Combinatorics of The Vaccine Design Problem: Definition and An Algorithm

Darko Kirovski, David Heckerman, and Nebojša Jojić
Microsoft Research

Contact: {darkok,heckerma,jojic}@microsoft.com

Technical Report MSR-TR-2007-148
November 2007

Microsoft Research
One Microsoft Way Redmond, WA 98052, USA
http://research.microsoft.com

Combinatorics of The Vaccine Design Problem: Definition and An Algorithm

Darko Kirovski, David Heckerman, and Nebojša Jojić
E-mail: darkok@microsoft.com

Microsoft Research

Abstract. Recently Nickle et al. introduced a new model of genetic diversity that summarizes a large input dataset into a short sequence containing overlapping subsequences from the dataset. This model has direct applications to rational vaccine design. In this paper we formally investigate the combinatorics of the vaccine optimization problem. Here the vaccine is constructed as a sequence S of amino-acids such that as many of the most frequently occurring epitopes found in mutated viruses are subsequences to S. We rigorously present the related design optimization problem, establish its complexity, and present a simple probabilistic algorithm to find an efficient solution. Our vaccine designs show improvement of over 20% in the coverage score over the previously best designs and produce over 15% shorter vaccines that achieve equivalent epitope coverage.

1 Introduction

Recent work in the rational design of HIV vaccines has turned to cocktail approaches with the intention of protecting against a set of variants of rapidly mutating viruses such as HIV [7]. One of the potential difficulties with this approach is vaccine size. Vaccines with a large number of nucleotides or amino-acids are difficult to deliver, expensive to manufacture, and more likely to cause autoimmune reactions.

Recently Nickle et al. introduced an approach for generating smaller vaccines that represent a wide genetic diversity [14]. The key to their approach is the use of a T-cell vaccine in which MHC-I epitopes (of length 8-11 amino-acids) *overlap*. This idea is illustrated in Figure 1. On the top of the figure is a list of MHC-I epitopes obtained from HIV strains found in a population of people. The vaccine candidate at the bottom of the figure covers each of these epitopes, exploiting their overlap. A color coding of the epitopes is used to highlight their overlap. The vaccine candidate is more than twice as short as one with no overlap. They call a vaccine candidate that exploits overlap an *epitome* as it epitomizes the many epitopes that went into its creation. Fisher et al. have recently proposed a similar strategy in [6].

In this paper we build upon their work and postulate the generic VACCINE DESIGN PROBLEM as a combinatorial optimization problem, demonstrate that it is NP-hard, and present an efficient algorithm that significantly surpasses previous designs in terms of epitope coverage and vaccine length. Finally, we argue in advantage of combinatorial search strategies for solving similar problems versus constructive heuristics guided by traditional machine learning and signal processing primitives as the latter do not take into account the randomness of the best design structures.

The optimization problem we address here is the discovery of a sequence S of amino-acids that covers the most MHC-I epitopes in a given set of viral sequences $\mathbf{P}$ from a population. We translate the biological problem into a computational one via the following definition of epitope coverage:

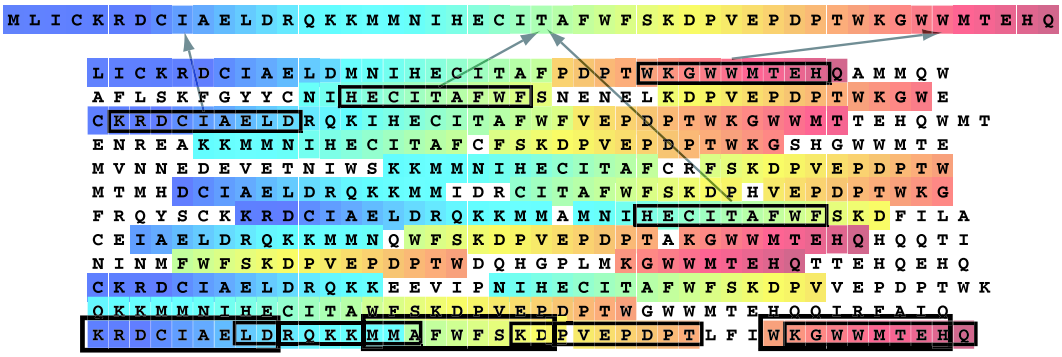

Fig. 1. A synthetic example of an epitome (top) and twelve amino-acid sequences (below) that it epitomizes in terms of their epitopes (select subsequences). A letter (amino-acid) is colored if and only if it is contained in at least one epitope. Boxes are used to show some epitopes and their mappings to the epitome. In the last sequence, *every* epitope is marked with a box to indicate that some epitopes overlap. The varying colors help to illustrate the mapping of the epitopes to the epitome.

