

Combinatorial Optimization in Rapidly Mutating Drug-Resistant Viruses

RICHARD H. LATHROP AND MICHAEL J. PAZZANI

{ rickl | pazzani }@uci.edu

*Department of Information and Computer Science,
University of California, Irvine, CA 92717 USA*

Editor:

Abstract. Resistance to chemicals is a common current problem in many pests and pathogens that formerly were controlled by chemicals. An extreme case occurs in rapidly mutating viruses such as Human Immunodeficiency Virus (HIV), where the emergence of selective drug resistance within an individual patient may become an important factor in treatment choice. The HIV patient subpopulation that already has experienced at least one treatment failure due to drug resistance is considered more challenging to treat because the treatment options have been reduced. A triply nested combinatorial optimization problem occurs in computational attempts to optimize HIV patient treatment protocol (drug regimen) with respect to drug resistance, given a set of HIV genetic sequences from the patient. In this paper the optimization problem is characterized, and the objects involved are represented computationally. An implemented branch-and-bound algorithm that computes a solution to the problem is described and proved correct. Data shown includes empirical timing results on representative patient data, example clinical output, and summary statistics from an initial small-scale human clinical trial.

Keywords: HIV, retrovirus, virus, clinical treatment, drug resistance, mutants, mutations, artificial intelligence, expert system

1. INTRODUCTION

Evolution consists of selective pressure acting on random sequence mutations that occur in the information bearing molecules of an organism. Within a population, individuals that thereby acquire a selective advantage will tend to propagate the advantageous changes to their progeny. Thus the population's gene pool will tend to shift over time to accommodate the selective pressure. Over geological time-scales, this evolutionary force has produced the great diversity of creatures, present and extinct, ever living on the face of the Earth.

The very long time-scales that generally characterize evolution may be shortened dramatically by human intervention. When human activity imposes a selective pressure on a population's gene pool, the survivors quickly adapt to resist the pressure. The example that concerns this paper, the long use of chemicals to control pests, has resulted in the emergence of wide-spread resistance to these chemicals in the pests they formerly controlled. Today we see newly emerged resistance to these chemicals at all levels of the pest phylogeny: rodents resist or learn to avoid poison; weeds tolerate herbicides; insects survive insecticides; parasites such as malaria resist anti-malarials; fungi persist in the presence of fungicides; bacteria resist antibiotics; and viruses thrive in spite of antivirals. The short time-scales involved in such adaptive or selective resistance have surprised many who believed that these pests had been controlled once and for all.

Many drug-resistant strains of important human pathogens have emerged and become growing medical problems. Bacteria and viruses typically have large populations, short generation times, and high mutation rates. Resistance in these pathogens can arise for a variety of reasons. For example, one species of bacteria has evolved a specialized protein that cleaves penicillin. Other bacteria have evolved specialized pumps that transfer antibiotics from inside to outside the cell. Bacteria may exchange copies of drug-resistance conferring DNA with each other as plasmids during conjugation. This paper concerns selective drug resistance in rapidly mutating viruses such as Human Immunodeficiency Virus (HIV). The viral genome is much simpler than the bacterial genome, and accordingly the mechanisms of viral drug resistance treated in this paper are much simpler than the complex bacterial mechanisms. This does not necessarily imply that viruses are easier to treat, however. The very simplicity of the viral genome means that there are fewer points of attack open to pharmaceutical agents. For example, the retrovirus HIV, the subject of this paper, has only three enzymes. Of these, only two are currently the target of U.S. Food and Drug Administration (FDA) approved antiviral drugs.

1.1. HIV Background

The treatment in this section is adapted from reference (Lathrop et al., 1998), which it follows. Human immunodeficiency virus causes progressive deterioration of the immune system leading almost invariably to AIDS and death from opportunistic cancers and infections. The information content of an HIV virus is contained in a set of genes encoded in its genome. Each gene is a sequence of bases or nucleotides of four varieties. A gene may be represented as a string over an alphabet of four characters, one character representing each nucleotide. The HIV genome ultimately causes the production of gene products, often proteins, important in the virus life cycle. A protein is a sequence of amino acids of twenty varieties, and may be represented as a string over an alphabet of twenty characters. Each amino acid in the protein is encoded by a block of three adjacent nucleotides in the genome, called a codon. The two proteins targeted by current FDA-approved drugs are called “reverse transcriptase” (RT) and “protease” (PRO).

The genome string must be copied from one generation to the next during the virus life cycle. Copying errors occur frequently, and are called mutations. Mutations can change the structure or function of the virus, and thus alter how it interacts with its environment. Mutant strains with genome sequences very similar to the patient’s current strain (close in Hamming or edit distance) appear spontaneously and continuously. In a full-blown case of AIDS, it is estimated that every single point mutation appears every day, every coordinated pair of point mutations appears once or more during the course of the infection, and even coordinated triples of point mutations may appear on occasion (Condra et al., 1995). The high rate of HIV viral mutation both makes development of a vaccine difficult and results in rapid positive selection for drug resistant mutant strains.

A drug typically works by blocking a key part of the virus life cycle. A drug resistant mutation occurs when a copying error in the viral genome so alters the virus that it can perform the targeted step of its life cycle even in the presence of the drug. In the continued presence of the drug the mutant strain may out-compete the dominant strain, and thereby may itself become the dominant stain in the patient. This is often called selec-

tive drug resistance, because the resistant mutant is selected for by the drug's presence. If unrecognized, the current treatment may lose its effect and the patient's condition may deteriorate. The resulting strain is more challenging to treat because the treatment options have been reduced. If the drug treatment is changed in response, the potential is present for a new drug resistant mutation to develop. The use of an increasing variety of drugs has led to virus strains increasingly resistant to multiple drugs simultaneously. Sadly, the increasing prevalence of drug resistant strains in the HIV global gene pool means that new patients may be infected by mutant strains that already have accrued resistance from previous hosts (Gu et al., 1994). For many reasons, it is important to avoid selecting for drug resistant mutants.

Combination treatments involving multiple drugs are one approach to avoiding drug resistance (Lange, 1995). If the virus mutates to resist one drug but still is inhibited by another, it may be suppressed or unviable. In this case the mutation may not be positively selected for. Combination treatments may contain up to four simultaneous drugs, but usually do not exceed three due to the potential for intolerable side-effects and toxicity. Combinations containing at least one protease inhibitor are referred to as Highly Active Anti-Retroviral Therapy (HAART). HAART typically results in a dramatic drop in viral load within two weeks, often sustained for long periods of time, but may fail eventually due to the development of drug resistance (Carpenter et al., 1996). Mutations still can occur under HAART, though the mutation rate is greatly decreased (Jacobsen et al., 1996).

Knowledge of the current or nearby mutants that are putatively resistant to one or more drugs may be valuable to a physician treating an HIV patient. In conjunction with HAART, such knowledge may help select a combination of drugs less likely to be resisted. A human physician may find it tedious to scan many long genetic sequences, be unfamiliar with the latest HIV drug resistant mutations reported, quickly tire of envisioning possible mutants, or have difficulty ranking the hundreds of treatment choices for each patient. On the other hand, the problem size and scope are quite reasonable for an automated system.

