A Pattern Matching Algorithm for Codon Optimization and CpG Motif-Engineering in DNA Expression Vectors

Ravi Vijaya Satya and Amar Mukherjee
School of Engineering and Computer Science
University of Central Florida
Orlando, FL 32816
rvijaya, amar@cs.ucf.edu

Udaykumar Ranga
Jawaharlal Nehru Center for Advanced
Scientific Research
Jakkur, Bangalore, India
udaykumar@jncasr.ac.in

Abstract

Codon optimization enhances the efficiency of DNA expression vectors used in DNA vaccination and gene therapy by increasing protein expression. Additionally, certain nucleotide motifs have experimentally been shown to be immuno-stimulatory while certain others immuno-suppressive. In this paper, we present algorithms to locate a given set of immuno-modulatory motifs in the DNA expression vectors corresponding to a given amino acid sequence and maximize or minimize the number and the context of the immuno-modulatory motifs in the DNA expression vectors. The main contribution is to use multiple pattern matching algorithms to synthesize a DNA sequence for a given amino acid sequence and a graph theoretic approach for finding the longest weighted path in a directed graph that will maximize or minimize certain motifs. This is achieved using $O(n^2)$ time, where n is the length of the amino acid sequence. Based on this, we develop a software tool.

Key Words: Codon optimization, immuno-modulatory motifs, multiple pattern matching, longest weighted path.

1. Introduction

DNA vaccines have revolutionized the field of vaccine technology by demonstrating the ability to induce humoral and cellular immune responses in experimental animals and humans [9]. Immunization of animals with plasmid DNA encoding a protein antigen was an accidental observation that eventually led the way to a novel strategy of immunization. DNA vaccines, also known as "naked DNA" or "nucleic acid" vaccines, encode the antigens of pathogenic organisms including viruses, bacteria, fungi and parasites [40]. The protein antigen is processed within the cell and presented by the MHC-I and –II pathways thereby eliciting specific immune responses essential for controlling pathogenic infections [2]. Although DNA vaccines have been successful in generating strong immune responses in smaller animal model such as mouse, they have not been as efficient in larger species such as primates and humans [23, 3, 25, 4]. Stimulating both the arms of the immune system is often desirable for efficient control of infectious diseases especially in the larger animals. In the case of recombinant protein vaccines, immune-enhancers technically known as adjuvants, such as Freund's adjuvants and Alum, are in use to enhance antigen specific immune responses. However, no such adjuvants are available for use in the context of DNA vaccines. The lack of suitable adjuvants for DNA vaccines is one important reason for the poor performance of the DNA vaccines in larger animals.

Nucleotide sequence encoding a foreign protein is directly placed under the control of a mammalian promoter to construct a DNA vaccine. Several amino acids are encoded by more than one triplet codon and different organisms have variable requirement for codon preference [13]. Cloning of a wild type gene from a parasite into a DNA vaccine often leads to insufficient levels of protein synthesis in the host cell as a result of codon bias between the species. Successful immunization with DNA vaccines requires high expression of cloned genes to synthesize large quantities of the foreign protein. For instance, the overall genetic content of Human Immunodeficiency Virus-1 is AT-rich, while that of the human beings is CG-rich. The codon frequency of the pathogenic DNA embedded into the mammalian expression vector may not be optimal for adequate protein expression in the host resulting in low level protein expression. A potential solution for the codon bias is to optimize the codon sequences of a gene to suit the requirements of the host without altering the original amino acid sequence of the protein [41, 16]. This approach has been successful in eliciting strong immune responses in several species of experimental animals [8,

28]. While immunization with synthetic genes, codon-optimized for mammalian expression stimulated strong immune response, immunization performed in parallel with wild type genes generated low or moderate levels of immune response [34, 36].

In addition to codon optimization of the synthetic genes, a range of molecular approaches is being evaluated to up-regulate immune responses generated by DNA vaccines. Co-expression of cytokine genes [20], co-stimulatory receptors [12, 35] or other immunemodulators [33], synthetic assembly of T-helper or CTL epitopes [6] and formulation with a variety of chemical adjuvants [11, 24, 33, 37, 39] have been some of the approaches reported. However, most of these approaches may not be suitable for human application due to toxic manifestations of the adjuvants. An ideal agent used as an adjuvant for DNA vaccine must enhance the immunogenicity without apparent cytotoxicity to the host. Engineering CpG islands into DNA vaccines has been one promising approach that showed enhanced immune responses [19].

The well-established immune-enhancing property of bacterial DNA has been mapped to sequence motifs consisting of un-methylated CpG dinucleotides flanked by base pairs in a specific context [18]. Two important differences between the bacterial and mammalian DNA enable the mammalian innate immune system to recognize the former as a foreign component. CpG motifs in bacteria are found at the expected 1:16 frequency; however, their frequency in mammals is 4 times less than expected. Bacterial CpG are non-methylated while those of mammals are mostly methylated. The mammalian immune system takes advantage of these two chemical differences between the bacterial and mammalian DNA to identify a bacterial infection and wage strong and rapid anti-bacterial immune responses [29].

CpG-mediated activation of the mammalian innate immune system has been extensively exploited in the vaccination technology. Co-injection of CpG containing oligonucleotides or empty vectors with protein antigens elicited potent immune response to the antigen. Methylation of the CpG motifs, on the other hand, abrogated immune response to the antigens suggesting that the CpG motifs possess adjuvant properties only when unmethylated or hypomethylated [[5, 17]. Since the DNA expression vectors used in genetic immunizations are usually grown in bacteria, several CpG motifs on the plasmids are not methylated possibly activating the innate component of the mammalian immune system. Presence of hypo-methylated CpG motifs is essential for induction of immune responses to the antigens encoded by the vectors.

While certain CpG motifs are immuno-stimulatory (CpG-S), enhancing immune responses when the host is vaccinated, others CpG motifs in a different nucleotide context are immuno-inhibitory or -neutralizing (CpG-N) abrogating antigen-specific immune response [18]. Engineering the nature and the frequency of the CpG motifs may be critical for the design of DNA expression vectors. DNA expression vectors are primarily used for two different applications, expression of a foreign gene in a host for vaccination or for correcting a genetic defect of the host [26].

