a region of the search space that is both self-consistent and characteristic of translational efficiency (Raiford III et al., 2006b, 2007a, and Section 4.2 of this dissertation). Solutions found in this way tend not to be perfectly self-consistent (rank-fitness scores somewhat less than 1) which indicates that the search has been brought into the neighborhood of an ideal solution. I have proven that there may not be an ideal solution at a given reference set size, but, at least for all un-confounded organisms tested, there are perfect solutions that are near the 1% size. This leads one to believe that the solutions provided by mSCCI can be
improved upon.

The first foray into searching for a better solution was through the use of a multi-objective genetic algorithm (Raiford III et al., 2007b, and Section 5.2 of this document). It showed that focusing on improving a solution’s ribosomal criterion (as opposed to its rank-fitness) yields the best solutions, but more importantly the solution (the identified reference set) is only a means to attaining a better set of weights.

This led to the insight that a search for a set of weights that more accurately reflects the placement of the highly expressed genes in a ranked listing of genes (by adaptation index) is, perhaps, a more appropriate strategy. Each highly expressed gene/protein is unique in its adherence to translational efficiency codon usage bias. A high degree of adherence in a region of the coding sequence ensures not only a higher ribosomal transversal rate (and therefore translational efficiency), but it also brings about a higher degree of accuracy in the transcription process (Precup and Parker, 1987). For this reason, highly conserved regions (regions where a more precise requirement for an exact amino acid exists) will exhibit a high degree of adherence to translational efficiency bias. In contrast, regions that represent a protein’s domain boundary will depart from optimal codon usage in order to slow-down the translational process, allowing an opportunity for the just-translated domain to fold before the next region is created (Thanaraj and Argos, 1996). These diverse adherence signatures found in highly expressed genes ensure that when their collective codon usage is calculated (by aggregating the degree to which each codon is preferred) the result will not necessarily exactly reflect the selective pressure to utilize that codon. This is particularly true in confounded genomes where non-adhering regions will tend toward the native GC(AT)-content and thus obscure further the underlying forces.

While this breakthrough is important and will lead to advances in the study of molecular evolution, there is still much to do.
6.2 Future Research

6.2.1 Make the Approach Available Online

In order to make these results more accessible to those whose research can most benefit from them, an automated tool should be designed. Such a tool would most appropriately be web-based and designed using a three-tier architecture (Figure 6.1). The translational efficiency bias for all known prokaryotic organisms could be pre-computed, and made available readily through such an interface. It would also be necessary for the utility to be capable of accepting uploaded complete or partial organismal genomic data to allow for the identification of translational efficiency bias in newly sequenced genomes. Finally, tools for visualizing RSQ landscapes and tools that would aid in understanding and characterizing an organism’s codon usage bias (e.g. what is the dominant bias? is there a discernable translational efficiency bias?) would need to be made available. The utility would need to be designed in such a way that its use would be intuitive and logically organized.

6.2.2 Extend the Study to Additional Organisms

As more expression data becomes available the performance of the GA and of mSCCI should be tested on additional organisms, particularly those that appear confounded to both SCCI and traditional CAI techniques. This additional data may add to, and make more conclusive, the evidence that the GA and mSCCI techniques indeed provide statistically more reliable predictions of expressivity rankings.

6.2.3 A Hidden Markov Model (HMM) Approach

While a useful concept in the abstract, “expressivity” is difficult to quantify – especially on a proteome-wide scale. Expression data available today (transcript abundance, protein abundance,
6.2. FUTURE RESEARCH

Figure 6.1: Three Tiered Architecture
and codon usage bias) are approximations of the underlying protein production rates. A hidden Markov model could utilize known abundance values (mRNA and protein) and estimated rates of degradation (and possibly other rates such as protein usage, mRNA degradation, ribosome availability, etc.) to determine the most probable underlying expression rates. Difficulties include gauging success and estimating certain rates under in vivo conditions.

![Diagram of various rates associated with expressivity](image)

Figure 6.2: Various Rates Associated with Expressivity

### 6.2.4 Biological Questions

The results from the various studies contained in this research have identified a number of open questions that are biological in nature. One such open question is the underlying assumptions regarding the reasons as to why the GA outperforms existing deterministic, tally-based methods for identifying translational efficiency bias.
6.2. FUTURE RESEARCH

6.2.4.1 Verify GA has Identified a Bias Associated with Translational Efficiency – That it Predicts Abundant tRNAs

While I have shown that the Direct Weight Search GA produces a gene ranking more consistent with that associated with a ranking of genes by expression level, the underlying assumption is that this method has identified the “true” translational efficiency bias – that is, the genomic adherence to a bias dictated by the abundance of the associated tRNAs. Future research should work to prove this assumption by examining tRNA concentrations and attempting to show whether a relationship exists between the concentrations and the GA-identified weights. If correct, large GA-identified weights should be associated with the most abundant tRNA species. In prokaryotic organisms this could be done (at least initially) with tRNA gene copy number as they are known to be predictors of tRNA abundance Ikemura (1981a); Percudani et al. (1997); Kanaya et al. (1999); Duret (2000).