Definition 1. ***Epitope coverage.*** *A sequence S is said to* cover *an epitope in a (single) viral sequence if the epitope is a substring of S.*[1]

Note that, if a viral sequence contains two copies of the same epitope, it can be covered only once. In contrast, if two viral sequences contain the same epitope, then the epitope can be covered in both sequences. Now, given $\mathbf{P}$, we can identify all the unique epitopes found in the set, and attach a frequency of occurrence f_i to each epitope i. Our optimization problem then becomes: given a set of epitope–frequency pairs, find a sequence S such that $\sum_{i \in S} f_i$ is a maximum.

This problem formulation makes several assumptions:

(1) we know what peptide sequences are MHC-I epitopes, i.e., what peptide sequences are presented with MHC-I molecules on the cell surface and trigger T-cell recognition;
(2) the infected cell processes the vaccine candidate so as to present every epitope on its surface; and
(3) a T-cell trained to recognize an epitope will only attach that epitope, i.e., there is no cross-reactivity. Jojić et al. show how these assumptions can be relaxed and still lead to the optimization problem we have just specified [8].

2 The Vaccine Design Problem

In this section we formally define the problem and evaluate its complexity. We start with a database $X = \{x_i, i = 1 \ldots N\}$ of N 10-amino-acids-long epitope sequences which appear in the strains of the target virus population.[2]

[1] Such a definition of the optimization problem can be augmented with additional models such as cross-reactivity, MHC binding affinity, etc.

[2] In practice, epitopes (good MHC binders) are not all known. We make a conservative assumption that every 10-mer is a potential epitope.

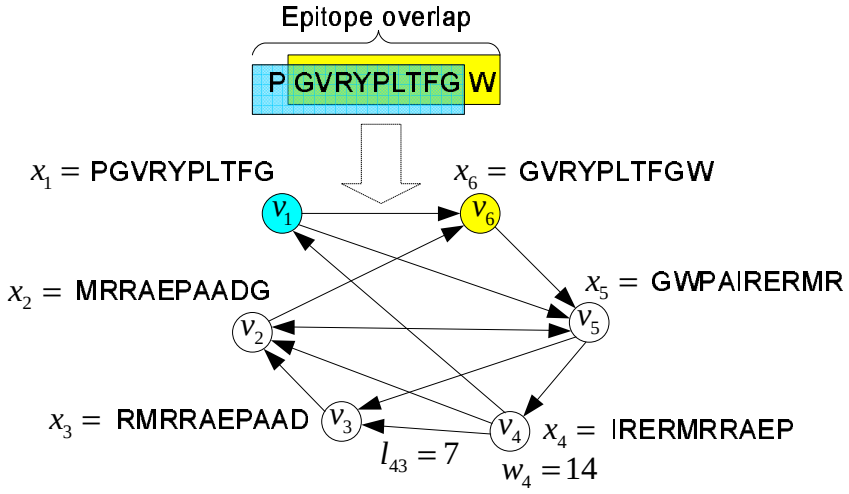

Fig. 2. An example of the epitope graph. Nodes represent epitopes, two nodes are connected by a vertex if they overlap.

2.1 The Model

Definition 2. ***Epitope Weight.*** *Each epitope x_i is weighted using a non-negative integer scalar $w_i \in \mathbb{Z}^*$ where w_i is proportional to the frequency of occurrence of x_i in the observed population of strains.*

Each epitope is denoted using a 10-symbol word $x_i \in \{\mathbb{A}\}^{10}$ where the symbols in $\mathbb{A}$ are drawn from the alphabet of 20 amino-acids.