1.2. A Physician's Advisor for HIV Drug Resistance

Previously we have described a rule-based artificial intelligence (AI) computer system intended to help an attending physician avoid HIV drug resistance in clinical treatment of individual HIV patients (Pazzani et al., 1997a, Pazzani et al., 1997b, Lathrop et al., 1998, Cimoch et al., 1998). The system, named CTSHIV (Customized Treatment Strategy for HIV), was designed primarily to assist in treatment of the subpopulation of patients that already have experienced an HIV treatment failure due to the emergence of drug resistance. These patients are considered more challenging to treat because their treatment options have been reduced. The system connects the scientific AIDS literature describing specific HIV drug resistances directly to the Customized Treatment Strategy of a specific HIV patient, thus mediating between the scientific literature and the patient's current infection. This is done by compiling rules from the scientific and medical literatures that relate genotypic sequence mutations to viral drug resistance. Rules are applied to HIV genetic sequences extracted from the patient and used to design a customized treatment strategy that attempts to avoid selection for known drug-resistant mutants. The hoped-for result is more precise

treatment of individual HIV patients, and a decreased tendency to select for drug resistant genes in the global HIV gene pool.

Knowledge about HIV drug resistant mutations is contained in a set of rules (currently numbering 55) extracted from the scientific and medical literature. Rules are of the form:

IF \langle antecedent \rangle THEN \langle consequent \rangle WITH \langle weight \rangle (references).

For example, one such rule in CTSHIV is:

IF the value of RT codon number 151 is ATG
(= it encodes methionine),
THEN infer resistance to AZT, ddI, d4T, and ddC
WITH weight = 1.0
(Iversen, et al., 1996)

Rules may mention several different codon numbers and values, and higher weighted rules (usually the more specific rules) supersede lower weighted rules (usually the more general) in cases where two or more rules apply. The weight associated with a rule is not a confidence as in many expert systems (Pazzani et al., 1997b, Jackson, 1990). Rather, it reflects the estimated level of resistance to a particular drug. Weights range from 0.1 (low resistance) to 1.0 (high resistance) based upon expert advice and the level of resistance reported in the literature.

The overall information processing architecture is diagrammed in Figure 1. Rules in the CTSHIV knowledge base encode knowledge about sequence mutations in the HIV genome that have been found to result in drug resistance in the HIV virus. Rules are applied to the actual HIV sequences of the virus strains infecting the specific patient undergoing clinical treatment in order to identify known drug resistant HIV mutant strains that already exist in the patient. A rule-directed search through mutation sequence space identifies nearby drug resistant mutant strains that are likely to be positively selected for by certain treatments. The possible combination drug treatment regimens currently approved by the FDA are considered and ranked by their estimated ability to avoid identified current and nearby drug resistant mutants. The highest-ranked treatments are suggested to the attending physician.

1.3. Related Work

This work rests upon a central foundational pillar of artificial intelligence: rule-based expert systems that are instantiated in the medical domain (for example, (Buchanan and Shortliffe, 1984)). For many such systems, a common diagnosis task is to identify the organism, from which treatment follows straightforwardly. Here, the organism is known to be HIV, but the treatment task is complicated by selective drug resistance.

Several AI applications have targetted HIV. An expert system based on experimental data from HIV patients (immunologic markers) has been used to diagnose opportunistic non-Hodgkin's lymphomas that may develop (Diamond et al., 1994). Knowledge-based systems have been applied to HIV patient medical record systems (Musen et al., 1995, Safran et al., 1996), monitoring of HIV patient protocols (Musen et al., 1996, Tu et al., 1995, Sonnenberg et al., 1994, Sobesky et al., 1994), and HIV patient assessment (Xu, 1996, Ohno-Machado et al.,

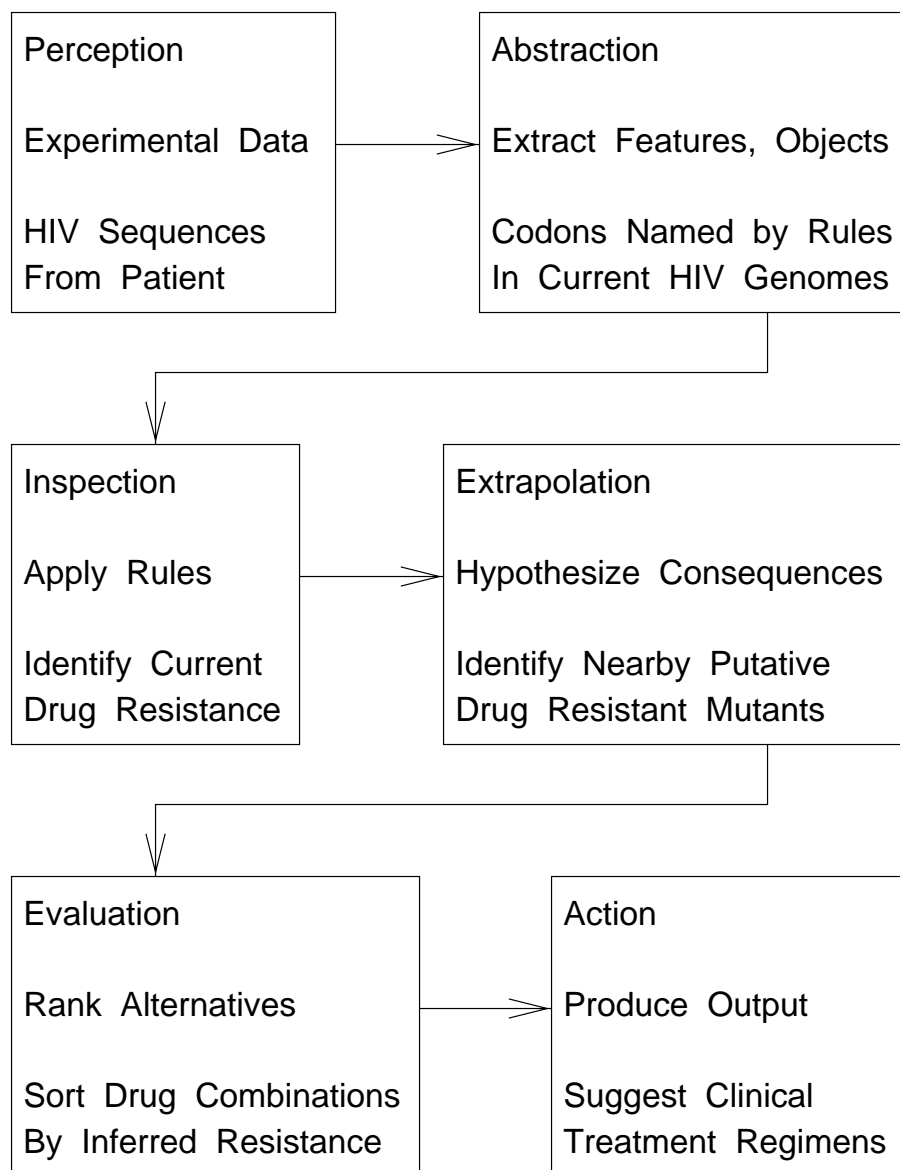


Figure 1. Application overview flowchart (after (Lathrop et al., 1998)).

