Analysis of Targeted and Combinatorial Approaches to Phage T7 Genome Generation

by

Alexander Mallet
B.S.E. Computer Science and Engineering
University of Pennsylvania, 1996

Submitted to the Computational and Systems Biology PhD Program
in Partial Fulfillment of the Requirements for the Degree of

Master of Science in Computational and Systems Biology
at the Massachusetts Institute of Technology

Signature of Author: _____

Computational and Systems Biology PhD program
October 20, 2006

Certified by: _____

Drew Endy
Assistant Professor of Biological Engineering
Thesis Supervisor

Accepted by: _____

Bruce Tidor
Professor of Biological Engineering and Electrical Engineering and Computer Science
Chairman, Computational and Systems Biology Graduate Committee

Analysis of Targeted and Combinatorial Approaches to Phage T7 Genome Generation

by

Alexander Mallet

Submitted to the Computational and Systems Biology PhD Program on
October 20, 2006 in Partial Fulfillment of the Requirements for
the Degree of Master of Science in Computational and Systems Biology

Abstract

I performed computational analyses of various approaches to generating re-engineered versions of the genome of bacteriophage T7. I analyzed a proposed design for a re-engineered genome by examining conservation of T7 genes across related phages, and looking for RNA secondary structure arising from the re-engineered genome that might contribute to unwanted regulation. In addition, I proposed two methods of generating libraries of T7 genomes, and implemented simulations showing that the proposed methods are theoretically feasible. I conclude with thoughts on how to further validate my proposed approaches to genome generation, and suggest a specific high-throughput method of characterizing rebuilt genomes.

Acknowledgements

First and foremost, I'd like to thank my wife Christina for encouraging me to take the plunge and follow my interest in biology by going to graduate school. Without her, none of this would have happened. Secondly, I am deeply grateful to my advisor Drew and the members of the Endy lab for providing instruction, inspiration and light-hearted diversion in just the right proportions. And, finally, I am indebted to Bruce Tidor for providing invaluable feedback and perspective on both personal and academic matters.

Table of Contents

List of Figures and Tables

Chapter 1: Introduction

Recent years have seen an increased focus on understanding how all the components of a biological system interact to produce a functional whole (1-3). This shift in focus has been accompanied by the realization that rigorously-specified quantitative models of the dynamics and control of system behavior are an essential aspect of an effective systems-level view of biological processes (1, 2). However, this "systems biology" approach has encountered considerable practical problems: hand-in-hand with the increase in quantitative modeling in biology has come the realization that many computational models, even for relatively simple, well-studied systems, do not agree very well with experimental data, or cannot correctly predict the effect of novel perturbations (4,5). In addition, it is increasingly appreciated that whole-genome sequences provide only a rough outline of the functional elements encoded on the genome, and require extensive further investigation to elucidate the necessary and sufficient combinations of elements, and the interactions between these elements, needed to produce a viable organism (6). It is also becoming apparent that filling in the gaps in our knowledge by the brute force expedient of "measuring everything" may not be practical because of sheer scale. A physically accurate model of a biochemical network may require modeling thousands of possible reactions (7), yet measuring all the associated reaction rates in order to parameterize the model is infeasible with current technology. Similarly, trying to establish a list of all essential combinations of parts, by determining all synthetic lethal combinations of k genes in an organism that has a total of N genes would require performing N-*choose*-k $= \frac{N!}{k!(N-k)!}$ knock-out experiments, a number that rapidly grows beyond the practical even for relatively small N and k.

In view of these difficulties in studying naturally-occurring biological systems, a potentially attractive alternative approach is to forward- or re-engineer an existing biological system, to construct surrogates that retain the system functions of interest, but make it easier to generate the data needed for better understanding. Here, the naturally-occurring system is dissected into a set of abstract, (putatively) independent parts of known function, and then reassembled *de novo* out of physical instantiations of the functions believed to be encoded by these parts. Reassembly allows the constructed system to be optimized for manipulation, dissection and analysis. These surrogate systems can be constructed in at least two ways: via an explicitly-specified redesign, yielding a single alternative system, or by combinatorially generating libraries of surrogate systems and then choosing library members of interest for further study.

Previous work on rearranging and extending the genome of the vesicular stomatitis virus (VSV) supplies an illustration of the single instance redesign approach applied to a small system. In a series of papers, 15 specific variants of the 5-gene VSV genome were constructed, both by permuting the natural gene order and by inserting entirely new genes (60-63). Characterizing the gene and protein expression profiles of these genome variants confirmed previous reports that gene order and transcriptional attenuation are the primary mechanisms of gene expression regulation among the non-segmented negative-strand RNA virus family that VSV belongs to. In addition, all constructed genomes were viable, which revealed the insensitivity of VSV to large-scale genomic rearrangements. Re-engineering the VSV genome thus helped to both confirm existing knowledge as well as generate new insights.

Targeted re-design of a single instance has also been applied to bacteriophage T7. T7 is a lytic phage that infects *Escherichia coli*, was originally isolated in 1944 (8), and has been

extensively studied over the last 60 years. The T7 genome consists of 39,937 bp of linear double-stranded DNA, with 3 major *E.coli* RNA polymerase promoters (termed "host promoters"), 17 T7 RNA polymerase promoters (termed "phage promoters"), 3 transcriptional terminators and 10 RNase III cleavage sites (12). Figure 1 shows the approximate genomic organization of these elements (with some elements omitted for clarity).

In order to generate a version of T7 that is more easily modeled and manipulated, Chan *et al.* split the T7 genome into 6 regions, designated *alpha* through *zêta*, and abstracted it into 73 functional parts (22), as shown in Figure 2. They then redesigned the genomic sequence to remove sequence overlaps between the parts, and bracketed each part with unique restriction sites to allow easy experimental manipulation of individual parts. The resulting genome was designated T7.1, and Figure 2c shows the detailed design of section *alpha* that emerged from this process. Chan *et al.* constructed sections *alpha* and *beta,* spanning the left 11.5kbp of the 40kbp genome, and combined them with the wild-type genome to produce the chimeric phages *alpha*-WT, WT-*beta*-WT, and *alpha-beta*-WT. The resulting chimerae were all viable, with growth characteristics comparable to the wild-type isolate. These results further illustrate the utility of the re-engineering approach in increasing our understanding of naturally-occurring systems, by confirming the hypothesis that no essential functionality is encoded in the overlapping elements of the wild-type T7 genome, and providing a proof-of-principle that the T7 genome can tolerate large-scale sequence changes designed to make it easier to model viral development and manipulate physical instances of the genome.

The work described in Chapters 2 and 3 further explores the construction of alternative T7 genomes. Members of the Endy lab have continued the line of research begun with the construction of T7.1 by designing an updated version of the T7 genome, designated T7.2. In

Chapter 2, I describe my analysis of various aspects of the T7.2 design, specifically the conservation profile of the genes that are part of T7.2, and potential regulation encoded in the secondary structure adopted by the genome as it is transcribed. The work described in Chapter 3 is motivated by the observation that one limitation of approaches generating a single target genome is that they inherently only probe a single point in the vast genome design space, and targeted construction of multiple instances is generally too labor-intensive to consider on a large scale. The ability to generate and characterize genomes in a more rapid fashion is thus highly desirable. In Chapter 3, I propose two methods for combinatorial generation of T7 genomes, via facilitated loss of multiple non-essential genes or gene order rearrangement, and analyze the feasibility of these methods via computational modeling.

Chapter 2: Analyzing the T7.2 Design

While the T7.1 genome is, in principle, a surrogate that is easier to understand, model and manipulate than the wild-type genome, it is not an ideal surrogate. Seventy percent of the built and tested *alpha-beta*-WT version of T7.1 still consists of wild-type genomic sequence, containing 32 genes coding for 36 putative proteins (out of 56 genes coding for 60 proteins in the entire genome). In addition, since no genes were eliminated in T7.1, the engineered genome still contains over 20 wild-type proteins that are non-essential, most of which are non-conserved, and many of which have not been assigned a function (13). It is thus easy to envision a version of the T7 genome that is more strongly optimized for ease of understanding than T7.1.

Members of the Endy lab have designed a genome labeled T7.2 (22), which encodes a more stringently-specified version of the T7 genome than T7.1. Like T7.1, T7.2 eliminates sequence overlaps between elements. In addition, to make it easier to construct accurate computational models of phage gene and protein expression, the T7.2 design standardizes the promoters, ribosome binding sites and RNase III sites to a small set of "canonical" instances of these regulatory elements (23). To eliminate elements of unknown function, the design also calls for the removal of 21 non-essential genes. Below, I describe my efforts to contribute to the work on T7.2 by computational analysis of several aspects of the proposed design.

Phylogenetic analysis: The T7.2 design calls for the removal of 21 non-essential, non-conserved genes. The initial list of non-conserved genes came from a review of the T7 family (13), but the review did not clearly specify the criteria used to judge conservation.

To obtain more explicit data about gene conservation in T7, and possibly refine the list of genes in T7.2, I analyzed the conservation of T7 genes across the family of T7-like phage (13): T3 (24), øA1122 (25), gh-1 (26), and øYeO3-12 (27). I first extracted the coding sequences of genes in these phages annotated as being similar to T7 genes and converted the DNA sequence to the encoded amino acid sequence. I then used BLAST (51) to generate pairwise alignments of the T7 amino acid sequence to each appropriate amino acid sequence from the other phages, and calculated pairwise percentage amino acid identities. Finally, I calculated the average amino acid identity between each T7 protein and the matching proteins in all the other phage genomes. The results are shown in Table 1.

Based on the data in Table 1, the 21 non-essential genes showing the least conservation, according to number of genomes they are conserved in and average amino acid identity with respect to T7, are *0.3, 0.4, 0.5, 0.6A/B, 0.7, 1.2, 1.4, 1.5, 1.8, 2.8, 3.8, 4.1, 4.2, 4.7, 5.3, 5.5, 6.3, 7, 7.7, 19.2, 19.3*. This list includes all T7 genes that are not conserved in any of its close relatives, those conserved in only one or two close relatives, and 8 of the 14 non-essential genes conserved in three out of four close T7 relatives. A phage genome based strictly on this list of genes to remove would differ from the genome specified by T7.2 by retaining genes *1.6*, *5.7* and *5.9*, and removing genes *0.3*, *0.7*, *1.2* and *5.5*. However, genes *0.3*, *0.7*, *1.2* and *5.5* have all been assigned a function, whereas *1.6* and *5.7* have no known function. Thus, inclusion of *0.3*, *0.7*, *1.2* and *5.5* is potentially more defensible than inclusion of *1.6*, *5.7* and *5.9*, and there is no compelling reason to update the T7.2 gene list.

Eliminating potential new secondary-structure based regulation: Genomes encode information not just at the linear sequence level, but also in RNA secondary structure, which can produce regulatory signals affecting processes like translation (29-31) and mRNA

stability (32, 33). In the T7 genome, the only regions known to adopt secondary structure affecting transcription and translation are the RNAse III and transcription termination sites (12), and the 5' and 3' UTR regions of gene *10* (34). However, replacing the wild-type RBS with a standardized RBS, as proposed in T7.2, could introduce new secondary structure that might inhibit ribosome binding and mRNA translation, by allowing pairing between the RBS and the beginning of the coding sequence.

To determine the extent of secondary structure introduced by the new RBS, I used RNAfold (52) to predict the folding energies of both wild-type and engineered RBS-CDS junctions. The folded sequences were 59 bp long and, for the engineered variant, consisted of the standardized (20 bp long) RBS assigned to the given gene in the T7.2 design and the first 13 codons of coding sequence. Similarly, 20 bp upstream of the ATG start codon and the first 13 codons of coding sequence were used for the wild-type variant. The length of sequence to fold was chosen based on the fact that most known secondary structure-based regulatory elements are relatively short, and also to limit the number of sequence variants that needed to be generated and evaluated.

The average predicted ΔG of folding was relatively high (i.e. little secondary structure was predicted) for both the engineered and wild-type RBS-CDS junctions. The average ΔG was actually higher with the engineered RBS than with wild-type RBS sequences, -1.5 kCal/mol for the engineered RBS versus -2.1 kCal/mol for the wild-type RBS sequences. However, there were instances when either the engineered RBS resulted in new extensive base pairing (shown in Figure 3), or the wild-type RBS itself led to extensive base pairing (shown in Figure 4).