Although the basic design of these two types of vectors is identical in several respects, their requirement for the presence and nature of CpG motifs is diagonally opposite. While genetic vaccines require CpG-S motifs for efficient stimulation of the host immune system, such a strong response must be avoided for long-term survival of the DNA expression vector intended for gene therapy. An ideal strategy for designing DNA vaccines must recruit as many CpG-S motifs as possible, concomitantly eliminating as many CpG-N motifs as possible without altering the original protein sequence of the genes. In contrast, design of a DNA expression vector intended for gene therapy must eliminate as many CpG-S motifs as possible and recruit as many CpG-N motifs as possible for the best result. Thus, depending on the application, some CpG motifs may be **desirable**, while certain others may be **undesirable**.

The software tool we report here is designed to help the researcher to engineer the composition of a gene with respect to codon usage and CpG motifs without modifying the original amino acid sequence of the protein. Codon optimization is advantageous for protein expression in a heterologous expression system such as *E.coli*, *Picchia*, *Saccharomyces*, *Baculo virus*, mammalian cells etc. The software could also engineer the content and nature of the CpG motifs recruited into a DNA expression vector in addition to optimizing the codon usage and resolve potential conflicts arising between these two requirements and find an optimal nucleotide sequence.

2. Prior Work and Summary of Contributions

Objective of this algorithm is to (1) identify the best triplet codon for codon optimization and (2) engineer the nature and content of the CpG motifs of a gene expressed from a genetic vector. Input for the software is the amino acid sequence of a gene of interest or the nucleotide sequence that needs genetic modification. The input amino acid sequence is reverse-translated to generate nucleotide sequence that is optimized for the codon and CpG content. During the process of reverse-translation, there is a one-to-many relationship between the amino acid sequence and the DNA sequences. A given amino acid sequence could correspond to an exponential number of DNA sequences with respect to the length of the given amino acid sequence. Our task here is to choose the particular sequence that has the desirable properties (maximum number of the desirable motifs, and the minimum number of the undesirable motifs) from this large number of possible DNA sequences. A brute force search on all the possible DNA sequences will take exponential time. The motifs (or the patterns) that need optimization are small DNA sequences, generally not exceeding eight nucleotides in length. If different motifs have to be optimized simultaneously, it is possible to have multiple motif occurrences (both desirable and undesirable) sharing the same position in the amino acid sequence. In such cases it may not be possible to have all the motif occurrences in the same DNA sequence. This problem will be explained in detail in Section 4.

McInerney[27] presented a program to perform the codon usage analysis on a sequence or a database of sequences. Other work in this area of bioinformatics is mostly on finding CpG islands within the genes and transcriptional elements that regulate gene expression. Ponger et al [30] developed a software package to locate CpG islands associated with the transcription start sites of the genes. Lin et al [22] presented software for locating non-overlapping maximum average segments in a given sequence. These publications analyzed DNA sequences and searched for nucleotide motifs with high CG content.

Our emphasis in this paper is to synthesize a DNA sequence that codes for a functional protein, with certain characteristics and optimization parameters. The main contribution is to use multiple pattern-matching algorithms to synthesize a DNA sequence for a given amino acid sequence and a graph theoretic approach for finding the longest weighted path in a directed graph that will maximize or minimize certain motifs as well as guarantee certain fitness factors of codon frequency usage for a particular species. This is achieved using $O(n^2)$ time and storage resources compared to the brute force algorithm that might take exponential amount of

resources, where n is the length of the amino acid sequence. The software tools developed for the purpose will find applications in the rapid development of vaccines and gene therapy.

In Section 3, we introduce the basic terminology. In Section 4, we give a precise combinatorial formulation of the problem in terms of pattern matching operations and present the main algorithm. Section 5 gives the complexity analysis. Section 6 gives a description of the software and in Section 7 we present the results.

3. Terms and definitions

The alphabet of a **DNA sequence** is denoted as $\Sigma_N = \{A,C,G,T\}^1$ where A,C,G,T are the four DNA molecules adenine, cytosine, guanine, and thymine, respectively. The alphabet for the **amino acid sequence** is $\Sigma_A = (A,R,N,D,C,Q,E,G,H,I,L,K,M,F,P,S,T,W,Y,V,stop)$, each letter indicating one of the 20 amino acids, and stop indicating the stop codon. For example, 'A' indicates Alanine, 'R' indicates Arginine, 'N' indicates Asparagine, etc. A **codon** c is a triplet of the DNA symbols in Σ_N. As the size of DNA alphabet is 4, there are 64 possible codons. In the context of amino acid translation from mRNA, each element of Σ_A gets mapped to a maximum of 6 codons[2]. This mapping is also referred to as the **genetic code**. Symbolically, the mapping of an amino acid $\sigma \in \Sigma_A$ can be denoted as a set $C(\sigma)$ of n_σ codons, where $1 \leq n_\sigma \leq 6$. The inverse-mapping associates an amino acid σ with a codon c, denoted by $C^{-1}(c) = \sigma$. The frequency of occurrence of members of $C(\sigma)$ is different for each species. This relative difference in the frequency of occurrence for each codon is called **codon bias**. The **codon usage table** for a species gives the codon bias information for that species[3]. For an amino acid σ, the codon bias (or the frequency information) is given by the set of fractions $F(\sigma)$, corresponding to $C(\sigma)$. For each $f_i \in$

$F(\sigma)$, $1 \leq i \leq n_\sigma$, $0 \leq f_i \leq 1$, and $\sum_{i=1}^{n_\sigma} f_i = 1$. When designing a DNA vaccine for a species, it is often desirable to have those codons that occur more frequently

in that species. Therefore, for each amino acid we generally choose the most frequent codon, henceforth referred to as the **best codon**. For an amino acid σ, the index of the best codon, b_σ is defined as $b_\sigma = \{i \mid f_i = MAX(F(\sigma))\}$. If there are more than one values for b_σ, we take one arbitrarily. It is also useful to have a measure of how desirable a codon is. This quantity is given by the **fitness** of the codon. The fitness of the ith codon, c_i of an amino acid σ is given by: $fitness(c_i) = \dfrac{f_i}{f_{b_\sigma}}$, the ratio of the frequency of the ith codon of σ to the frequency of the best codon of σ. Given a list of codons, the **average fitness** of the list of codons is defined as the arithmetic mean of the *fitness* values of all codons in the list.

patterns, $aasp(M^1)$, $aasp(M^2)$, $aasp(M^3)$ will also be empty strings, but their head and/or tail portions will be non-null. This implies that, for such an *aasp*, every position in the amino acid sequence is a possible occurrence of the motif. The three possible codon encodings of a motif M and their corresponding three amino acid search patterns are shown in Table 1.

