Optimizing Restriction Site Placement for Synthetic Genomes

Pablo Montes[1] Heraldo Memelli[1] Charles Ward[1]
Joondong Kim[2] Joseph S. B. Mitchell[2] Steven Skiena[1]

[1]Department of Computer Science
Stony Brook University

[2]Department of Applied Mathematics and Statistics
Stony Brook University

21st Annual Symposium on Combinatorial Pattern Matching

What do we want to do?

Problem Definition

Modify virus-length genomes to introduce large numbers of evenly spaced unique restriction sites while preserving their amino-acid sequence.

What do we want to do?

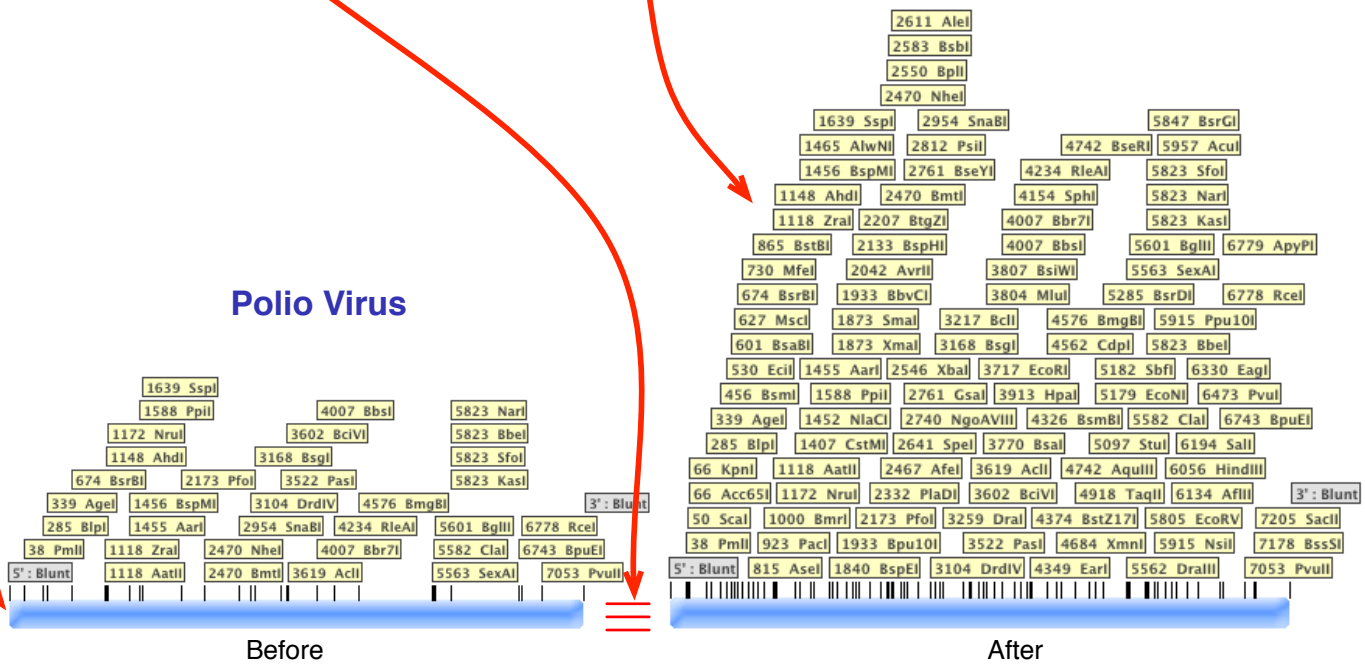
Problem Definition

Modify virus-length genomes to introduce large numbers of evenly spaced unique restriction sites while preserving their amino-acid sequence.

What do we want to do?

Problem Definition

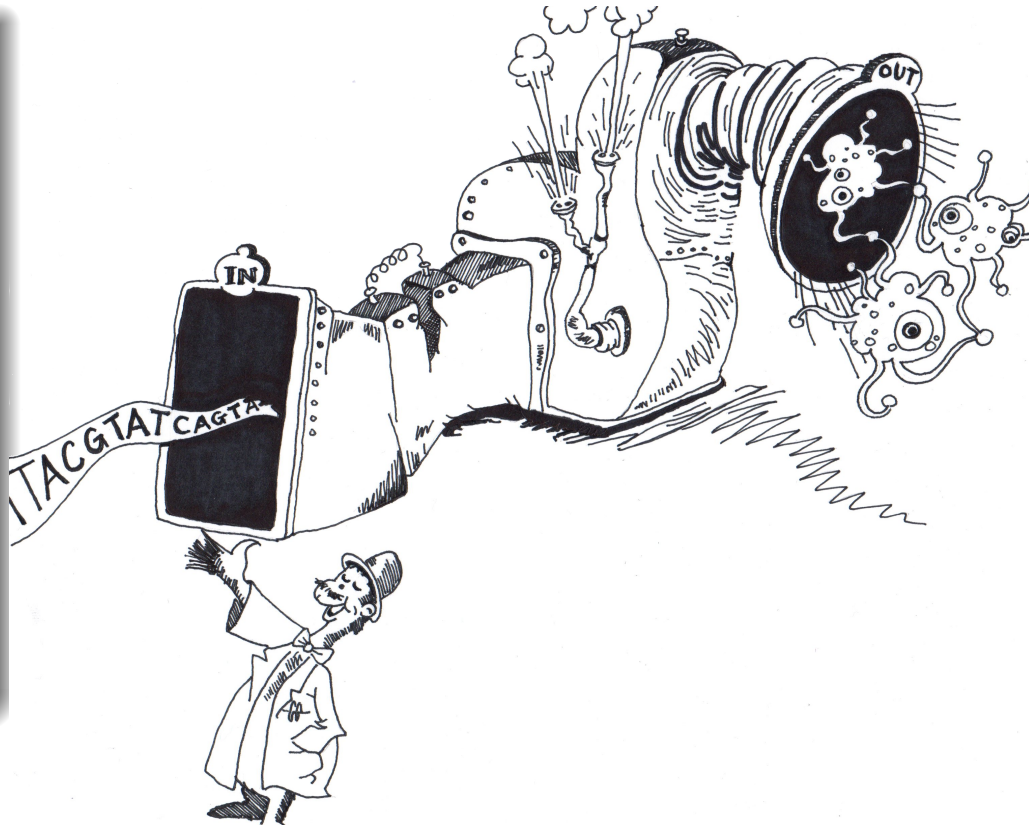
Modify virus-length genomes to introduce large numbers of evenly spaced unique restriction sites while preserving their amino-acid sequence.