6.2.4.2 Possible Causes of a Distinct Ridge Configuration

Another open question is the underlying cause of the distinct-ridge configuration found in some confounded organisms (e.g. Nostoc). Why do many organisms have a coincident configuration (merged ridges) while some exhibit distinctly different bias ridges (between some dominant bias ridge and one associated with translational efficiency)? Could there have been changes in environment, separated in evolutionary time, that drove first one bias, and then the other? Could such consecutive application of selective force cause such a divergence, and if so, what sort of environmental changes could these be? These are interesting and difficult to answer questions that could lead to an increased understanding of the nature of molecular evolution and natural selection.

6.2.4.3 Investigation of Whether Escherichia coli Underwent a Large-Scale HGT Event

Section 3.1.4 illustrated that the RSQ landscape for E. coli is dominated by two peaks, one associated with translational efficiency and another related to a set of genes believed to have been
horizontally transferred into the genome. Others have noted this large concentration in codon usage space of horizontally transferred genes (Carbone et al., 2003), but have avoided commenting on its significance. The genes all share a similar, highly self-consistent, bias (by definition, genes in elevated regions of the RSQ space are self-consistent). This is an indication that the genes all originated from the same organism or at least a closely related group of organisms. Such large-scale horizontal gene transfer events would be an important finding, and if not taken into account, might skew other research, such as studies of *E. coli* lineage and proteomic homology. One would need to identify the genes that comprise this cluster, and then show that they do, indeed fall in contiguous segments of the genome. Next, an analysis of codon usage bias of other organisms would need to be performed in order to identify the source genome(s) of these genes. Additionally, a homology search (BLAST) could be performed in which a search for organisms whose proteins share a high degree of similarity would need to be performed in order to confirm the findings. Finally, a phylogenetic comparison could be made to determine where in the ancestral history these HGT events took place. The genome for *Escherichia coli* has been extensive studied with respect to horizontal gene transfer Lawrence and Ochman (1998). A careful comparison with the findings of such research would be necessary to validate this proposed codon usage analysis.

An additional benefit of such analysis could be the development of a technique for estimating the time at which a gene was introduced into a genome by lateral gene transfer. The degree to which the amelioration process has proceeded (gene evolves into adherence to the native genomes codon usage bias) could be measured and compared to other methods of determining evolutionary time.

6.2.4.4 Eukaryotes

The analysis in this work is limited to prokaryotic organisms because they are the organisms in which translational efficiency bias has been found to exist (*Saccharomyces cerevisiae* has also been shown to exhibit translational efficiency bias). A single celled organism generally has a
6.2. **FUTURE RESEARCH**

May 27, 2008

single expression signature. It is only during times of stress or unusual cellular activities (such as cell replication) that gene expression requirements change. In eukaryotic organisms (and to some extent in Archaea) this is not the case. There are differing expression-level needs in different regions of the organism (e.g. the eye has different expression requirements than the liver).

An interesting area for further research is a determination of whether there might be a small set of expression signatures that could exploit differing regions of codon usage bias within an eukaryotic organism’s genome? In other words, might not different profiles in tRNA abundance cause certain genes to be more highly expressed in the presence of a particular profile, and might not selective pressure cause preferential usage of the associated codons (to the locally abundant tRNAs) in the associated genes (that are locally highly expressed)? There is support for this theory. Non-mammalian genes introduced into a mammalian genome must have their codon usage adjusted in order for them to be expressed at high levels (Levy et al., 1996; Zolotukhin et al., 1996; Wells et al., 1999; Zhou et al., 1999). This is a clear indication that codon usage bias has at least some effect on expression rates. This method is known as ”mammalianization” or ”humanization.”

### 6.2.5 Algorithmic Research

#### 6.2.5.1 Removal of Multiple Biases, Simultaneously

There are several known sources of bias in genomic data. Those mentioned in this research include strand, content (GC- or AT-), translational efficiency, and bias introduced through horizontal gene transfer. A completely automated approach to identifying translational efficiency bias would remove the effects of all these biases from the search process. The techniques employed in this work remove the effects of GC(AT)-content from the search for translational efficiency bias.