As an example, if $M = ATCGAT$, $M^1 = ATCGAT$, $aasp(M^1) = ID$, $head(M^1) = null$, $tail(M^1) = null$; $M^2 = TCG$, $aasp(M^2) = S$, $head(M^2) = A$, $tail(M^2) = AT$; and $M^3 = CGA$, $aasp(M^3) = R$, $head(M^3) = AT$, $tail(M^3) = AT$. For example, searching for the motif $M = ATCGAT$ in the amino acid sequence ...*LSI*..., we find $S = assp(M^2)$. $head(M^2) = A$ and $tail(M^2) = AT$. L, the amino acid that precedes S, has two codons that end with A ($head(M^2)$),

Table 1. Head and tail of amino acid search patterns

$M^1 = (m_1 m_2 m_3) \cdots (\cdots m_{\lfloor l/3 \rfloor * 3})$ $aasp(M^1) = a_1 a_2 \ldots \ldots a_{\lfloor l/3 \rfloor}$, where $a_1 = C^{-1}(m_1 m_2 m_3)$,etc.	$head(M^1) = null$	$tail(M^1) = null$ if $l \bmod 3 = 0$ $tail(M^1) = m_l$ if $l \bmod 3 = 1$ $tail(M^1) = m_{l-1} m_l$ if $l \bmod 3 = 2$
$M^2 = (m_2 m_3 m_4) \cdots (\cdots m_{\lfloor (l-1)/3 \rfloor * 3 + 1})$ $aasp(M^2) = a_1 a_2 \ldots \ldots a_{\lfloor (l-1)/3 \rfloor}$, where $a_1 = C^{-1}(m_1 m_2 m_3)$, etc.	$head(M^2) = m_1$	$tail(M^2) = null$ if $l \bmod 3 = 1$ $tail(M^2) = m_l$ if $l \bmod 3 = 2$ $tail(M^2) = m_{l-1} m_l$ if $l \bmod 3 = 0$
$M^3 = (m_3 m_4 m_5) \cdots (\cdots m_{\lfloor (l-2)/3 \rfloor * 3 + 2})$ $aasp(M^3) = a_1 a_2 \ldots \ldots a_{\lfloor (l-2)/3 \rfloor}$, where $a_1 = C^{-1}(m_3 m_4 m_5)$,etc.	$head(M^3) = m_1 m_2$	$tail(M^3) = null$ if $l \bmod 3 = 2$ $tail(M^3) = m_l$ if $l \bmod 3 = 0$ $tail(M^3) = m_{l-1} m_l$ if $l \bmod 3 = 1$

A **motif** is a short DNA sequence of length l, denoted as $M = m_1 m_2 m_3 \ldots \ldots m_l$, $m_i \in \Sigma_N, 1 \le i \le l$. We will consider immuno stimulatory and immune neutralizing (CPG-S and CPG-N) sequences as possible motifs. A motif occurs as part of a long DNA sequence and as such will correspond to a sequence of codons. Since the beginning of the codon boundary could be aligned with any one of the first, second, or third positions in the motif, this gives rise to three possible codon encodings for the motif. In each codon encoding, the codons that lie completely within the motif can be translated to an amino acid sequence. These amino acid sequences will be referred to as **amino acid search patterns** (henceforth referred to as *aasp*). The parts of the motif (less than three nucleotide beginning or the end of the motif that do not completely correspond to a codon are referred to as the **head** or **tail**, respectively, of the motif. When $l < 5$, all or some of M^1, M^2, M^3 may be null. The corresponding amino acid search

and *I*, the amino acid that follows *S*, has three codons that begin with AT ($tail(M^2)$), as shown below.

L	S	I
CTA	TCG	ATA
TTA		ATC
		ATT

Selecting any codon from the first column and any codon from the third column will result in an occurrence of *ATCGAT*. Therefore, there are *2*3 = 6* unique combination of codons that result in the occurrence of *ATCGAT* at the location in the amino acid sequence beginning with *L* and ending with *I*. These will be referred to as the possible **occurrences** of the motif. The codons corresponding to an occurrence will be referred to as **constituent codons** for that occurrence. We call the collection of all possible occurrences of all motifs as the **potential occurrences**, as all of these occurrences might

not be retainable in the final solution. As noted in Section 1, some of these occurrences may be desirable while others may be undesirable; we assign a **desirability value** of +1 and –1, respectively , for these occurrences. In general, these values could be arbitrary.

An additional step of pre-processing is required for undesirable motifs. We want to avoid occurrence of an undesirable motif. However, we need to make sure that the undesirable motifs do not re-appear in the final solution as a result of selecting the best codons for amino acids that are not part of a desirable motif. We achieve this by considering all alternate codon combinations of each potential occurrence of an undesirable motif, and treating them as potential occurrences with a *desirability* value of zero. We call these alternate potential occurrences as **neutral occurrences**. In the above example, if *ATCGAT* were an undesirable motif, out of a total of 6*6*3 = 108 possible encodings for the subsequence *LSI*, 108-6 = 102 codon combinations do not result in the occurrence of *ATCGAT*. Each one of these 102 combinations will be registered as a *neutral occurrence*.

The **weight** of an *occurrence* is the sum of the *average fitness* of the constituent codons and the **desirability** for that motif. The **total weight** of a DNA sequence is the sum of the individual *weights* of all the occurrences from the given set of motifs that are part of the DNA sequence.

4. The Algorithm

4.1 Problem Definition

Given an **amino acid sequence** $S_A = a_1a_2a_3.............a_J$, and a **list of motifs** $M = \{M_1, M_2, M_3, ..., M_k\}$, with their corresponding *desirability* values, $D = \{D_1, D_2, D_3, ..., D_k\}$, the problem is to find the **codon sequence, S_C** $= c_1c_2c_3.............c_J$, where each $c_i \in C(a_i)$, $1 \le i \le J$, and the **DNA sequence, S_D** $= d_1d_2d_3.............d_{3*J}$, corresponding one-to-one to S_C, where $c_i = d_{3i-2}d_{3i-1}d_{3i}$, such that: 1)$S_D$ has the maximum **total weight** possible; and 2)Every codon c_i in S_C that is not part of an occurrence (desirable, undesirable or neutral) of a motif in M is the *best codon* for the corresponding amino acid a_i. In other words, we need to find the exact mapping of the amino acid sequence into a DNA sequence that has the maximum weighted occurrences of the motifs in M.