Where is this useful?

Synthetic Biology

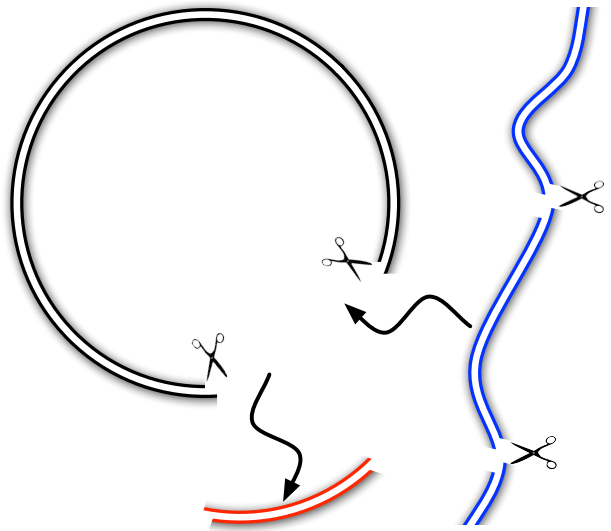
- Enables us to construct DNA molecules to specification.
- Is an emerging field due to declining costs for synthesizing long DNA sequences (Under 60 cents per base).



Why is this useful?

Gene Cloning

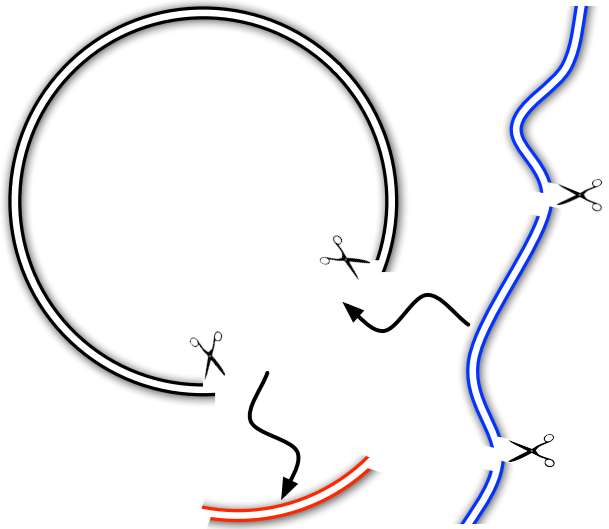
- Is a fundamental operation in microbiology.
- Uses restriction enzymes to cut plasmids for insertion/removal of DNA fragments.
- To use a restriction enzyme, the place where it cuts must be unique.
- Unique restriction sites regularly distributed along the genome gives more flexibility.



Why is this useful?

Genome Refactoring

Restructuring the genome of an organism into a sequence that behaves the same in its natural environment while being easier to manipulate in the laboratory.



History

2002 Wimmer's group at Stony Brook synthesized the first virus from scratch (about 10 Kilo-bases).

2005 Endy's group refactored the genome of the T7 phage (a virus) to make it easier to manipulate in the laboratory.

2010 Venter announced the first synthetic bacteria (about 1 Mega-bases).

What are biologists doing with this?

- Developing experimental vaccines for polio and flu (to appear in the July issue of Nature Biotechnology).
- Synthesizing and reengineering an animal pathogen to create a vaccine (first attempt to work on a commercial target without successful vaccines).

To summarize

- Synthetic biology is an exciting and emerging field.
- This problem is useful in practice for biologists.
- The need to design new sequences to specification leads to a variety of new algorithmic problems on sequences.

Contributions

- Theoretical
  - Problem abstraction
  - NP-completeness
  - Hardness of approximation
  - 2-approximation algorithm
  - Dynamic programming algorithm
- Experimental
  - Greedy approach
  - Weighted bipartite matching
  - Dynamic programming in blocks
  - Experimental results that show that large numbers of regularly-spaced unique restriction sites can be engineered into viral genomes.

Outline

Recall

Problem Definition

Modify virus-length genomes to introduce large numbers of evenly spaced unique restriction sites while preserving their amino-acid sequence.

Virus-length genomes

- The complete genetic information (DNA) of an organism.
- String on $\Sigma = \{A, C, G, T\}$.
- Both genes and non-coding sequences.
- In the order of 7 to 20 Kilo-bases.

TGCAAAGCTTGATGTTCGCACGCATTCTACCTGGAGCTAAGCTTACCAT

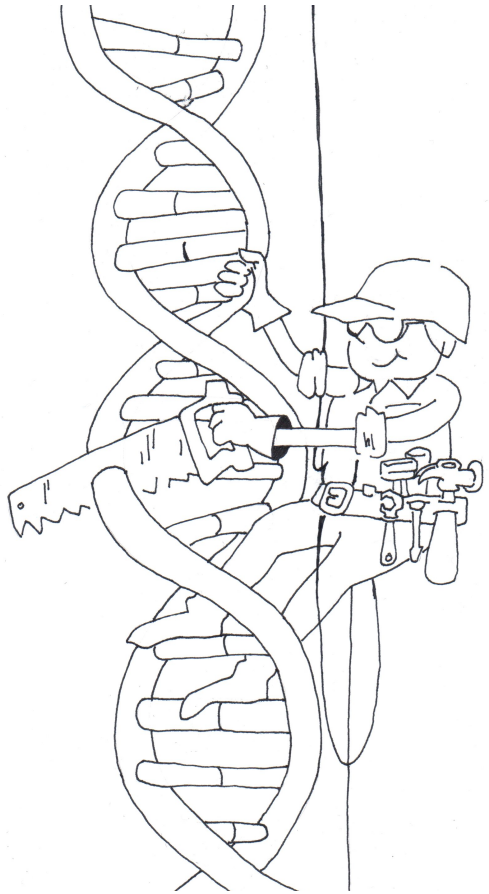
Restriction enzymes

Definition

Proteins that recognize specific DNA sequences and cut DNA at or near **all occurrences** of those sites.