We construct the weighted epitope overlap graph $\mathcal{G}(V, W, E, L)$ as follows. For each epitope x_i we create a vertex $v_i \in V$, where V is the set of vertices in $\mathcal{G}$. Consequently, we have $|V| = N$. Next, we connect two nodes v_i and v_j using a directed edge $e_{ij} = v_i \rightarrow v_j \in E$ if the corresponding x_i and x_j "overlap."

Definition 3. ***Epitope Overlap.*** *Epitopes $x_i = \{a_1 \dots a_{10}\}$ and $x_j = \{b_1 \dots b_{10}\}$ "overlap" if $\{a_{11-k} \dots a_{10}\} = \{b_1 \dots b_k\}$, where all $a_i, b_i \in \mathbb{A}$ and k is a positive non-zero integer $k \in \mathbb{Z}^+$.*

E denotes the set of all directed edges in $\mathcal{G}$. We consider only the maximum overlap for an ordered pair of epitopes. For example, epitopes x_1=PGVRYPLTFG and x_6=GVRYPLTFGW overlap at nine positions in a resulting sequence "PGVRYPLTFGW." We do not consider the inferior overlap "PGVRYPLTFGVRYPLTFGW." Also we do not honor the inverse overlap. For example, x_7=WGFTLPYRVG (amino-acids in x_3 are inversely ordered amino-acids of x_6) does not overlap with x_1 as the groove into which epitopes bind are not symmetric about its center.

Each vertex $v_i \in V$ is weighted with the corresponding w_i; we define $W = \{w_1, \dots, w_N\}$.

Finally, each directed edge e_{ij} is weighted with a non-negative integer $l_{ij} \in \mathbb{Z}^*$ that quantifies the depth of the overlap. From Definition 3, we derive the following definition:

Definition 4. ***Overlap Depth.*** *Overlap depth for two epitopes x_i and x_j represents the largest k for which x_i and x_j "overlap."*

In the example in Figure 2, we conclude that $l_{16} = 9$. The set of all edge weights is denoted as L. By default, two vertices v_i and v_j that are not connected have $l_{ij} = 0$.

2.2 Optimization Objective

We impose the paramount design objective using the following problem definition:[3]

PROBLEM: MAX-WEIGHT LENGTH-CONSTRAINED PATH (MLP)
INSTANCE: Graph $\mathcal{G}(V, W, E, L)$.
SOLUTION: A permutation $\pi : \{1 \dots M\} \rightarrow \{1 \dots M\}$ of a cardinality-M subset $S \subset V$ such that $10 + \sum_{i=1}^{M-1} \left[10 - l_{s(\pi(i))s(\pi(i+1))}\right] = K$, where $K = \text{const}$.
MEASURE: $\lambda = \sum_{i=1}^{M} w_{s(i)}$.

In the problem statement, M is a variable that depends upon the selected path $\pi(S) \in \mathcal{G}$ and K.

By setting this objective, we impose that from a given population of strains, using the overlap method we construct a vaccine of given length K amino-acids such that it maximizes λ, the number of epitopes that frequently occur in these strains. There is no biological proof that this criterion is optimal, but it is considered reasonable by several researchers [13, 14].

Theorem 1. MAX-WEIGHT LENGTH-CONSTRAINED PATH *is NP-hard.*

Proof. (sketch) We define a simple polynomial transformation $f() : \mathcal{G}(V, W, E, L) \rightarrow \mathcal{G}'(V, Z, E, L)$ such that for each node $v_i \in V$ it sets the corresponding $w_i = 1$ and for each edge $e \in E$ it introduces a constant edge weight $z(e) = 1 \in Z$. A polynomial time algorithm that finds an optimum solution to MLP on $f(\mathcal{G})$ would also solve the equal-edge-weight variant of the LONGEST WEIGHT-CONSTRAINED PATH problem on $\mathcal{G}'$. The latter problem has been proven to be NP-hard via the KNAPSACK problem in [1] (§2.4, pp.25).■

2.3 Discussion

In this subsection, we present a short discussion on several important assumptions exhibited in our model. First, the assumption that all 10-mers in X are possible epitopes is not realistic; thus, the number of possible epitopes could be reduced before applying the optimization algorithm. One way to address this issue in the problem model is to define weighted coverage over X where each k-mer is weighted by the probability that it is an epitope [8]. Interestingly, a model that would address this additional constraint, would, in concept, remain equivalent to the one presented in Subsection 2.1. As a result, a vertex weight w_i would be computed as a product of:

- the number of times the corresponding 10-mer x_i has occurred in $\mathbf{P}$, *and*
- the probability that x_i is an epitope. An algorithm that aims to quantify this probability has been introduced in [10].