We find all possible occurrences of all the motifs, including neutral occurrences for the undesirable motifs. Out of these, if two occurrences (of the same motif or of different motifs) extend over the same position in S_A, and require different codons for that position, then the two occurrences have a *conflict* and cannot co-exist. We have to select the largest weighted subset of these occurrences that are mutually non-conflicting.

4.2 The Algorithm Steps

Step1: Pre-processing:

A)Form a list L_p of search patterns consisting of all three search patterns $aasp(M_i^1)$, $aasp(M_i^2)$ and $aasp(M_i^3)$ for each motif $M_i \in M$, $1 \le i \le k$, in the list of motifs. Also, enumerate the head and tail for all three alignments of each motif. Each entry in L_p is an element of a data structure having the following members:

- *codon_list*, the list of codons that lie entirely within the motif.
- *search_pattern*, the *aasp* corresponding to the *codon_list*.
- *length*, the size of the *codon_list*. (note: *length* may be 0 for short motifs).
- *head*, the corresponding head – a DNA sequence which is either null, or 1 or 2 characters in length.
- *tail*, the corresponding tail – a DNA sequence which is either null or 1 or 2 characters in length.
- *desirability*, the desirability value for the corresponding motif, +1 or –1.

B) Build an Aho-Corasick keyword tree [1] *ACT* for all the amino acid search patterns.

- A node in the keyword tree might correspond to multiple search patterns with different *head*s and *tail*s, henceforth referred to as the *list_of_matches*, denoted by the indices into L_p.

- When the length of a motif is less than five nucleotides, some amino acid search patterns might be empty strings. i.e., they will have non-empty head and/or tail, but empty search pattern. In the keyword tree, the root itself will correspond to such empty amino acid search patterns.

Step2: Finding all potential occurrences: In this step, we build the list of potential occurrences, *P*. Each entry in *P* will have the following attributes:

- **pBegin** is the position in S_A at which the match starts.
- **pEnd**, the position in S_A at which the match ends.
- A list of codons L_c of size $pEnd-pBegin+1$. The codon at position 'r' in L_c indicates rth codon in the match.
- The **weight** of the occurrence. We will generate the list P in **sorted** order in ascending values of $pBegin$; if two occurrences have same $pBegin$ value, they are sorted in ascending value of $pEnd$. The following algorithm traverses the tree according to the given amino acid sequence.

Algorithm 4.2.1 find_potential_occurrences

Inputs: The Aho-Corasick keyword tree *ACT*, the amino acid sequence S_A, list of search patterns L_p

Output: The list of potential occurrences, P, sorted.

Initialize $P \leftarrow null$, $cur\_node \leftarrow ACT.root$
for $q=1$, until $q \leq J$, traverse ACT:
    if $cur\_node.child[a_q]$!= *null*
        $cur\_node \leftarrow cur\_node.child[a_q]$
    else
        $cur\_node \leftarrow cur\_node.failure\_link$
 if $cur\_node.list\_of\_matches$ is not empty,
/*Corresponding aasp has occurred. Check if the motif corresponding to the aasp can actually occur. The head and tail are compared with all the codons of the amino acids on either side aasp*/
 for each entry i in the list, do the following:
   a. Enumerate position p, the starting position of $L_p(i).search\_pattern$ in S_A.
   b. $pBegin \leftarrow p$, $pEnd \leftarrow q$
   c. **if** $(L_p(i).head \neq null)$ /* head is not null */
      i. $pBegin \leftarrow pBegin-1$
      ii. form a set of codons C_{head} such that $L_p(i).head$ is a suffix of each $c_{head} \in C_{head}$, $C_{head} \subseteq C(a_{p-1})$
   d. **if** $(L_p(i).tail \neq null)$ /* tail is not null */
      i. $pEnd \leftarrow pEnd+1$
      ii. form a set of codons C_{tail} *such that* $L_p(i).tail$ is a suffix of each $c_{tail} \in C_{tail}$, $C_{head} \subseteq C(a_{q+1})$
   e. **for** each unique combination of $c_{head} \in C_{head}$ and $c_{tail} \in C_{tail}$, insert an entry to the list of potential occurrences, P, with the entries $pBegin$, $pEnd$, $L_p(i).codon\_list$, and the corresponding *weight* of the occurrence. Note: C_{head} and/or C_{tail} will be empty if $L_p(i).head$ and/or $L_p(i).tail$ are null.
 if a non-zero number of occurrences were added in the above step, and *if $L_p(i).desirability < 0$*, insert an entry in P for each unique combination of codons selecting one codon each from the sets $C(S_A(pBegin))$, ……, $C(S_A(pEnd))$. i.e, insert all neutral occurrences corresponding to the current undesirable occurrence into P.

Example: Let us assume we are trying to maximize the number of occurrences of the motif *ATCGAT* in the amino acid sequence *MEPRVID*. The three amino acid search patterns, $aasp(M^1)$, $aasp(M^2)$ and $aasp(M^3)$ are *ID*, *S*, and *R*, respectively. Searching for them, we find *R* at position 4, and *ID* at positions 6-7. *R* is $aasp(M^3)$, $head(M^3)$ is *AT*, $tail(M^3)$ is *T*, therefore we have to see if any codons ending with *AT* can occur at position 3 in *MEPRVID*, and if any codons beginning with *T* can occur at position 5. The amino acid at position 3 is *P*, which has the codons [*AGA, AGG, CGA, CGC, CGG* and *CGT*], none of which end with *AT*, therefore it is not possible for *ATCGAT* to occur from positions 3-5. Coming to the next match, *ID* at positions 6-7, *ID* =$aasp(M^1)$. $head(M^1) = null$, $tail(M^1)$ is null, therefore *ID* at positions 6-7 can be added to the list of matches. The corresponding entry in P will be: $pBegin \leftarrow 6$, $pEnd \leftarrow 7$, list of codons, $L_C \leftarrow$ *ATC,GAT*.