Terminology

- **Recognition pattern:** DNA sequence recognized by a restriction enzyme.
- **Restriction site:** Actual location in the entire DNA sequence where a restriction enzyme cuts.



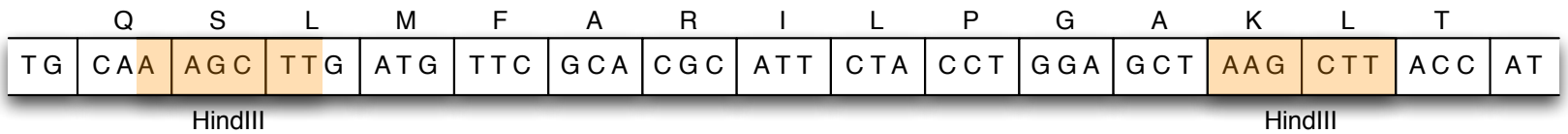
Restriction enzymes

- Over 3000.
- More than 600 available commercially.
- We focus on *Type II* restriction enzymes.
  - Cut within the recognition pattern.
  - 4 to 8 bases.
  - Predominantly used in biotechnology.

Enzyme	Recognition Pattern
EcoRI	GAATTC
HindIII	AAGCTT
BamHI	GGATCC
TaqI	TCGA
NotI	GCGGCCGC
PovII	CAGCTG

Amino-acid sequence

- A gene is a DNA sequence which acts as a template for building a specific protein.
- From the start of a reading frame, each *codon* (group of 3 consecutive bases) maps to a single amino-acid.
- There are $4^3 = 64$ possible codons.
- There are 20 standard amino-acids.



Amino-acid sequence

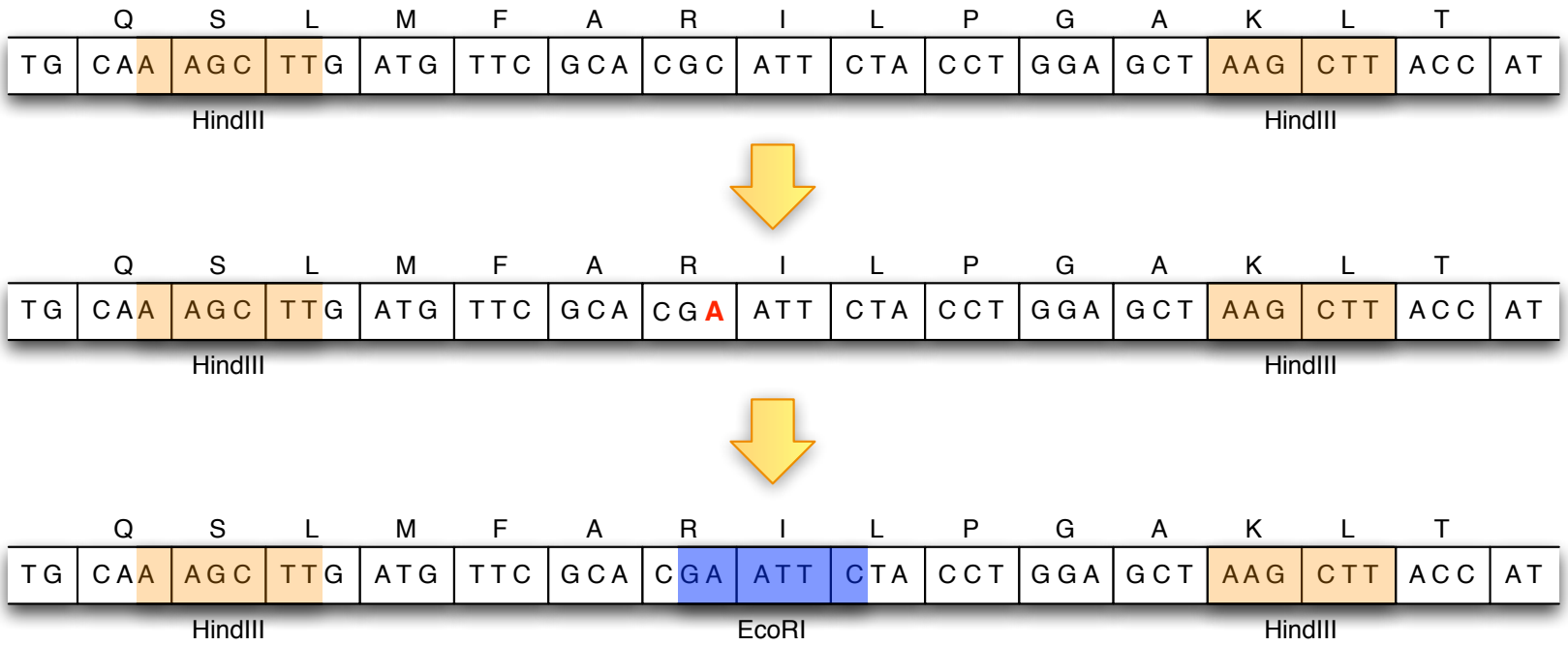
Redundancy in the genetic code

Amino Acid	Synonymous Codons
Isoleucine (I)	ATT, ATC, ATA
Leucine (L)	CTT, CTC, CTA, CTG, TTA, TTG
Valine (V)	GTT, GTC, GTA, GTG
Phenylalanine (F)	TTT, TTC
Methionine (M)	ATG
Cysteine (C)	TGT, TGC
Alanine (A)	GCT, GCC, GCA, GCG
Glycine (G)	GGT, GGC, GGA, GGG
Proline (P)	CCT, CCC, CCA, CCG
Threonine (T)	ACT, ACC, ACA, ACG
Serine (S)	TCT, TCC, TCA, TCG, AGT, AGC