Next, the presented model does not reflect the phenomenon of cellular processing of epitopes. Processing of a k-mer is thought to be mostly influenced by relatively small (5-10 amino-acids long) flanking regions on either side of the k-mer [17]. Thus, to handle the processing, one can

[3] The definition format is adopted from a comprehensive existing compendium of NP optimization problems [15]. It is straightforward to derive definitions according to alternate formats such as the Garey-Johnson format [16].

simply increase the size of the considered k-mers [18]. Similarly, we can model the effect of cross-reactivity by noting that a k-mer x_i in the vaccine will cover k-mer x_j if x_i and x_j are similar, where similarity is defined by some cross reactivity model (e.g., [9]). By introducing multi-k-mer weights over V, one would incorporate this additional constraint. In this case, by adding a single k-mer to the vaccine, the weights of all similar k-mers would be added to the overall measure λ. In a more complex model, the added weights of similar k-mers would be initially scaled proportional to the level of their similarity.

Because the above adjustments do not fundamentally alter the optimization algorithm, we adopt the model presented in Subsection 2.1 in the remainder of this article for both simplicity and brevity of presentation.

3 An Efficient Algorithm

The collected strain databases for the HIV virus certainly pose sufficient difficulty for exact solvers. Just as in the case of many computational biology problems, here also we are significantly more interested in the solution than in the algorithm. As a consequence, in our proposed algorithm we trade off speed for solution quality. To that extent we propose a simple least-constraining most-constrained probabilistic heuristic preceded by a constraint analyzer which aims at simplifying problem's search space.

3.1 Search Space Reduction

The first step in the developed algorithm is to preprocess the input epitopes in order to reduce the overall search space. The key idea is to merge two epitopes into a longer sequence if there exists a strong force between them to appear jointly in virus strains. We perform the reduction as follows. We first sort all epitopes in X in decreasing order of $g(x_i) = w_i/h(x_i)$. Function $h(x_i)$ returns the current length of the sequence x_i. Initially we have $(\forall x_i \in X)h(x_i) = 10$. We process the resulting sorted list of sequences starting from vertex x_i with the highest $g(x_i)$. Then, we find a group of vertices G such that $l_{ji} > \vartheta = \text{const.}$ for any $x_j \in G$ and then identify the sequence $x_j \in G$ with the largest $g(x_j)$. Next, we find a group of vertices G' such that $l_{jk} > \vartheta$ for any $x_k \in G'$ and then identify the sequence $x_k \in G'$ with the largest $g(x_k)$. If $x_i \equiv x_k$ then we merge x_i and x_j into a single epitope x_m of length $h(x_m) = h(x_i) + h(x_j) - l_{ij}$, remove x_i and x_j from X and insert x_m into X. We repeat this procedure until there exists a pair of vertices in X that could merge according to these requirements.

The constant ϑ is a threshold on the overlap. The purpose of this filter is to exclude merging nodes that have a shallow overlap in preprocessing – such vertices are connected in the search phase of the algorithm.

The reduction procedure is sub-optimal for arbitrary input. Its key objective is to attach epitopes that match well in terms of depth of overlap and frequency of occurrence in the observed strain population. Although examples where it performs sub-optimally could be constructed, in our experiments its benefits, primarily reduction of $|X|$ of 7% (from 860 to 800), were worthwhile considering the proximity of the obtained final solution to an optimistic upper bound.

3.2 A Simple MLP Solver

We developed a probabilistic least-constraining most-constrained algorithm to find the best vaccine design. The key idea was to generate random paths in $\mathcal{G}$ using a lottery-scheduling-based search

strategy [2] and a set of computationally inexpensive cost functions. The algorithm is detailed using the pseudo-code in Figure 3.

We use lottery scheduling (LS) as the fundamental selection process in the algorithm [2]. LS is a simple method of selecting an item x_i from a set of items X such that the probability of its selection is proportional to a certain normalized criterion function $\alpha(x_i)\left[\sum_{\forall x \in X}\alpha(x)\right]^{-1}$. The selection process can be done in $\mathcal{O}(\log_2|X|)$ via a simple binary tree. We represent this procedure using a function $\mathrm{LS}(X, \alpha())$ which returns a member of X.