I also investigated the possibility of eliminating secondary structure at RBS-CDS junctions altogether. For each T7.2 gene, I generated all possible DNA sequence variants of

the first 12 codons after the start codon, allowing up to 3 alternative codons at each position. The allowed codons were the top three most frequently used codons in *E.coli,* based on published data on tRNA abundance and codon usage (53). I then prepended the standardized RBS assigned to the gene, predicted the folding energy of each shuffled sequence, and retained the sequence with the highest ΔG (i.e. least amount of secondary structure). Figure 5a shows the results of these calculations. As can be seen, some RBS-shuffled CDS sequences are predicted to have a ΔG of folding equal to 0.0 kCal/mol, and the ΔG of the RBS-CDS junction with the lowest ΔG of all the shuffled genes is still relatively high, at -5.4 kCal/mol. This RBS-CDS belongs to gene *14.3*, and has the structure shown in Figure 5b, showing very little basepairing that could potentially disrupt ribosome binding and translation. From the results above, codon-shuffling allows elimination of almost all predicted secondary structure.

Based on the hypothesis that eliminating secondary structure would eliminate the potential for translational inhibition at the RBS-CDS junction, all T7.2 genes with a predicted energy of folding less than -9.0 kCal/mol at the RBS-CDS junction were updated to incorporate the shuffled coding sequence resulting in the least predicted secondary structure. The -9.0 kCal/mol cutoff was chosen manually, by looking at the energies of structures predicted to have extensive basepairing (>= 10 basepairs).

Finding existing potential secondary-structure based regulation: To find regions in the protein-coding regions of the T7 genome that might encode secondary structure-based regulation, I utilized the DicodonShuffle algorithm developed by Katz and Burge. This algorithm generates variants of an mRNA sequence that preserve the encoded amino acid sequence, codon usage and dinucleotide composition (35). By predicting the secondary

structure adopted by these variants and using their folding energies to establish the expected background distribution of folding energies, it is possible to estimate whether the folding energy of the wild-type sequence differs significantly from what would be expected at random (i.e. in the absence of selection for secondary structure); such sequences are candidates for encoding biologically-relevant information in their secondary structure.

By combining a C implementation of the DicodonShuffle algorithm and the source code for the RNAfold package (52) into a single C program, I obtained a program that allowed the efficient generation and folding of shuffled sequence variants. I used this program to generate 1000 shuffled variants of each protein-coding RNA sequence in the T7 genome and calculated the predicted energy of folding for each position of a window sliding across the sequence. I gathered data for 50, 60 and 70 bp windows, with a step size of 10 bp between window positions. For each window position, I calculated the average and standard deviation of folding energies of the sequence variants, and then derived a z-score for the folding energy of the wild-type sequence. To find segments of the wild-type sequence that have folding energies that are significantly different from the background distribution, I looked for segments with a z-score >= -2.5758, which corresponds to sequences in the top 0.5% of the energy distribution (Figure 6 shows an example of the distribution of folding energies obtained; as can be seen, the distribution is approximately normal, justifying the use of a z-score to evaluate the significance of a particular folding energy).

Table 3 shows all 50, 60 and 70 bp regions that had highly significant folding energies. As the data show, whether a particular segment of sequence has a folding energy above the cutoff is highly dependent on the window size – most regions are not considered significant for more than one window size. Thus, the regions of most immediate interest are the ones which do exhibit a significant amount of predicted secondary structure across multiple

window sizes for the same, or closely-spaced, starting positions. There appear to be 3 such regions in the T7 genome (highlighted in Table 3): positions 1030-1130 in gene *8,* positions 380-470 in gene *15*, and positions 2380-2470 in gene *16*. None of these locations is near the transcriptional terminators, RNAse III sites or other sites known to have secondary structure-based regulatory functions in T7 (12, 34). The minimum-energy secondary structures predicted for these 3 regions by the Mfold server (58, 59) are shown in Figure 7. As expected, these sequences show extensive basepairing, and thus may affect translation (29-31).

Based on this analysis, the three regions listed above seem to be the best initial candidates for codon-shuffling to remove potential "cryptic" regulation encoded by the RNA secondary structure of protein-coding regions of the T7 genome. Should the T7.2 work ever proceed to the point of constructing regions of the genome that include genes *8, 15* or *16*, it would be worth considering codon-shuffling these regions to remove/reduce their secondary structure, if this can be done without introducing significant new structure in nearby sequences.

Chapter 3: Library-based Approaches to Genome Generation

As mentioned earlier, there are limits to the scalability of the re-engineering approach based on constructing specific genome instances. Below, I describe and analyze two potential approaches to generating libraries of T7 genomes with reduced or re-ordered gene sets.

Design of a "lossy" genome: A systems-level understanding of a biological entity requires knowing which combination of parts is essential to system function. However, determination of synthetic lethal subsets of genes by direct deletion of gene sets is largely infeasible due to the combinatorial explosion of possible gene subsets to delete. To allow efficient generation of a large number of genomes with reduced gene sets, one possibility is to construct a genome that is prone to gene deletions, and evolve it over many generations to allow accumulation of gene deletions. Construction of such a "lossy" genome could utilize the fact that direct repeats in the T7 genome can recombine during T7 DNA replication, leading to deletion of the intervening sequence, as initially reported by (37-39). Recent data also confirms the phenomenon of recombination between repeats: experimental evolution of the T7.1 genome resulted in elimination of several of the repeats introduced into the T7.1 genome (IJ Molineux, personal comm.).

This mechanism of sequence deletion could possibly be exploited to design a lossy genome, by extending the T7.2 genome design to include repeat regions around all remaining non-essential genes. The repeat-enriched genome could then be subjected to multiple rounds of evolution to generate genomes with differing gene sets (Figure 8), and isolates exhibiting growth and fitness characteristics that differ significantly from the progenitor phage could be sequenced. Analysis of the final set of sequenced genomes would

then allow determination of which genes have been lost over the course of the experiment and hence are dispensable (or important) for viability.

To determine whether this approach was even theoretically feasible i.e. would result in enough genomes with reduced gene sets, I simulated the rate of gene loss across multiple cycles of phage growth. Specifically, I simulated the effect of inserting repeats between each of the T7 genes *1.1 – 1.8*, and then subjecting the resulting phage genome to multiple growth cycles, interspersed with serial dilutions and transfers, similar to the protocol described in (43, 44). This set of genes was chosen for being a mix of genes of known and unknown functions, and thus potentially a good candidate set for investigation using my proposed genome construction scheme.

My simulation was based on the following assumptions:

- The same repeat is inserted between all eight genes, leading to multiple possible recombination events
- The repeats are 20bp long
- The recombination rate per lysis cycle varies linearly with the distance between repeats, with the following recombination rates used to calculate the probability of a particular recombination event: 1 in 1600 for repeats 100bp apart, 1 in 8000 for repeats 900bp apart, and no recombination if repeats are > 1100 bp apart. These recombination rates are based on data from (56, 57).
- Each gene is 350bp long i.e. the distance between repeats is 350bp; this corresponds to the actual average length of genes *1.1 – 1.8*.
- The burst size is 100 i.e. 100 new virions are produced per each infected cell, based on (9). This corresponds to the parameter *b* below.

- There are 3 cycles of viral growth and cell lysis between each serial dilution and transfer; this is represented by the parameter *r* below.
- The probability of multiple recombination events occurring in a single genome during one viral growth cycle is low enough to be negligible

The algorithm for calculating gene loss across multiple cycles of phage growth is described below, where a "genome family" is defined as a set of genomes with a particular number of genes (regardless of what the actual genes are):

1. Start with a founder population consisting of a single genome family, with all 8 genes
2. During each round of cell culture, for each genome family *k* (ie consisting of genomes with *k* genes and *k+1* repeats), with *k* decreasing from 8 to 0:
    a. Calculate the number of progeny phage expected as $N_{new} = N^{k}_{current} * b^{r}$, where N_{new} is the total number of new progeny phages from genome family *k*, $N^{k}_{current}$ is the current number of phages in genome family *k*, *b* is the burst size and *r* is the number of cell lysis cycles
    b. Calculate the probability distribution of a particular number of genes being lost from a member of genome family *k*
    c. Use the probability distributions created in the previous step to calculate the partition of the N_{new} phages into phages with $i <= k$ genes i.e. set $N^{i}_{new} = N_{new} * p^{k}_{i}$, where N^{i}_{new} is the new number of phages with *i* genes and p^{k}_{i} is the probability of a genome with *k* genes losing *(k-i)* genes to result in a genome with *i* genes
    d. Update the number of phages in each genome family with $j <= k$ genes with the numbers calculated in the previous step ie set $N^{j}_{current} = N^{j}_{current} + N^{j}_{new}$

3. "Dilute" the phage population by proportionally dividing up the number of phages in each genome family to maintain a constant phage population size. (This is the equivalent of performing a serial dilution and transfer of the lysate from a viral culture into a fresh culture of the host cells).
4. Repeat step 2 for the desired number of serial transfers
5. The final result is the number of phages in each genome family

The probability distribution for losing *i* genes from a *k*-gene genome was calculated by calculating the probability of each of the *(k+1)-choose-2* = $\frac{k(k+1)}{2}$ possible recombination events, based on the distance between repeats, and summing up the probabilities for the number of genes eliminated by each possible recombination event.

The results of this simulation are shown in Figure 9 and Table 2. As the data show, the estimated rate of recombination, and hence gene loss, between 20bp repeats is too low to allow easy generation and isolation of genomes that have lost multiple genes. However, increasing the recombination rate by a factor of 50-fold would result in a phage population in which virtually all phages have lost at least one gene, and the majority have lost multiple genes. Since the rate of recombination increases 500-fold when the repeat length increases from 20bp to 10bp (38), it seems reasonable to think that the desired 50-fold increase in recombination rate could be achieved by increasing the repeat length to 30 or 40bp. Thus, the repeat-based approach to constructing a lossy genome seems at least theoretically feasible.

Generating shuffled genomes: Due to the relatively slow rate of entry of the T7 genome into an *E.coli* cell during infection (14, 15), there are large differences in the total time available

for transcription and translation of genes that are widely separated on the genome. In addition, the promoter organization on the genome leads to genes being transcribed from differing numbers of promoters (12). Thus, the overall amount of mRNA and protein produced by a gene is affected both by the strength of genomic regulatory elements driving its transcription and translation, and its position on the genome.

A systems-level understanding of the effect of genomic organization on the T7 lifecycle would be reflected in an ability to accurately model the effects that reordering genomic elements has on gene expression. The work by Endy *et al.* to characterize reorganized genomes (21) was a step in this direction, but was limited in the amount of data that could be gathered, due to the large amount of work needed to construct these genomes. Thus, the possibility of being able to easily generate and characterize many rearranged genomes, and use the generated data to refine our model, is appealing.

One possible method of generating a library of permuted genomes is to use a combination of the DNA shuffling technique pioneered by WP Stemmer (45, 46) and work by Tsuge *et al.* that demonstrated efficient *in vitro* assembly of multiple DNA fragments in a designed order and orientation (47). In DNA shuffling, DNA sequences containing regions of homology but differing from each other by, for example, point mutations, are fragmented by DNase I treatment and then allowed to reassemble by multiple cycles of annealing and extension in the presence of DNA polymerase. The regions of homology guide the reassembly, resulting in a shuffling of the sequences as depicted in Figure 10a. Tsuge *et al.* were able to assemble multiple genes in a designed order and orientation by ligating together gene sequences with protruding sequences at both ends; the protruding ends determined the order of assembly, as shown in 8b.

Combining a homology-based approach to shuffling sequences with a mechanism allowing control of the order and orientation of reassembly would allow creation of a library of elements assembled in various orders. Such a library could be constructed by generating an ensemble of individual elements flanked by protruding sequences and allowing these sequences to guide assembly via annealing and ligation. An example of this, applied to three elements, denoted E1 – E3, is depicted in Figure 11.