Step3: Mapping to a directed acyclic graph: In this step, we find a non-conflicting subset of the occurrences in P that has the maximum over-all weight. This can be done by mapping the problem to that of finding the critical path in a directed acyclic graph and finding the nodes in the critical path, as follows:

1. Each occurrence in P is represented by a node v in a weighted directed acyclic graph $G = (V,E)$, where V and E denote the vertex and the directed edge set of G. The node weight w_i of a node v_i, is given by the weight of the corresponding occurrence, $P(i)$. i.e, $w_i \leftarrow P(i).weight$, $1 \leq i \leq |P|$, $w_i \in W$, where W is the weight vector corresponding to P.

2. Start with an empty edge list E. For each i, $1 \leq i \leq |P|$, and j, $i < j \leq |P|$, if $P(i).pEnd < P(j).pBegin$ or if $P(i)$ and $P(j)$ have the same codon encodings for all positions r, $P(j).pBegin \leq r \leq \min(P(i).pEnd, P(j).pEnd)$, then v_i, v_j will have an edge between them. i.e., $E \leftarrow E \cup (v_i,v_j)$.

Now, we can solve for the longest weighted path (critical path) in G, and insert the nodes in the longest path in a list L_v. The longest weighted path can be found by the critical path algorithm [7].

4.3 Translation of the amino acid sequence to a DNA sequence

Once we have a set of occurrences of motifs that do not conflict with each other, the next step is to do the actual mapping from the amino acid sequence to a codon (or DNA) sequence. At each position in S_A that is not part of any occurrence, we select the best codon for the corresponding amino acid. At positions in S_A that are part of an occurrence, we select the codon that is required for the occurrence.

Let L_v be the final list of occurrences.
1. **for** each $a_i \in S_A$, $1 \leq i \leq J$,
   $c_i \leftarrow$ best codon $(C(a_i))$
2. **for** all i, $1 \leq i \leq |L_v|$,
   for all j, $L_v(i).pBegin \leq j \leq L_v(i).pEnd$, /* j is between pBegin and pEnd for the occurrence $L_v(i)$ */
   $c_j \leftarrow L_v(i).L_c[j-pBegin]$ /*the codon sequence for the occurrence of $L_v(i)$ */

At the end of the second step, we will have the output codon sequence, $S_C = c_1c_2c_3\ldots\ldots c_J$. This can be converted into the output DNA sequence S_D by replacing each codon with the corresponding DNA triplet.

5. Complexity Analysis

Step1-Pre-processing: Enumerating the amino acid search patterns takes constant time for each motif (assuming that the upper bound for the length of each motif is a constant). There are k motifs, therefore, this step takes $O(k)$ time. Building the search tree is linear, therefore takes $O(k)$ time.

Step2- Enumeration of all potential occurrences: Finding the potential occurrences $O(L+3k+|P|)$, according to the Aho-Corasick Algorithm. The actual number of matches considered is greater than $|P|$, but still in $O(|P|)$. As the length of each motif is bound by a constant, so is the number of all possible codon combinations for each motif. Therefore, $|P|$ is in $O(L)$. Inserting an entry into P takes $log|P|$ time, as P is a sorted list. Therefore, the overall complexity for this step is $|P|log|P|$.

Step3-Mapping P to a directed acyclic graph (DAG):

a. Constructing the graph – involves checking the upper triangle of the adjacency matrix of the graph - takes $O(|P|.(|P|-1)/2) \sim O(|P|^2)$ time.

b. Finding the critical path takes $O(|V| + |E|)$ time. Here, $|V| \leftarrow |P|$. The graph is expected to be dense in general, therefore, $|E| \leftarrow |P|^2$. Therefore, this part takes $O(|P|^2)$ time.

As the number of *potential occurrences* found, $|P|$, is in $O(J)$, the overall complexity of the algorithm is $O(J^2)$.

6. Software

The software is available both as a console program, and as a windows application with a GUI. The program requires four input files: a file containing the input amino acid sequence, a file containing the codon bias information, and a file each for the desirable and undesirable motifs. The *desirability* values can be supplied in the input files. If desirability values are not supplied in the input files, a default value of *+1* is assigned for desirable motifs, and *–1* for undesirable motifs. Additionally, it is made sure that each occurrence of an undesirable motif has a *weight* less than zero, by setting it to a small –ve value if it is zero. The amino acid sequence file should be in the FASTA format, containing a single sequence. The output is in ASCII text format, listing different solutions with varying degrees of codon optimization. The GUI writes the formatted results to an html file. The sample out put of the program is shown in figure-3.

7. Results

We tested the validity of the program on different proteins. The results are shown in Table 2. These test cases were run with list of 33 desirable motifs and 5 undesirable motifs. Each motif was any where between 3 to 8 nucleotides long. In the sequences that were tested, all the occurrences of undesirable motifs could be eliminated. However, in some cases, it might be impossible to eliminate all occurrences of undesirable motifs.

Table 3 shows the average codon fitness, which is a measure of the degree of codon optimization. It can be observed that the average fitness could be improved by approximately 30% for all amino acid sequences tested.

Table 2: Results - CpG-S and CpG-N motifs

Amino acid sequence (length)	CpG-S motfs found	CpG-S motfs retained	CpG-N motifs found	CpG-N motifs eliminated
Tat (72)	80	7	66	66
C3d (302)	224	30	273	273
M23809 (226)	164	24	209	209

Table 3: Results – average fitness

Amino acid sequence (length)	Wild type	Maximizing CpG-S motifs	Minimizing CpG-N motifs	Both max. and min.
Tat (72)	0.671	0.948	0.981	0.930
C3d (302)	0.672	0.935	0.985	0.922
M23809 (226)	0.654	0.950	0.980	0.937

The analysis of one of these proteins, a 72 amino acid regulatory protein, Tat, of the Human Immunodeficiency Virus type-1 (HIV-1), is presented here. The Tat protein of HIV-1 plays critical role in regulating viral gene expression [15]. Additionally, Tat also modulated several functions of the host and a wide range of pathogenic properties has been ascribed to this viral protein. Considering the pathogenic significance of Tat, this viral antigen is being used as an important component of the viral vaccines, to elicit cell-mediated and humoral immune responses [31].