Less closely related are knowledge-based systems that apply qualitative modeling and process simulation to HIV laboratory systems (Sieburg, 1994, Ruggiero et al., 1994). A quasispecies equation has been used to show that the pretreatment frequency of HIV resistant virus depends on the number of point mutations between wild-type and mutant virus, the selective disadvantage of the resistant mutant and intermediate mutants, and the mutation rate (Ribeiro et al., 1998). Artificial life techniques were used to study the interaction between the design of effective drugs and HIV-1 protease resistance mutations, and to characterize general features of inhibitors that are effective in overcoming resistance (Rosin et al., 1998). To our knowledge the CTSHIV system is the first to use HIV sequence data from HIV patients to estimate current and nearby drug resistant mutants and suggest treatment combinations to avoid both.

1.4. *The Underlying Combinatorial Optimization Problem*

Of interest to this paper is the underlying combinatorial optimization problem that the system confronts: to rank order the available combination treatments according to some heuristic ranking function f reflecting inferred (or putative) drug resistance in the patient's current or nearby virtual mutant viral sequences. This gives rise to a triply nested combinatorial optimization problem: find the minimum over drug combinations of the maximum over nearby mutants of the minimum resistance to drugs in the combination.

1. For a given patient, identify the drug combinations that most strongly suppress a population of mutants centered on the patient's current viral strains;
2. For a given patient and drug combination, do this by identifying the most-resistant mutant in the population; and
3. For a given patient, drug combination, and mutant, do this by identifying the least-resisted drug.

2. METHODS

This section develops notation and formalizes the problem. Sets and tuples are denoted by capital italic, sets of sets by capital script.

2.1. *FDA-approved Combination Treatments*

A treatment regimen D is a set or combination of FDA-approved drugs that might be prescribed by a physician treating an AIDS patient in the clinic. Sometimes it is necessary to eliminate some drugs entirely, e.g., if the patient cannot tolerate certain side-effects. Some drugs should not be given together, so it is also necessary to eliminate some otherwise possible drug combinations. Let \mathcal{T}_{FDA} be the set of permissible treatment regimens for a given patient, and let \mathcal{T}_{FDA}^k be those members of \mathcal{T}_{FDA} that contain exactly k drugs.

2.2. Sequences, Genes, and Genomes

Let s be any DNA sequence, $s[i]$ be the i -th character of s , S be any set of DNA sequences, and S_0 be specifically the original input set of pre-aligned sequences cloned from the viral strains currently infecting the patient. The sequence alphabet is $\{A, C, G, T, *\}$, where the letters represent DNA bases, and “*” represents any base and collects all degeneracies. This yields $5^3 = 125$ possible codon values, and consequently a codon may be represented by a seven-bit byte. The function *codons* maps a string representing DNA to its codons, and assumes a reading frame inherited by pre-alignment to a reference string (currently HXB2 (Fisher et al., 1985)).

Let G_{FDA} be the set of genes targeted by FDA-approved drugs, currently protease (PRO) and reverse transcriptase (RT). For $s \in S$ and gene $g \in G_{FDA}$, let *genestring*(g, s) return the substring of s that encodes g . A virtual viral genome h assembled from a set of input sequences S is defined to be any concatenation of sequences *genestring*(g, s) extracted from S such that (1) every gene $g \in G_{FDA}$ is present exactly once and in order, and (2) if any gene extracted from sequence $s \in S$ is present, all genes extracted from s are present. This allows all virtual genomes that are consistent with S .

2.3. Rules

Let R be the current set of rules. For rule $r \in R$, let *drugs*(r) be the set of drugs mentioned by r . Let *gene*(r) $\in G_{FDA}$ be the gene targetted by r . For treatment regimen $D \in \mathcal{T}_{FDA}$, let *rules*(D) be the set of rules that mention a drug in D .

Each rule r has a weight *wt*(r) and an antecedent *ant*(r). Each weight *wt*(r) is a floating point number that represents an heuristic estimate of resistance, with 0.0 indicating no resistance (e.g., wild type), 1.0 indicating very strong resistance, and intermediate values indicating intermediate degrees of resistance. Each antecedent *ant*(r) is a set of conditions, all of which must be satisfied for the rule r to fire. Each antecedent condition $A \in \text{ant}(r)$ is a pair $A = \langle i, C_i \rangle$, where i is a codon index in *gene*(r) and C_i is the set of codon values that would satisfy condition A if found at codon index i there. C_i is usually the set of all codons that encode the same amino acid residues as the ones mentioned in the literature paper cited by the rule r , because the current drugs function mostly at the protein level while mutation occurs at the genome level.

Rules are organized into rule groups. Within each rule group, only the satisfied rule of highest weight fires on any given sequence s . This constraint is intended to capture the situation wherein a single point mutation may confer some resistance occurring in isolation, but confers more resistance when occurring in coordination with certain other mutations at other sites. In such a case, one wishes to acknowledge the higher weight for the more specific rule, but not double-count the more general rule for a single point mutation. In practice, this usually means that more specific rules supersede more general rules, as is common in rule-based expert systems (Buchanan and Shortliffe, 1984, Jackson, 1990).

The function *apply*(r, s) yields *wt*(r) if rule r fired on sequence s and 0 otherwise. Subject to the rule group constraint, the rule r is permitted to fire on sequence s just in case all antecedent conditions $A = \langle i, C_i \rangle \in \text{ant}(r)$ satisfy $c[i] \in C_i$, where $c = \text{codons}(\text{genestring}(\text{gene}(r), s))$ is the string of codons for the gene targetted by rule r as found in sequence s . That is, all

conditions in the antecedent must be satisfied, but each condition would be satisfied by finding any codon value from C_i at codon i in sequence s .

2.4. Virtual Mutants

Nearby resistant mutants are generated by a backward-chaining search through mutation sequence space, beginning with the patient's current HIV sequences. At each step, a sequence that does not fire a rule is used to generate several new sequences that do. The new sequences are identical except that codon positions mentioned by the rule are modified so that the rule does fire. They represent nearby virtual mutants that resist the drugs mentioned by the rule.

The Hamming distance $d(b_1, b_2)$ between two possibly degenerate bases b_1, b_2 is 1 if their intersection is null and 0 if it is non-null. Hamming distances between other objects are defined by the natural extension. For example, the Hamming distance $d(h, H)$ between an viral genome h and a set of viral mutants H is the minimum distance between h and any member of H .

Let d_{max} be the maximum Hamming distance between viral genomes initially constructed from the patient sequences and any other virtual mutants considered (currently $d_{max} = 3$). Thus, d_{max} is the maximum number of simultaneous coordinated mutations allowed. Let $M(S, d)$ abbreviate the set of mutants at Hamming distance d or less from any genome constructible from sequences S . $M(S_0, 0)$ is the set of genomes initially constructible from S_0 , and $M(S_0, d_{max})$ is the set of accessible nearby mutants considered.