The algorithm creates L distinct paths over $\mathcal{G}$ and chooses the one with the best total weight λ. We use a simple least-constraining most-constrained heuristic to construct each path. First, we select the starting node in the path $\mathit{\Pi} = \{v\} = \mathrm{LS}(V, \phi())$ where $\phi(v_i) \equiv \frac{w_i}{g(v_i)}$. Then, we concatenate iteratively new nodes to $\mathit{\Pi} = \{\pi_H, \ldots, \pi_T\}$ until the length of the resulting sequence is equal to or greater than K. Each new vertex is concatenated as follows. For both the head π_H and the tail π_T of the path we compute the concatenation candidates $v_H = \mathrm{LS}\ (V - \mathit{\Pi}, \varrho_H(\pi_H))$ and $v_T = \mathrm{LS}\ (V - \mathit{\Pi}, \varrho_T(\pi_T))$. Functions $\varrho_H()$ and $\varrho_T()$ are defined as:

$$\varrho_H(v, \pi_H) \equiv \frac{\max\limits_{v \in V - \mathit{\Pi}} y_{v\pi_H} w_v}{\left[g(v) - \max\limits_{v \in V - \mathit{\Pi}} y_{v\pi_H}]\right]\left[1 + \max\limits_{v \in V - \mathit{\Pi}} y^2_{\pi_H v}\right]} \tag{1}$$

$$\varrho_T(v, \pi_T) \equiv \frac{\max\limits_{v \in V - \mathit{\Pi}} y_{\pi_T v} w_v}{\left[g(v) - \max\limits_{v \in V - \mathit{\Pi}} y_{\pi_T v}]\right]\left[1 + \max\limits_{v \in V - \mathit{\Pi}} y^2_{v\pi_T}\right]}. \tag{2}$$

The $\varrho()$-functions quantify heuristically how attracted two vertices are. The most constrained vertices in the current remainder of nodes $V - \mathit{\Pi}$ with high overlap at the head or the tail of $\mathit{\Pi}$ as well as high weight tend to increase the output of $\varrho()$. On the other hand, the cost function is relaxed if the candidate vertex has a high overlap with a vertex in $V - \mathit{\Pi}$ (see second term in the denominator). Thus, we enforce that less constraining head/tail is chosen while concatenating candidate vertices. Once candidate vertices v_H and v_T are identified we determine which one will be appended to $\mathit{\Pi}$ using another round of lottery scheduling that uses the corresponding $\varrho()^3$ function to establish the probability of occurrence. We constructed our search algorithm with the aim to rapidly produce new candidate paths which have a high likelihood of producing high λ. Thus, simple iterative search can be used to produce the final resulting path $\pi(S)$.

4 Experiments

In this section, we evaluate candidate vaccines constructed using our approach under various assumptions. We compare the optimization score of our vaccine candidates with those of other designs and an optimistic upper bound. The data we use are a set of 197 clade B HIV sequences taken from GenBank (numbers available on request). Each HIV sequence in the dataset was obtained from a different person.

In the first experiment, we assume that all 10-mers from each HIV sequence are epitopes. Under these conditions, the optimal vaccine is one that maximizes the coverage of all 10-mers found in the virus population. In Figure 4 (left), we plot this coverage of the HIV-1 gag region as a function of vaccine length for the epitome. There are three sets of results provided: one returned by the greedy epitome design approach [8], results obtained using our MLP algorithm, and an optimistic upper bound. We computed the upper bound by assuming that $(\forall i \in V)(\forall j \in V, j \neq i) y_{ij} = 9$

A Simple MLP SOLVER
Input: $\mathcal{G}$, number of search iterations L.