One important limitation of using the wild type Tat gene directly as a vaccine component without any modification is the sub-optimal codon content of the viral gene for mammalian immunization. Selective use of specific codons for protein translation is a characteristic feature of several species, a phenomenon called codon bias. Recent approaches of enhancing the immune responses against antigens have laid emphasis on protein translation efficiency, which is a function of the base composition of the pathogen. Codon usage of the Tat protein of HIV-1 is strikingly different from that of the human genes. The AT content of HIV-1 Tat is approximately 54% while that of the host human genome is about 35%, relatively low in AT content. Sub-optimal codon use could result in compromised protein translation

efficiency in a mammalian context and may reflect in the generation of a poor immune response against the AT-rich gene.

The characteristic features of codon use of the Tat gene in the virus are depicted in figure-2. The 72 amino acids encoding Tat exon-1 consisting of 219 bp represent the consensus sequence of the HIV-1 subtype-C strains. We used the 72 amino acid sequence (and the stop codon) as an input for optimizing the nucleotide sequence for mammalian expression. Alternatively, it is also possible to use the wild type nucleotide sequence of the gene directly obtained from the virus as the input. The results (the occurrences of desirable and undesirable motifs) were manually verified.

Using the software, we identified a total of 37 codons that could be replaced by the corresponding best codons to optimize the codon content of this gene to match that of the mammalian genome. Of note, optimization of the codons is fairly a simple task, provided the objective is only to replace the sub-optimal codons with the best ones, as is the case with several of the substitutions. For instance, serine at the position 16 in the viral gene is encoded by the codon AGT that has a fitness value of only 0.62, the lowest of all the 6 codons available for this amino acid (Figure 2). Substituting AGT at this position with AGC (fitness value 1.0) is likely to improve the average fitness of the viral gene for mammalian protein expression.

The algorithm identified seven potential motifs where CpG-S motifs could be engineered into the gene. One of these motifs, comprising of the amino acid residues S57 and A58 (AGC GCT), is naturally present in the viral gene (Figure 2). However, a possible conflict was identified between the possible codon optimization of these amino acid residues and not disrupting the CpG-S motif in this sequence. The codon GCT (fitness 0.67) encoding for alanine could have been substituted by the best codon GCC (fitness 1.0) for improved average fitness of the gene. Importantly, use of GCT in the gene design allowed recruiting a potential CpG-S island that could have far more beneficial effect on immunization potential of the DNA vaccine. The average fitness of this codon combination (AGC GCT) 0.835 ((1.0 + 0.67) / 2), as opposed to that of best codon combination 1.0, may not significantly modulate the protein translation efficiency.

The algorithm also identified a second site (L35 and V36) in the viral gene where substituting the original viral codons CTA GTT (average fitness 0.275) with the best codons CTG GTG (average fitness 1.0) would improve the overall fitness of the gene. In contrast, replacing the original codons with others, CTC GTA (fitness 0.36),

would permit recruiting a putative CpG-S motif into the gene, though at the expense of average fitness of the gene (Figure 2). At the moment, it is not clear how to resolve a conflict that may arise between optimizing the codon content of a gene and recruiting CpG-S motifs into the sequence. Most often, the codons useful in CpG motif design are not the ones with the highest fitness value thus leading to a potential disagreement between the diagonally juxtaposed requirements.

Optimization of Tat sequence finally resulted in a nucleotide sequence that was modified in a total of 37 codons (Figure 2). The AT content of the gene changed from 54% of the wildtype gene to 43% in the codon-optimized version thus resembling closely that of the mammalian context. The overall fitness value of the codon-optimized gene also improved from 0.67 to 0.93 suggesting a possible improvement at the level of protein translation.

Using the algorithm, we designed several variants of the Tat gene that differed from one another in the number of CpG islands engineered and the extent of codon content optimized. Experiments are presently underway to evaluate the antigenic potential of these Tat expression vectors in different mouse strains.

Acknowledgements

References

[1]   Aho A. and Corasick M: Efficient string matching: an aid to bibliographic search. Comm. CAN, 18:333-40,1975.

[2]   Bagarazzi ML, Boyer JD, Ayyavoo V, Weiner DB: Nucleic acid-based vaccines as an approach to immunization against human immunodeficiency virus type-1. Curr Top Microbiol Immunol 226:107-143 (1998).

[3]   Boyer JD, Cohen AD, Vogt S, Schumann K, Nath B, Ahn L, Lacy K, Bagarazzi ML, Higgins TJ, Baine Y, Ciccarelli RB, Ginsberg RS, MacGregor RR, Weiner DB: Vaccination of Seronegative Volunteers with a Human Immunodeficiency Virus Type 1 env/rev DNA Vaccine Induces Antigen-Specific Proliferation and Lymphocyte Production of beta-Chemokines. J Infect Dis 181:476-483 (2000).

[4]   Calarota S, Bratt G, Nordlund S, Hinkula J, Leandersson AC, Sandstrom E, Wahren B: Cellular cytotoxic response induced by DNA vaccination in HIV-1-infected patients. Lancet 351:1320-1325 (1998).

[5]   Chen Y, Lenert P, Weeratna R, McCluskie M, Wu T, Davis HL, Krieg AM: Identification of methylated CpG motifs as inhibitors of the immune stimulatory CpG motifs. Gene Ther 8:1024-1032 (2001).

[6]   Ciernik IF, Berzofsky JA, Carbone DP: Induction of cytotoxic T lymphocytes and antitumor immunity with DNA vaccines expressing single T cell epitopes. J Immunol 156:2369-2375 (1996).

[7]   Dolan A., Aldous J.: Networks and algorithms, John Wiley and Sons, New York, NY, 1993. pp. 366-386.

[8]   Egan MA, Charini WA, Kuroda MJ, Schmitz JE, Racz P, Tenner-Racz K, Manson K, Wyand M, Lifton MA, Nickerson CE, Fu T, Shiver JW, Letvin NL: Simian immunodeficiency virus (SIV) gag DNA-vaccinated rhesus monkeys develop secondary cytotoxic T-lymphocyte responses and control viral replication after pathogenic SIV infection. J Virol 74:7485-7495 (2000).

[9]   Gurunathan S, Klinman DM, Seder RA: DNA vaccines: immunology, application, and optimization*. Annu Rev Immunol 18:927-974 (2000).

[10] Gusfield D: Algorithms on Strings Trees and Sequences. Cambridge Univeristy Press, 1997.

[11] Hamajima K, Sasaki S, Fukushima J, Kaneko T, Xin KQ, Kudoh I, Okuda K: Intranasal administration of HIV-DNA vaccine formulated with a polymer, carboxymethylcellulose, augments mucosal antibody production and cell- mediated immune response. Clin Immunol Immunopathol 88:205-210 (1998).