As shown, arbitrary permutations of elements could be generated via the appropriate ordering of overhang sequences annealing to each other and being ligated together. The length of the assembly could be controlled via the 5' and 3' "caps", which are sequences that stop extension in either the 5' or 3' direction, by having an overhang on only one end. In addition, the overhang sequences can be designed to have a unique pairing, thereby avoiding "cross-talk" that could lead to arbitrary-length assemblies despite the presence of the capping sequences. The caps would also allow amplification and purification of generated assemblies: PCR primers specific to the caps can be used to amplify only capped assemblies, which can then be purified via gel electrophoresis and extraction of bands of the appropriate length. Thus, this approach could allow generation of permuted T7 genome segments by shuffling individual segments containing one or more genes.

One potential problem with my proposed library construction scheme is that the desirable assemblies, specifically the ones containing a complete set of the shuffled genes, with no repeated genes, may be a very small fraction of the total assembly pool. Analytically, if k genes are being shuffled, there are k^k possible k-gene assemblies, of which $k!$ contain no repeated genes (hereafter called "complete" assemblies). Thus, complete assemblies make up $\frac{k!}{k^k}$ of the total assembly pool, which may be a relatively small fraction. However,

purification of complete assemblies can be optimized by observing that assembly length provides a rough guide to the contained genes. Thus, it is possible to enrich for complete assemblies by running the assembly pool out on a gel and extracting only bands that are approximately the length of a complete assembly, thereby generating a pool of "restricted length" assemblies.

To estimate the enrichment provided in this manner, I simulated shuffling 5 segments covering the region from gene *1* to gene *3.5* in the T7.2 design. This region contains 5 essential genes that are the main contributors to regulating transcription and duplication of the phage genome (genes *1, 2, 2.5, 3, 3.5*), and are thus a good candidate set for exploring the impact of genome ordering on phage gene expression. The boundaries of the 5 segments were as shown in Figure 12. I simulated the generation of one million assemblies via the algorithm described below.

I calculated the fraction of complete assemblies, as well as the fraction of assemblies that had lengths within 10% of the length of a complete assembly (the "restricted length" assemblies), and would be indistinguishable from complete assemblies on a gel. The fraction of complete assemblies was 3.8% (in agreement with the analytical solution), and the fraction of restricted-length assemblies was 21.8%; thus, complete assemblies make up 3.8/21.8 = 17.4% of the restricted-length assemblies, an approximately 4.5-fold enrichment.

Presumably, this technique could be made even more effective by optimizing the lengths of the shuffled elements to maximize the difference in length between complete assemblies and all other assemblies. It thus seems reasonable to assume that a ligation reaction containing the appropriate DNA sequences, coupled with length-based enrichment for complete assemblies will allow generation, isolation, and subsequent characterization of T7 variants with permuted gene orders.

Simulation algorithm:

L is the number of allowed ligation reactions per assembly and was set to 100; *pCapping* is the probability of a ligation reaction adding a cap to the current assembly and was set to 0.1.

- For *N* iterations
    - Seed the assembly by uniformly picking a random starting segment and assigning it 5' and 3' linkers
    - For the allowed number *L* of ligation reactions per assembly:
        - Uniformly generate a random number *randNum* in the range 0 to 1
        - If *randNum* < *pCapping* and the assembly can be capped on the 5' end, cap the assembly on the 5' end
        - Else If (*randNum* >= *pCapping* && *randNum* < *2*pCapping)* and the assembly can be capped on the 3' end , cap the assembly on the 3' end
        - Else
            - Uniformly generate a segment and 5' and 3' linkers
            - If the new segment can be ligated to the current assembly on the 5' or 3' end (i.e. the segment's 5' linker matches the 3' linker of the assembly, or vice versa), add it to the assembly
        - If the assembly is complete (i.e. is capped on the 5' and 3' ends, and has the desired number of segments), terminate this set of ligation reactions
- Generate statistics for the number of complete and restricted-length assemblies

Chapter 4: Conclusions and Future Work

Portions of the T7.2 genome containing some of my proposed changes to reduce RNA secondary structure at the RBS-CDS boundary have been commercially synthesized and are currently being assembled. Assuming the designed genome passes the most basic test, namely whether it leads to a viable phage, the phage encoded by this genome will need to be carefully characterized to determine whether the desired ability to more accurately model the T7 system has been achieved.

The simulations described in Chapter 3 show that my proposed approaches to generating libraries of T7 genomes are theoretically feasible. The obvious next step is thus to attempt to validate these approaches experimentally. Construction of a genome that easily loses genes (i.e. the "lossy" genome described in chapter 3) is predicated on being able to increase the rate of recombination between direct repeats by about 50-fold above the recombination rate measured between 20bp repeats. Whether this increase can be achieved via my suggestion of using longer repeats can be tested by constructing genomes with longer repeats and measuring their recombination rate via the method described in (38). Should the results appear encouraging, construction of a genome with multiple such repeats can then be attempted either via commercial synthesis of the desired genome or manual insertion of repeats into, for example, the T7.1 genome. This genome can then be evolved and isolates sequenced to determine whether gene loss is occurring at an acceptable rate.

My suggested method of obtaining genomes with shuffled gene orders can tested on a small scale at first, by attempting to shuffle 2 or 3 segments. If each segment includes an essential gene, only assemblies containing all segments will lead to viable phages, thereby allowing the use of plaque formation as a strong screen for selecting phages that have incorporated complete assemblies. These phages can then be partially sequenced (for

example, via sequencing microarrays), or otherwise characterized (by PCR, for example), to determine whether the desired shuffling of segment order has occurred. It should be noted that even if these experiments show that the number of segments that can be effectively shuffled at one time is relatively small, the method can be applied in a hierarchical fashion to shuffle larger portions of the genome. For example, suppose one experiment shuffles genes *1*-3, a second experiment shuffles genes *4-6,* and a third experiment shuffles genes *7-9*. One complete assembly from each of these experiments can then be used in a fourth shuffling experiment, thereby permuting the order of genes *1-9*. Although not all possible permutations are accessible via this hierarchical approach, it should allow generation of genomes with significantly changed gene ordering.

Once the re-engineered genomes have been constructed, whether according to a specific design or in a combinatorial manner, their utility to the scientific and engineering enterprise will in large part be determined by how easy it is to characterize them, for example, how quickly they can be sequenced and their gene expression profile measured. If this data cannot be generated quickly enough to allow rapid testing of desired characteristics, or refinement of existing models of the system, the appeal of re-engineering genomes rapidly diminishes. From this perspective, the use of microarrays for high-throughput sequencing and gene expression measurements of the generated T7 instances is appealing. In particular, I conducted an initial survey of microarray manufacturers and found that, at the time of this writing, Nimblegen arrays custom-designed for T7 seem to offer the ability to sequence T7 genomes variants, and measure gene expression profiles, at a reasonable price. Future work on re-engineering T7 would presumably benefit from continuing this line of investigation.

Table 1: T7 genes conserved in close relatives of T7. Numbers in parentheses are average amino acid identity, P. *T7 genes not conserved in any other genome: 0.4, 0.5, 1.4, 4.1, 2.8.*

T7 genes conserved in 4 genomes (29)	T7 genes conserved in 3 genomes (14)	T7 genes conserved in 2 genomes (8)	T7 genes conserved in 1 genome (2)
P>=80%: 8 (84.5), 3.5 (83.5), 5 (83.5), 5.7 (82.3), 4A (80.3), 17.5 (80.3) *70%<= P < 80%:* 19 (79.0), 1 (78.5), 3 (77.5), 18 (77.0), 2.5 (75.5), 16 (75.3), 6 (75.3), 12 (71.75) *60%<=P<70%:* 11 (69.5), 15 (68.8), 10A (67.8), 18.5 (67.8), 17 (65.0),1.3 (64.8), 13 (62.0), 6.5 (61.8), 14 (61.5), 18.7 (61.3), 9 (60.0) *40%<=P<60%:* 6.7 (59.3), 7.3 (54.3) , 2 (49.8), 1.1 (47.8)	*P >= 80%:* 5.9 (81.7), 4.5 (81.3) *50%<=P<80%:* 19.5 (75.0), 1.6 (71.0), 4.3 (61.0), 1.7 (59.3), 19.2 (57.7), 1.8 (55.7), 1.2 (54.0), 19.3 (53.3) *P<50%:* 5.5 (46.0), 1.5 (33.3), 0.3 (31.7), 6.3 (18.3)	*P >= 80%:* 4B (86.0) *P <80%:* 7 (66.0), 3.8 (65.0), 0.7 (44.0), 4.2 (42.0), 5.3 (25.0), 0.6A (45.0), 0.6B (25.0)	*P >= 80%:* 7.7 (99.0), 4.7 (87.0)

Tables 2a-d: Population fractions after 20 serial transfers. a) 1x recombination rate b) 10x recombination rate c) 50x recombination rate d)100x recombination rate

a)

Genes lost	0	1	2	3	4	>= 5
Fraction of pop.	0.89	0.067	0.034	0.0052	0.0004	~0.0

b)

Genes lost	0	1	2	3	4	5	>=6
Fraction of pop.	0.31	0.26	0.23	0.12	0.05	0.01	~0.0

c)

Genes lost	0	1	2	3	4	5	6	7	8
Fraction of pop.	0.001	0.011	0.048	0.125	0.21	0.26	0.22	0.12	0.005

d)

Genes lost	<3	3	4	5	6	7	8
Fraction of pop.	~0.0	0.004	0.03	0.14	0.33	0.465	0.03

Table 3: T7 coding regions with predicted folding energies in the top 0.5% of the energy distribution. Position is in bp, relative to each gene's start codon.

Gene	Position	Length	ΔG(kJ/mol)	z-score	Gene	Position	Length	ΔG(kJ/mol)	z-score
gene 0.3					gene 3.5				
	280	50	-14.2	-3.64111		340	70	-30.4	-2.73884
	280	60	-14.2	-2.96326	gene 3.8				
gene 0.6B						0	60	-14.7	-2.59481
	180	50	-16.8	-2.61144		20	60	-18.3	-2.61263
	190	60	-17.6	-2.71198	gene 4B				
	190	70	-23.3	-2.62879		150	60	-14.42	-3.19413
	230	50	-15.86	-2.71517		150	70	-18.52	-3.3531
	240	70	-19.9	-3.07339		160	50	-13.12	-2.96982
	250	70	-20.7	-2.82969		960	70	-18.1	-2.60082
	260	60	-16.5	-2.59225		1000	60	-21.7	-3.09524
	270	50	-16.5	-3.05621		1110	60	-18.9	-2.97165
	270	60	-18.2	-2.94809		1110	70	-19.6	-2.57929
gene 0.7					gene 4A				
	540	50	-13.1	-2.80733		340	60	-14.42	-3.03881
	540	70	-16.8	-3.12602		340	70	-18.52	-3.34022
	550	60	-16.8	-3.51762		350	50	-13.12	-2.97091
	550	70	-21.5	-2.94958		1000	60	-19.7	-2.94623
gene 1						1180	70	-23	-2.67322
	460	60	-19.1	-2.73702		1190	60	-23	-3.33644
	1150	50	-13.5	-2.73767		1300	60	-18.5	-3.08238
	1980	60	-22.2	-2.61018		1300	70	-18.5	-2.57852
gene 1.2					gene 4.2				
	120	50	-12.6	-2.832		10	50	-12	-2.70659
gene 1.3					gene 4.3				
	90	50	-13.2	-2.75489		50	70	-25.4	-2.58209
	100	50	-14.9	-2.81011	gene 4.7				
	800	60	-21	-2.62639		50	50	-8.8	-2.59975
gene 1.6					gene 5				
	70	50	-13.1	-2.60906		100	60	-29.1	-2.68677
	70	50	-13.1	-2.60906		580	50	-17.8	-2.70545
gene 1.7						1010	50	-17.3	-2.97194
	300	50	-11.3	-2.70597		1010	60	-17.8	-2.92285
	430	70	-22.5	-3.64224		1050	50	-16.5	-2.9778
gene 1.8						1170	50	-15.7	-2.94948
	60	60	-11.6	-2.60548		1260	50	-16	-3.08574
	60	70	-16.6	-2.76666	gene 5.5				
gene 2.5						210	50	-9.6	-3.09156
	330	60	-12.3	-2.63428	gene 6				
	330	60	-12.3	-2.63428		380	60	-22.4	-2.73383
gene 3						540	70	-24.1	-2.80288
	20	70	-26.6	-3.45901		570	70	-28.3	-3.04026
	30	60	-22.7	-3.56705		780	60	-18.7	-2.58139
	30	70	-22.7	-3.56958	gene 7.3				
						20	60	-15.2	-3.31563

