Designing Better Phages
(Extended Abstract)

Steven S. Skiena*
Department of Computer Science
State University of New York
Stony Brook, NY 11794-4400
skiena@cs.sunysb.edu

1 Introduction

Bacteriophages are viruses which inject their own DNA into a host bacterium. The phage then takes over the genetic machinery of the cell to produce more phages, eventually killing the bacterium in the process. For this reason, bacteriophages have long been proposed as an antibacterial agent. Indeed, phages have been widely used in the former Soviet Union to fighting infections since the 1930's [14].

There is increasing interest in medical applications of bacteriophages, motivated by the growing problem of antibiotic resistance. Although promising, several hurdles must be overcome before bacteriophages can successfully compete with antibiotics as antibacterial agents. First, phages tend to be narrowly targeted, i.e. a particular phage can only successfully invade a narrow range of host cell types. Second, mammalian circulatory systems are adept at removing phages from the blood stream. Third, crude phage lysates often contain toxins to the host organism. However, recent work towards isolating long-circulating bacteriophages [10] shows considerable promise in overcoming these hurdles.

We propose a complementary approach to building better phages, by engineering them to avoid a primary bacterial defense mechanism. *Restriction enzymes* cut unmethylated DNA at specific patterns, or *cutter sequences*. Such enzymes, isolated from bacteria, have become a fundamental tool in manipulating DNA in the laboratory [8, 9]. However, their primary biological role is as a bacterial defense mechanism against phages. Bacteria race to deploy restriction enzymes so as to cut the injected phage DNA into fragments before it can become biologically active. The greater the number of *restriction sites*, or occurrences of the given cutter sequence on the phage genome, the more vulnerable they are to the given restriction enzyme. A convincing body of evidence [3, 6, 7, 18] shows that natural selection has worked to reduce the number of restriction sites in phage genomes.

We propose further increasing the effectiveness of a given therapeutic phage by altering its genome to minimize the number of restriction sites for a given set of cutter sequences. One must be careful not to change the phenotype of the resulting phage, however. We propose to exploit the known redundancy in triplet code by which the $4^3 = 64$ different codons (sequences of three consecutive nucleotide bases) in DNA either terminate transcription or map to one of the 20 different

*This work is partially supported by NSF Grant CCR-9625669 and ONR Award N00149710589.

amino acids (or residues) which build up proteins. We seek the coding sequence which minimizes the number of restriction sites while still coding for the desired protein.

Paradoxically, it is the non-coding regions (often called 'junk DNA') which conservatively should be preserved verbatim from the wild-type encoding. Although these non-coding regions are less well understood, they contain most of the critical binding sites for regulatory elements and have other, still unknown, functions. Therefore, our program leaves these intergenic regions intact. In fact, the intergenic regions of bacteriophages are typically very short, a tribute to the evolutionary advantage of minimizing genome length in phages.

Our computational results are very encouraging – we demonstrate that there is sufficient flexibility in the genetic code to remove the overwhelming majority of restriction sites from genes, even when subjecting the phage to a battery of several dozen enzymes. Indeed, it is feasible to significantly protect any phage against the cutter sequences of *all* known restriction enzymes. Although there is a particular suite of cutter sequences specific to each bacterial host, such an approach eliminates the need to carefully assay the enzymes specific to a specific bacterium, and may increase the range of possible hosts for a given phage. Indeed, the IncPα plasmids are remarkable both for their broad host range and small number of restriction sites [23]. The laboratory technology needed to alter the genome of a phage to specification is available, and would clearly be worth employing if it could substantially improve the performance of a therapeutic agent. We are now pursuing laboratory implementation of these ideas.

Our paper is organized as follows. In Section 2, we present additional biological background on restriction enzymes, phages, and the triplet code. In Section 3, we formalize our problem of minimizing restriction sites, and present an efficient dynamic programming algorithm to solve it for constant-length cutter sequences, as occur in nature. Section 4 proves the hardness of the problem as the length of the cutter-sequences increases, justifying our algorithmic approach. In Section 5, we perform a series of experiments on a variety of phages and enzyme sets, which support our claims that a large fraction of restriction sites can be effectively removed without changing the phenotype of a phage. Finally, in Section 6, we propose a simple evolutionary model which explains why our technique can hope to generate more robust phages than the product of billions of years of natural selection.

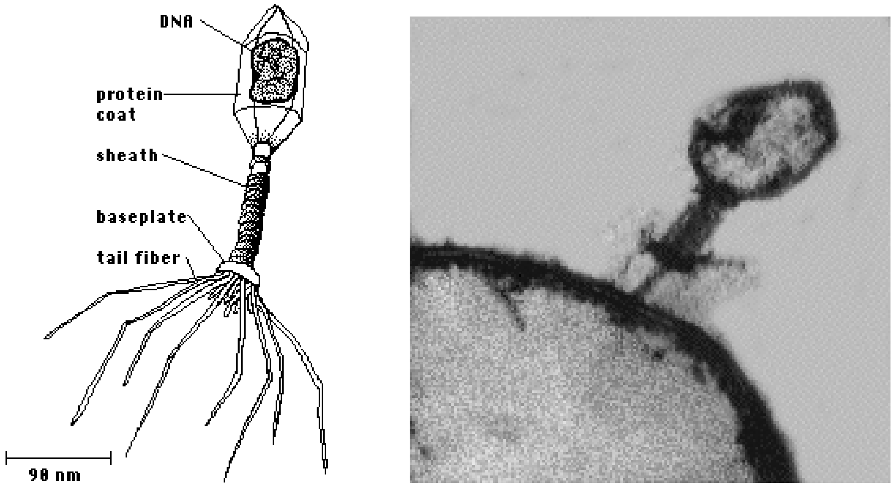

Figure 1: (a) bacteriophage schematic diagram, and (b) a micrograph of a phage invading a host.

