

Discrete Mathematics in Bioinformatics

HAO Bailin

T-Life Research Center, Fudan University
Shanghai 200433, China

and

Institute of Theoretical Physics, CAS
Beijing 100080, China

<http://www.itp.ac.cn/~hao/>

(PDF files of some published papers)

Biological Knowledge Needed Today

1. DNAs are symbolic sequences over
Alphabet $\Sigma = \{a, c, g, t\}$
 $|\Sigma| = 4$
2. Proteins are symbolic sequences over
Alphabet $\Sigma = \{A, C, \dots, Y\}$
 $|\Sigma| = 20$
- [3. DNAs store information]
- [4. Proteins perform biological function]
- [5. The **Central Dogma** of Molecular
Biology: DNA makes RNA makes Protein]

Central Theme: Composition in terms of K-strings

0. From alphabet $\Sigma = \{a, c, g, t\}$ or $\Sigma = \{A, C, \dots, Y\}$ to Σ^K
1. Natural generalization of
g+c contents, CpG islands
Amino acid frequency
2. Higher resolution:
DNA: $K = 10 \rightarrow 1\,048\,576$ strings
Protein: $K = 5 \rightarrow 3\,200\,000$ strings
3. Longer correlations taken into account
Higher-order Markov models
5. Species-specificity, gene-specificity enhanced
6. Transition from randomness to
determinism: primers, probes,
markers, etc.

Compositional Analysis of Genomic Data

1. Avoided and under-represented K-strings
Visualization and direct counting
A piece of neat mathematics:
Combinatorics and language theory
(Published and PDF available)
2. Repeats in prokaryote complete genomes;
Species-specific avoidance signature
Failure to infer phylogeny
3. Phylogenetic trees from complete genomes
No alignments. Compositional distance
J. Mol. Evol. **58** (2004) 1-11;
Mol. Biol. Evol. **21** (2004) 200-206;
Nucl. Acids Res. **32**(7) (July 2004) W45
4. Decomposition and reconstruction of
protein sequences and number of
Eulerian loops
(arXive.org: physics/0103028)
5. Finding genes in the rice genome

Three Case-Studies in Biology-Inspired Mathematics

From Real Biological Data
To Neat Mathematical Solutions

1. Goulden-Jackson cluster method
in combinatorics
2. Factorizable language and **minDFA**
3. Number of Eulerian loops in a graph

The E. coli Genome

A DNA loop made of 4 639 221 letters
from $\Sigma = \{a, c, g, t\}$

Look at (overlapping) strings of
length $K = 8$.

There are altogether $4^K = 65\,536$
possible string types.

Do they all exist in E. coli?

Expectation for a random sequence:
Each type would appear about

$$\frac{4639221}{65536} \approx 71 \text{ times.}$$

What happens in reality?

The simplest string-counting problem.

Allocation of Counters for $K = 1$ to 3

| | |
|----------|----------|
| <i>g</i> | <i>c</i> |
| <i>a</i> | <i>t</i> |

$K=1$

| | | | |
|-----------|-----------|-----------|-----------|
| <i>gg</i> | <i>gc</i> | <i>cg</i> | <i>cc</i> |
| <i>ga</i> | <i>gt</i> | <i>ca</i> | <i>ct</i> |
| <i>ag</i> | <i>ac</i> | <i>tg</i> | <i>tc</i> |
| <i>aa</i> | <i>at</i> | <i>ta</i> | <i>tt</i> |

$K=2$

| | | | | | | | |
|------------|------------|------------|------------|------------|------------|------------|------------|
| <i>ggg</i> | <i>ggc</i> | <i>gcg</i> | <i>gcc</i> | <i>cgg</i> | <i>cgc</i> | <i>ccg</i> | <i>ccc</i> |
| <i>gga</i> | <i>ggt</i> | <i>gca</i> | <i>gct</i> | <i>cga</i> | <i>cgt</i> | <i>cca</i> | <i>cct</i> |
| <i>gag</i> | <i>gac</i> | <i>gtg</i> | <i>gtc</i> | <i>cag</i> | <i>cac</i> | <i>ctg</i> | <i>ctc</i> |
| <i>gaa</i> | <i>gat</i> | <i>gta</i> | <i>gtt</i> | <i>caa</i> | <i>cat</i> | <i>cta</i> | <i>ctt</i> |
| <i>agg</i> | <i>agc</i> | <i>acg</i> | <i>acc</i> | <i>tgg</i> | <i>tgc</i> | <i>tcg</i> | <i>tcc</i> |
| <i>aga</i> | <i>agt</i> | <i>aca</i> | <i>act</i> | <i>tga</i> | <i>tgt</i> | <i>tca</i> | <i>tct</i> |
| <i>aag</i> | <i>aac</i> | <i>atg</i> | <i>atc</i> | <i>taq</i> | <i>tac</i> | <i>ttg</i> | <i>ttc</i> |
| <i>aaa</i> | <i>aat</i> | <i>ata</i> | <i>att</i> | <i>taa</i> | <i>tat</i> | <i>tta</i> | <i>ttt</i> |