Table 3 continued:

Gene	Position	Length	ΔG(kJ/mol)	z-score	Gene	Position	Length	ΔG(kJ/mol)	z-score
gene 7.7					Gene 15 contd	*410*	*50*	*-18.6*	*-3.4317*
	160	60	-17.2	-2.77539		*410*	*60*	*-22.3*	*-3.44345*
	190	60	-14.3	-2.85222		450	60	-19.1	-2.88564
gene 8						820	50	-16.7	-2.72836
	310	60	-15.8	-2.71693		1080	70	-20.6	-2.88472
	490	70	-22.7	-2.88735		1470	50	-16.1	-3.39772
	500	50	-17.1	-2.69651		1750	60	-17.8	-2.57624
	880	60	-21.7	-2.93767		1760	50	-13.4	-2.73324
	1030	*70*	*-24.7*	*-3.00268*		1760	60	-14.7	-2.92349
	1040	*50*	*-13.9*	*-2.81419*		2040	70	-17.6	-2.61361
	1040	*60*	*-22.1*	*-3.41487*		2050	60	-15.8	-2.70918
	1040	*70*	*-30.1*	*-3.43546*		2200	44	-11.9	-2.87741
	1050	*50*	*-18.7*	*-2.81444*	gene 16				
	1050	*60*	*-26.9*	*-2.75018*		580	60	-21.1	-3.08343
	1060	*50*	*-19.3*	*-3.02176*		580	70	-22.8	-2.95061
	1060	*70*	*-26.8*	*-3.3436*		600	50	-14.4	-2.76394
	1330	50	-25.1	-2.81657		1030	70	-24.1	-3.50266
gene 9						*2380*	*70*	*-24.5*	*-3.1373*
	210	60	-24.2	-2.64901		*2390*	*60*	*-21.8*	*-2.7308*
	210	70	-30.4	-3.59211		*2390*	*70*	*-27.4*	*-3.35353*
	220	50	-23.9	-3.05274		*2400*	*50*	*-20.9*	*-2.94204*
	220	60	-24.7	-2.94155		*2400*	*60*	*-23.9*	*-3.28397*
	440	50	-16.5	-2.60493		*2400*	*70*	*-25.1*	*-2.92623*
	690	60	-21.8	-2.80847		*2410*	*50*	*-16*	*-2.68987*
gene 10A						2580	50	-14.8	-2.96855
	810	70	-18.7	-2.70405		2610	60	-20.3	-2.69359
	830	60	-18.3	-3.64819	gene 17				
	850	70	-21.3	-2.68848		80	50	-15	-2.62606
	860	60	-21.3	-3.05664	gene 18.5				
	860	70	-26.3	-2.89422		80	50	-10.6	-2.86737
gene 12						80	60	-12.4	-2.82055
	650	50	-17.4	-2.79309	gene 19				
gene 13						400	50	-16.6	-2.76465
	1260	70	-22.2	-2.72393		640	50	-16.3	-3.30984
	2000	70	-26.6	-2.74194		1010	60	-16.8	-2.99748
	2400	70	-25.1	-2.92623		1020	50	-10.9	-2.91289
gene 14									
	210	60	-19.4	-3.30136					
gene 15									
	380	*70*	*-30.2*	*-3.94986*					
	390	*60*	*-19.8*	*-2.86099*					
	390	*70*	*-22.3*	*-2.62666*					
	400	*50*	*-17.1*	*-3.33304*					
	400	*60*	*-19.9*	*-3.25392*					
	400	*70*	*-23.3*	*-2.8975*					

Figure 1: T7 genome organization. Vertical green lines with half bars: host promoters; vertical blue lines with half bars: phage promoters; vertical orange lines with full bars above the genome: transcriptional terminators; vertical purple lines with full bars below the genome: RNAse III sites.

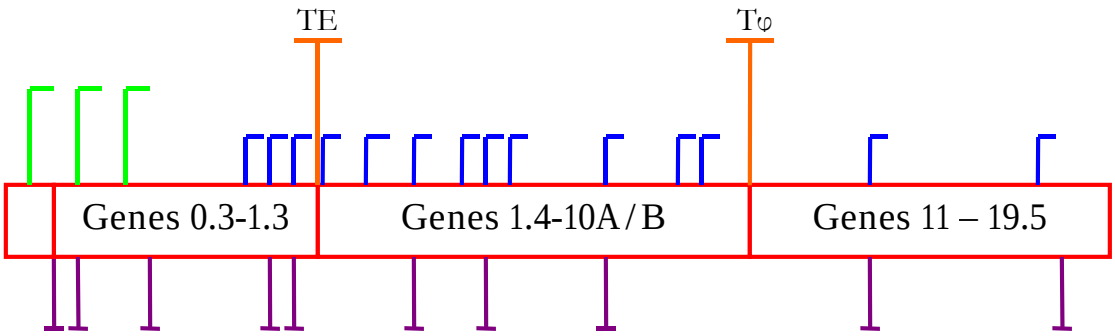


Figure 2: T7.1 genome design. Partial reproduction of Figure 2 from Chan *et al* (22). (A) The wild-type genome was split into 6 sections, *alpha* through *zêta*, using 5 restriction sites unique across the natural sequence. (B) Wild-type section *alpha* genetic elements: protein coding regions (blue), RBSs (purple), promoters (green), RNAse III recognition sites (pink), a transcription terminator (yellow) and others (gray). Images are not to scale, but overlapping boundaries indicate elements with shared sequence. The five useful natural restriction sites across section *alpha* are shown (black lines). (C) T7.1 section *alpha* parts. Parts are given integer numbers, 1-73, starting at the left end of the genome. Unique restriction sites bracket each part (red/blue lines, labeled D[part #]L/R]. Added unique restriction sites (purple lines, U[part #]) and part length (# base pairs, open boxes) are shown.

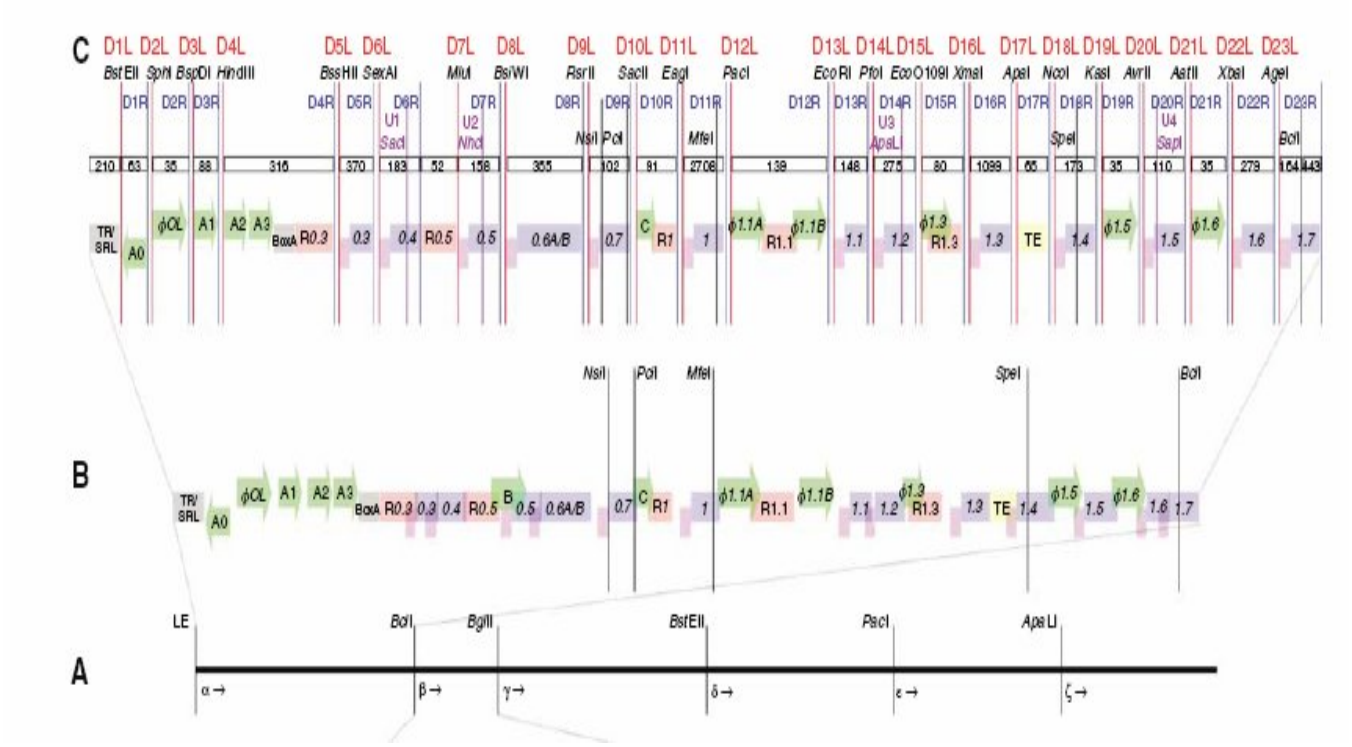
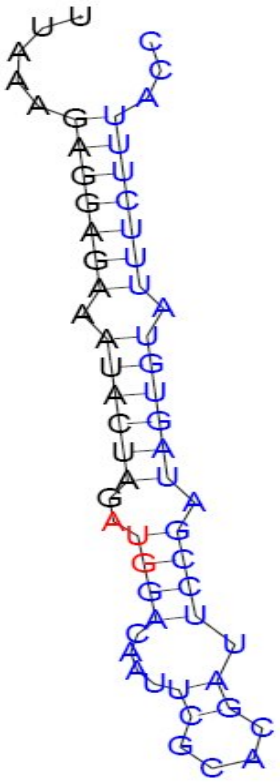


Figure 3. Predicted RNA structures at standardized RBS-CDS junction. Start codons in red. (A) Predicted RNA structure for gene 4A with standardized RBS. B) Predicted RNA structure for gene 11 with standardized RBS.

A)



B)

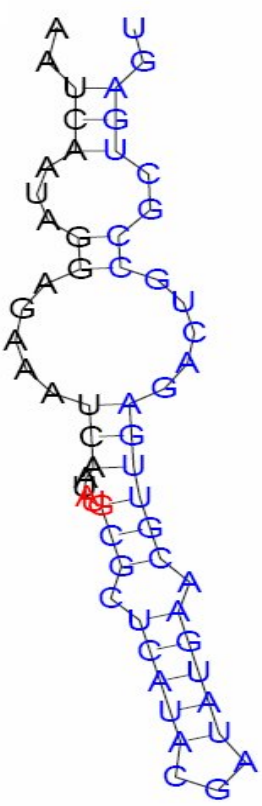


Figure 4: Predicted RNA structures at wildtype RBS-CDS junction. Start codons in red. (A) Predicted structure for gene 17 (B) Predicted structure for gene 15

(A) (B)