2 Restriction Enzymes, Phages, and the Genetic Code

2.1 Restriction Enzymes

Restriction enzymes recognize and cut DNA molecules at particular patterns. For example, the enzyme *Eco*RI cuts at the pattern *GAATTC*. Enzymes are indigenous to specific bacteria, with the name of each enzyme denoting the order of discovery within the host organism (e.g. *Eco*RI was the first restriction enzyme discovered in E. Coli). The first restriction enzyme was discovered by Hamilton Smith in 1970 [12]. Restriction enzymes cut double-stranded DNA, and are classified as either type I or type II depending whether they cleave the strand at specific locations, with those that do (type II) significantly more important for biotechnology. Enzymes can be further classified according to whether they leave behind blunt or sticky ends, i.e. double- or single-stranded end fragments. See Lander [8] for an early survey on restriction enzymes.

The problem of constructing restriction maps, i.e. identifying all the locations of restriction sites on a genome, was a critical component to early sequencing projects. For example, [9] provides a list of all restriction sites for the plasmid pBR322 under 81 different enzymes. Many interesting combinatorial and algorithmic problems have arisen in the context of technologies for restriction mapping [4, 13, 19, 20].

Rebase [17] maintains a complete list of all known restriction enzymes, including cutter sequences, literature references, and commercial availability. As of January 1, 2001, 3487 different enzymes were known, defining at least 255 distinct cutter sequences. Cutter sequence lengths range in length from 2 to 15 bases. Although most enzymes cut at specific oligonucleotide base patterns, other enzymes recognize multiple sequences by allowing variants at specific base positions. For example, the cutter AACNNNNNNGTGC matches from the 5' to 3' end any sequence starting AAC, ending GTGC where they are separated any sequence of exactly six bases.

2.2 Bacteriophages

Bacteriophages were among the first organisms whose genomes were completely sequenced [21], a tribute both to their importance and their short genome lengths. Bacteriophages can breed rapidly, multiplying to several hundred daughter phages in about 20 minutes [22]. Bacteriophage genomes can be either linear or circular. Table 1 displays the characteristics the bacteriophages employed in our study. They vary substantially in length (from 3,569 to 48,502 base pairs) and gene content (from 4 to 102 genes). Almost all of them have very little non-coding DNA – none of the phages in Table 1 have more than 40% non-coding DNA, with many phages below 10%.

There is abundant evidence that both phage genomes and bacterial hosts evolve to avoid restriction enzyme cut sites. Sharp [18] established smaller than expected number of restriction sites in phage genomes and attributed this to natural selection. A second study of five bacteriophage genomes [6] shows four of them significantly depleted in the short palindromic sequences typically associated with restriction sites. Further the T4 phage, the lone exception to this pattern, derives protection from the modification of cytosine to hydroxymethylcytosine.

Table 1 also provides measures of the number of short palindromes in the coding and non-coding regions of each phage, as well as the expected number of such palindromes by averaging 10 random permutations of the genome. Comparing the actual and expected counts for palindromic subsequences in Table 1 shows that 4- and 6-length palindromes are substantially depleted in almost all of our phages. The greater depletion of 6-palindromes presumably reflects the greater likelihood for a random mutation to disrupt the site. There is no clear pattern as to whether depletion occurs more or less frequently inside coding regions.