$K=3$

The algorithm:

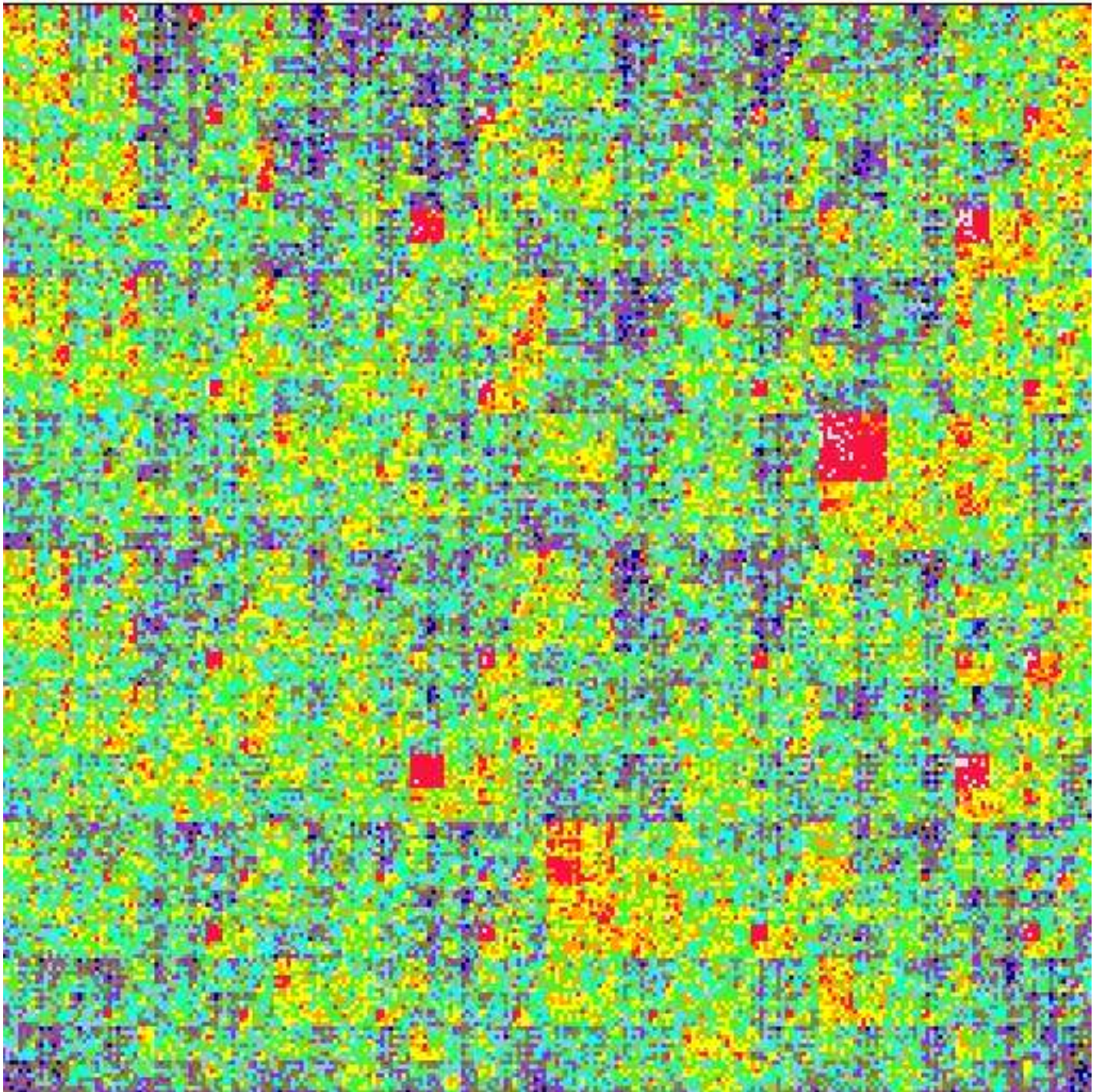
$$M^{(1)} \equiv M = \begin{pmatrix} g & c \\ a & t \end{pmatrix}.$$