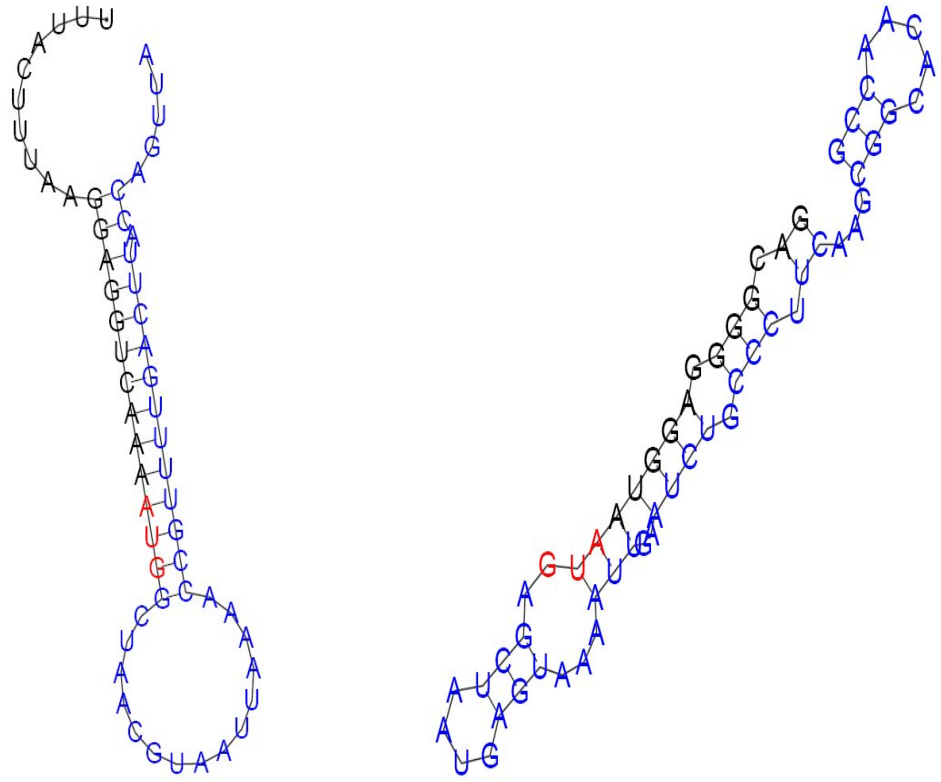
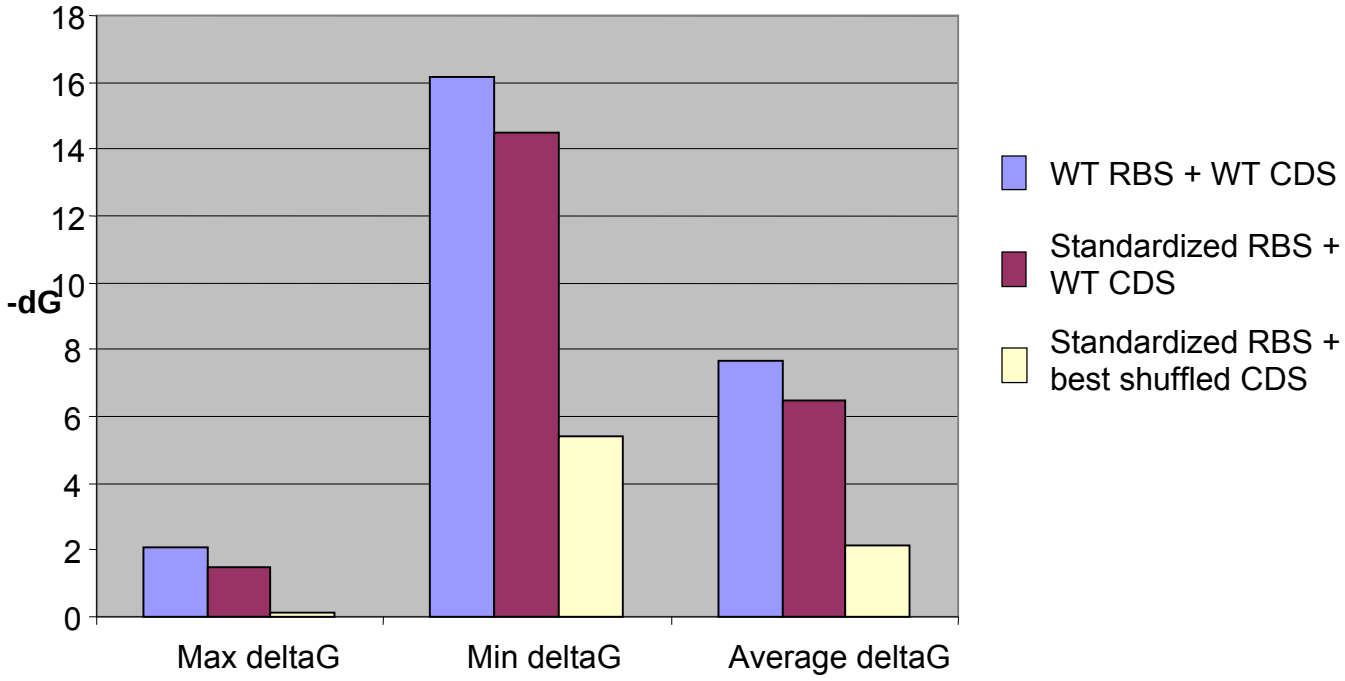


Fig 5. Effects of codon-shuffling. (A) Lowest, highest and average -ΔG across all genes in 7.2, for various RBS-CDS combinations. WT = wild-type. (B) Predicted RNA structure for gene 14.3; ΔG = -5.4 kCal/mol

(A)



(B)

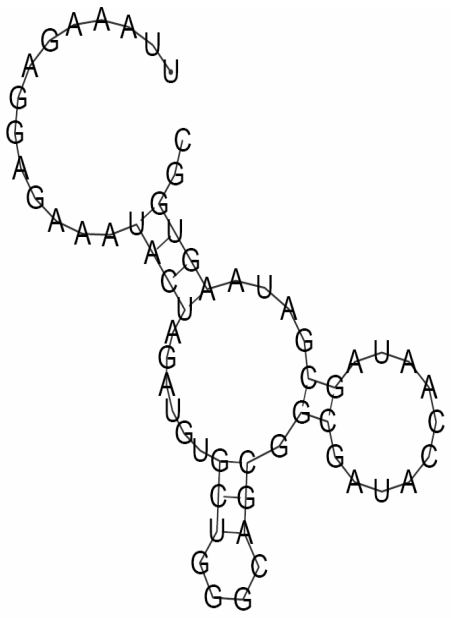


Figure 6: Histogram of shuffled segment folding energies for positions 60-110 of gene 1

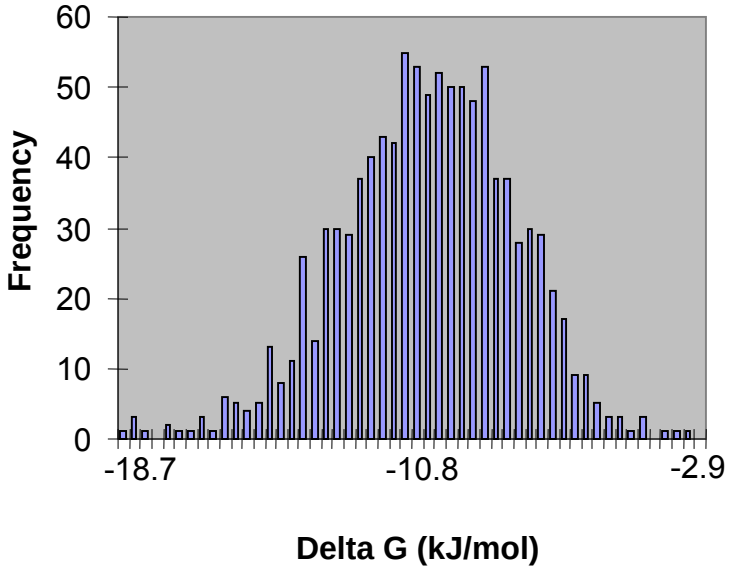


Figure 7: Predicted secondary structures for regions of T7 genome with significant folding energies.