	Genome			Coding regions					Non-coding regions					Palindrome	
		Genes			4-palins		6-palins			4-palins		6-palins		Depletion	
Phage	Leng.	Cnt	Lay	len	real	exp	real	exp	len	real	exp	real	exp	ratio	rnk
186	30624	46	4	28380	1845	1771.7	321	466.0	2244	126	141.9	26	31.9	1.040	25
A118	40834	72	2	38461	2239	2707.2	561	732.1	2373	157	177.7	48	46.8	1.219	11
Alteromonas	10079	23	2	9318	597	595.2	113	151.4	761	55	53.3	13	15.0	1.047	24
C2	22172	39	2	20532	1026	1461.3	218	388.7	1640	106	127.9	30	38.1	1.461	3
Chp1	4877	12	6	2927	170	205.6	56	50.4	1950	88	129.7	29	32.1	1.218	12
Cp-1	19343	28	2	16715	788	1116.2	133	284.7	2628	113	201.2	22	51.5	1.566	1
D29	49136	77	2	44191	3200	3208.2	419	856.7	4945	296	329.1	59	89.9	1.128	20
G4	5577	11	4	5295	254	333.8	52	82.5	282	254	333.8	52	82.5	1.360	5
HK022	40751	35	2	24781	1429	1546.4	290	399.2	15970	853	998.6	180	259.0	1.164	17
HK97	39732	60	2	34641	1964	2157.2	366	535.9	5091	275	324.6	48	83.4	1.169	16
HP1	32355	41	2	29645	1509	1966.4	329	507.1	2710	171	188.9	55	54.2	1.316	8
L2	11965	14	2	9140	624	718.6	145	206.3	2825	233	226.6	69	64.8	1.136	19
MAV1	15644	15	2	14115	942	1174.1	276	347.1	1529	115	140.6	38	41.7	1.242	10
MS2	3569	4	2	3246	195	201.4	55	51.7	323	16	22.5	4	5.2	1.040	26
MX1	4215	4	2	4017	249	262.5	70	64.2	198	9	12.6	2	3.3	1.038	27
Mu	36717	53	3	33276	1825	2061.9	277	518.9	3441	221	218.2	51	56.2	1.203	14
N15	46375	60	2	41847	2491	2624.6	555	655.0	4528	295	282.9	75	70.5	1.064	22
NL95	4248	4	2	4053	272	258.3	80	64.3	195	10	14.9	2	2.4	0.934	28
P2	33593	42	2	30748	1875	1917.0	326	487.5	2845	145	180.0	32	44.6	1.106	21
P4	11624	13	2	9457	595	588.1	113	145.8	2167	125	144.0	27	35.0	1.062	23
PZA	19366	27	2	18181	979	1204.8	203	315.4	1185	72	83.4	16	24.5	1.282	9
SPP1	44007	102	4	38883	2101	2338.4	413	570.1	5124	308	325.0	51	81.0	1.154	18
Streptococcus	40739	38	2	32311	1638	2203.8	350	569.0	8428	528	580.2	122	156.4	1.330	7
bIL170	31754	63	2	28806	1664	2149.2	335	583.9	2948	207	248.2	47	70.2	1.354	6
bIL67	22195	37	2	19846	1000	1407.6	204	384.5	2349	146	193.8	39	55.4	1.469	2
fs-2	8651	9	2	7334	354	465.8	54	118.5	1317	64	86.0	14	22.9	1.426	4
lambda	48502	67	2	40842	2248	2556.3	383	633.7	7660	425	495.5	79	126.9	1.216	13
psiM2	26111	31	2	22588	1170	1364.4	292	345.0	3523	184	224.6	48	50.6	1.172	15
Avg. ratios					1.16619		1.48705			1.24163		1.38094			

Table 1: Characteristics of the phages used in this study.

There has been a succession of statistical results of the distribution of patterns in DNA sequences, reflecting the increasing body of available sequence data [3, 6, 7, 15]. Indeed, it has been established that bacterial genomes avoid cut sites associated with *their own* type II restriction enzymes [3]. This depletion helps protect the bacteria against the occasional failure of methylated systems. Patterns corresponding to enzymes in other bacterial species also tend to be depleted, suggesting the lateral transfer of restriction-modification systems across species.

2.3 The Triplet Code

Because there are more codons (64) than residues (20 plus the stop symbol), there is inherent redundancy in the triplet code for proteins. Certain residues (e.g. methionine) have only a single corresponding codon, while other residues (e.g. arginine) have as many as 6 corresponding codons. This freedom implies that the number of possible DNA sequences coding for a given protein grows exponentially in the length of the protein. For example, a typical modest-size plant hemoglobin of 147 residues (accession number AB010831) may be coded for by 1.7×10^{75} distinct DNA sequences [1]. We seek the coding which minimizes the total number of restriction sites for a given set of enzyme cutter sequences.

We note that codon usage patterns vary significantly among organisms, and efficient translation requires conforming to these distributions. Thus, removing potential restriction sites leads to improved fitness in one respect (lower probability of DNA degradation), but at a possible decrease in fitness in another (less efficient translation). Our goal here is to demonstrate the potential win in designing a “super-phage”. Our methods can be generalized to significantly respect codon distributions, with what we expect will be little degradation in performance.

We have previously employed dynamic programming methods to exploit the redundancy of the genetic code. In [1], we investigate the degree to which alternate encodings of a given protein can effect the strength of the RNA secondary structure. An interesting connection between these problems is that rRNA genes are typically in the regions least depleted of palindromes in bacterial genomes, perhaps reflecting the tradeoff between the secondary structure necessary for RNA function and the tendency for restriction enzymes to nick such structures [11].

3 Minimizing Restriction Sites

The previous discussion motivates the following *minimum restriction site genome* problem:

Input: The DNA sequence S of a given phage P, a set of genes $G = \{g_1, \ldots, g_k\}$ each defined by an interval on S, a set of restriction enzymes $E = \{e_1, \ldots, e_m\}$, where each enzyme is associated with a given cutter sequence, and an integer x.
Output: A DNA sequence S' where $|S| = |S'|$, such that for each gene $g_i \in G$ the corresponding interval in S and S' code for identical proteins, all non-coding regions of S are conserved in S', and the number of restrictions sites in S' over all enzymes in E is at most x.