$$M^{(2)} = M \times M = \begin{pmatrix} g & c \\ a & t \end{pmatrix} \times \begin{pmatrix} g & c \\ a & t \end{pmatrix}.$$

$$M^{(3)} = M \times M \times M.$$

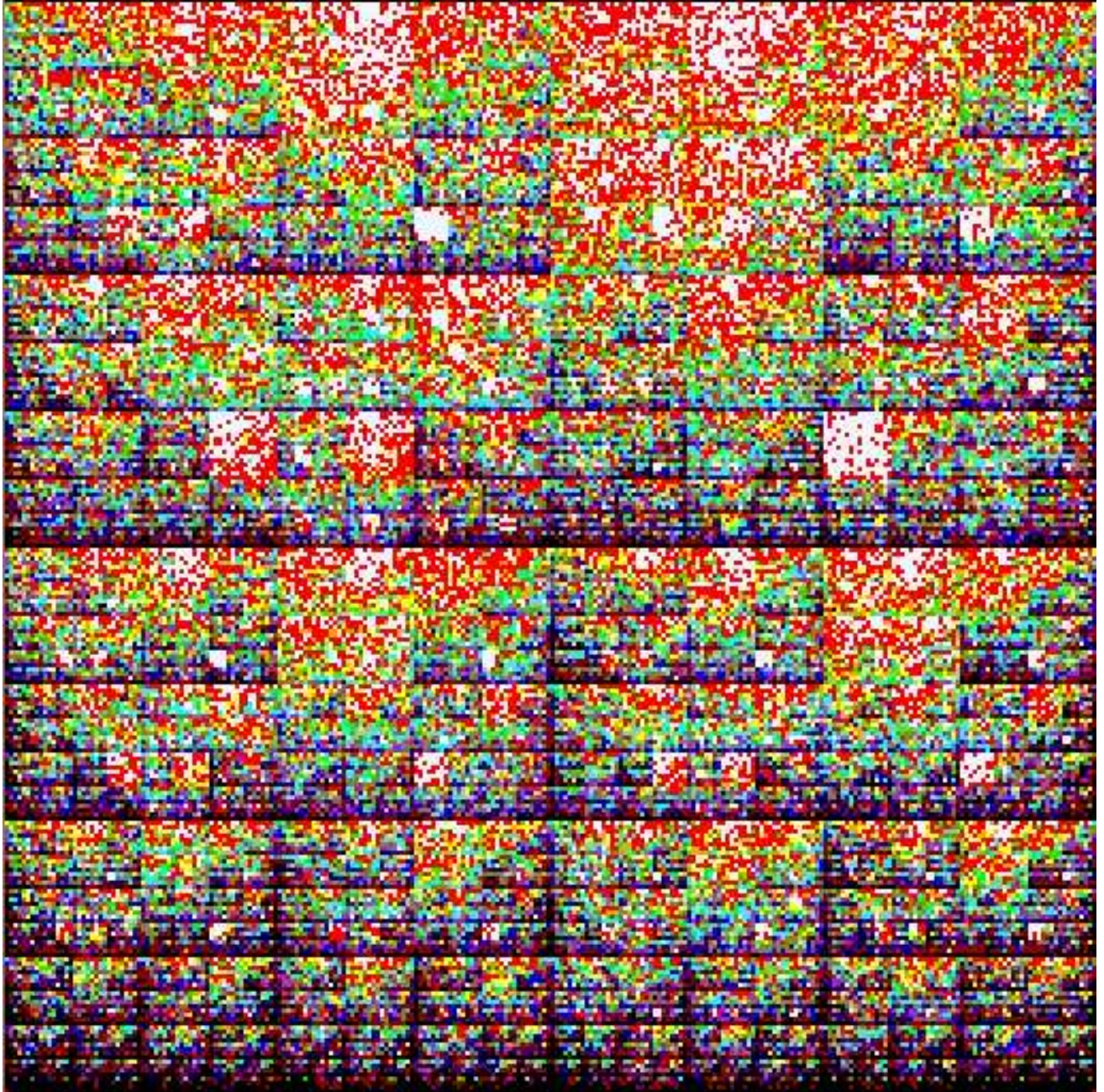
$$M^K = M \times M \times \dots \times M.$$