1 **while** $L > 0$
2 Set path $\Pi = \text{LS}(V, \phi())$.
3 **while** $\sum_{x \in \Pi} g(x) < K$
4 π_H and π_T are the head and the tail of Π.
5 Head-candidate $v_H = \text{LS}\ (V - \Pi, \varrho_H(\pi_H))$.
6 Tail-candidate $v_T = \text{LS}\ (V - \Pi, \varrho_T(\pi_T))$.
7 Add-on $a = \text{LS}\ (\{v_H, v_T\}, \{\varrho_H(v_H, \pi_H)^3, \varrho_T(v_T, \pi_T)^3\})$.
8 **if** $a = v_H$ **then** $\Pi = \{v_H, \Pi\}$
9 **else** $\Pi = \{\Pi, v_T\}$.
10 **if** $\lambda = \sum_{\forall v_i \in \Pi} w_i > \lambda_{max}$
11 **then** current best path $\Pi_{max} = \Pi$, set $\lambda_{max} = \lambda$.
12 L=L-1.

Lottery Scheduling LS
Input: Set X, objective function $\alpha() : \{x \in X\} \rightarrow \mathbb{R}$.

1 Compute $(\forall x_i \in X) a_i = \alpha(x_i)$
2 Generate random number r within $[0, \sum_{i=1}^{|X|} a_i]$.
3 Find j such that $\sum_{i=1}^{j} a_i \leq r < \sum_{i=1}^{j+1} a_i$.
4 **return** x_i.

Fig. 3. Pseudo-code of the developed MLP Solver.

and then taking $\lambda^* = \sum_{v_i \in \Pi^*} w_i$, where Π^* is a path created in descending order of weights in $\mathcal{G}$. Clearly this upper bound is not likely to be reached in a real-life solution as the maximum depth of coverage (i.e., 9) between two epitopes can be achieved only for at most 20 other epitopes in $\mathcal{G}$. We compute the improvement of the developed MLP solver against [8] by reporting $[\lambda(GE) - \lambda(MLP)]\,[\lambda(GE) - \lambda^*]^{-1}$, where index GE denotes results produced by the reference greedy algorithm for epitome construction [8]. For example, for vaccine length $K = 618$ we obtained a 25.3% improvement with respect to the greedy epitomes. We do not comment on the optimality of the obtained solutions as we did not use an exact solver for either of the K-spots due to the involved problem complexity. Another way to report results is to compare the vaccine lengths obtained using the two methods with identical coverage. In order to achieve $\lambda \left[\sum_{v_i \in V} w_i\right]^{-1} = 0.8821$, we have $K_{MLP} = 618$ and $K_{GE} = 711$ amino-acids, an improvement of 15% over the existing method. It is important to notice that the improvements are more significant with the increase in vaccine size.

Finally, the problem representation proposed in this experiment is independent of flanking regions and assumes no cross-reactivity. Jojić et al. have demonstrated problem definition adjustments to address these constraints [8].

From the same experiment, we report the progress of our algorithm as the number of iteration increase. Figure 4 (right) illustrates the improvement in the best found result for $K = 618$ as L increases compared to the result obtained by a single iteration of the greedy algorithm. We ran the MLP solver for $L \sim 10^8$ iterations and recorded the moments at which the best results were computed. We can observe that the greedy result is achieved within one second of run-time. The 3.2GHz Pentium machine we used in the experiment, produced ≈ 200paths/sec.