We propose the following dynamic programming algorithm for the minimum restriction site genome problem. Let $M[i, W]$ be the minimum number of enzyme cuts possible to encode the first i bases of S, where the last l bases of S' are defined by the string $W = w_1 w_2 \ldots w_l$. Then:

$$M[i, w_1 w_2 \ldots w_l] = \min_{\alpha \in ACGT} M[i-1, \alpha w_1 w_2 \ldots w_{l-1}] + cuts(w_1 w_2 \ldots w_l)$$

where $cuts(W)$ is the number of restriction sites created by the implicit enzyme set whose cutter matches a suffix of W. Further, $M[i,W]=\infty$ if W is not a legal encoding for the l bases ending at i.

The number of restriction sites in the optimal sequence, $M[n]$, is found by testing all possible suffix windows, i.e.:

$$M[n]=\min_{i=0}^{4^l-1} M[n,W_i]$$

Theorem 1 *The minimum restriction site genome problem can be solved in $O((n+m)\cdot 1.817^l)$ time, where l is the length of the longest cut sequence.*

Proof: The correctness of the recurrence relation follows provided the state window length l is at least as long as the longest cutter in the enzyme set. Each cut site is counted exactly once, since $cut(W)$ only counts suffix cuts, and all possible windows are optimized over.

A naive time analysis would be $O(nml4^l)$ time, since at each sequence position 4^l windows must be considered, and at each window (1) the legality of the window must be established for all m genes and (2) the number of suffix cuts for each enzyme established. In fact, there can only be at most $3\times 6^{l/3}=O(1.817^l)$ distinct legal windows of length l at any single position, since the most heavily represented residue is assigned only six of the 64 possible codons.

Quickly identifying the number of suffix cuts can be done using a suffix tree data structure [5], over all legal windows at a position. Details appear in the full paper. ▮

There are a variety of real-world complications which must be considered in redesigning genomes to avoid restriction sites:

- *Overlapping genes* – It is fairly common for two or more genes to share a given region on a genome. Sometimes multiple overlapping genes all reside in the same frame and terminate on the same stop codon. These provide no additional constraints over the longest gene in the set, as all must code for the same amino acid sequence in the common region.

  However, many phages also have overlapping genes residing in different reading frames. In such cases, we are restricted to generating encodings which respect the genes in all the reading frames containing a given position.

- *Genes on alternate strands* – All but seven of our test phages (specifically Chp1, G4, MS2, MX, N95, fs-2, and psi-M2) exhibited genes on both the upper and lower strands of the genome. Genes on both strands must be conserved. We are restricted to using encodings which respect all active reading frames covering a given position, with codings on the lower strand appropriately complemented.

  The layers column in Table 1 records the maximum number of genes to cover some portion of the given genome. Multiplicities are due to overlapping gene and genes on alternate strands.

- *Non-palindromic cutters* – A substantial majority of cutters for restriction enzymes are palindromic, meaning that the cutter is equal to its reverse complement. For palindromic cutters, there is no cut on the bottom strand which is not matched with one on the upper strand, and hence the bottom strand can be disregarded from our computation. However, for non-palindromic enzymes, we must explicitly avoid the presence of the reverse complement of the cutter to avoid cutting the alternate strand.

Although the dynamic programming recurrence above is conceptually simple, an efficient implementation requires care. Our implementation maintain legal windows as packed bits in a long integer. All known cutters in Rebase are at most 15 bases long, and thus a 32 bit integer suffices to encode the window at 2 bits per symbol. We maintain a sparse dynamic program (since most possible windows are illegal) by using an array of hash tables, and use linear search (instead of suffix trees) to count enzyme cuts. Our implementation, which is 2000 lines of C, can run each of our phage/enzyme set pairs in a few minutes on a modest desktop workstation.

4 Hardness for Long Cutters

Although our algorithm is efficient enough to easily handle all known restriction enzymes, it heavily depends upon the fact that each enzyme cutter sequence is short. In fact,

Theorem 2 *The minimum restriction site genome problem is APX-complete for general enzyme lengths.*

Proof: We use a reduction from maximum satisfiability (MAX-SAT). In MAX-SAT, we are given a set of CNF clauses $C = \{c_1, \ldots, c_k\}$ defined over a set of boolean variables $V = \{v_1, \ldots, v_n\}$.

We reduce this to the minimum restriction site genome problem. Suppose the wild-type genome S consists of n bases, each of which is allowed to be one of two possibilities (say either A or T but not C or G). Let the state of the ith base reflect the truth assignment of the ith variable. Each clause will be satisfied unless the relevant bases are exactly set so the appropriate literals are negated. For each clause, we will construct an enzyme of length n, with don't care 'N' bases at every position except the relevant bases for the clause, which will be set such that the enzyme will match iff the truth assignment does not satisfy the resulting clause. Thus the assignment which minimizes the number of matching enzymes maximizes the number of satisfied clauses, giving the reduction.