Amino Acid	Synonymous Codons
Tyrosine (Y)	TAT, TAC
Tryptophan (W)	TGG
Glutamine (Q)	CAA, CAG
Asparagine (N)	AAT, AAC
Histidine (H)	CAT, CAC
Glutamic acid (E)	GAA, GAG
Aspartic acid (D)	GAT, GAC
Lysine (K)	AAA, AAG
Arginine (R)	CGT, CGC, CGA, CGG, AGA, AGG
Stop codons	TAA, TAG, TGA

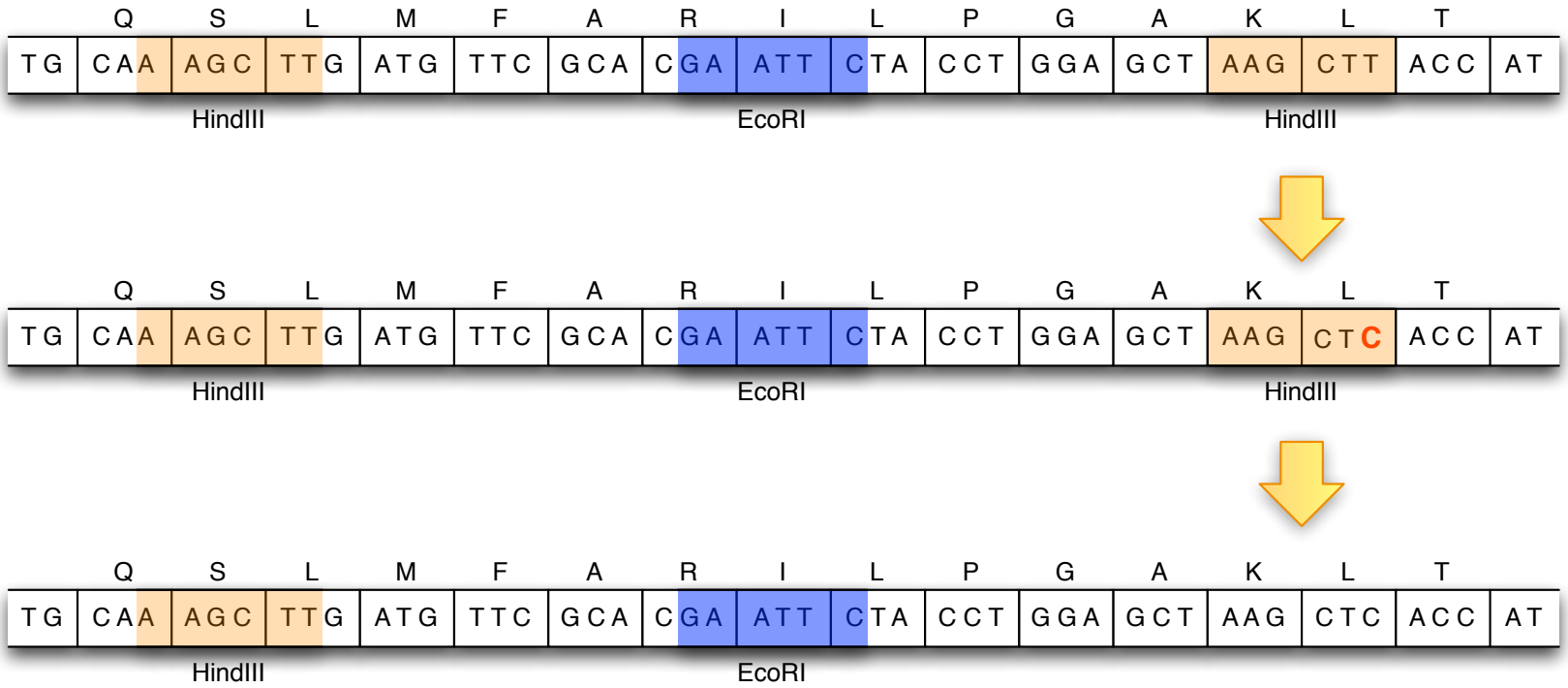
Restriction site insertion

EcoRI	GAATTC
Arginine (R)	CGT, CGC, CGA, CGG, AGA, AGG



Restriction site deletion

Leucine (L)	CTT, CTC, CTA, CTG, TTA, TTG



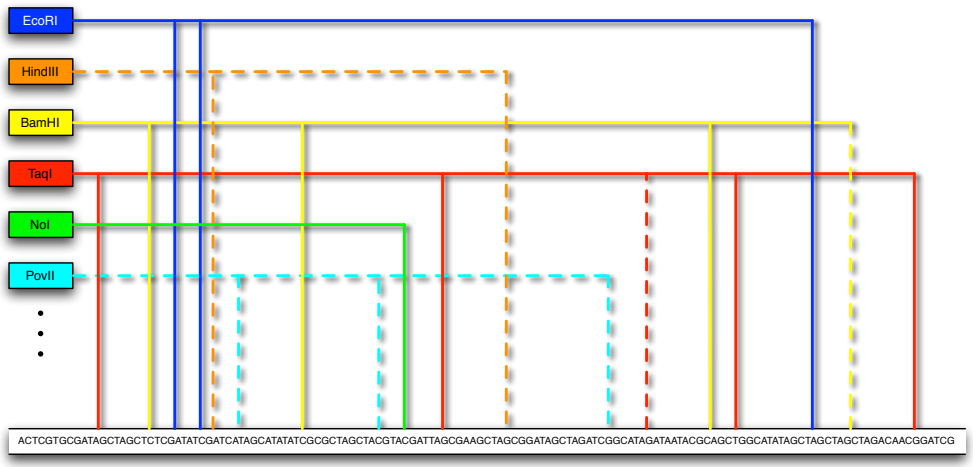
In short

- Given
  - DNA sequence.
  - List of locked regions.
  - List of reading frames (genes).
  - List of restriction enzymes and their recognition pattern.
- We want to:
  - Modify the sequence while preserve its amino-acid sequence (by substitution of synonymous codons).
  - Minimize the maximum gap between unique restriction sites.
  - Maximize the number of unique restriction sites.
  - Minimize number of base changes.