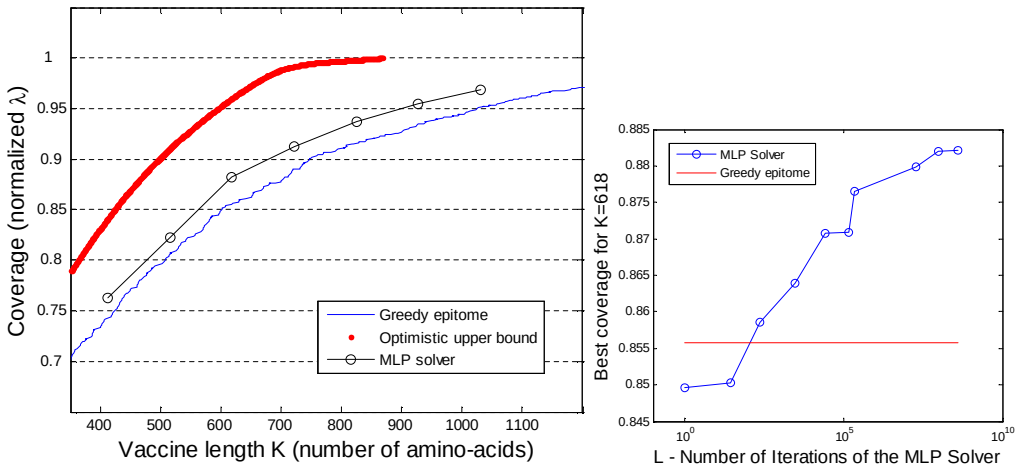

Fig. 4. (left) Coverage of 10-mers from 197 clade B HIV protein sequences by vaccines of various lengths and types: greedy epitome, results obtained by our MLP solver, and an optimistic upper bound. Note that the MLP solver performed $L \sim 10^8$ iterations only during the search for the best $K = 618$ amino-acids vaccine. In all other tests, it performed $L \sim 10^6$ iterations. (right) A single run of the MLP solver for $L \sim 10^8$ iterations. We recorded the moments at which the best results were computed. The result obtained by the greedy epitome algorithm is also illustrated.

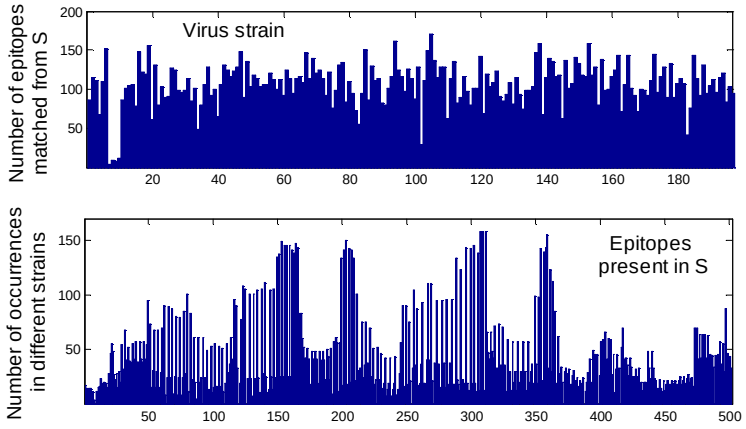

Fig. 5. Two diagrams describing the efficacy of the new vaccine design. We singled out the best $K = 618$ amino-acids long vaccine design S with normalized epitope coverage of $\lambda = 0.8821$ and plotted: (top) how many epitopes from the final vaccine design S appear in the 197 HIV strains collected from distinct individuals and (bottom) the number of occurrences of a given epitope in distinct virus strains.

Finally, we plot two histograms in Figure 5. The top diagram identifies how many epitopes from the final vaccine design S (with $K = 618$ amino-acids and normalized epitope coverage of $\lambda = 0.8821$) appear in the 197 HIV strains collected from distinct individuals. One can observe that all strains are covered by S which points to the efficacy of the new vaccine design methodology. Note that there exist four strains in the databank (with indices 7–10) for which most of their genotype is still not uncovered – understandably we recorded poor coverage on these strains. Finally, most of the other strains have more than 30 containing epitopes present in the target vaccine design. A potential design improvement from this perspective could be achieved by readjusting the optimization goal to provide maximum-minimum coverage of distinct strains. We plan to address this issue in future work. The bottom plot identifies the number of occurrences of a given epitope in distinct virus strains. Again we used the best $K = 618$ amino-acids vaccine design to present the data. The resulting vaccine covered 501 out of 860 identified epitopes. From the diagram one can note that several dozens of epitopes appeared in the vast majority of individual strains.

5 Conclusions and Future Work

We have formally defined the optimization goal behind a promising vaccine design approach that exploits overlap in MHC-I epitopes to create vaccines that cover a large fraction of a viral diversity. We have shown that the problem of finding optimally compact vaccines can be modeled as the MAXIMUM-WEIGHT LENGTH-CONSTRAINED PATH problem, we have proven that this problem is computationally intractable, and introduced an efficient algorithm to find near-optimal vaccine candidates. By applying our MLP solver to the GenBank dataset, we demonstrate quantitatively that our technique produces vaccine candidates with significantly larger coverage of potential epitopes than previous methods that include a greedy heuristic with a similar design objective and alternative approaches based on cocktails of observed strains, cocktails of consensus strains, or cocktails of tree centers [8]. With respect to [8], we obtained improvements in coverage in excess of 20% for equivalent vaccine length and 15% shorter vaccine designs for vaccines of equivalent coverage.