The hardness of approximation follows from the fact that maximum satisfiability is APX-complete [16]. ▮

5 Experimental Results

To test the effectiveness of our algorithm, and measure the extent to which the triple code allows us to remove restriction sites, we optimized each of the phage genomes of Table 1 against a variety of enzyme cutter subsets, described in Table 2. This enzyme subsets can be partitioned into three general classes:

- *All cutter sequences* – These sets consist of all known sequences of particular length. In all sets, cutter sequences of length greater than eight have been ignored. In particular, we consider the set of all simple cutters (ALL-SIMPLE), all cutters including ambiguous bases (NON-SIMPLE), and all simple cutters of length 4, 6, and 8 (ALL-SIMPLE-4, -6, and -8).

- *Dense organism-specific sequences* – Different suites of enzymes constitute different bacteria defense mechanisms. These sets constitute the host-specific sets of enzymes for the six genus with the largest known enzyme sets, ranging in size from 49 to 26 distinct cutters each. These are *Pseudomonas*, *Escherichia*, *Streptomyces*, *Helicobacter*, *Neisseria*, and *Thermus*.

- *Sparser organism-specific sequences* – Here we consider the host-specific enzyme sets for the next five largest known enzyme sets, ranging in size from 11 to 20 enzymes. The smaller number of enzymes may well reflect less extensive research more than weaker restriction systems.

	Cutter	Exp. cuts	Distribution by Length (self-palindromic/irregular)						
Enzyme Set	count	per site	2	3	4	5	6	7	8
NON-SIMPLE	67	.2017	0 / 0	0 / 0	1 / 1	9 / 8	10 / 26	7 / 4	0 / 1
ALL-SIMPLE	138	.3017	0 / 2	1 / 0	3 / 14	20 / 0	29 / 56	3 / 0	0 / 10
ALL-SIMPLE-4	17	.0781	0 / 0	0 / 0	3 / 14	0 / 0	0 / 0	0 / 0	0 / 0
ALL-SIMPLE-6	85	.0278	0 / 0	0 / 0	0 / 0	0 / 0	29 / 56	0 / 0	0 / 0
ALL-SIMPLE-8	10	.0002	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 0	0 / 10
Pseudomonas	49	.0447	0 / 0	0 / 0	0 / 4	3 / 0	3 / 34	1 / 2	0 / 2
Escherichia	37	.0391	0 / 0	0 / 0	0 / 1	4 / 2	5 / 22	0 / 3	0 / 0
Streptomyces	34	.0228	0 / 0	0 / 0	0 / 1	3 / 0	1 / 20	3 / 1	0 / 5
Helicobacter	29	.1030	0 / 0	0 / 0	1 / 10	4 / 6	1 / 7	0 / 0	0 / 0
Neisseria	26	.0666	0 / 0	0 / 0	0 / 4	6 / 4	0 / 12	0 / 0	0 / 0
Thermus	26	.0656	0 / 0	0 / 0	0 / 6	7 / 1	4 / 7	1 / 0	0 / 0
Haemophilus	20	.0795	0 / 0	1 / 0	0 / 6	5 / 2	1 / 5	0 / 0	0 / 0
Vibrio	20	.0213	0 / 0	0 / 0	0 / 2	1 / 1	2 / 12	1 / 1	0 / 0
Enterobacter	18	.0178	0 / 0	0 / 0	0 / 0	2 / 1	1 / 13	0 / 1	0 / 0
Citrobacter	17	.0156	0 / 0	0 / 0	0 / 0	1 / 1	2 / 12	0 / 1	0 / 0
Salmonella	11	.0171	0 / 0	0 / 0	0 / 1	2 / 1	1 / 5	0 / 1	0 / 0

Table 2: Characteristics of the enzyme sets used in this study.

The genus included in this category are *Haemophilus*, *Vibrio*, *Enterobacter*, *Citrobacter*, and *Salmonella.*

The cut sequences associated with the enzyme subsets in Table 2 have been distinguished into those which are *self-palindromic* (i.e. are identical to their reverse complement), and those which are instead *irregular*. The expected cuts field of Table 2 gives the expected number of cuts per site a random sequence with equiprobable bases would endure if exposed to the given set of enzymes. Larger number of irregular enzymes with shorter cut sequences increases the expected number of cuts.

Our optimization results are described in Tables 3 and 4, grouped according to enzyme class. The results for sparse organism-specific sequences have been omitted because of lack of space, but will appear in the full version. The results for fixed length cutters are very encouraging, demonstrating that we have the freedom to remove every instance of every 8-cutter from the coding regions of every phage we considered. We are able to remove almost 98% of all 6-cutter sites or almost 94% of all 4-cutter sites. A few phages, particularly G4 and SPP1, prove unusually resistant to optimization, leaving between 10% and 20% of all sites. Combining *all* simple cutters begins to overwhelm our ability to remove sites, but even here we typically remove 80% or more of the original sites. There seems to be no significant difference in our ability to remove simple or ambiguous cutter sequences.

Table 4 documents our success at removing organism-specific sets of sites. It is interesting to note that the expected site density of the different sets does not correspond directly the number of distinct cutters. The *Heliobacter* set has more than twice as much expected site density as *Pseudomonas*, even though it has roughly half as many cutters (29 vs. 49). The secret is that *Pseudomonas* has longer cutters on average than *Heliobacter.* However, we are able to remove about 90% of all wild-type sites in most cases.