Preprocessing

Restriction site map

- Identifies the position of existing (or potential) restriction sites in a DNA sequence for a set of restriction enzymes.
- Can be built using standard pattern matching techniques (Aho-Corasick algorithm).



Outline

Problem abstraction

Decision version

Definition (URSPP)

- **Input:**
  - $\mathcal{S} = \{S_1, \ldots, S_m\}$
  - $S_i \subseteq \{1, \ldots, n\}$
  - $\lceil \frac{n}{m+1} \rceil \leq k \leq n$
- **Output:** Does there exist a single element s_i in all S_i such that the maximum gap between adjacent elements of $\{0, n+1, s_1, \ldots, s_m\}$ is at most k?

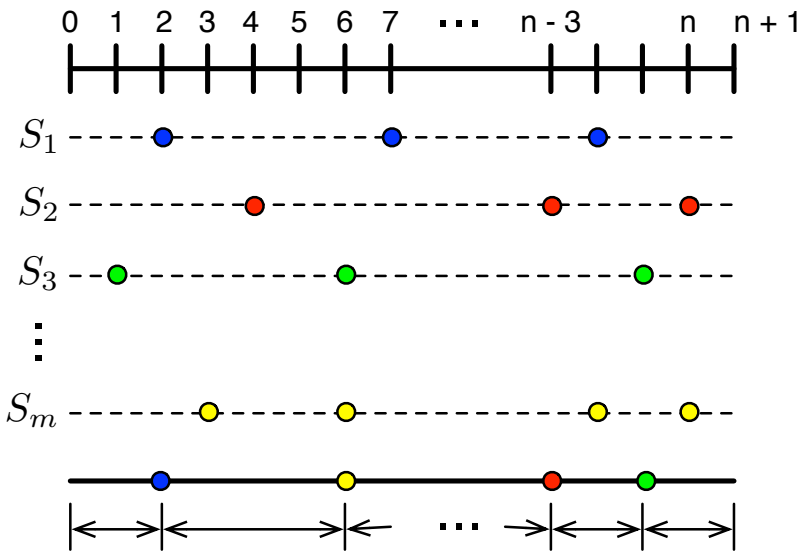
- Each S_i consists of the existing or potential restriction sites for a specific restriction enzyme.
- We want to choose a single restriction site for each restriction enzyme.
- So that adjacent unique restriction sites are no more than k bases apart.

Problem abstraction

Decision version

Definition (URSPP)

- **Input:**
  - $\mathcal{S} = \{S_1, \ldots, S_m\}$
  - $S_i \subseteq \{1, \ldots, n\}$
  - $\lceil \frac{n}{m+1} \rceil \leq k \leq n$
- **Output:** Does there exist a single element s_i in all S_i such that the maximum gap between adjacent elements of $\{0, n+1, s_1, \ldots, s_m\}$ is at most k?



NP-completeness

Theorem

The decision version of URSPP *is NP-complete.*

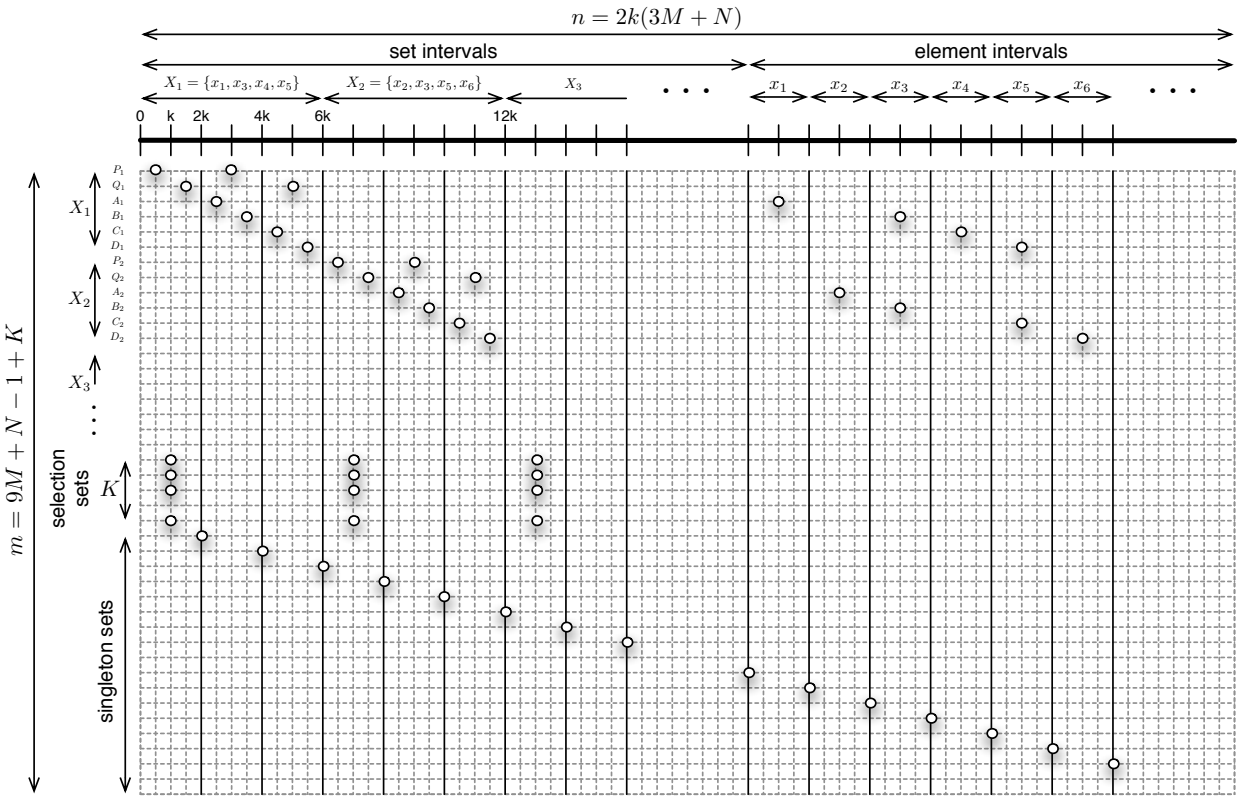
Sketch of proof.