Let $I(r)$ be the set of codon indices that are mentioned in any antecedent condition of r , and let $I(R) = \cup_{r \in R} I(r)$. Given genome h and rule r , let $invert(r, h)$ be the set of genomes that agree with h except at codon indices that are mentioned in any antecedent condition of r , where they agree with a codon value from the condition.

$$invert(r, h) = \{x | (\forall i \in I(r), x[i] \in C_i) \text{ and } (\forall i \notin I(r), x[i] = h[i])\} \quad (1)$$

These virtual mutants are near h but trigger r , and so putatively resist $drugs(r)$.

In the implementation, a virtual mutant is encoded as a feature vector of codon values at codon positions mentioned in $I(R)$. As the current rule set mentions 31 different codon positions, a virtual mutant currently may be represented compactly and efficiently by a vector of 31 bytes, or < 256 bits. This will scale linearly with $|I(R)|$.

2.5. Heuristic Functions

Two different heuristic functions are used to model mutant resistance: *CurrWt* for modelling current resistance weight, and *MutScore* for modelling nearby mutant resistance. An overall ranking function, f , combines these to rank order the possible treatment regimens. Additionally, lower bound functions h and h_0 are used internally in the branch-and-bound algorithm.

2.5.1. *Current Weight* For $g \in G_{FDA}$ let n_g be the number of sequences in S that encode g . Let $D \in \mathcal{T}_{FDA}$. Define

$$w(s, D) = \sum_{r \in rules(D)} apply(r, s) \tag{2}$$

$$CurrWt(S, D) = \sum_{g \in G_{FDA}} \sum_{s \in S} w(genestring(g, s), D) / n_g \tag{3}$$

$CurrWt(S, D)$ is the average total current rule weight of S with respect to D .

Under this model, the total current level of resistance to a multi-drug combination is the sum of the current resistances according to each rule that is triggered by the patient's current HIV sequences. When this measure is low for a given combination treatment, there is little identified genotypic resistance to any drug in the combination. The effect of this is to identify drug combinations that have little or no current resistance and therefore attack the virus strongly.

2.5.2. *Mutation Score* Let $D \in \mathcal{T}_{FDA}$. Let

$$w_{min}(x, D) = \min_{y \in D} w(x, y) \tag{4}$$

$$m(x, S, D) = \begin{cases} 0, & \text{if } d(x, S) > d_{max} \\ 0, & \text{if } w_{min}(x, D) = 0 \\ d_{max} - d(x, S) + \min\{1, w_{min}(x, D)\}, & \text{otherwise} \end{cases} \tag{5}$$

$$MutScore(S, D) = \max_{x \in M(S, d_{max})} m(x, S, D) \tag{6}$$

$MutScore(S, D)$ is zero if every mutant within Hamming distance d_{max} of S has zero resistance to at least one drug in D . Otherwise, (1) its integer part is d_{max} minus the Hamming distance to the nearest mutant that resists every drug in D , and (2) its fractional part is the rule weight of the least resisted drug in D on the most resistant such mutant.

Under this model, a mutant resists a drug combination only as strongly as it resists the least-resisted drug in the combination, and a drug combination suppresses a virus population only as strongly as it suppresses the most-resistant member of the population. When this measure is low, most identified nearby virtual mutants are putatively suppressed by at least one drug in the combination. The effect of this is to identify nearby mutants that resist every drug in a combination, and drug combinations such that no nearby mutant resists every drug.

2.5.3. *Overall Ranking Function* The drug combinations are ranked by a function f that combines $CurrWt$ and $MutScore$ to rank drug combination D relative to other drug combinations. Currently we use Euclidean distance,

$$f(D) = \sqrt{CurrWt^2(S_0, D) + MutScore^2(S_0, D)} \tag{7}$$

When f is near or at zero there is little or no identified resistance from either source, while increasing positive values indicate increasing resistance. The best ranked combinations represent a satisficing compromise that attempts to avoid both current and nearby drug resistance simultaneously. Any other function f' that is non-decreasing in *MutScore* could be used instead of f without changing the analysis below.

2.5.4. Lower Bound on Ranking Function The branch and bound algorithm below requires a lower bound on f . Let $worst(D)$ be any fixed virtual mutant accessible from S_0 , i.e., $worst(D) \in M(S_0, d_{max})$, and let

$$h_0(x, D) = \sqrt{CurrWt^2(S_0, D) + m^2(x, S_0, D)} \quad (8)$$

$$h(D) = h_0(worst(D), D) \quad (9)$$

By construction,

$$h(D) \leq f(D) \text{ whenever } worst(D) \in M(S_0, d_{max}) \quad (10)$$

$$h(D) = f(D) \text{ whenever } worst(D) = \operatorname{argmax}_{x \in M(S_0, d_{max})} m(x, S_0, D) \quad (11)$$

Note that if some other function f' that is non-decreasing in *MutScore* were chosen to rank treatment regimens instead of f , then corresponding functions h'_0 and h' could be derived by substituting m for *MutScore* in the definition of f' .

3. RESULTS

This section contains an algorithm that solves the problem, a proof of correctness, timing results on representative patient data, example patient output from a deployed application, and summary statistics from an initial small-scale human clinical trial.

3.1. Algorithm

For each number of drugs at or below some maximum (currently 4), the computational task is to sort the possible drug combination treatment regimens D according to the sort function f and enumerate the top few highest-ranked alternatives. The main algorithm is given below. The basic data structure is a priority queue that holds drug combination objects. Primitive queue operations include Queue-Clear, Queue-Insert, Queue-Pop, and Queue-Top. The queue Q is sorted by the lower bound function h . At the N -th outer loop iteration, steps 4–21, it holds drug combinations with exactly N drugs.

1. **procedure** MAIN(S_0 , Max-Ndrugs, Max-Nsuggests):
2. **begin**
3. $Q \leftarrow$ Make-Priority-Queue();
4. **for** Ndrugs \leftarrow 1 **until** Max-Ndrugs **do**
5. **begin**
6. Initialize-Queue(Q , Ndrugs, S_0);

```

7.          for NSuggests ← 1 until Max-Nsuggests do
8.              begin
9.                  for X ← NOT(FAILURE) until X = FAILURE do
10.                     begin
11.                         D ← Queue-Pop(Q);
12.                         X ← Find-Worse-Mutant(D);
13.                         if X ≠ FAILURE then
14.                             begin
15.                                 worst(D) ← X;
16.                                 Queue-Insert(D,Q, h(D));
17.                             end
18.                         end
19.                     Suggest(D, Ndrugs, NSuggests);
20.                 end
21.             end
22.         end

```

At each inner loop iteration, steps 9–18, the currently best-ranked drug combination D is popped from the queue. The mutation space $M(S_0, d_{max})$ is searched to find a new mutant x such that $h_0(x, D) > h(D)$. If this search succeeds then x replaces $worst(D)$, D is reinserted into the queue sorted by h , and the algorithm continues with the next inner loop iteration. Otherwise the search fails and the algorithm suggests D as an optimal combination under f ; continuing from this point will enumerate further combinations in order of optimality. The implementation uses additional bookkeeping and indexing, not shown below, to avoid redundant or unnecessary processing steps.