The primary factor affecting how successfully restriction sites can be removed from a given genome is a function of how densely the restriction sites are packed. This density is a function of both the number of enzymes, and the length of their cutting sequences more than a function of the sequence. We anticipate that there is sufficient freedom in the genetic code to remove essentially all sites if they occur with sufficiently low density.

	4-cutters		6-cutters		8-cutters		All simple		Non-simple	
Phage	wild	opt	wild	opt	wild	opt	wild	opt	wild	opt
186	2514	61	695	7	9	0	10311	1872	5937	697
A118	2114	45	716	8	12	0	8156	1957	3485	580
Alteromonas	658	11	151	2	4	0	2489	567	1190	175
C2	963	27	323	2	4	0	3354	890	1593	186
Chp1	151	18	80	6	0	0	633	173	249	35
Cp-1	780	45	221	15	1	0	3191	802	1458	195
D29	5126	70	1523	8	9	0	19985	3120	13178	1320
G4	314	60	115	14	0	0	1343	440	681	138
HK022	1840	28	661	5	3	0	8377	1575	4896	555
HK97	2482	60	869	8	8	0	11176	2287	6369	876
HP1	1597	29	533	4	6	0	7070	1523	3056	489
L2	492	13	178	0	3	0	1627	354	732	160
MAV1	854	15	267	1	7	0	2499	452	893	101
MS2	241	21	108	4	1	0	1060	216	593	81
MX1	272	7	121	1	0	0	1206	186	677	76
Mu	2585	51	736	9	4	0	11273	2292	6438	895
N15	3267	59	1211	5	9	0	14324	2751	8493	1047
NL95	339	5	126	1	0	0	1357	213	789	100
P2	2396	43	690	4	4	0	10010	1887	5520	698
P4	782	19	215	2	4	0	3086	611	1889	243
PZA	959	29	307	2	7	0	3758	819	1656	225
SPP1	2603	263	736	61	5	0	9752	2745	5193	1205
Streptococcus	1484	40	560	5	2	0	6310	1529	2798	427
bIL170	1355	40	446	5	5	0	4704	1176	2177	258
bIL67	931	20	287	2	3	0	3146	834	1404	166
fs-2	429	10	119	2	0	0	1884	412	1028	155
lambda	3005	89	987	10	9	0	13249	2775	7508	1107
minimum	0.0136		0.0000		0.000		0.1542		0.1002	
average	0.0386		0.0209		0.000		0.2189		0.1413	
maximum	0.1910		0.1217		0.000		0.3276		0.2320	

Table 3: Number of restriction sites cut in coding regions by given enzyme sets for both wild-type and optimized phages.

	Pseudomonas		Escherichia		Streptomyces		Helicobacter		Neisseria		Thermus	
Phage	wild	opt	wild	opt	wild	opt	wild	opt	wild	opt	wild	opt
186	946	42	652	84	335	48	2915	274	2182	335	1977	348
A118	370	20	308	46	167	25	1786	330	1099	272	1859	330
Alteromonas	133	3	60	3	18	1	550	63	417	87	517	92
C2	123	6	102	14	40	6	886	167	359	55	869	96
Chp1	54	1	41	4	30	3	184	31	96	16	127	21
Cp-1	52	2	60	12	9	2	723	153	374	56	587	75
D29	3301	159	3237	322	1928	194	7107	589	5081	699	4207	665
G4	103	15	63	12	29	9	434	91	219	46	271	72
HK022	937	50	894	112	538	67	2404	206	2018	290	1637	286
HK97	1120	65	1037	145	618	86	3149	334	2557	447	2214	459
HP1	316	18	287	47	171	27	1433	226	1077	229	1543	276
L2	148	24	145	52	92	32	449	83	270	73	503	79
MAV1	111	5	89	17	43	7	454	80	276	50	845	66
MS2	143	12	110	10	77	9	348	49	209	38	211	51
MX1	154	14	120	27	79	18	394	40	227	39	280	39
Mu	1174	70	1001	145	544	84	3044	339	2616	435	2244	473
N15	1630	101	1500	207	896	125	4055	396	3434	547	2825	539
NL95	174	12	142	24	91	16	451	44	268	53	300	57
P2	956	39	636	71	294	42	2774	272	2087	346	1961	366
P4	316	14	252	26	129	16	828	81	700	130	671	132
PZA	176	4	141	13	71	4	1076	163	560	94	914	123
SPP1	870	125	792	188	461	101	2920	645	1889	530	2033	550
Streptococcus	261	27	276	70	115	38	1572	270	785	180	1522	229
bIL170	166	6	159	19	54	9	1143	242	544	82	1245	140
bIL67	88	4	76	10	22	3	821	158	315	47	833	81
fs-2	92	3	54	2	19	1	476	62	357	74	415	91
lambda	1334	79	1137	173	636	106	3765	441	3014	565	2651	574
psiM2	945	56	922	116	534	62	2295	308	1420	211	1368	213
minimum	0.0185		0.0370		0.0526		0.0828		0.1375		0.0781	
average	0.0624		0.1486		0.1603		0.1442		0.1842		0.1738	
maximum	0.1621		0.3586		0.3478		0.2209		0.2805		0.2705	

Table 4: Number of restriction sites cut in coding regions by enzyme sets in given bacteria for both wild-type and optimized phages.