(A) Bases 1030-1130 of gene *8;* predicted ΔG = -39.1 kJ/mol

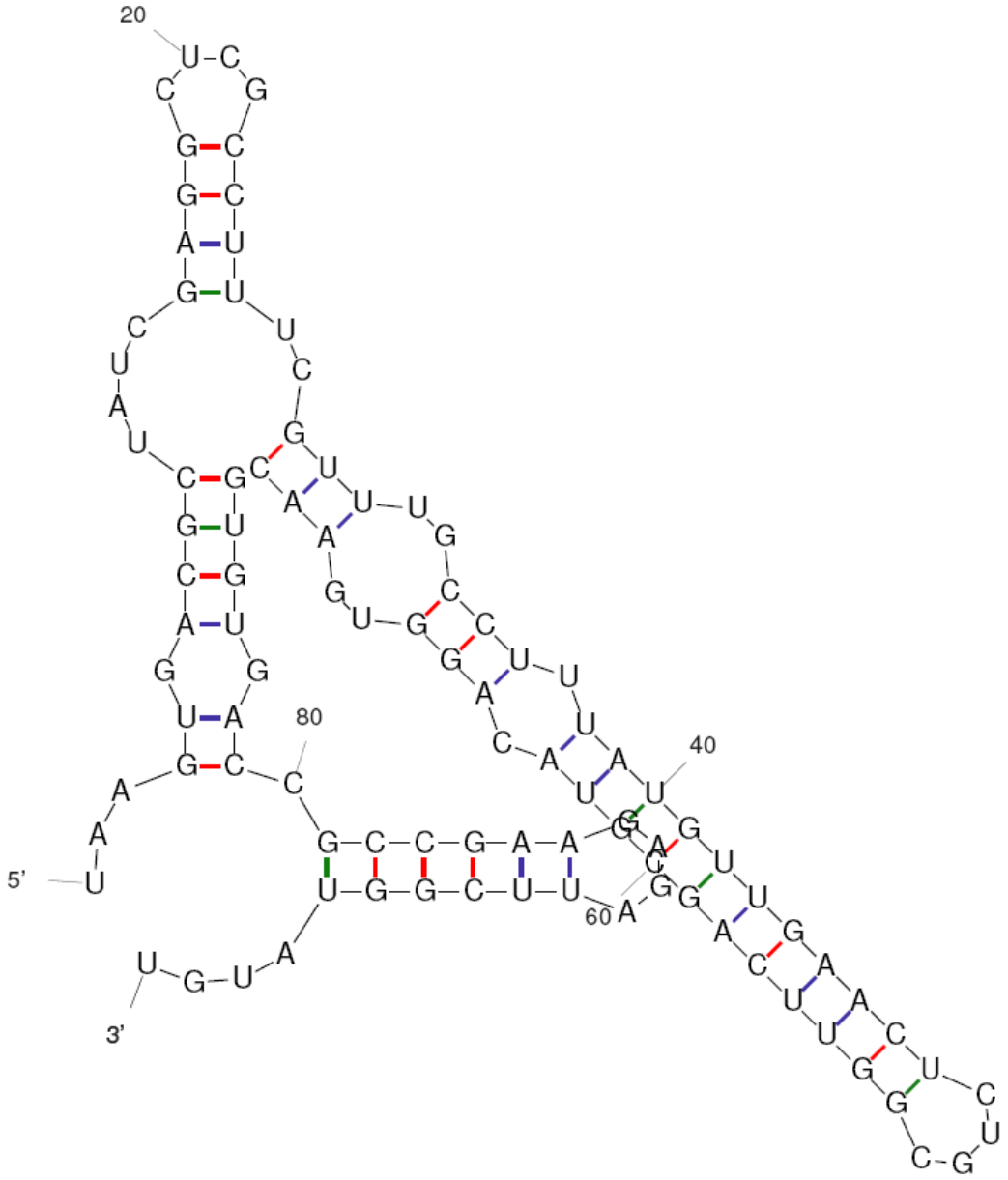


Figure 7B) Bases 380-470 of gene *15*; predicted ΔG = -37.8 kJ/mol

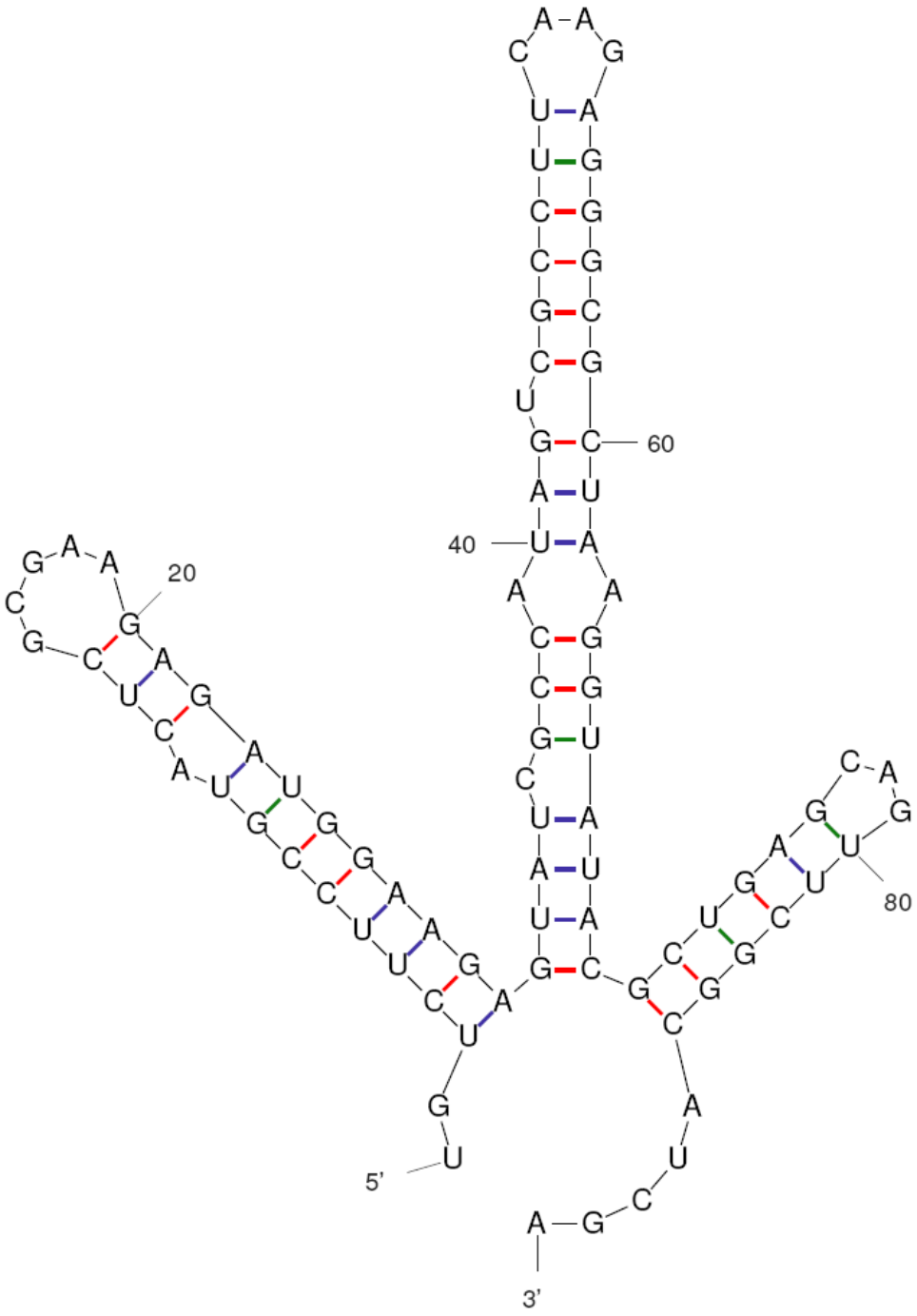


Figure 7C) Bases 2380-2470 of gene *16*; predicted ΔG = -31.0 kJ/mol

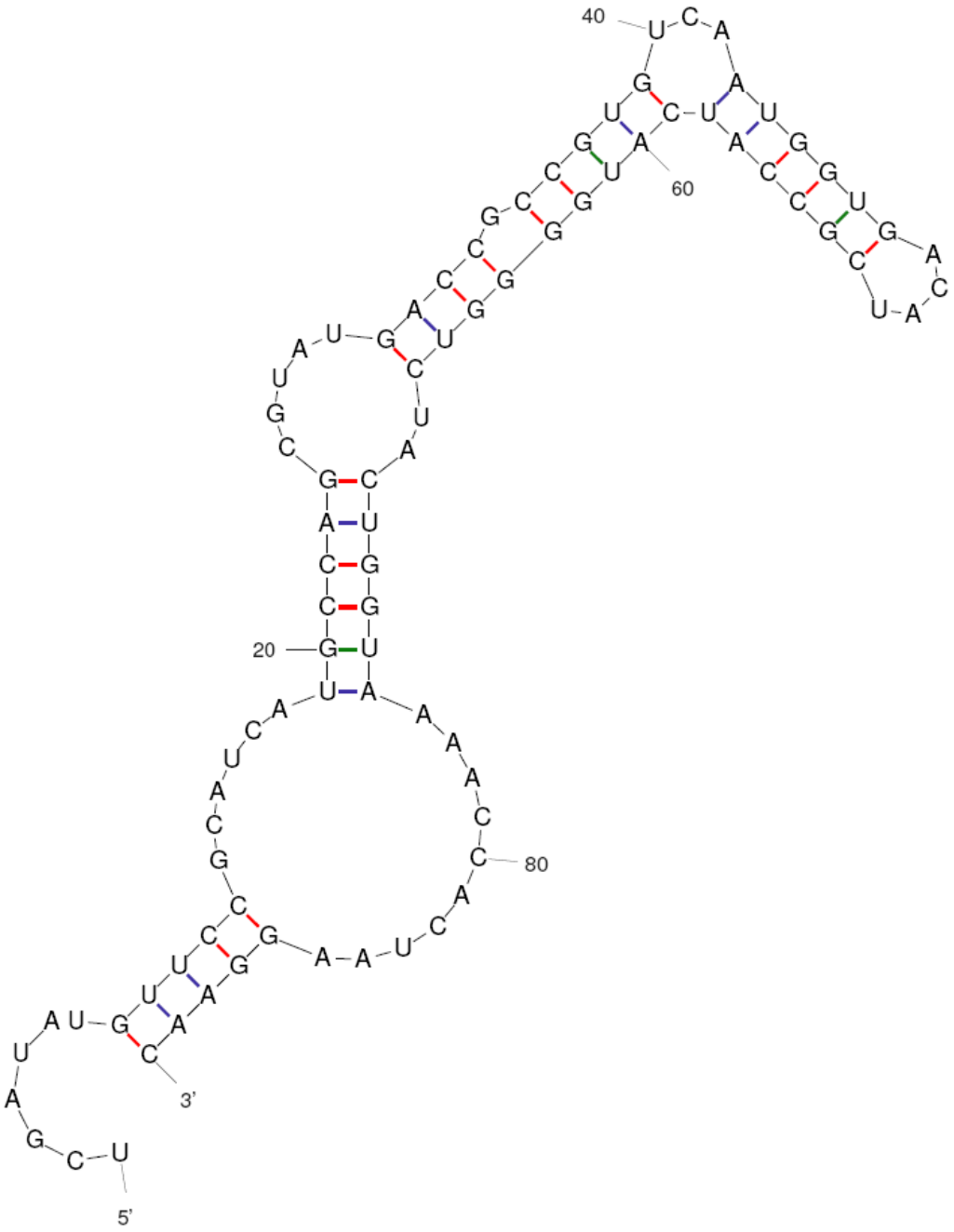


Figure 8: Effects of recombination between direct repeats. Recombination during genome replication can lead to a library of genomes with differing gene sets.

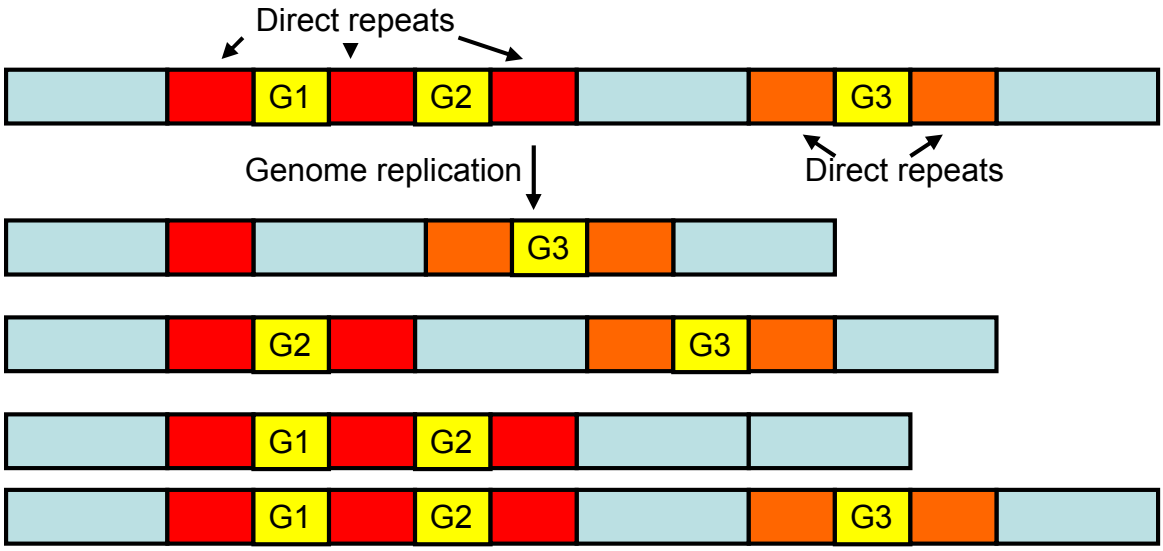


Figure 9: Distribution of genome population after 20 serial transfers

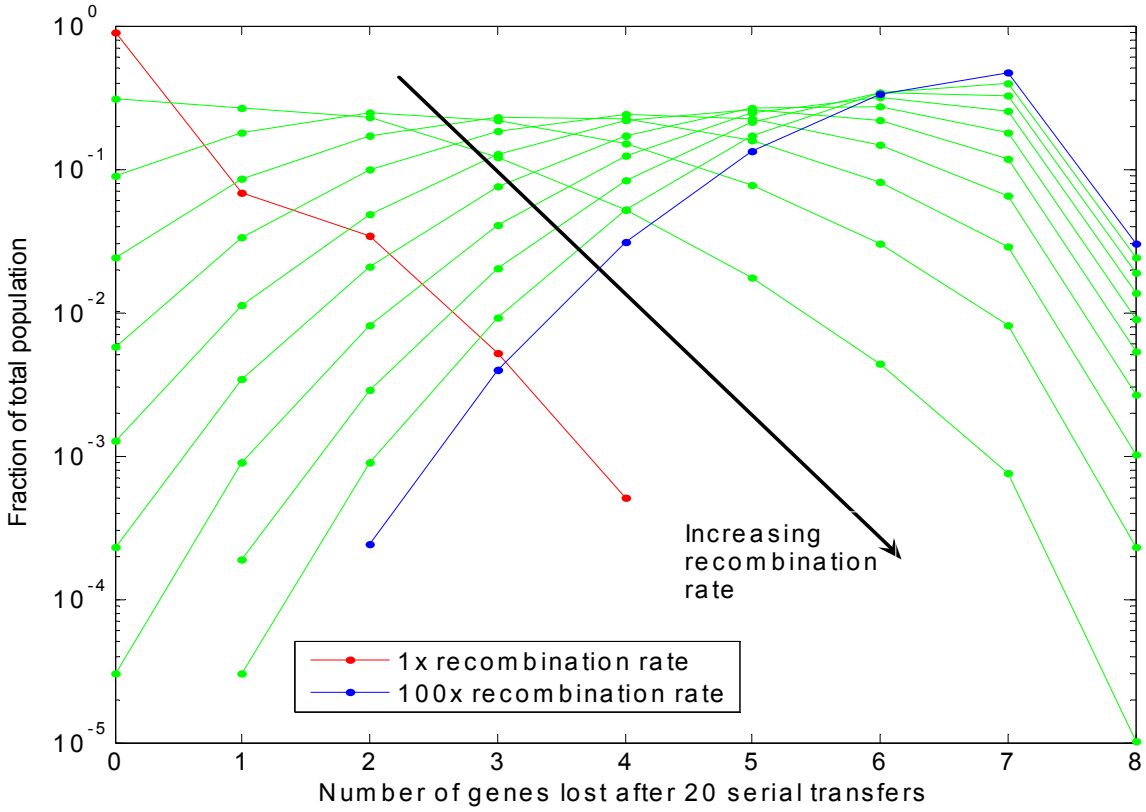
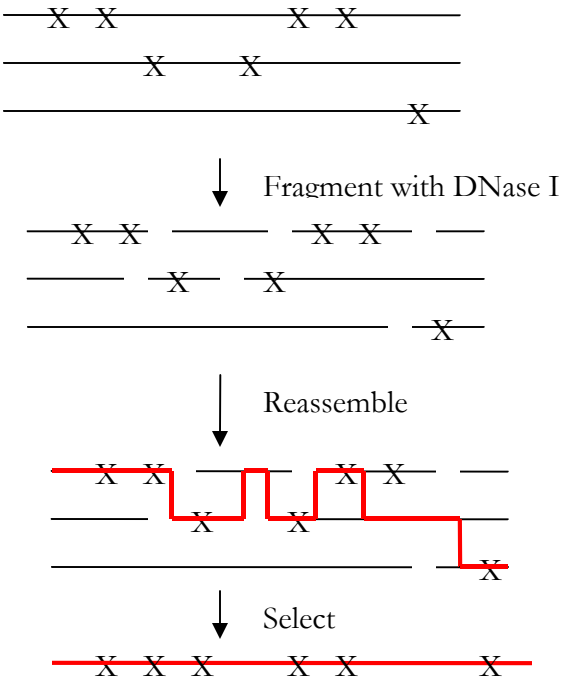


Figure 10: DNA shuffling and ligation. (A) DNA shuffling of homologous sequences. X: point mutation. (B) Ordered assembly of multiple genes via ligation.