- Clearly in NP
- NP-hard by a reduction from Set Cover where $|X_i| = 4$.

□

NP-completeness

- There exists a SET COVER of size K if and only if there exists a selection of s_i's for URSPP with maximum gap k.



Hardness of approximation

Theorem

The optimization version of URSPP *cannot be approximated within factor* $3/2$.

Sketch of proof.

- There exist a set cover of size K if and only if there exists a selection of s_i's for URSPP with maximum gap less than $3k/2$.
- In any suboptimal solution of the URSPP instance, the gap size is at least $3k/2$.

2-approximation algorithm

Theorem

The URSPP *optimization problem has a polynomial-time 2-approximation.*

Sketch of proof.

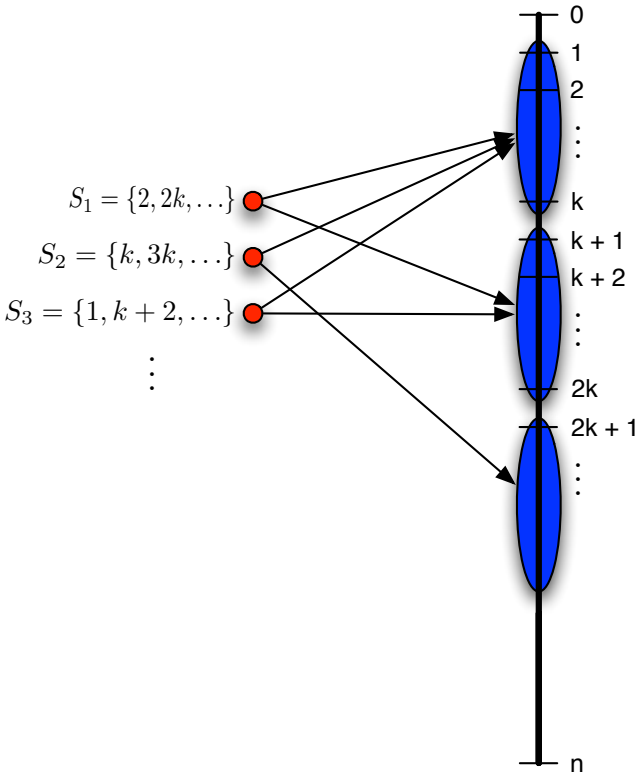
- For a given k, we can run an algorithm that will report:

  success and provide selections such that the maximum gap is at most $2k - 1$.

  failure it is impossible to make selections such that gaps are of size at most k.

2-approximation algorithm

- "red": S_i.
- "blue": k-element sets $\{jk+1, jk+2, \ldots, jk+k\}$, for $j = 0, 1, 2, \ldots, \lfloor \frac{n}{k} \rfloor - 1$.
- Edge from red to blue if and only if S_i contains an element in the set $\{jk+1, jk+2, \ldots, jk+k\}$.



We have showed

- The definition of the Unique Restriction Site Placement Problem (URSPP)
- That URSPP is NP-complete.
- That URSPP cannot be approximated within factor 3/2.
- A 2-approximation algorithm for the optimization version of URSPP.

Outline

Dynamic programming algorithm

- Let $C[\mathcal{S}, i, j]$ be the length of the minimum possible maximum gap in the range $[0, \ldots, i]$ where $\mathcal{S}$ is the set of available restriction enzymes, and the last restriction site was placed at position j.
- Let $\textsc{Enzymes}(\mathcal{S}, i)$ return a set with all the restriction enzymes in $\mathcal{S}$ that cut at location i, or $\emptyset$ if there is no restriction site at that location.

$$C[\mathcal{S}, i, j] = \begin{cases} j & \text{if } i < 0, \\ \min \left\{ \begin{array}{l} \min\limits_{e \in \textsc{Enzymes}(\mathcal{S}, i)} \left\{ \max \left\{ \begin{array}{l} C[\mathcal{S} \setminus \{e\}, i-1, i], \\ j - i \end{array} \right\} \right\}, \\ C[\mathcal{S}, i-1, j] \end{array} \right. & \text{otherwise.} \end{cases}$$

Dynamic programming algorithm

$$C[\mathcal{S}, i, j] = \begin{cases} j & \text{if } i < 0, \\ \min \left\{ \begin{array}{l} \min\limits_{e \in \textsc{Enzymes}(\mathcal{S}, i)} \left\{ \max \left\{ \begin{array}{l} C[\mathcal{S} \setminus \{e\}, i-1, i], \\ j - i \end{array} \right\} \right\}, \\ C[\mathcal{S}, i-1, j] \end{array} \right. & \text{otherwise.} \end{cases}$$

- Intuitively, for each possible location we can either use one of the restriction enzymes that cut at this location or choose not to cut. For each of these options we find the best way to use the remaining restriction enzymes in the remaining part of the sequence.
- $O(2^m n^2)$.

Outline

Heuristic approach

- Two phases:
  1. **Deletion:** Create a unique restriction site for each enzyme that appears in the genome initially.
  2. **Insertion:** Create a new restriction site for enzymes that do not appear in the genome initially.

Deletion of restriction sites

Heuristic

1. First consider enzymes with fewest restriction sites.
2. Randomly keep one restriction site and delete all other.
3. Lock the kept restriction site.

- Randomization makes the final distribution of restriction sites quite uniform throughout the sequence.
- Discard enzymes that cannot be used.

Insertion of restriction sites

After deletion

- After the deletion phase we are left with **gaps:** sequences of contiguous bases between unique restriction sites.
- Use the restriction enzymes that do not currently appear in the genome by creating restriction sites for them in order to reduce the size of these gaps.