The effects of bias introduced by horizontal gene transfer could be removed by simply removing candidate genes prior to running the algorithm. This is somewhat complicated by the fact that current techniques for HGT candidacy identification require the removal of highly expressed genes.
6.2. FUTURE RESEARCH

from consideration. This circular requirement (HGT removal from translational efficiency determination and removal of high translational efficiency adherents from HGT determination) must be overcome if the effects of bias introduced through HGT is to be accomplished. The fact that HGT genes appear to have little effect on the iterative reference set search employed by SCCI is heartening. The SCCI algorithm locates the dominant bias. The reference set that is representative of the dominant bias is generally free from the effects of other biases unless they are nearby in the codon usage space, and then only marginally. Other algorithmic techniques, such as correspondence analysis (MCU) (Kanaya et al., 1999), may be more susceptible to the effects of these genes. They tend to rely on the detection of an axis of greatest variance within the codon usage data, and the trajectory of this axis is altered whenever these genes are present. Whether the ordering of genes projected onto the axis is altered is a matter for future research, but clearly this HGT candidate genes could have an effect.

The removal of strand bias might be done in a similar manner to the removal of GC(AT)-content bias. It would simply be a matter of determining an appropriate method for establishing $\beta$ (Equation 4.1) that sets the amount of penalty to impose on genes that adhere more strongly to this confounding bias. Strand bias is the result of transcription induced C→T transitions on the nontranscribed strand (Francino et al., 1996; Francino and Ochman, 1997; Beletskii and Bhagwat, 1998) and/or the replication induced increased usage of G+T on the leading strand (Lafay et al., 2000, 1999; Rocha et al., 1999). Under transcription induced bias the non-template strand genes will have higher than normal C-content in the presence of strand bias. If a measure of the degree to which a gene deviates from expected C-content can be devised, then it could be used to determine a $\beta$ appropriate for directing the search away from genes experiencing high strand bias. Under replication induced bias deviation from expected G+T usage could be used.
6.2.5.2 Better Fitness Function

During the search for weights described in Section 5.3 there were often solutions early in the search that had better correlations with expression data than the final solutions (Figure 5.11). This may be an indication that even better correlations can be discovered with a suitable fitness function. A good topic for future research is the discovery of such a function. It might require a multi-objective approach or it may involve some measure yet to be discovered. A naive approach would involve simply running the GA with many different fitness functions and observe the behavior of the correlation results. A more scientific approach would involve positing a reason for the early spike, devising a measure that would exploit the underlying cause, and test the hypothesis experimentally.

6.3 Bringing it all Together: Flow Chart

Found herein are descriptions of many techniques for bias discovery (FOP, MCU, CAI, etc.) and several automated methods that are resistant to the confounding effects of GC(AT)-content (mSCCI, multi-objective GA, direct weight discovery GA). Which method to use, and when to use it, can be confusing. It depends upon many conditions such as whether the sequence information has been annotated (gene function determined, etc.), how time dependent the need for the result is (GAs can be time consuming), and whether or not the target genome is confounded by GC(AT)-content. Figure 6.3 describes the tests that can be run, and the order in which to run them, to determine a likely gene ordering according to gene expressivity.

Probably the best approach is to first run SCCI and determine the content criterion. This will determine whether the genome is confounded by GC(AT)-content. If it is, then it will be necessary to use some alternative technique for determining translational efficiency bias. One of the easiest ways to accomplish this task is to employ the traditional Sharp and Li (1987) method. One could use either experimentally determined highly expressed genes or genes known to be,
generally, highly expressed (RPCGs or HEDB genes). The solution can be improved upon by running the mSCCI algorithm, and if the need is not highly time dependent, the mSCCI result can be improved upon by using the genetic algorithm for weight identification (using the mSCCI weights as a starting point).

If the genome exhibits distinct ridges in the RSQ landscape one can attain a perfect RSQ score by running a greedy hill-climber algorithm using the mSCCI result as a starting point. One can usually tell if this condition exists by examining the content and ribosomal criterion of the SCCI solution. If it yields a content criterion \( \geq 0.7 \) or \( \leq -0.7 \) and a ribosomal criterion \( \leq -1.0 \) then there is probably a distinct ridge configuration. The dominant bias is GC(AT)-content, and it generates a gene listing virtually opposite that of a listing based upon translational efficiency.

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 global selective pressures associated with translational efficiency and the way in which this pressure affects genomic evolution.
6.3. BRINGING IT ALL TOGETHER: FLOW CHART

May 27, 2008

Figure 6.3: Flow Chart

Run SCCI

Expression data available?