3.1.1. Initialization In the initialization phase for N drugs, each N -drug combination D is initialized by setting (1) $worst(D)$ to the worst current genome in $M(S_0, 0)$; (2) $to-expand(D)$ to a list of the elements of $M(S_0, 0)$; and (3) $to-check(D)$ to the empty list. After initialization, drug combinations with low current weight rise to the top of the queue and are processed first. Drug combinations with high current weight may remain below the top of the queue for the entire program execution, and so may never be processed.

```

1. procedure Initialize-Queue(Q, Ndrugs,  $S_0$ ):
2. begin
3.     Queue-Clear(Q);
4.     for D in  $\mathcal{T}_{FDA}^{Ndrugs}$  do
5.         begin
6.              $X \leftarrow M(S_0, 0)$ ;
7.              $to-check(D) \leftarrow \text{EMPTY}$ ;
8.              $to-expand(D) \leftarrow X$ ;

```

```

9.            $worst(D) \leftarrow \operatorname{argmax}_{x \in X} h_0(x, D);$ 
10.          Queue-Insert(D, Q,  $h(D)$ );
11.          end
12. end

```

3.1.2. Primitive search step The primitive search step for D is to find a worse virtual mutant than the current $worst(D)$, or else to prove that there is none. For each D , this leads to a separate search through mutation sequence space. First, D is checked against mutants that have been generated since the last time D was considered. Because the top-ranked treatments often share several drugs, there is often overlap in their generated mutants. Next, existing mutants are expanded under rules for D to generate more new mutants. As the search progresses, drug combinations that successfully avoid putative resistant mutants will tend to remain at or near the top of the queue and so will receive the most processing time, while those that encounter a close resistant mutant will tend to sink below the top of the queue and so may not be processed again.

```

1. procedure Find-Worse-Mutant(D):
2. begin
3.   X  $\leftarrow$  Check-Mutants(D);
4.   if X = FAILURE
5.     then X  $\leftarrow$  Expand-Mutants(D);
6.   return X;
7. end

```

The virtual mutants are stored on one singly threaded list (or an array) common to all treatments D . For a given D , the methods *to-check*(D) and *to-expand*(D) return a pointer to D 's current place on the list. Below, function List-Pop(L) returns the list element currently pointed to by L and advances the pointer. Function List-Top(L) returns the element without advancing the pointer. Function List-Pull(X, L) adds the elements of list X to the end of the list pointed to by L. List-Empty(L) is true iff L points to the empty list. Where L is a list of virtual mutants, Delete-Redundant(L) returns the elements of L that have not been generated before.

```

1. procedure Check-Mutants(D):
2. begin
3.   for () until List-Empty(to-check( $D$ )) do
4.     begin
5.       X  $\leftarrow$  List-Pop(to-check( $D$ ));
6.       if  $h_0(X, D) > h(D)$ 
7.         then return X;
8.     end
9.   return FAILURE;
10. end

```

New virtual mutants are generated by expanding existing mutants, using the rule inversion *invert()* described above. This confers additional putative resistance against some drug in D to the expanded mutant, and thus constitutes a possible new pessimal mutant for D . As new mutants are generated, they are added to the end of the common list. In the pseudocode this is denoted as *to-check(all)* and *to-expand(all)* to indicate that it is common to all treatments D .

```

1. procedure Expand-Mutants(D):
2. begin
3.   for () until List-Empty(to-expand(D)) do
4.     begin
5.        $X \leftarrow \text{List-Top}(\textit{to-expand}(D));$ 
6.       for R in rules (D) do
7.         begin
8.            $L \leftarrow \text{Delete-Redundant}(\textit{invert}(X, R));$ 
9.           if NOT(List-Empty(L)) then
10.            begin
11.              List-Pull(L, to-check(all));
12.              List-Pull(L, to-expand(all));
13.              return Find-Worse-Mutant(D);
14.            end
15.          end
16.          List-Pop(to-expand(D))
17.        end
18.      return FAILURE;
19.    end

```

3.2. Proof of Correctness

LEMMA

Find-Worse-Mutant(D) returns FAILURE if and only if $\textit{worst}(D) = \text{argmax}_{x \in M(S_0, d_{max})} h_0(x, D)$.

Proof: \Rightarrow Assume Find-Worse-Mutant(D) returns FAILURE. Then both Check-Mutants(D) and Expand-Mutants(D) must have returned FAILURE. Consequently we must have examined the transitive closure of *invert* over *rules(D)* within the set $M(S_0, d_{max})$, and so the argmax is known by direct enumeration for that particular D .

\Leftarrow Assume $\textit{worst}(D) = \text{argmax}_{x \in M(S_0, d_{max})} h_0(x, D)$. Then the test, $h_0(X, D) > h(D)$ in Check-Mutants(D), must fail for every X. This test conditionalizes the only RETURN in either Check-Mutants(D) or Expand-Mutants(D) that does not return FAILURE, and consequently every available RETURN must return FAILURE. Find-Worse-Mutant(D) always returns, so it must return FAILURE because no other possibility is open. ■

THEOREM

Suppose the algorithm is started with patient sequences S_0 , and suggests drug combination D with $|D| = k$ as the highest-ranked k -drug combination.

Then D is globally optimal because $\forall D' \in \mathcal{T}_{FDA}^k, f(D') \geq f(D)$.

Proof: The proof of correctness proceeds by a *reductio ad absurdum* argument. Assume to the contrary that $\exists D' \in \mathcal{T}_{FDA}^k, f(D') < f(D)$. Then

$$f(D') < f(D) \quad , \text{ by assumption} \quad (12)$$

$$= h(D) \quad , \text{ because } \textit{worst}(D) \text{ is worst for } D \text{ by Lemma} \quad (13)$$

$$\leq h(D') \quad , \text{ because } D \text{ preceded } D' \text{ on the queue} \quad (14)$$

$$\leq f(D') \quad , \text{ because } h \text{ is a lower bound on } f \quad (15)$$

which is a contradiction. Consequently the assumption that some other $D' \in \mathcal{T}_{FDA}^k$ had a better ranking than D must be false, and so D must be optimal under f . ■

3.3. Timing Results

The algorithm currently is implemented in Common Lisp. The results described here were obtained on a 110 MHz SPARCstation 5 desktop workstation. Currently there are 12 drugs approved by the FDA for HIV, with several more expected in the near- to mid-term future. The current 12 result in 407 different combination treatments of four or fewer drugs, because some drugs should not be used together. The input experimental data on each patient's current infection typically consists of five reverse transcriptase sequences and five protease sequences, a total of 7,980 letters of HIV genetic information. Each HIV genome gives rise to roughly 6,000 putatively drug-resistant virtual mutants as described below, resulting in a typical problem search space size currently about 12 million drug combination/virtual mutant pairs per patient. This will increase combinatorically as more drug-resistant mutations are reported and more drugs and drug targets for HIV are approved by the FDA.