[12] Ihata A, Watabe S, Sasaki S, Shirai A, Fukushima J, Hamajima K, Inoue J, Okuda K: Immunomodulatory effect of a plasmid expressing CD40 ligand on DNA vaccination against human immunodeficiency virus type-1 [In Process Citation]. Immunology 98:436-442 (1999).

[13] Ikemura T: Codon usage and tRNA content in unicellular and multicellular organisms. Mol Biol Evol 2:13-34 (1985).

[14] Jareborg N, Durbin R, 'Alfresco--a workbench for comparative genomic sequence analysis', Genome Res 2000 Aug;10(8):1148-57.

[15] Jeang,K.T., H.Xiao and E.A.Rich: Multifaceted activities of the HIV-1 transactivator of transcription, Tat. J Biol Chem Dev. 274, 28837-28840 (1999).

[16] Kim CH, Oh Y, Lee TH: Codon optimization for high-level expression of human erythropoietin (EPO) in mammalian cells. Gene 199:293-301 (1997).

[17] Klinman DM, Yamshchikov G, Ishigatsubo Y: Contribution of CpG motifs to the immunogenicity of DNA vaccines. J Immunol 158:3635-3639 (1997).

[18] Krieg AM: CpG motifs in bacterial DNA and their immune effects. Annu Rev Immunol 20:709-760 (2002).

[19] Krieg AM, Yi AK, Schorr J, Davis HL: The role of CpG dinucleotides in DNA vaccines. Trends Microbiol 6:23-27 (1998).

[20] Lee AH, Suh YS, Sung YC: DNA inoculations with HIV-1 recombinant genomes that express cytokine genes enhance HIV-1 specific immune responses [In Process Citation]. Vaccine 17:473-479 (1999).

[21] Lee MK, Lynch ED, King MC: SeqHelp: a program to analyze molecular sequences utilizing common computational resources. Genome Res 1998 Mar;8(3):306-12.

[22] Lin YL, Huang X, Jiang T, Chao KM: MAVG: locating non-overlapping maximum average segments in a given sequence. Bioinformatics 2003 Jan;19(1):151-152.

[23] Liu MA, McClements W, Ulmer JB, Shiver J, Donnelly J: Immunization of non-human primates with DNA vaccines. Vaccine 15:909-912 (1997).

[24] Lodmell DL, Ray NB, Ulrich JT, Ewalt LC: DNA vaccination of mice against rabies virus: effects of the route of vaccination and the adjuvant monophosphoryl lipid A (MPL(R)). Vaccine 18:1059-1066 (2000).

[25] MacGregor RR, Boyer JD, Ugen KE, Lacy KE, Gluckman SJ, Bagarazzi ML, Chattergoon MA, Baine Y, Higgins TJ, Ciccarelli RB, Coney LR, Ginsberg RS, Weiner DB: First human trial of a DNA-based vaccine for treatment of human immunodeficiency virus type 1 infection: safety and host response. J Infect Dis 178:92-100 (1998).

[26] Manthorpe M, Cornefert-Jensen F, Hartikka J, Felgner J, Rundell A, Margalith M, Dwarki V: Gene therapy by intramuscular injection of plasmid DNA: studies on firefly luciferase gene expression in mice. Hum Gene Ther 4:419-431 (1993).

[27] McInerney JO: GCUA: general codon usage analysis. Bioinformatics 1998;14(4):372-3.

[28] Nagata T, Uchijima M, Yoshida A, Kawashima M, Koide Y: Codon optimization effect on translational efficiency of DNA vaccine in mammalian cells: analysis of plasmid DNA encoding a CTL epitope derived from microorganisms. Biochem Biophys Res Commun 261:445-451 (1999).

[29] Ozinsky A, Underhill DM, Fontenot JD, Hajjar AM, Smith KD, Wilson CB, Schroeder L, Aderem A: The repertoire for pattern recognition of pathogens by the innate immune system is defined by cooperation between Toll-like receptors. Proc Natl Acad Sci U S A 97:13766-13771 (2000).

[30] Ponger L, Mouchiroud D: CpGProD: identifying CpG islands associated with transcription start sites in large genomic mammalian sequences. Bioinformatics 2002 Apr;18(4):631-3.

[31] Rusnati,M. and M.Presta: HIV-1 Tat protein: a target for the development of anti-AIDS therapies. Drugs Fut. Dev. 27, 481-493 (2002).

[32] Sasaki S, Fukushima J, Hamajima K, Ishii N, Tsuji T, Xin KQ, Mohri H, Okuda K: Adjuvant effect of Ubenimex on a DNA vaccine for HIV-1 [published erratum appears in Clin Exp Immunol 1998 May;112(2):354]. Clin Exp Immunol 111:30-35 (1998a).

[33] Sasaki S, Sumino K, Hamajima K, Fukushima J, Ishii N, Kawamoto S, Mohri H, Kensil CR, Okuda K: Induction of systemic and mucosal immune responses to human immunodeficiency virus type 1 by a DNA vaccine formulated with QS-21 saponin adjuvant via intramuscular and intranasal routes. J Virol 72:4931-4939 (1998b).

[34] Stratford R, Douce G, Zhang-Barber L, Fairweather N, Eskola J, Dougan G: Influence of codon usage on the immunogenicity of a DNA vaccine against tetanus. Vaccine 19:810-815 (2000).

[35] Tsuji T, Hamajima K, Ishii N, Aoki I, Fukushima J, Xin KQ, Kawamoto S, Sasaki S, Matsunaga K, Ishigatsubo Y, Tani K, Okubo T, Okuda K: Immunomodulatory effects of a plasmid expressing B7-2 on human immunodeficiency virus-1-specific cell-mediated immunity induced by a plasmid encoding the viral antigen. Eur J Immunol 27:782-787 (1997).

[36] Uchijima M, Yoshida A, Nagata T, Koide Y: Optimization of codon usage of plasmid DNA vaccine is required for the effective MHC class I-restricted T cell responses against an intracellular bacterium. J Immunol 161:5594-5599 (1998).