(A)



(B)

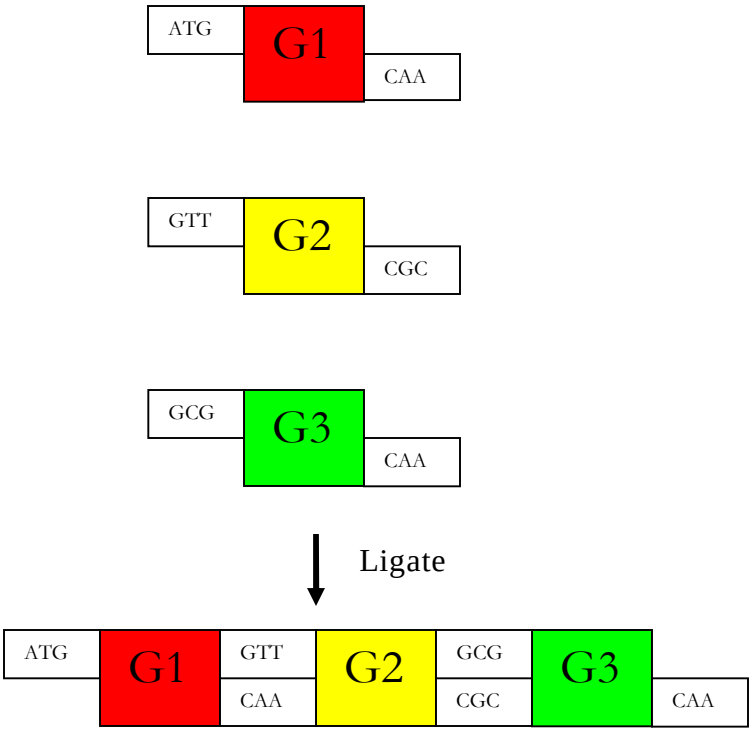


Figure 11: Ligation of fragments to generate permuted element assemblies. E1-3: elements being permuted; Lxy: linker between position x and position y; (Lxy)': complementary sequence to Lxy; LC: left cap; RC: right cap. The linkers guide ordered assembly of the fragments.

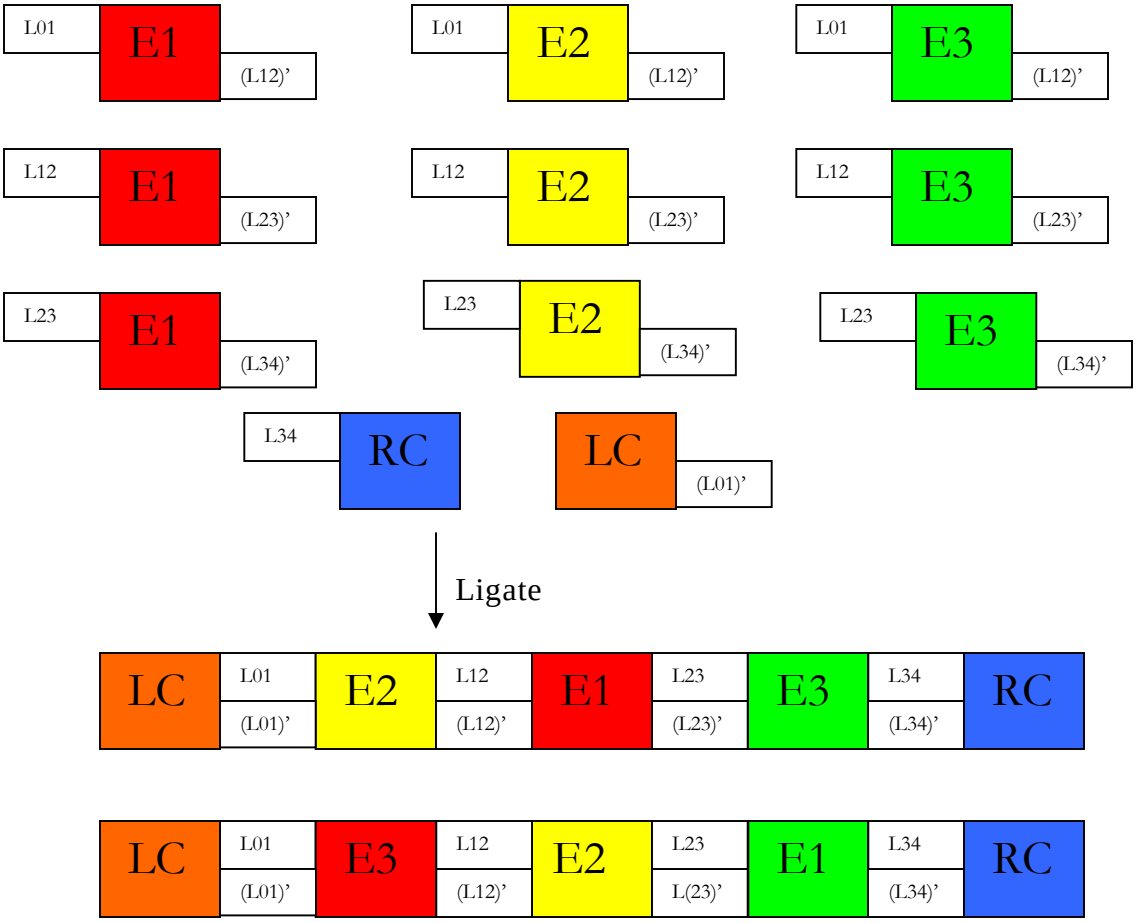
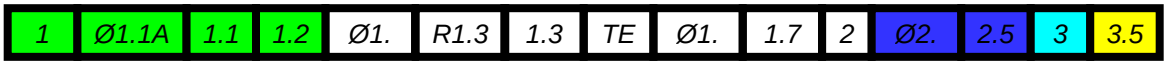


Figure 12: Example segmentation of the T7.2 region spanning genes *1-3.5*. Coloring indicates elements belonging to the same segment.



References

1. Kitano, H.. Computational systems biology. *Nature*. 420, 206-10 (2002).
2. Kitano, H.. Looking beyond the details: a rise in system-oriented approaches in genetics and molecular biology. *Curr Genet*. 41, 1-10 (2002).
3. Kirschner M.W. The meaning of systems biology. *Cell* 121, 503-4 (2005).
4. Arkin A.P., Gilman A. Genetic "code": representations and dynamical models of genetic components and networks. *Annu Rev Genomics Hum Genet.* 3, 341-69 (2002).
5. Endy D., Brent R. Modelling cellular behavior. *Nature* 409, 391-5 (2001).
6. Ooi S.L. *et al.* Global synthetic-lethality analysis and yeast functional profiling. *Trends Genet*. 22, 56-63 (2006).
7. Lok L., Brent R. Automatic generation of cellular reaction networks with Moleculizer 1.0. *Nat Biotechnol*. 23, 131-6 (2005).
8. Demerec M., Fano U. Bacteriophage-resistant mutants in *Escherichia coli. Genetics* 30, 119-136 (1945).
9. Studer F.W. The genetics and physiology of bacteriophage T7.*Virology* 39, 562-74 (1969).
10. Studier F.W., Maizel J.V. Jr. T7-directed protein synthesis. *Virology* 39, 575-86 (1969).
11. Studier F.W. Identification and mapping of five new genes in bacteriophage T7. *J Mol Biol* 153, 493-502 (1981).
12. Dunn J.J., Studier F.W. Complete nucleotide sequence of bacteriophage T7 DNA and the locations of T7 genetic elements. *J Mol Biol.* 166, 477-535 (1983).
13. Molineux I.J. The T7 Group, pp 277-301. In: R. Calendar (ed.), *The Bacteriophages*, second edition. Oxford University Press (2006).

14. Garcia L.R., Molineux I.J. Transcription-independent DNA translocation of bacteriophage T7 DNA into Escherichia coli. *J Bacteriol.* 178, 6921-9 (1996).
15. Garcia L.R., Molineux I.J. Rate of translocation of bacteriophage T7 DNA across the membranes of Escherichia coli. *J Bacteriol.* 177, 4066-76 (1995).
16. Cheetham G.M., Steitz T.A. Structure of a transcribing T7 RNA polymerase initiation complex. *Science* 286, 2305-9 (1999).
17. Yin Y.W., Steitz T.A.. Structural basis for the transition from initiation to elongation transcription in T7 RNA polymerase. *Science* 298, 1387-95 (2002).
18. Ma K., Temiakov D., Anikin M., McAllister W.T. Probing conformational changes in T7 RNA polymerase during initiation and termination by using engineered disulfide linkages. *Proc Natl Acad Sci USA* 102, 17612-7 (2005).
19. Imburgio D., Rong M., Ma K., McAllister W.T. Studies of promoter recognition and start site selection by T7 RNA polymerase using a comprehensive collection of promoter variants. *Biochemistry* 39, 10419-30 (2000).
20. Dunn J.J., Studier F.W. Effect of RNAase III, cleavage on translation of bacteriophage T7 messenger RNAs. *J Mol Biol.* 99, 487-99 (1975).
21. Endy D., You L., Yin J., Molineux I.J.Computation, prediction, and experimental tests of fitness for bacteriophage T7 mutants with permuted genomes. *Proc Natl Acad Sci USA* 97, 5375-80 (2000).
22. Chan L.Y., Kosuri S, Endy D. Refactoring bacteriophage T7. *Mol Syst Biol.* 1, 2005.0018. (2005).
23. T7.2. http://openwetware.org/wiki/T7.2

24. Pajunen M.I., Elizondo M.R., Skurnik M., Kieleczawa J., Molineux I.J. Complete nucleotide sequence and likely recombinatorial origin of bacteriophage T3. *J Mol Biol.* 319, 1115-32 (2002).

25. Garcia E., Elliott J.M., Ramanculov E., Chain P.S., Chu M.C., Molineux I.J. The genome sequence of Yersinia pestis bacteriophage phiA1122 reveals an intimate history with the coliphage T3 and T7 genomes. *J Bacteriol.* 185, 5248-62 (2003).

26. Kovalyova I.V., Kropinski A.M. The complete genomic sequence of lytic bacteriophage gh-1 infecting Pseudomonas putida--evidence for close relationship to the T7 group. *Virology* 311, 305-15 (2003).

27. Pajunen M.I., Kiljunen S.J., Soderholm M.E., Skurnik M. Complete genomic sequence of the lytic bacteriophage phiYeO3-12 of Yersinia enterocolitica serotype O:3. *J Bacteriol.* 183, 1928-37 (2001).

28. Zavriev S.K., Shemyakin M.F. Influence of the deletions of A2-A3 promoters or a terminator of early genes upon the rate of T7 DNA entrance into Escherichia coli cell. *FEBS Lett.* 131, 99-102 (1981).

29. Thanaraj T.A., Argos P. Ribosome-mediated translational pause and protein domain organization. *Protein Sci.* 5, 1594-612 (1996).

30. Klionsky D.J., Skalnik D.G., Simoni R.D. Differential translation of the genes encoding the proton-translocating ATPase of Escherichia coli. *J Biol Chem.* 261, 8096-9 (1986).

31. Guisez Y., Robbens J., Remaut E., Fiers W. Folding of the MS2 coat protein in Escherichia coli is modulated by translational pauses resulting from mRNA secondary structure and codon usage: a hypothesis. *J Theor Biol.* 162, 243-52 (1993).

32. Chen L.H. *et al.* Structure and function of a bacterial mRNA stabilizer: analysis of the 5' untranslated region of ompA mRNA. *J Bacteriol.* 173, 4578-86 (1991).