Portrait of *E. coli* at $K=8$



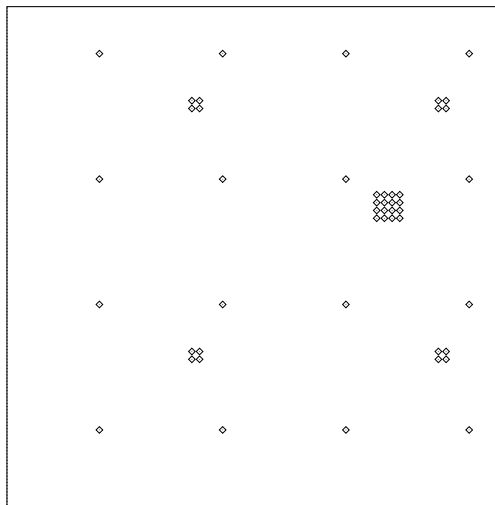
***Escherichia coli* K-12 ($K=8$)**

Portrait of *M. jannaschii* at $K=8$

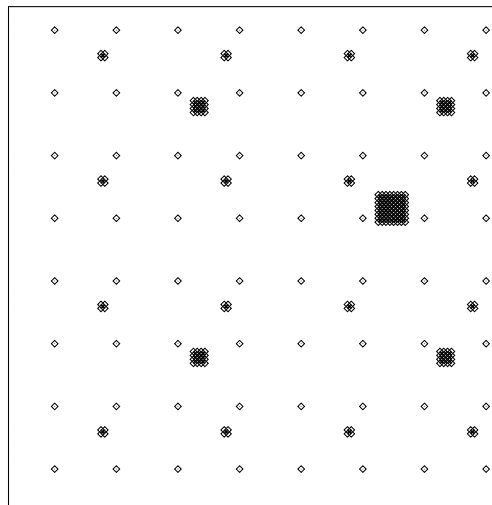


***M. jannaschii* ($K=8$)**

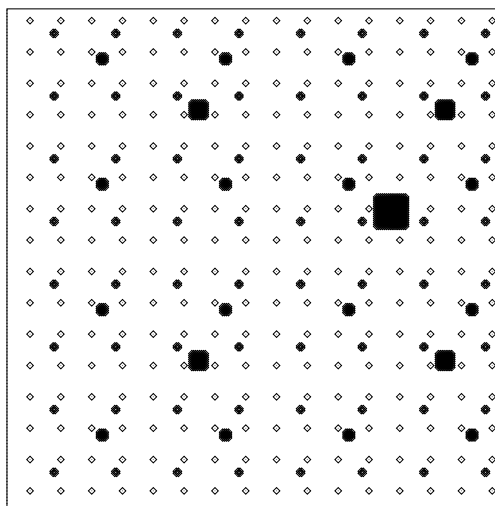
Location of ctag-tagged Strings



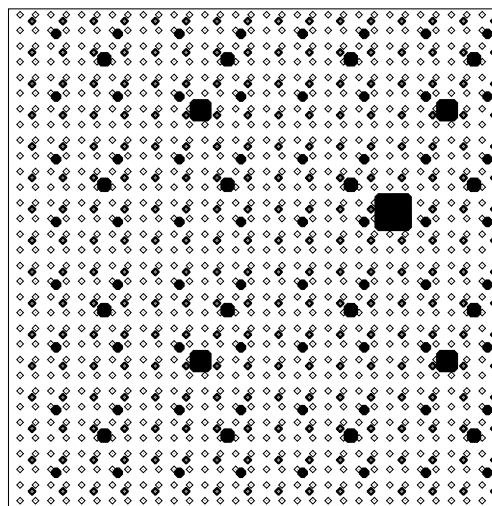
K=6



K=7

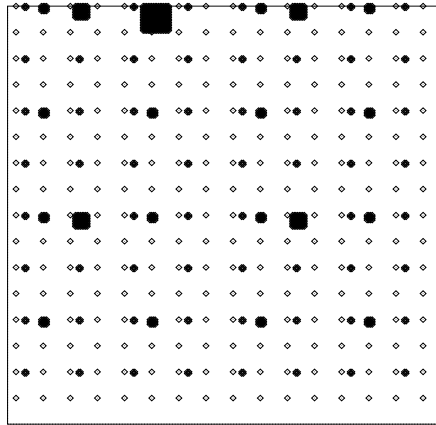


K=8