Insertion of restriction sites

Heuristic

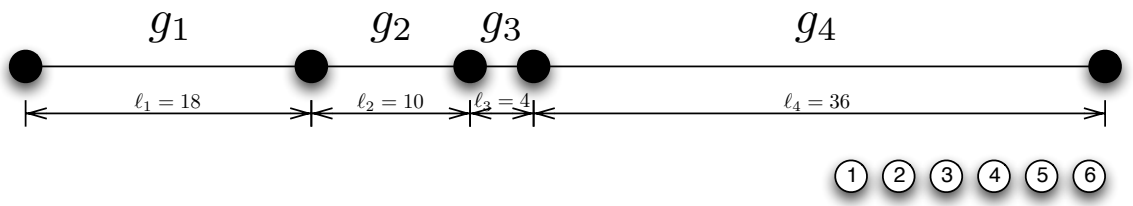
1. Find the ideal insertion points.
2. Insert the enzymes as close as possible to these ideal insertion points.
   - Greedy approach.
   - Weighted bipartite matching.
   - Dynamic programming in blocks.

Insertion of restriction sites

Ideal insertion points

Given

- Gaps: $\{g_1, g_2, \ldots, g_n\}$.
  - g_i has length ℓ_i.
  - g_i contains k_i separators (an therefore is composed of $k_i + 1$ segments). Initially $k_i = 0$.
- Separators: $\{s_1, s_2, \ldots, s_m\}$.

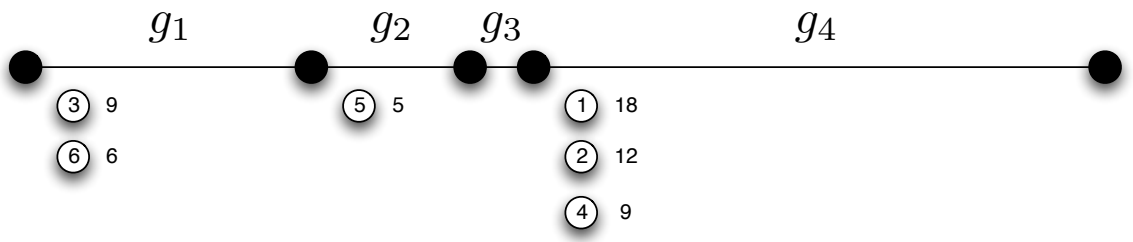


Insertion of restriction sites

Ideal insertion points

Algorithm

- As long as there are unused separators
  - Place the next available separator, s_j, in the gap g_i whose ratio between its length and the number of segments it is composed of $(\ell_i/(k_i+1))$ is highest.
  - increment k_i by one (without making a commitment in terms of the exact position in which s_j should be placed).
- Place separators by evenly dividing the length of the gap by the number of segments composing the gap.

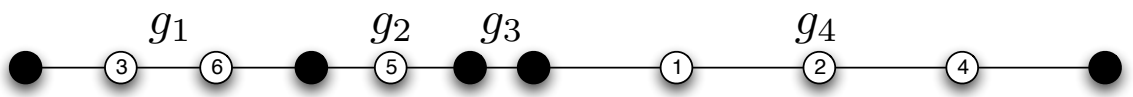


Insertion of restriction sites

Ideal insertion points

Algorithm

- As long as there are unused separators
  - Place the next available separator, s_j, in the gap g_i whose ratio between its length and the number of segments it is composed of $(\ell_i/(k_i+1))$ is highest.
  - increment k_i by one (without making a commitment in terms of the exact position in which s_j should be placed).
- Place separators by evenly dividing the length of the gap by the number of segments composing the gap.



Insertion of restriction sites

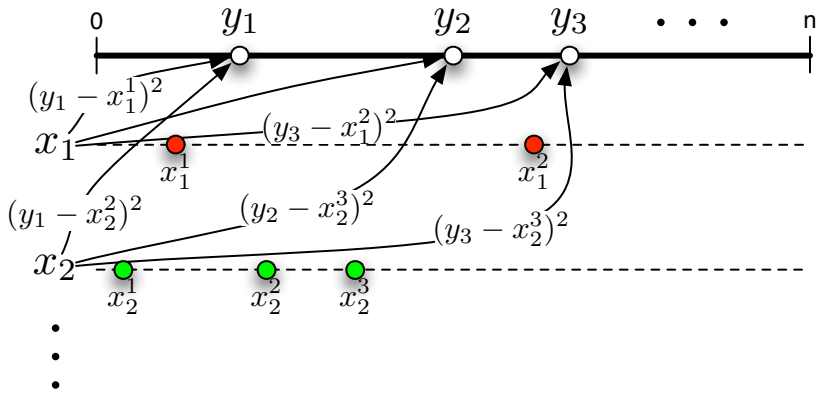
Greedy approach

- At each step try to insert the restriction enzyme with the least number of potential restriction sites, as close to an ideal insertion point as possible.
- Remove both, the selected enzyme and the selected ideal insertion point, from their corresponding lists.
- Iterate until all restriction enzymes are inserted.

Insertion of restriction sites

Weighted bipartite matching

- Restriction enzymes: $\{x_1, x_2, \ldots\}$.
- Ideal insertion points: $\{y_1, y_2, \ldots\}$.
- Add an edge, e_{ij}, from each x_i to every y_j.
- $w(e_{ij})$ is the squared distance between y_j and x_i's restriction site that is closest to y_j.



Insertion of restriction sites

Dynamic programming in blocks

- First we run the algorithm to find the optimal placement of a feasibly small set of X enzymes, then for the following X, and so on until we have covered all enzymes.
- Two ordering for considering enzymes:
  - Fewest possible restriction sites.
  - Most possible restriction sites.

Heuristic overview

1. Deletion of existing restriction sites.
2. Insertion of restriction sites.
    1. Find ideal insertion points.
    2. Insert enzymes as close as possible to the ideal insertion points using three different approaches:
        - Greedy.
        - Weighted Bipartite Matching.
        - Dynamic Programming in Blocks.