Yes

Run traditional CAI using highly expressed genes

No

Time permitting run GA maximizing true reference set gene placement (particularly if content criterion >0.7 (< -0.7)

Run SCCI

Content criterion >0.7 (< -0.7)?

Yes

Run traditional CAI using highly expressed genes

No

Time permitting run GA maximizing true reference set gene placement (particularly if content criterion >0.7 (< -0.7)

Ribosomal protein coding genes known?

Yes

Run mSCCI using RPCGs to determine alpha

No

Time permitting run GA maximizing HEDB gene placement (particularly if content criterion >0.7 (< -0.7)

Ribosomal (or HEDB) criterion ≥1?

Yes

Run mSCCI using HEDB genes to determine alpha

No

Run mSCCI using HEDB genes to determine alpha

Done


BIBLIOGRAPHY


Appendix A: Degeneration Table  
(Codon-Amino Acid-Abbreviation Cross-reference)

Table A.1: Degeneracy Table. Codon to amino acid cross-reference (single and three letter abbreviation for amino acid)

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>U</td>
<td>C</td>
<td>A</td>
<td>G</td>
</tr>
<tr>
<td>U</td>
<td>UU</td>
<td>Ser</td>
<td>UAU</td>
</tr>
<tr>
<td>U</td>
<td>UC</td>
<td>Ser</td>
<td>UAC</td>
</tr>
<tr>
<td>U</td>
<td>UA</td>
<td>Tyr</td>
<td>UAA</td>
</tr>
<tr>
<td>U</td>
<td>UG</td>
<td>Cys</td>
<td>UGA</td>
</tr>
<tr>
<td>C</td>
<td>CU</td>
<td>Pro</td>
<td>CAU</td>
</tr>
<tr>
<td>C</td>
<td>CC</td>
<td>Pro</td>
<td>CGU</td>
</tr>
<tr>
<td>A</td>
<td>AU</td>
<td>Ile</td>
<td>AAC</td>
</tr>
<tr>
<td>A</td>
<td>AC</td>
<td>Thr</td>
<td>GAG</td>
</tr>
<tr>
<td>G</td>
<td>GU</td>
<td>Val</td>
<td>GAU</td>
</tr>
<tr>
<td>G</td>
<td>GC</td>
<td>Ala</td>
<td>GAC</td>
</tr>
<tr>
<td>G</td>
<td>GA</td>
<td>Ala</td>
<td>GAA</td>
</tr>
</tbody>
</table>

204
Appendix B: Description of PCA

The following is a common sense, big picture description of the purpose and implementation of principal components analysis (PCA). The description contains just enough detail so that one could implement this procedure if necessary. PCA identifies a set of vectors that represent the axes of greatest variance in multivariate data. It will find a number of such axes equal to the original dimensionality (e.g. in 64 dimensional RSCU data it will find 64 axes). The most common use of PCA is reduce dimensionality down to something that is viewable (two or three dimensions) while retaining as much of the original meaning for the data as possible. Typically the original data is arranged in matrix form with rows representing the data points and the columns representing the dimensions in the data, as in the RSCU data for genes 1 through n (Figure B.1).

The general procedure for calculating PCA is to first generate a covariance matrix based upon the original data, then generate Eigenvectors on that data. These Eigenvectors are the principal components. The Eigenvector with the largest associated Eigenvalue is the axis associated with the greatest variance. In MatLab this procedure is as simple as the following command:

$$[\text{EVS, VALS}] = \text{PRINCOMP}(\text{DATA})$$

Where EVS contains the Eigenvectors and VALS contains the Eigenvalues. To reduce the dimensionality the original data is projected onto the first two or three Eigenvectors. This is done through the performance of a dot product between the original data and the Eigenvector in question (Figure B.2 and Equations B.1 and B.2).

$$
\begin{align*}
&c_1 & c_2 & c_3 & \cdots & c_{59} \\
g_1 & f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,59} \\
g_2 & f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,59} \\
g_3 & f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,59} \\
&\vdots & \vdots & \vdots & \vdots & \vdots \\
g_n & f_{n,1} & f_{n,2} & f_{n,3} & \cdots & f_{n,59}
\end{align*}
$$

Figure B.1: RSCU Matrix. Genes are rows and columns are frequencies of codons. The original data points are the genes.
B.1 Covariance Matrix

A vector (first principal component) that extends along the axis of greatest variance is desired. For this reason it is logical to begin with a covariance matrix.