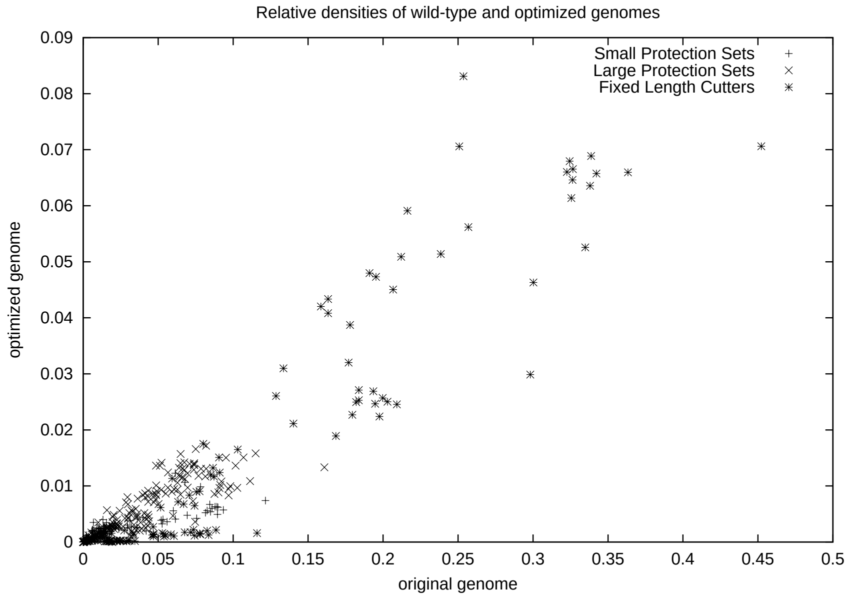

Figure 2: Density of sites in optimized vs. wild-type genomes.

To study the effect of site density on the performance of our algorithm, in Figure 2 we plot the wild-type site density versus the optimized site density for each enzyme set/phage pair. The resulting curve appears to be growing somewhat super-linearly. Certainly the heavy concentration on the x-axis suggests that we are likely to remove all sites if the density is low enough.

The spreading of the point cloud shows that expected site density is not a perfect predictor of the number of optimized sites, but that the variance grows as the density grows. This presumably reflects the fact that certain cutter sequence sets are more "interconnected" than others, meaning that pairs of complementary enzymes make it difficult to erase one site without adding another. It would be interesting to make a more detailed study of the cutter sequences associated with given organisms, to see if such complementary sets tend to be selected for.

6 Eliminating Restriction Sites through Evolution

Since eliminating restriction sites from phage genomes confers protection to a bacteriophage, it is reasonable to ask why billions of years of evolution hasn't already removed as many such sites as possible. Evolution in phages, like other viruses, can happen rapidly because of short generation times and the large number of phages that can be produced in each generation. Indeed, previous work in developing robust phages [10] has focused on using natural selection to isolate the longest-lived members of a phage population. In this section, we introduce a simple mathematical model which suggests that natural selection will be unable to remove a substantial fraction of the possible restriction sites from phage genomes, thus motivating our design approach to doing so.

We model the interaction between a given phage/host pair as a binary string of length n, where n is the length of the genome of the phage. A zero at a given position denotes the presence of a restriction site, while a one denotes the absence of a site at that position. Mutations are modeled by flipping bits in this binary string. Thus the most viable organism is one represented by a string of all 1's.

The evolution of such a genome can be modeled as a random walk on a n-dimensional hypercube

[2], since the set of all 2^n binary strings ordered by inclusion defines a hypercube of dimension n. Each point mutation takes us one step along an edge of the hypercube.

We can model such a random walk by a $(n+1)$-state finite automata, where the ith state corresponds to the set of $\binom{n}{i}$ strings with exactly i 1-bits. In each generation at state i, there is a given probability $p_{m,i}$ of a mutation flipping one of n bits. Each such mutation will transform a 0-to-1 with probability $(n-i)/i$ and a 1-to-0 with probability i/n.

Mutations to state $(i+1)$ should be selected for, because they contain one less restriction site than the genome at state i. Let p_{cut} be the probability that a given restriction site on the phage genome is cut by an enzyme during infection. Thus any forward mutation to state $i+1$ will survive with probability $(1-p_{cut})^{n-i-1}$, or $(1-p_{cut})^2$ times more than any given backwards mutation.