The implemented system was run on 43 randomly chosen HIV patients' sequences. The mean was 68 seconds and the median was 62 seconds. Typically patients are seen in the clinic every three months for routine monitoring, as well as in between routine visits if the need arises. Consequently, this throughput rate would support a maximum of approximately 100,000 patients under the current configuration. This is comfortably above our current patient population, though in the future a publicly accessible WWW server and projected advances in sequencing technology may considerably increase both the patient base and the burden of the computational analysis per patient.

3.4. Example Output on Patient Clinical Data

The final result of the optimization problem treated here is the five highest-ranked combinations of 1, 2, 3, and 4 drugs. This section gives some examples of the final output that is provided to the physician in a clinical setting. In actual use, several additional displays are generated that are not shown here.

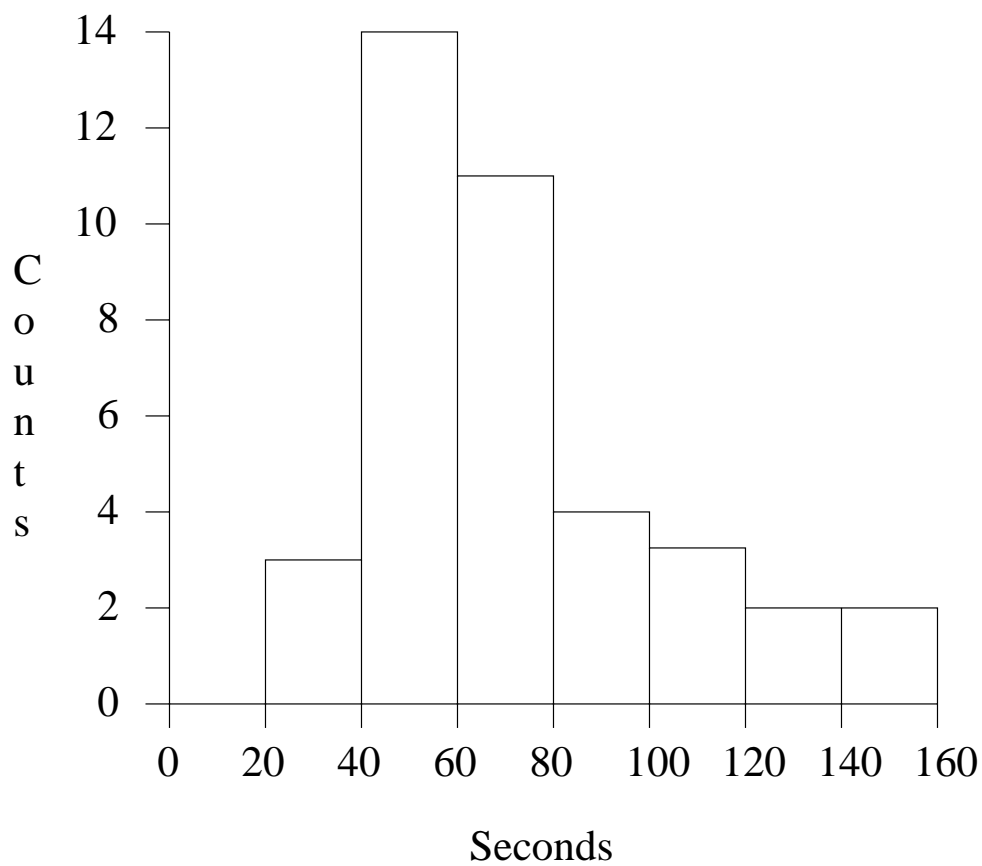


Figure 2. Timing histogram. The implemented system was run on 43 randomly chosen HIV patients' sequences. The x -axis shows the number of elapsed seconds, the y -axis the number of trials that consumed that many seconds.

Table 1 shows 3-drug combinations suggested for an HIV patient. This is the primary display used by an attending physician when making a choice of treatment regimen for a given patient. The five top-ranked 3-drug combinations are shown ordered by f from top to bottom. The next highest-ranked RT-only combination is also generated for comparison purposes. Each combination is listed along with its estimated cost, its values of $CurrWt$ and $MutScore$, and the maximum possible value of w_{min} at 0–3 coordinated base pair changes (0–3 Mut).

Table 2 shows an example nearby most-resistant mutant for the top-ranked treatment in Table 1. It illustrates a putative mutational pathway from the patient's current sequences, which already have slight resistance to Nelfinavir (NFV), to a virtual mutant that also resists the other drugs in the combination. The codon indices and values involved, the rules that induced the mutation through inversion, and the drugs resisted at each step, are shown.

3.5. Small-scale Clinical Trial Summary

The first HIV patient data was run through the CTSHIV system in June, 1996. In February, 1997, the application began its first round of human clinical trials on 14 HIV patients at the University of California, Irvine, and at the Orange County Center for Special Immunology as a satellite site, under the auspices of the California Collaborative Treatment Group (CCTG). Informed consent was obtained using a form approved by the UC Irvine Institutional Review Board. All patients had detectable viral load at baseline (mean \log_{10} load of 4.67 ± 2.16), and failure of at least one previous antiviral treatment regimen due to the emergence of drug resistance. These patients, already expected to be infected by drug-resistant strains of HIV, are considered among the most challenging to treat.

Results from these small-scale trials have been encouraging (Cimoch et al., 1998, Lathrop et al., 1998). As shown in Table 3, 12 patients completed one year of trials (2 patients withdrew prior to completion). After one year of treatment, nine patients who had failed at least one prior treatment regimen had an undetectable viral load (9 complete responders, 64% of enrollees, 75% of completers), and one other patient had $\approx 25\times$ viral load reduction (10 partial responders, 71% of enrollees, 83% of completers).

4. DISCUSSION

We have described a triply nested combinatorial optimization problem that arises in computational attempts to optimize HIV patient treatment protocol (drug regimen) with respect to drug resistance, given a set of HIV genetic sequences from the patient:

1. Identify the drug combinations that most strongly suppress a population of mutants centered on the patient's current HIV strains;
2. For a given drug combination, identify the most-resistant mutant in the population; and
3. For a given drug combination and mutant, identify the least-resisted drug.

