Journal of Discrete Algorithms 16 (2012) 104–112

Contents lists available at SciVerse ScienceDirect

Journal of Discrete Algorithms

www.elsevier.com/locate/jda

Efficient codon optimization with motif engineering ☆

Anne Condon *, Chris Thachuk

Department of Computer Science, University of British Columbia, Vancouver, BC, V6T 1Z4, Canada

ARTICLE INFO

Article history:
Available online 3 May 2012

Keywords:
Codon optimization
Dynamic programming
Sequence design
Motif engineering

ABSTRACT

It is now common to add synthetic protein coding genes into cloning vectors for expression within non-native host organisms. Codon optimization is the task of choosing a sequence of codons that specify a protein so that the chosen codons are those used with the highest possible frequency in the host genome, subject to certain constraints, such as ensuring that occurrences of pre-specified “forbidden” motifs are minimized. Codon optimization supports translational efficiency of the desired protein product, by exchanging codons which are rarely found in the host organism with more frequently observed codons. Motif engineering, such as removal of restriction enzyme recognition sites or addition of immuno-stimulatory elements, is also often necessary. We present an algorithm for optimizing codon bias of a gene with respect to a well motivated measure of bias, while simultaneously performing motif engineering. The measure is the previously studied codon adaptation index, which favors the use, in the gene to be optimized, of the most abundant codons found in the host genome. We demonstrate the efficiency and effectiveness of our algorithm on the GENCODE dataset and provide a guarantee that the solution found is always optimal. The implementation and source code of our algorithm are freely accessible at http://www.cs.ubc.ca/labs/beta/Projects/codon-optimizer.

1. Introduction

Gene synthesis is now an economical and technically viable option for the construction of non-natural genes. Synthetic genes can be novel or derivatives of those found in nature. In either case, the expression levels of these genes, when inserted into the genome of a host organism, depend on many factors. One important factor is the bias of codon usage, relative to the host organism [8,11,18]. Note that for each amino acid in a protein, there may be many (up to six) valid codons, as given by the genetic code. Loosely speaking, the codon bias of a gene for the protein measures how well—or poorly—codons used in the gene match codon usage in the genome of a host organism (we describe specific measures later in this paper). In a study by Lithwick and Margalit [12] on the effects of sequence-dependent features associated with prokaryotic translation, the authors concluded that codon bias is the feature most highly associated with the level of protein expression. It was demonstrated by Kane [11] that usage of rare codons, especially when clustered, is detrimental to protein expression levels [11]. Gao et al. [4] noted that a key obstacle to DNA-based vaccines for the human immuno-deficiency virus (HIV) is low expression levels of HIV genes in mammalian cells, which they attribute to rare codon usage and AU-rich elements. These studies indicate that it is desirable to choose codons that have high usage in a host's genome, in order to ensure that designed genes are maximally expressed within the host.

☆ A preliminary version of this work appeared in the Proceedings of the 2011 International Workshop on Combinatorial Algorithms (IWOCA 2011) (Condon and Thachuk, 2011 [2]).

* Corresponding author.

E-mail addresses: condon@cs.ubc.ca (A. Condon), cthachuk@cs.ubc.ca (C. Thachuk).

1570-8667/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jda.2012.04.017

In addition to optimizing the codon bias of a gene, relative to the genome of a host, it is often desirable or necessary to add or remove certain motifs, i.e., subsequences, via silent mutation: altering the DNA sequence, provided the amino acid sequence it encodes remains unchanged. This is common practice and important for the elimination of restriction enzyme recognition sites of a host organism [7,16]—or inclusion of these elements for diagnostic purposes—and removal of polyhomomeric repeat regions [19]. Exclusion of these elements can be seen as a hard constraint. Yet, in other instances, removal or addition of motifs can be treated more naturally as optimization criteria to be minimized or maximized. This is the case, for instance, with immuno-regulatory CpG motifs in mammalian expression vectors [14], where it is desirable to minimize immuno-inhibitory elements and maximize immuno-stimulatory motifs. In the remainder of this work, we will refer to inclusion or exclusion of motifs, via silent mutation, as motif engineering.

A number of published software tools are capable of codon optimization, i.e., of choosing a sequence of codons that specify a protein so that the product of the frequencies of the chosen codons in the host genome is as high as possible, subject to certain constraints such as exclusion or inclusion of certain subsequences. These tools include DNA Works [9], Codon Optimizer [3], GeMS [10], Gene Designer [19], JCat [5], OPTIMIZER [13], the Synthetic Gene Designer [20], UpGene [4] and a method by Satya et al. [14]. Some of these methods also consider the other problem considered here, motif engineering. Of these, only the method of Satya et al. provides a mathematical guarantee of finding an optimal solution to codon optimization when one exists. However, their method—based on the graph theoretic approach of finding a critical path—runs in $O(n^2)$ time and space, where n is the length of the DNA sequence being optimized. Skiena [16] presents an efficient algorithm for minimizing forbidden motifs when choosing a sequence of codons that specify a protein, assuming that the length of forbidden sequences is bounded by a constant. Skiena also shows that a decision version of the problem is NP-complete when there is no bound on the length of forbidden sequences. However, Skiena's work does not address codon optimization. In this work, we propose the first linear time and space codon optimization algorithm that is guaranteed to find an optimal solution and that also satisfies motif engineering constraints. We have focused our attention on optimizing codon usage according to the Codon Adaptation Index (CAI). The index, originally proposed by Sharp and Li [15], is based on the premise of each amino acid having a 'best' codon for a particular organism. This perspective evolved from the observation that protein expression is higher in genes using codons of high fitness and lower in genes using rare codons [7]. It is believed that this is due to the relative availability of tRNAs within a cell.