Equilibrium will be reached where the forward and backwards transitions occur with equal probability, i.e. at the state i_{eqi} where

$$p_{m,i}(1-p_{cut})^2\frac{i-1}{n} = p_{m,i}\frac{(n-i)}{n} \tag{1}$$

which implies

$$i_{eqi} \approx \frac{n}{(1-p_{cut})^2+1} \tag{2}$$

If each restriction site almost always lethal, i.e. $p_{cut} \sim 1$, then natural selection would drive $i_{eqi} \rightarrow n$ as suggested by the model. If restriction sites had no deleterious effect, i.e. $p_{cut} = 0$, all 2^n strings are equally favored and we predict $n/2$ 1's in the genome. We have seen that typical phage genomes can have dozens of restriction sites, hence p_{cut} must be fairly low for any significant phage population to survive.

Studies show [23] that between 1/10th to 1/100,000th of bacteria survive a phage infection. Assuming a survival value of 1/1000 for a phage with ten restriction sites yields $p_{cut} \approx 0.5$, which places equilibrium at removing only 80% of all restriction sites. Similar calculations for a phage with 100 restriction sites yields $p_{cut} \approx 0.06$ and equilibrium at only 53% of all restriction sites.

Thus we conclude that eliminating a small number of restriction sites from a genome may have such a minor impact on viability that these mutations will not be significantly selected for. However, without these intermediate steps, global optimization cannot be achieved. Under such a model, a globally optimized phage may be exponentially more likely to survive, and hence significantly more valuable as a therapeutic agent.

References

[1] B. Cohen and S. Skiena. Optimizing RNA secondary structure over all possible encodings of a given protein. In S. Miyano, R. Shamir, and T. Takagi, editors, *Currents in Computational Biology.* Universal Academy Press, 2000.

[2] P. Diaconis, R. Graham, and J. Morrison. Asymptotic analysis of a random walk on a hypercube with many dimensions. *Random Structures and Algorithms*, 1:51–72, 1990.

[3] M. Gelfand and E. Koonin. Avoidance of palindromic words in bacterial and archaeal genomes: A close connection with restriction enzymes. *Nucleic Acids Research*, 25:2430–2439, 1997.

[4] L. Goldstein and M.S. Waterman. Mapping DNA by stochastic relaxation. *Adv. in Applied Math.*, 8:194–207, 1987.

[5] D. Gusfield. *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology.* Cambridge University Press, 1997.

[6] S. Karlin, C. Burge, and A. Campbell. Statistical analysis of counts and distributions of restriction sites in DNA-sequences. *Nucleic Acids Research*, 20:1363–1370, 1992.

[7] S. Karlin, J. Mrazek, and A. Campbell. Compositional biases of bacterial genomes and evolutionary implications. *J. of Bacteriology*, 179:3899–3913, 1997.

[8] E. Lander. *Analysis with Restriction Enzymes*, pages 35–51. CRC Press, Boca Raton, FL, 1989.

[9] T. Maniatis, E. F. Fritsch, and J. Sambrook. *Molecular Cloning, a laboratory manual.* Cold Spring Harbor Laboratory, Cold Spring Harbor NY, 1982.

[10] C. Merril, B. Biswas, R. Carlton, N. Jensen, G. Creed, S. Zullo, and S. Adhya. Long-circulating bacteriophage as antibacterial agents. *Proc. Natl. Acad. Sci. USA*, 93:3188–3192, April 1996.

[11] K. Mizuuchi, B. Kemper, J. Hays, and R. Weisberg. T4 endonuclease-vii cleaves holliday structures. *Cell*, 29, 1982.

[12] D. Nathans and H. Smith. Restriction endonucleases in the analysis and restructuring of DNA molecules. *Ann. Rev. Biochemistry*, 44, 1975.

[13] L. Newberg and D. Naor. A lower bound on the number of solutions to the probed partial digest problem. *Advances in Applied Math.*, 14(2), 1993.

[14] L. Osborne. A Stalinist antibiotic alternative. *The New York Times Magazine*, pages 50–55, February 6, 2000.

[15] E. Panina, A. Mironov, and M. Gelfand. Statistical analysis of complete bacterial genomes: Avoidance of palindromes and restriction-modification systems. *Molecular Biology*, 34:215–221, 2000.

[16] C. Papadimitriou and M. Yannakakis. Optimization, approximation, and complexity classes. *J. Comput. System Sci.*, 43:425–440, 1991.

[17] R. Roberts. Rebase: the restriction enzyme database. http://rebase.neb.com.

[18] P. Sharp. Molecular evolution of bacteriophages – evidence of selection against the recognition sites of host restriction enzymes. *Molecular Biology and Evolution*, 3:75–83, 1986.

[19] S. S. Skiena and G. Sundaram. A partial digest approach to restriction site mapping. *Bulletin of Mathematical Biology*, 56:275–294, 1994.

[20] M. Stefik. Inferring DNA structures from segmentation data. *Artificial Intelligence*, 11:85–114, 1978.

[21] M. Waterman. *Mathematical Methods for DNA Sequences.* CRC Press, Boca Raton, FL, 1989.

[22] J. Watson, M. Gilman, J. Witkowski, and M. Zoller. *Recombinant DNA.* Scientific American Books, second edition edition, 1992.

[23] B. Wilkins, P. Chilley, A. Thomas, and M. Pocklington. Distribution of restriction enzyme recognition sequences on broad host range plasmid RP4: Molecular and evolutionary implications. *J. Molecular Biology*, 258:447–456, 1996.