The vaccine design is a flexible representation of HIV (and other pathogen) diversity and can accommodate several extensions. Our design model can be adjusted to include additional constraints that pertain to the expressiveness of epitopes in a vaccine strain. To that extent several adjustments could be readily included such as a cross-reactivity submodel [9], a model that quantifies the uncertainty about whether a peptide sequence is an MHC-I epitope [10], a model that accounts for the influence of flanking regions on epitope presentation, a model that associates viral mutations with individuals' HLA types [11], and physics-based T-cell binding models. As another example, problems with immunodominance may be attenuated by delivering components of a cocktail in different vectors [12]. The epitome can be optimized for this format. As yet another example, blocking virtually all evolutionary pathways in a protein segment may prove more effective than blocking many but not all pathways in a full protein. Creating vaccines that achieve such full blockage can be constructed by directing the algorithm to concentrate on a particular segment of the protein.

Finally, we argue that combinatorial optimization techniques excel in problem statements such as the Vaccine Design Problem because of their ability to explore search spaces efficiently. As optimal designs in such search spaces often have certain degree of randomness associated with their structure, greedy heuristics guided by traditional signal processing and machine learning algorithms are typically unable to find such structures.

References and Notes

1. A. Isto. Interactive Knapsacks: Theory and Applications. Ph.D. dissertation, Tietojenksittelytieteiden laitos, Acta Electronica Universitatis Tamperensis; A-2002-13, 2002.
2. C.A. Waldspurger and W.E. Weihl. Lottery Scheduling: Flexible Proportional-Share Resource Management. USENIX Symposium on Operating Systems Design and Implementation, 1994.
3. S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by Simulated Annealing. Science, Vol.220, no.4598, pp.671–680, 1983.
4. F. Glover and M. Laguna. Tabu Search. Kluwer, 1997.
5. N.A. Barricelli. Esempi numerici di processi di evoluzione. Methodos, pp.45–68, 1954.
6. W. Fischer, et al. Polyvalent vaccines for optimal coverage of potential T-cell epitopes in global HIV-1 variants. Nature Medicine, Vol.13, (no.1), pp.100–6, 2007.
7. A. McMichael and T. Hanke. The quest for an AIDS vaccine: Is the CD8+ T-cell approach feasible? Nature Reviews Immunology, Vol.2, pp.283–291, 2002.
8. N. Jojić et al. Using epitomes to model genetic diversity: Rational design of HIV vaccine. Advances in Neural Information Processing Systems, MIT Press, Cambridge MA, 2005.
9. D. McKinney et al. Recognition of variant HIV-1 epitopes from diverse viral subtypes by vaccine induced CTL. Journal of Immunology, Vol.173, pp.1941–1950, 2004.
10. D. Heckerman et al. Leveraging information across HLA alleles/supertypes improves epitope prediction. RECOMB, 2006.
11. C. Moore et al. Evidence of HIV-1 adaptation to HLA-restricted immune responses at a population level. Science, Vol.296, pp.1439–1443, 2002.
12. F. Rodriguez et al. Immunodominance in virus-induced CD8(+) T-cell responses is dramatically modified by DNA immunization and is regulated by gamma interferon. Journal of Virology, Vol.76, no.9, pp.4251–4259, 2002.
13. A. DeGroot et al. HIV vaccine development by computer assisted design: The GAIA vaccine. Vaccine, Vol.23, pp.2136–2148, 2005.
14. D.C. Nickle, et al. Coping with viral diversity in HIV vaccine design. To appear in PLoS Computational biology, 2007.
15. P. Crescenzi and V. Kann. A compendium of NP optimization problems. Available on-line at: `http://www.nada.kth.se/~viggo/problemlist`.
16. M.R. Garey and D.S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman & Co., NY, 1979.
17. R. Draenert, et al. Immune selection for altered antigen processing leads to cytotoxic T-lymphocyte escape in chronic HIV-1 infection. Journal of Experimental Medicine, Vol.199, pp.905–15, 2004.
18. A. Milicic, et al. CD8+ T-cell epitope-flanking mutations disrupt proteasomal processing of HIV-1 Nef. Journal of Immunology, Vol.175, pp.4618–26, 2005.