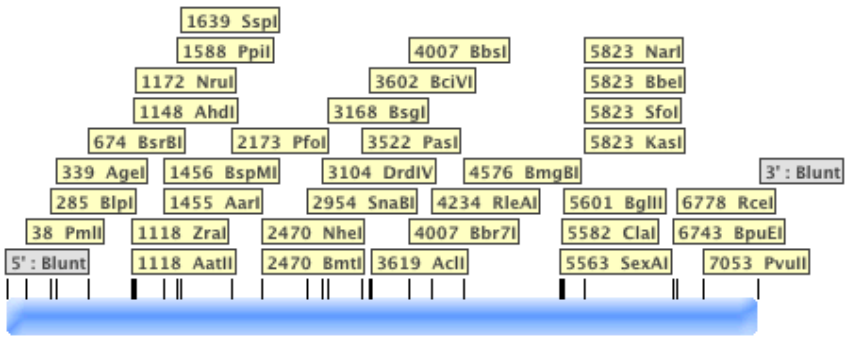
Baseline

1. Compute a gap-length g which would generate evenly spaced positions throughout the genome, based on the total number of restriction enzymes which either appear or can be created in the genome.
2. Attempt to introduce a unique restriction site (either by insertion or by deletion) as close to g as possible, say at position p.
3. Move to position $p + g$ and repeat.

Experimental Results

Virus	Metric	Initial	Baseline	After Removal	Greedy	Weighted Bipartite	DP (fewest)	DP (most)
Polio	base changes	0	141	120	207	216	194	189
Virus	unique enzymes	35	40	81	104	105	104	110
(7.5K)	max. gap	982	537	685	459	240	269	269
Equine	base changes	0	90	158	188	196	186	190
Arteritis	unique enzymes	24	29	80	90	91	88	92
Virus	max. gap	3,575	1,866	949	671	714	498	413
(12.8K)								
λ Phage	base changes	0	149	371	383	384	384	384
(48.5K)	unique enzymes	18	28	77	82	82	82	82
	max. gap	10,085	6,288	3,091	2,954	2,954	2,954	2,954

Polio virus



Before

After

We have showed

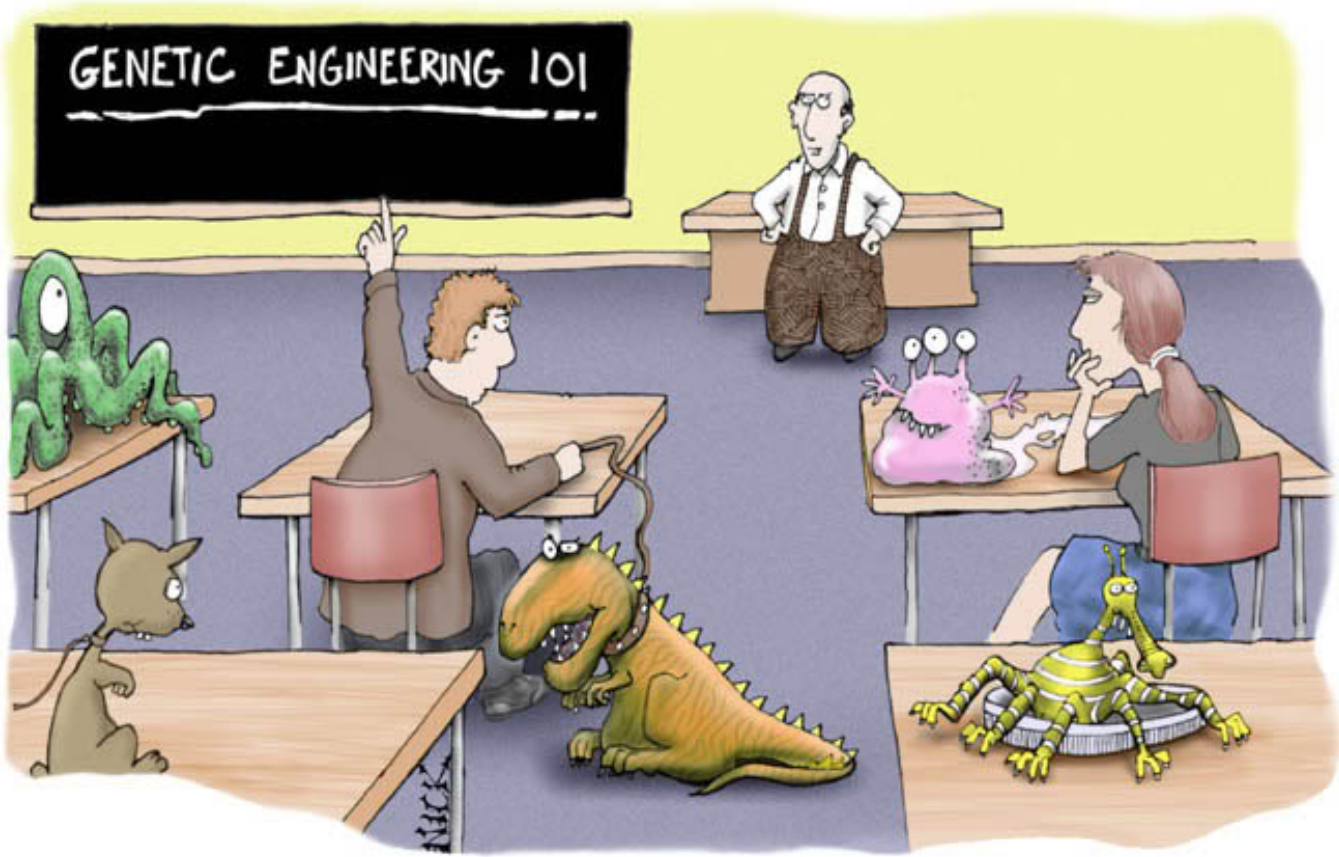
- Three different heuristics.
- A baseline algorithm.
- That we can insert three to four times more unique restriction enzymes than the baseline algorithm.
- That it is possible to reduce the maximum gap three to nine-fold.

Outline

Future work

- Minimize number of base changes.
- Optimize the use of cheap enzymes.
- Define a more clever policy to handle restriction site deletion to assure that the codon bias is not disrupted.
- Give a polynomial time algorithm to solve the problem of **how to optimally transform the original sequence into the final sequence** or prove it NP-hard (or both).
- Improve the tool for distribution.
- Close the gap between the upper and lower bounds on the approximation factor (2 versus 3/2).

Thank you



"Okay—is there anybody ELSE whose homework ate their dog?"