33. Belasco J.G., Chen C.Y. Mechanism of puf mRNA degradation: the role of an intercistronic stem-loop structure. *Gene* 72, 109-17 (1988).

34. Mertens N., Remaut E., Fiers W. Increased stability of phage T7g10 mRNA is mediated by either a 5'- or a 3'-terminal stem-loop structure. *Biol Chem.* 377, 811-7 (1996).

35. Katz L., Burge C.B. Widespread selection for local RNA secondary structure in coding regions of bacterial genes. *Genome Res.* 13, 2042-51 (2003).

36. Posfai G. *et al.*. Emergent properties of reduced-genome Escherichia coli. *Science* 312, 1044-6 (2006).

37. Yang Y., Masker W. Deletion during recombination in bacteriophage T7. *Mutat Res.* 349, 21-32 (1996).

38. Pierce J.C., Kong D, Masker W. The effect of the length of direct repeats and the presence of palindromes on deletion between directly repeated DNA sequences in bacteriophage T7. *Nucleic Acids Res.* 19, 3901-5 (1991).

39. Kong D., Masker W. Deletion between directly repeated DNA sequences measured in extracts of bacteriophage T7-infected Escherichia coli. *J Biol Chem.* 268, 7721-7 (1993).

40. Studier F.W. The genetics and physiology of bacteriophage T7. *Virology* 39, 562-74 (1969).

41. Smith H.O., Hutchinson C.A. 3rd, Pfannkoch C, Venter J.C. Generating a synthetic genome by whole genome assembly: phiX174 bacteriophage from synthetic oligonucleotides. *Proc Natl Acad Sci USA.* 100, 15440-5 (2003).

42. Tian J. *et al.*. Accurate multiplex gene synthesis from programmable DNA microchips. *Nature* 432, 1050-4 (2004).

43. Bull J.J., Badgett M.R., Rokyta D, Molineux I.J. Experimental evolution yields hundreds of mutations in a functional viral genome. *J Mol Evol.* 57, 241-8 (2003).

44. Springman R., Badgett M.R., Molineux I.J., Bull J.J. Gene order constrains adaptation in bacteriophage T7. *Virology* 341, 141-52 (2005).

45. Stemmer W.P. Rapid evolution of a protein in vitro by DNA shuffling. *Nature* 370, 389-91 (1994).

46. Stemmer W.P. DNA shuffling by random fragmentation and reassembly: in vitro recombination for molecular evolution. *Proc Natl Acad Sci USA* 91, 10747-51 (1994).

47. Tsuge K., Matsui K., Itaya M. One step assembly of multiple DNA fragments with a designed order and orientation in Bacillus subtilis plasmid. *Nucleic Acids Res.* 31, e133 (2003).

48. Sanger F., Nicklen S., Coulson A.R. DNA sequencing with chain-terminating inhibitors. *Proc Natl Acad Sci USA* 74, 5463-7 (1977).

49. Dramanac R. *et al.*. Sequencing by hybridization (SBH): advantages, achievements, and opportunities. *Adv Biochem Eng Biotechnol.* 77, 75-101 (2002).

50. Endy D., Kong D., Yin J. Intracellular kinetics of a growing virus: a genetically structured simulation for bacteriophage T7. *Biotechnol. Bioeng.* 55, 375-389 (1997).

51. Tatusova T.A., Madden T.L. Blast 2 sequences - a new tool for comparing protein and nucleotide sequences. *FEMS Microbiol Lett.* 174, 247-250 (1999).

52. Hofacker L.L. *et al.* Fast Folding and Comparison of RNA Secondary Structures. *Monatshefte f. Chemie* 125, 167-188 (1994

53. Dong H., Nilsson L., Kurland C.G. Co-variation of tRNA abundance and codon usage in Escherichia coli at different growth rates. *J Mol Biol.* 260, 649-63 (1996).

54. Olsthoorn R.C., van Duin J. Evolutionary reconstruction of a hairpin deleted from the genome of an RNA virus. *Proc Natl Acad Sci USA* 93, 12256-61 (1996).

55. Garcia L.R. Characterization of bacteriophage T7 DNA entry into *Escherichia coli.* Dissertation, The University of Texas at Austin (1996).

56. Scearce L.M. *et al.*, Deletion mutagenesis independent of recombination in bacteriophage T7. *J. Bacteriol.* 173, 869-78 (1991)

57. Kong D., Masker W. Deletion between direct repeats in T7 DNA stimulated by double-strand breaks. *J. Bacteriol.* 176, 5904-11 (1994).

58. Zuker, M. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Res.* 31, 3406-15, (2003)

59. Mathews D.H. *et al*, Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *J. Mol. Biol.* 288, 911-940 (1999).

60. Wertz G.W, *et al.* Gene rearrangement attenuates expression and lethality of a nonsegmented negative strand RNA virus. *Proc Natl Acad Sci USA* 95, 3501-3506 (1998).

61. Ball L. A., *et al.* Phenotypic consequences of rearranging the P, M, and G genes of vesicular stomatitis virus. *J. Virology* 73, 4705-4712 (1999).

62. Wertz G.W. *et al.*. Adding genes to the RNA genome of vesicular stomatitis virus: positional effects on stability of expression. *J. Virology* 76, 7642-7650 (2002).

63. Novella I.S., *et al.* Fitness analyses of vesicular stomatitis strains with rearranged genomes reveal replicative disadvantages. *J. Virology* 78, 9837-9841 (2004).

Appendix A: Programs used for analysis and simulation

The programs used to generate the results described above are, with a single exception, written in Python, and designed to run on a Windows machine. Each program is commented extensively enough that it will hopefully be reasonably easy to understand.

Phylogenetic analysis of T7 genes:

Program: *comparegenes.py:*

Dependencies: There are unfortunately quite a few dependencies here, because I was experimenting with keeping all my data in the MySQL database when I wrote this bit of code. You'll need to install:

- MySQL database, available at http://mysql.com/
- Python interface to MySQL, available at http://sourceforge.net/projects/mysql-python

Once these programs are all installed, you'll need to create the necessary database and database tables, and populate them via the following steps:

- Create a MySQL database called "t7rebuild"
- Create the necessary database tables by running

  python createt7tables.py --user <your MySQL user name> --pwd <your MySQL password>
- Populate the database tables by running

  python loadt7tables.py --user <your MySQL user name> --pwd <your MySQL password> --file t7_stripped.gb --file t3_stripped.gb --file gh-1_stripped.gb --file phiA1122_stripped.gb --file phiYe03-12_stripped.gb

After the database has been created and the tables populated, you can run *comparegenes.py* as described below.

Usage: *python comparegenes.py --user <your MySQL user name> --pwd <your MySQL password>*

Output: Self-explanatory.

Analysis of RBS-CDS secondary structure:

Program: *findbestss.py*

Dependencies:

- Cygwin, a Linux emulation environment available from http://www.cygwin.com, must have been installed to c:\cygwin
- RNAfold.exe, available from http://www.tbi.univie.ac.at/~ivo/RNA/windoze, must have been copied to c:\cygwin\bin

Usage: *python findbestss.py --input t7_stripped.gb [--output <output file name>] [--top <number of seq variants to retain] [--noshuffle]*.

Warning: this program takes a long time [~24 hours, on my laptop] to run to completion. The *input* parameter specifies a GenBank file that will be parsed to extract the T7 gene sequences. The *output* parameter controls the prefix given to the FASTA output files produced; the prefix defaults to "bestss" ("best secondary structure") if not specified. The *top* parameter specifies the number of sequence variants to retain, and defaults to 50. The *noshuffle* parameter can be used if you only want to generate data for the WT RBS + WT CDS and T7.2 RBS + WT CDS sequences, and not generate any shuffled sequences.

Output: Produces a set of files named *bestss_<gene number>.fa* eg *bestss_1.fa*. These .fa output files are in FASTA format and contain a set of RBS-CDS sequences and their associated ΔG's of folding. The first sequence in each file is always the wild-type RBS + wild-type CDS and the second sequence is always the T7.2 RBS + wild-type CDS. Subsequent sequences are the T7.2 RBS + shuffled CDS variants, sorted from least to most secondary structure.

The format of each file is:

>[gene number]_[sequence index]_[ΔG of folding]
[actual DNA sequence]
>[gene number]_[sequence index]_[ΔG of folding]
[actual DNA sequence]

For example,

>1_0_-4.300000
taactggaagaggcactaaaatgaacacgattaacatcgctaagaacgacttctctgac
>1_1_-8.800000
ttaaagaggagaaatactagatgaacacgattaacatcgctaagaacgacttctctgac
>1_2_-0.100000
ttaaagaggagaaatactagatgaatacgataaatatagccaaaaatgacttcagcgat

are first three sequences for gene 1. The first sequence (WT RBS + WT CDS) has a ΔG of folding of -4.3kJ/mol, the second sequence (T7.2 RBS + WT CDS) has a ΔG of folding of -8.8kJ/mol, and the third sequence (T7.2 RBS + shuffled CDS) has a ΔG of folding of -0.1kJ/mol. The third sequence has the highest ΔG (i.e. least secondary structure) of all the gene 1 CDS variants.

Analysis of secondary structure in protein-coding RNA:

Program: *ShuffleAndFold.exe*. This program is written in C and needs to be compiled; the list of files is given in "Dependencies" below. I used Microsoft Visual C++ Express Edition, available free at http://msdn.microsoft.com/vstudio/express/visualc/default.aspx, to compile it. .

Dependencies:

- All the files in the ShuffleAndFold\H, ShuffleAndFold\lib and ShuffleAndFold\Progs directories.

- The Microsoft Windows Platform SDK, available at

  http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/.

Usage: *shuffleandfold.exe <input file> <window size> <step size> [<number of sequences to generate>]*

The <*input file*> parameter must specify a file that has the format

[gene name]:[gene DNA sequence]
[gene name]:[gene DNA sequence]
…
The file *t7genes.txt* is in the appropriate format.

The <*window size*> parameter controls the size of the sliding window and the <*step size*> parameter specifies how far the window is moved on successive steps. The <*number of sequences to generate*> parameter can be used to specify how many shuffled sequences to generate via the DicodonShuffle algorithm and fold via RNAfold; it defaults to 100.

Output: Two files, *t7bias_all_<window size>_<step size>.txt* and *t7bias_sig_<window size>_<step size>.txt*.

For each gene, the *t7bias_all_<window size>_<step size>.txt* contains the folding energies for all window positions across all sequence variants. The first folding energy listed for each window position is that of the wild-type sequence. The format of this file is

Sequence:[gene name]
Position [zero-based start pos-end pos]
[Folding energy for WT seq] [Folding energy for seq variant 1] [Folding energy for seq variant 2] …
…

For example, a typical entry might look like

Sequence:gene 4.2
Pos 0-50
-8.000 -5.610 -7.440 -8.100 -7.400 -9.500 -5.300 -6.100 -9.400 -8.100 -8.600 -9.200 -9.000 -8.700 -6.600 -5.300 -5.860 -6.600 -4.420 -8.300 -6.500 -6.000 -8.600 -7.300 -7.460

indicating that the energy of folding of positions 0-50 of the wild-type sequence of gene *4.2* is -8.0kJ/mol, the energy of folding of positions 0-50 of the first sequence variant is -5.61kJ/mol etc.

The *t7bias_sig_<window size>_<step size>.txt* file contains only data about the wild-type sequence windows considered to have statistically-significant energies of folding. The format of this file is

Sequence:[gene name]
WT seq in position [start pos-end pos] has z-score [z-score] ([ΔG in kJ/mol])
[WT DNA sequence]
…

For example, a typical entry might look like

Sequence:gene 4.2
WT seq in pos 10-60 has z-score -2.801749 (-12.000000)
TCGCCCCGTTTCTATTACTGACCTACGTGGTTCTGGCGCACTACGCCAAC

indicating that positions 10-60 of gene *4.2*, with sequence TCGCC..AAC, have a z-score equal to -2.801749, and a folding energy of -12kJ/mol.

Simulating gene loss in a lossy genome:

Program: *simrecomb.py*

Dependencies: None

Usage: *python simrecomb.py*

Output: Self-explanatory.

Simulating genome shuffling via ligation:

Program: *simulateligation.py*

Dependencies: None

Usage: *python simulateligation.py*

Output: Self-explanatory