[37] Verschoor EJ, Mooij P, Oostermeijer H, van der Kolk M, ten Haaft P, Verstrepen B, Sun Y, Morein B, kerblom L, Fuller DH, Barnett SW, Heeney JL: Comparison of immunity generated by nucleic acid-, MF59-, and ISCOM- formulated human immunodeficiency virus type 1 vaccines in rhesus macaques: evidence for viral clearance [In Process Citation]. J Virol 73:3292-3300 (1999).

[38] Wang S, Liu X, Fisher K, Smith JG, Chen F, Tobery TW, Ulmer JB, Evans RK, Caulfield MJ: Enhanced type I immune response to a hepatitis B DNA vaccine by formulation with calcium- or aluminum phosphate [In Process Citation]. Vaccine 18:1227-1235 (2000).

[39] Wang TT, Cheng WC, Lee BH: A simple program to calculate codon bias index. Mol Biotechnol 1998 Oct;10(2):103-6.

[40] Wolff JA, Malone RW, Williams P, Chong W, Acsadi G, Jani A, Felgner PL: Direct gene transfer into mouse muscle in vivo. Science 247:1465-1468 (1990).

[41] zur MJ, Chen MC, Doe B, Schaefer M, Greer CE, Selby M, Otten GR, Barnett SW: Increased expression and immunogenicity of sequence-modified human immunodeficiency virus type 1 gag gene. J Virol 74:2628-2635 (2000).

	M[1]	E[2]	P[3]	V[4]	D[5]	P[6]	N[7]	L[8]	E[9]	P[10]	W[11]	N[12]
wt	ATG	GAG	CCA	GTA	GAT	CCT	AAC	CTA	GAG	CCC	TGG	CAT
co	---	---	---	---	---	---	---	--G	---	---	---	---

A

	H[13]	P[14]	G[15]	S[16]	Q[17]	P[18]	K[719]	T[10]	A[21]	C[22]	N[23]	N[24]
wt	CAT	CCA	GGA	AGT	CAG	CCT	AAA	ACT	GCT	TGC	AAC	AAC
co	--C	--C	---C	---C	---	--C	--G	--C	--C	---	---	---

B

	C[25]	Y[26]	C[27]	K[28]	H[29]	C[30]	S[31]	Y[32]	H[33]	C[34]	L[35]	V[36]
wt	TGT	TAT	TGT	AAA	CAC	TGT	AGC	TAC	CAT	TGT	CTA	GTT
co	--C	--C	--C	--G	---	--C	---	---	--C	--C	--G	--G

	C[37]	F[38]	Q[39]	T[40]	K[41]	G[42]	L[43]	G[44]	I[45]	S[46]	Y[47]	G[48]
wt	TGC	TTT	CAG	ACA	AAA	GGC	TTA	GGC	ATT	TCC	TAT	GGC
co	---	--C	---	--C	--G	---	C-G	---	--C	AG-	--C	---

C

	R[49]	K[50]	K[51]	R[52]	R[53]	Q[54]	R[55]	R[56]	S[57]	A[58]	P[59]	P[60]
wt	AGG	AAG	AAG	CGG	AGA	CAG	CGA	CGA	AGC	GCT	CCT	CCA
co	--A	---	---	A-A	---	---	A--	A--	---	---	---	---

	S[61]	S[62]	E[63]	D[64]	H[65]	Q[66]	N[67]	L[68]	I[69]	S[70]	K[71]	Q[72]
wt	AGT	AGT	GAG	GAT	CAT	CAA	AAT	CTT	ATA	TCG	AAG	CAA
co	--C	--C	---	--C	--C	---	---	---	---	--S	---	--G

Figure 1. The wild type Tat amino acid sequence

Codon	A				B		C	
	H[13]	P[14]	G[15]	S[16]	L[35]	V[36]	S[57]	A[58]
Wt	CAT (0.70)	CCA (0.83)	GGA (0.72)	AGT (0.62)	CTA (0.17)	GTT (0.38)	AGC (1.0)	GCT (0.67)
Best	CAC (1.0)	CCC (1.0)	GGC (1.0)	AGC (1.0)	CTG (1.0)	GTG (1.0)	AGC (1.0)	GCC (1.0)
CpG	-----	-----	-----	-----	CTC (0.48)	GTA (0.24)	AGC (1.0)	GCT (0.67)

Figure 2. Codon optimization at three different places in the Tat amino acid sequence

Maximizing the desirable Motifs:

AT content: 50% Avg Fitness: 0.967748%

CAC TTG ATT GTT ACC CCA TCG GGC TGC GGC GAA CAA AAT ATG ATT GGC ATG ACC CCA ACC

GTT ATT GCA GTT CAC TAT **CTC GAC** GAA ACC GAA CAA TGG GAA AAA TTC GGC TTG GAA AAA

CGC CAA GG**G GCG TT**G GAA TTG ATT AAA AAA GGC TAT ACC CAA CAA TT**G GCG TT**C CGC CAA

CCA TCG TC**G GCG TT**C GC**A GCG TT**C

Minimizing the Undesirable Motifs:

AT content: 55% Avg Fitness: 0.96046%

CAC TTG ATT GTT AC*C CCA TCA GGC T*GC *GGC GAA C*AA AAT AT*G ATT GGC A*TG AC*C CCA ACT*

*GTT AT*T *GCA GTT C*AC TAT TTG GAC GA*A ACT GAA C*AA TGG GAA AA*A TTT GGC T*TG GAA AA*A*

*CGC CAA GGC G*CA TTG GAA TTG ATT AA*A AAA GGC T*AT ACC CAA CAA TTG GCA TT*C CGT CA*A

*CCA T*CG *AGC GCA T*TC *GCA GCA T*TC

Maximizing Desirable and Minimizing Undesirable Motifs:

AT content: 53% Avg Fitness: 0.94195%

CAC TTG ATT GTT AC*C CCA TCA GGC T*GC *GGC GAA C*AA AAT AT*G ATT GGC A*TG AC*C CCA ACT*

*GTT AT*T *GCA GTT C*AC TAT **CTC GAC** GA*A ACT GAA C*AA TGG GAA AA*A TTT GGC T*TG GAA AA*A*

*CGC CAA GGC G*CA TTG GAA TTG ATT AA*A AAA GGC T*AT ACC CAA CAA TT**G GCG TT**C *CGT CA*A

*CCA T*CG TC**G GCG TT**C GC**A GCG TT**C

Figure 3. Sample output of the program for a part of C3d sequence (desirable occurrences are shown in bold and neutral occurrences are shown in italics)