The optimization problem was characterized, and the objects involved were represented computationally. An implemented branch-and-bound algorithm that computes a solution to

Suggested protocols with 3 drugs:	CurrWt	MutScor	0 Mut	1 Mut	2 Mut	3 Mut
A5 SQV NFV D4T:	0.06	0.1	0.0	0.0	0.0	0.1
B3 SQV DEL D4T:	0.00	0.2	0.0	0.0	0.0	0.2
C3 SQV NEV D4T:	0.00	0.4	0.0	0.0	0.0	0.4
D4 SQV DEL AZT:	0.00	0.6	0.0	0.0	0.0	0.6
E4 SQV NEV AZT:	0.00	0.6	0.0	0.0	0.0	0.6
RF3 DEL DDI AZT:	0.08	1.2	0.0	0.0	0.2	0.9

Table 1. Example 3-drug output from HIV patient “AA,” showing a favorable resistance profile. For the highest-ranked treatment, current resistance (CurrWt) and nearby mutation score (MutScor) are small, and only a weakly-resistant mutant appears even out to Hamming distance three (3 Mut). The letters A–F identify treatments. Treatment F is the best RT-only treatment (indicated by the prefixed letter R). Digits after the letters indicate cost codes (0 = \$0 to \$200, . . . , 3 = \$600 to \$800, 4 = \$800 to \$1000, 5 = \$1000 to \$1200, . . . , per month estimated average wholesale cost). Drug abbreviations: AZT = Zidovudine, D4T = Stavudine, DDI = Didanosine, DEL = Delavirdine, NFV = Nelfinavir, NEV = Nevirapine, SQV = Saquinavir. (After (Lathrop et al., 1998).)

	CurrWt	MutScor	0 Mut	1 Mut	2 Mut	3 Mut
A5 D4T NFV SQV:	0.06	0.1	0.0	0.0	0.0	0.1
Current: (NFV) RT 151:CAG->ATG by R11 (D4T) PRO 90:TTG->ATG by R28 (SQV)						

Table 2. Example output for HIV patient “AA” showing one example of the closest mutants inferred to most resist every drug in the top-ranked 3 drug combination of Table 1. Three letters must change simultaneously. Currently NFV is resisted; changing two letters at RT 151 resists D4T and changing one at PRO 90 resists SQV. (After (Lathrop et al., 1998).)

RESPONDERS:

- 9 = complete; no detectable viral load at completion
- 1 = partial; viral load reduction $\approx 25\times$ at completion

NON-RESPONDERS:

- 2 = treatment failure at completion
- 2 = withdrawn (1 death, 1 disappeared)

OF 14 ENROLLEES:

- 64% = 9/14 had no detectable viral load
- 71% = 10/14 were responders

OF 12 COMPLETERS:

- 75% = 9/12 had no detectable viral load
- 83% = 10/12 were responders

Table 3. Summary of small-scale human clinical trials: outcome of 14 patients after 1 year of treatment. Detailed per-patient trial outcomes are reported in (Cimoch et al., 1998).

the problem was described and proved correct. Empirical timing results on representative patient data, example clinical output, and summary statistics from an initial small-scale human clinical trial, were given.

Currently, execution speed is an issue for this application only because excessive execution times make the web-based server bog down considerably. The current patient load is light, and for now execution times of a minute per patient are acceptable. The problem search space size currently is not large, and other more straight-forward approaches currently might do as well. On the other hand, routine multiple sequencing of patient infections and widespread use of genomic information in patient treatment can be expected in the future. The search space size will increase combinatorically as more resistant mutations are reported in the literature and more drugs are approved by the FDA. Branch-and-bound algorithms typically have excellent scaling properties, and are usually able to exploit parallel and distributed computing paradigms. Consequently, it appears that the algorithm described above is adequate for the current task and should scale well as the search space size increases and the patient load goes up.

The overall ranking function f used above is purely heuristic, and alternatives would be welcome. It has the property that values near or at zero indicate little or no putative resistance, either current or nearby, and increasing positive values indicate increasing resistance. Consequently, the best ranked combinations represent a satisficing compromise along both metrics simultaneously. However, any combination of *CurrWt* and *MutScore* that is non-decreasing in *MutScore* could be used to sort alternative treatment protocols. The analysis above applies unchanged for any other ranking function f' that is non-decreasing in *MutScore*.

It is difficult to analyze the computational complexity of the algorithm because branch and bound search is inherently exponential. In the worst case, every element of the search space must be examined, and no savings arises. However, if the heuristic functions are well suited to the domain problem, the pruning and hence the computational savings may be considerable. Here, implicit pruning of a combination treatment occurs whenever the current resistance is high or a strongly resistant nearby mutant is discovered. Both these conditions will tend to make the treatment sort away from the queue top under the heuristic function, and consequently computational effort will go to other combination treatments instead.

There are important limitations of the approach above. Sequence-based rules capture only part of the domain knowledge about drug resistance, albeit a clinically useful part. Drug resistance may arise for other domain-specific reasons that cannot be represented easily as rules. The rule representation can only approximate temporal dependencies such as apparently seen in the peculiar “chain-initiator” mutation (Boucher et al., 1992, Gurusinghe et al., 1995, Kellam et al., 1992) involving codons 41, 67, 70, 215, and 219 of reverse transcriptase and the sequential nature of that subsequent progression; however, this is currently poorly understood and would be problematic for any clinical advisor program. Current sequencing techniques may provide only partial or no information about minority strains. The rule set is only as complete as current scientific knowledge allows. Currently it may be possible to infer when resistance is likely to occur, based on genome sequences actually seen in the patient that correspond to resistance-conferring mutations described in the scientific literature. However, it is impossible to guarantee the non-existence of an unsuspected resistant mutant.

Nonetheless, for cases where a treatment regimen has failed due to the development of drug resistance, the application may help the attending physician to base the next choice of treatment regimen on scientific principles and experimental data. In the future we envision that sequencing technological advances will result in the routine availability of hundreds or thousands of pathogen sequences per patient per visit at a cost of pennies, and that computational analysis of pathogen sequences will be an important component guiding treatment choice in most clinical settings, regardless of disease. This work represents one small enabling step toward that goal.

Acknowledgments

Darryl See, Jeremiah Tilles, Paul Cimoch, and Edison Schroeder initiated the biomedical foundations of CTSHIV. Nick Steffen, Miriam Raphael, Sophia Deeds-Rubin, Wei Wang, Ranjit Iyer, and Yi Cao assisted in developing the web pages. Darryl See, Douglas Richman, and Edison Schroeder helped extract the original knowledge base from the AIDS scientific literature. Doug Cable and Winnie Huang co-supervised the first CTSHIV clinical trials. Data entry and analysis were done by Richard Haubrich and Allen McCutchan at the University of California at San Diego. Catherine Diamond and Carol Kemper are supervising the next round of clinical trials. John Gennari and Nick Steffen are revising the expert system framework. Joanne Luciano was an early advocate of computer-based customized treatment for individual patients. Tom Gingeras contributed useful practical insights and advice. We gratefully acknowledge the technical assistance of Tonya Clark and Marikel Chatard in cloning, PCR, and RNA/DNA extraction. Comments from Nick Steffen, Yi Cao, and the blind reviewers improved the presentation.

Funding was provided by Roche Molecular Systems, the California University-wide AIDS Research Program through the California Collaborative Treatment Group (CCTG), and the National Science Foundation under grant IRI-9624739.

CTSHIV is available from University/Industry Research and Technology, 380 University Tower, University of California, Irvine, CA, 92717 USA.