\[
\begin{align*}
    g_1 & \cdot f_{1,1} & f_{1,2} & f_{1,3} & \cdots & f_{1,59} & b_{1,1} & Z'_{i,1} \\
    g_2 & \cdot f_{2,1} & f_{2,2} & f_{2,3} & \cdots & f_{2,59} & b_{1,2} & Z'_{i,2} \\
    g_3 & \cdot f_{3,1} & f_{3,2} & f_{3,3} & \cdots & f_{3,59} & \cdot b_{1,3} & Z'_{i,3} \\
    \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
    g_n & \cdot f_{n,1} & f_{n,2} & f_{n,3} & \cdots & f_{n,59} & b_{1,n} & Z'_{i,n}
\end{align*}
\]

Figure B.2: Graphical representation of \( X \cdot b_1 = Z'_i \). This projects the original data onto the first principal component.

\[
\begin{align*}
    X \cdot b_1 & = Z'_1 \\
    X \cdot b_2 & = Z'_2
\end{align*}
\]

Next, a more in depth examination of the underlying math will be presented. This can be skipped if a high-level understanding is all that is required.

B.1 Covariance Matrix

A vector (first principal component) that extends along the axis of greatest variance is desired. For this reason it is logical to begin with a covariance matrix.

\[
\begin{align*}
    var(x) & = \frac{\sum (x_i - \bar{x})(x_i - \bar{x})}{n - 1} \\
    cov(x) & = \frac{\sum (x_i - \bar{x})(y_i - \bar{x})}{n - 1}
\end{align*}
\]

Cell 1,1 of the covariance matrix contains the covariance between codon 1 and codon 1 (i.e. the variance of codon 1). Cell 1,1 of the covariance matrix contains the covariance between codon 1 and codon 2.

B.2 Eigenvector

Next, generate Eigenvectors and values from the above covariance matrix.

Given a matrix \( A \) and an Eigenvector \( \mathbf{v} \) and an Eigenvalue \( \lambda \) the following formula holds.

\[
A \mathbf{v} = \lambda \mathbf{v}
\]

An alternative name for the Eigenvalue is proper or characteristic value. With a little algebra, the following equation falls out:
B.3. DETERMINANTS

May 27, 2008

\[ A \mathbf{v} = \lambda \mathbf{v} \quad (B.6) \]
\[ A \mathbf{v} - \lambda \mathbf{v} = 0 \quad (B.7) \]
\[ A \mathbf{v} - \lambda \mathbf{I} \mathbf{v} = 0 \quad (B.8) \]
\[ (A - \lambda \mathbf{I}) \mathbf{v} = 0 \quad (B.9) \]

So given the Eigenvalues one should be able to solve for the Eigenvector using the above equation. In order to solve for the Eigenvalues a determinant must be utilized.

B.3 Determinants

To solve the above equation a determinant must be employed. Determinants can be used to determine if a matrix is invertible. The nomenclature is as follows:

\[ A^{-1} = \frac{1}{\det A} \quad (B.10) \]
\[ \det A = |A| \quad (B.11) \]

B.4 Characteristic Equation and Polynomial

The determinant of the reduced equation is known as a characteristic polynomial:

\[ P(\lambda) = \det(A \mathbf{v} - \lambda \mathbf{I}) \mathbf{v} = 0 \quad (B.12) \]

As an example if there were an \( A \) defined as follows:

\[ A = \begin{pmatrix} a & b \\ c & d \end{pmatrix} \quad (B.13) \]

Solve for the \( \lambda \) by performing the following operations:

\[ A - \lambda \mathbf{I} = \begin{pmatrix} a & b \\ c & d \end{pmatrix} - \begin{pmatrix} \lambda & 0 \\ 0 & \lambda \end{pmatrix} = \begin{pmatrix} a - \lambda & b \\ c & d - \lambda \end{pmatrix} \quad (B.14) \]
\[ \det(A - \lambda \mathbf{I}) = ad - a\lambda - d\lambda + \lambda^2 - bc = 0 \quad (B.16) \]
\[ P(\lambda) = \lambda^2 - (a + d)\lambda + (ad - bc) = 0 \quad (B.17) \]
If solve for $\lambda$ will get 2 roots, $\lambda_1$ and $\lambda_2$. Now that the Eigenvalues have been acquired one can solve for the Eigenvector ($v$ below).

\[(A - \lambda I)v = 0 \quad \text{(B.18)}\]

Eigenvectors (calculated on a covariance matrix) have the useful property of being orthogonal to one another. This is due to the symmetric nature of a covariance matrix. They also are of unit length.
Appendix: C++ GA Software Design

Figure C.1: Unified Modeling Language (UML) Chart for C++ GA Code