We also provide an experimental study of the performance of our algorithm on a biological data set comprising 3157 coding sequence regions of the GENECODE subset of the ENCODE dataset [17]. The implementation and source code of our algorithm are freely accessible at http://www.cs.ubc.ca/labs/beta/Projects/codon-optimizer.

The remainder of this paper is structured as follows. In the Preliminaries section, we formally define the problem of codon optimization. We detail the general objectives of the problem, and formalize the goals of motif engineering. We then present our algorithm, providing a proof of correctness and time and space analysis. In the Empirical Results section, we describe the performance of our algorithm, both in terms of run-time efficiency and also in terms of the quality of optimization achieved. Finally, we conclude with a summary of our major findings and directions for future work.

2. Preliminaries

A DNA strand is a string over the alphabet of DNA. A *codon* is a triple over the DNA alphabet and therefore there are at most $4^3 = 64$ distinct codons. An *amino acid sequence* is a string over the alphabet of amino acids, $\Sigma_{AA} = \{Ala, Arg, Asn, Asp, Cys, Glu, Gln, Gly, His, Ile, Leu, Lys, Met, Phe, Pro, Ser, Thr, Trp, Tyr, Val, stop\}$ with each symbol representing an amino acid and the special symbol '*stop*' denoting a string terminal. We assume there is a predetermined ordering of amino acids, for example, lexicographic.

Therefore, we can represent an amino acid sequence A as a sequence of integers, with $A = \alpha_1, \alpha_2, \ldots, \alpha_{|A|}$, where $1 \leqslant \alpha_k \leqslant 21$, for $1 \leqslant k \leqslant |A|$. For example, the amino acid sequence of the problem instance in Fig. 1 can be represented as $A = 19, 11, 1, 19$. The *genetic code* is a mapping between amino acids and codons. However, as there are 64 possible codons and only 20 amino acids (plus one stop symbol), the code is degenerate, resulting in a one-to-many mapping from each amino acid to a set of corresponding codons.

The process of gene translation can be thought of more naturally as a mapping of codons to amino acids, however, we define our mapping as the inverse for convenience. We denote the set of codons for the ith amino acid by $\lambda(i)$. Therefore, for the first amino acid, namely Ala, $\lambda(1) = \{\mathtt{GCA}, \mathtt{GCC}, \mathtt{GCG}, \mathtt{GCU}\}$. Similarly, the second amino acid is Arg and $\lambda(2) = \{\mathtt{AGA}, \mathtt{AGG}, \mathtt{CGA}, \mathtt{CGC}, \mathtt{CGG}, \mathtt{CGT}\}$, and so on. We let $\lambda_j(i)$ be the jth codon in the set $\lambda(i)$, $1 \leqslant j \leqslant |\lambda(i)|$, where again we use lexicographic ordering. Therefore, we can define a DNA encoding for an amino acid sequence as a sequence of codon indices. Again consider the problem instance in Fig. 1. For the Leucine amino acid (Leu) which is the 11th amino acid in lexicographic order, $|\lambda(11)| = 6$ and $\lambda_3(11)$ is the codon CUG. The DNA sequence UAC CUC GCC UAC can be represented by the codon index sequence $S = 1, 2, 2, 1$; that is, $\lambda_1(19)\lambda_2(11)\lambda_2(1)\lambda_1(19) = \mathtt{UAC}\ \mathtt{CUC}\ \mathtt{GCC}\ \mathtt{UAC}$. We refer to S as the *codon design*.

A *codon's frequency* is the number of times that it appears in nature, divided by the total number of times that all codons corresponding to the same amino acid appear in nature. By "in nature", we mean codon frequencies present in some reference sequence or set of sequences such as a genome or set of genomes. As an example, if for some amino acid index i, $|\lambda(i)| = 2$, and the codon $\lambda_1(i)$ is observed 37 times in nature, while $\lambda_2(i)$ is observed 63 times, we can define the frequency of $\lambda_1(i)$ to be $\frac{37}{37+63} = 0.37$. Let $\rho_j(i)$ denote the frequency of the jth codon of the ith amino acid, $1 \leqslant j \leqslant |\lambda(i)|$.