References

- C.A. Boucher, E. O'Sullivan, J.W. Mulder, C. Ramautarsing, P. Kellam, G. Darby, J.M. Lange, J. Goudsmit, and B.A. Larder. Ordered appearance of zidovudine resistance mutations during treatment of 18 human immunodeficiency virus-positive patients. *J. Infect. Dis.*, 165:105–110, 1992.
- B. Buchanan and E. Shortliffe, editors. *Rule-based expert systems: The MYCIN experiments of the Stanford Heuristic Programming Project*. Addison-Wesley, Reading, MA, 1984.
- C. Carpenter, M. Fischl, S. Hammer, M. Hirsch, D. Jacobsen, D. Katzenstein, J. Montaner, D. Richman, M. Saag, R. Schooley, M. Thompson, et al. Antiretroviral therapy for HIV infection in 1996. *J. American Medical Assoc.*, 276:146–154, 1996.
- P.J. Cimoch, D.M. See, M.J. Pazzani, W.M. Reiter, R.H. Lathrop, W.A Fasone, and J.G. Tilles. Application of a genotypic driven rule-based expert artificial intelligence computer system in treatment experienced HIV-infected patients. Immunologic and virologic response. In *Proc. of the 12th World AIDS Conf.*, page extended abstract #32297, Geneva, Switzerland, 1998.
- J. Condra, W. Schief, O. Blahy, L. Gabryelski, D. Graham, J. Quintero, A. Rhodes, H. Robbins, E. Roth, M. Shivaprakash, D. Titus, et al. *In vivo* emergence of HIV-1 variants resistant to multiple protease inhibitors. *Nature (London)*, 374:569–571, 1995.

- L. Diamond, D. Nguyen, H. Jouault, M. Imbert, and C. Sultan. An expert system for the interpretation of flow cytometric immunophenotyping data. *J. of Clinical Computing*, 22:50-58, 1994.
- A. Fisher, L. Collaiti, R. Ratner, R. Gallo, and F. Wong-Staal. A molecular clone of HTLV III with biologic activity. *Nature (London)*, 316:262-265, 1985.
- Z. Gu, Q. Gao, H. Fang, M. Parniak, B. Brenner, and M. Wainberg. Identification of novel mutations that confer drug resistance in the human immunodeficiency virus polymerase gene. *Leukemia*, 8S1:5166-5169, 1994.
- A.D. Gurusinghe, S.A. Land, C. Birch, C. McGavin, D.J. Hooker, G. Tachedjian, R. Doherty, and N.J. Deacon. Reverse transcriptase mutations in sequential HIV-1 isolates in a patient with AIDS. *J. Med. Virol.*, 46:238-243, 1995.
- P. Jackson. *Introduction to Expert Systems*. Addison-Wesley, Reading, MA, 1990.
- H. Jacobsen, M. Hanggi, M. Ott, I. Duncan, S. Owen, M. Andreoni, S. Vella, and J. Mous. *In vivo* resistance to a human immunodeficiency type-1 proteinase inhibitor. *J. Infect. Diseases*, 173:1379-1387, 1996.
- P. Kellam, C. Boucher, and B.A. Larder. Fifth mutation in human immunodeficiency virus type 1 reverse transcriptase contributes to the development of high-level resistance to zidovudine. *Proc. Natl. Acad. Sci. USA*, 89:1934-1938, 1992.
- J. Lange. Triple combinations: present and future. *J. of AIDS and Human Retrovirology*, 10 Suppl 1:S77-82, 1995.
- R.H. Lathrop, N.R. Steffen, M.P. Raphael, S. Deeds-Rubin, M.J. Pazzani, P.J. Cimoch, D.M. See, and J.G. Tilles. Knowledge-based avoidance of drug-resistant HIV mutants. In *Proc. of the Tenth Conf. on Innovative Applications of Artificial Intelligence*, Madison, WI, 1998. AAAI Press, Menlo Park, CA, USA. Invited to appear in *AI Magazine*.
- M. Musen, S. Tu, A. Das, and Y. Shahar. EON: a component-based approach to automation of protocol-directed therapy. *J. Amer. Medical Informatics Assoc.*, 3:367-388, 1996.
- M. Musen, K. Wieckert, E. Miller, K. Campbell, and L. Fagan. Development of a controlled medical terminology: knowledge acquisition and knowledge representation. *Methods of Information in Medicine*, 34:85-95, 1995.
- W. O'Brian, P. Hartigan, D. Martin, J. Esinhart, A. Hill, S. Benoit, M. Rubin, M. Simberkoff, and J. Hamilton. Changes in plasma HIV-1 and CD4+ lymphocyte counts and the risk of progression to AIDS. *New England J. Medicine*, 334:426-431, 1996.
- L. Ohno-Machado, E. Parra, S. Henry, S. Tu, and M. Musen. AIDS 2: A decision-support tool for decreasing physician's uncertainty regarding patient eligibility for HIV treatment protocols. In *Proc. of the 17th Annual Symp. on Computer Applications in Medical Care*, pages 429-433, Washington, D.C., 1993.
- M. Pazzani, R. Iyer, D. See, E. Shroeder, and J. Tilles. CTSHIV: A knowledge-based system in the management of HIV-infected patients. In *Proc. of the Intl. Conf. on Intelligent Information Systems*, 1997.
- M. Pazzani, D. See, E. Shroeder, and J. Tilles. Application of an expert system in the management of HIV-infected patients. *J. of AIDS and Human Retrovirology*, 15:356-362, 1997.
- R.M. Ribeiro, S. Bonhoeffer, and M.A. Nowak. The frequency of resistant mutant virus before antiviral therapy. *J. AIDS*, 12:461-465, 1998.
- C.D. Rosin, R.K. Belew, G.M. Morris, A.J. Olson, and D.S. Goodsell. Computational coevolution of antiviral drug resistance. *Artif. Life*, 4:41-59, 1998.
- C. Ruggiero, M. Giacomini, O. E. Varnier, and S. Gaglio. A qualitative process theory based model of the HIV-1 virus-cell interaction. *Computer Methods and Programs in Biomedicine*, 43:255-259, 1994.
- C. Safran, D. Rind, D. Sands, R. Davis, J. Wald, and W. Slack. Development of a knowledge-based electronic patient record. *M.D. Computing*, 13:46-54, 1996.
- H. Sieburg. Methods in the Virtual Wetlab I: rule-based reasoning driven by nearest-neighbor lattice dynamics. *AI in Medicine*, 6:301-319, 1994.
- M. Sobesky, C. Michelet, R. Thomas, and P. LeBeux. Decision making system. *J. Clinical Computing*, 22:20-26, 1994.
- F. Sonnenberg, C. Hagerty, and C. Kulikowski. An architecture for knowledge-based construction of decision models. *Medical Decision Making*, 14:27-39, 1994.
- S. Tu, H. Eriksson, J. Gennari, Y. Shahar, and M. Musen. Ontology-based configuration of problem-solving methods and generation of knowledge-acquisition tools. *AI in Medicine*, 7:257-289, 1995.
- L. Xu. An integrated rule- and case-based approach to AIDS initial assessment. *Intl. J. of Bio-Medical Computing*, 40:197-207, 1996.