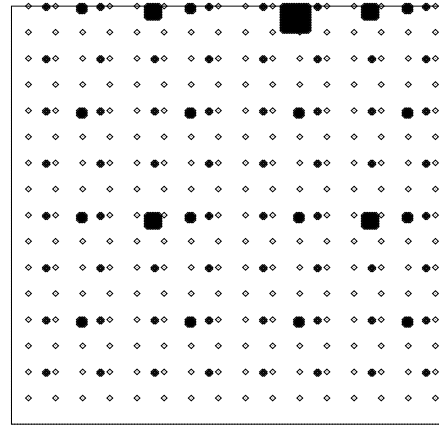


K=9

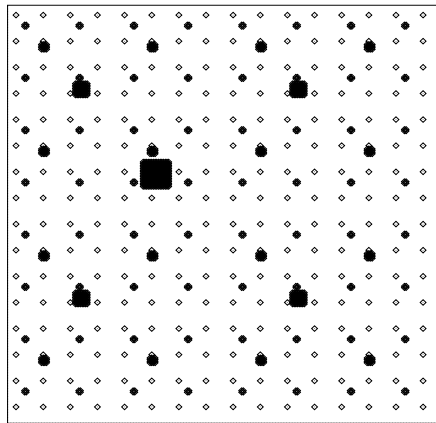
Some “4-String”-tagged Templates ($K = 8$)



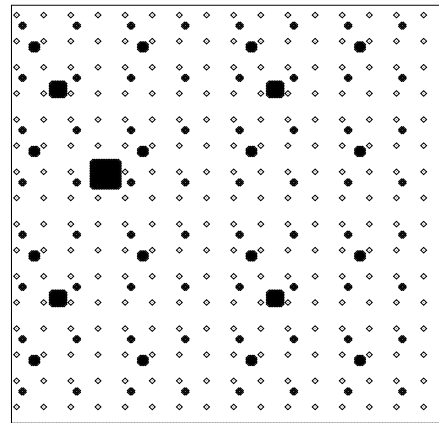
gcgc



cgcg

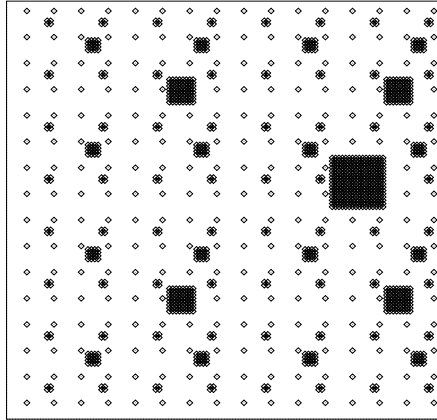


gtac

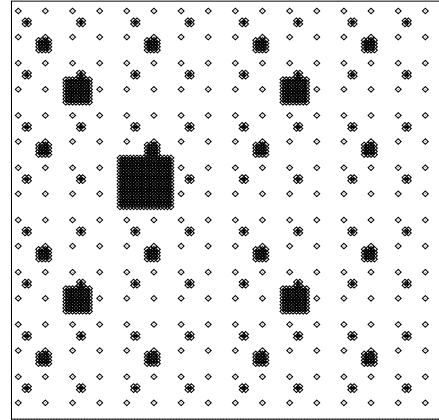


gatc

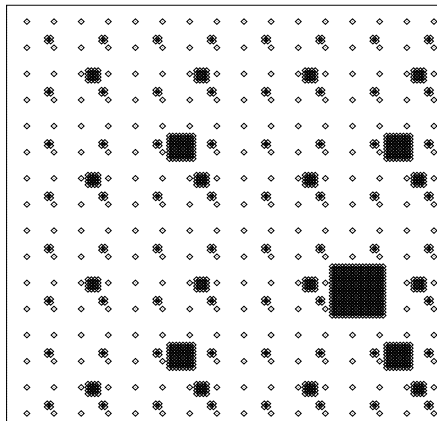
Some “3-String”-tagged Templates ($K = 8$)



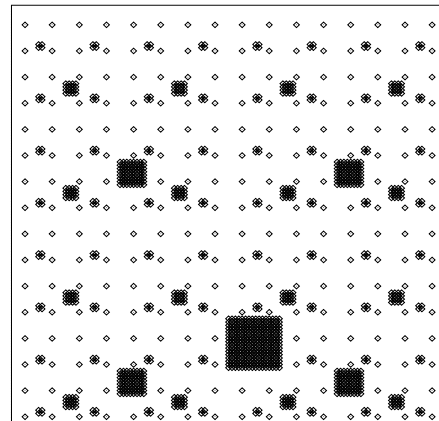
cta



gta



tca



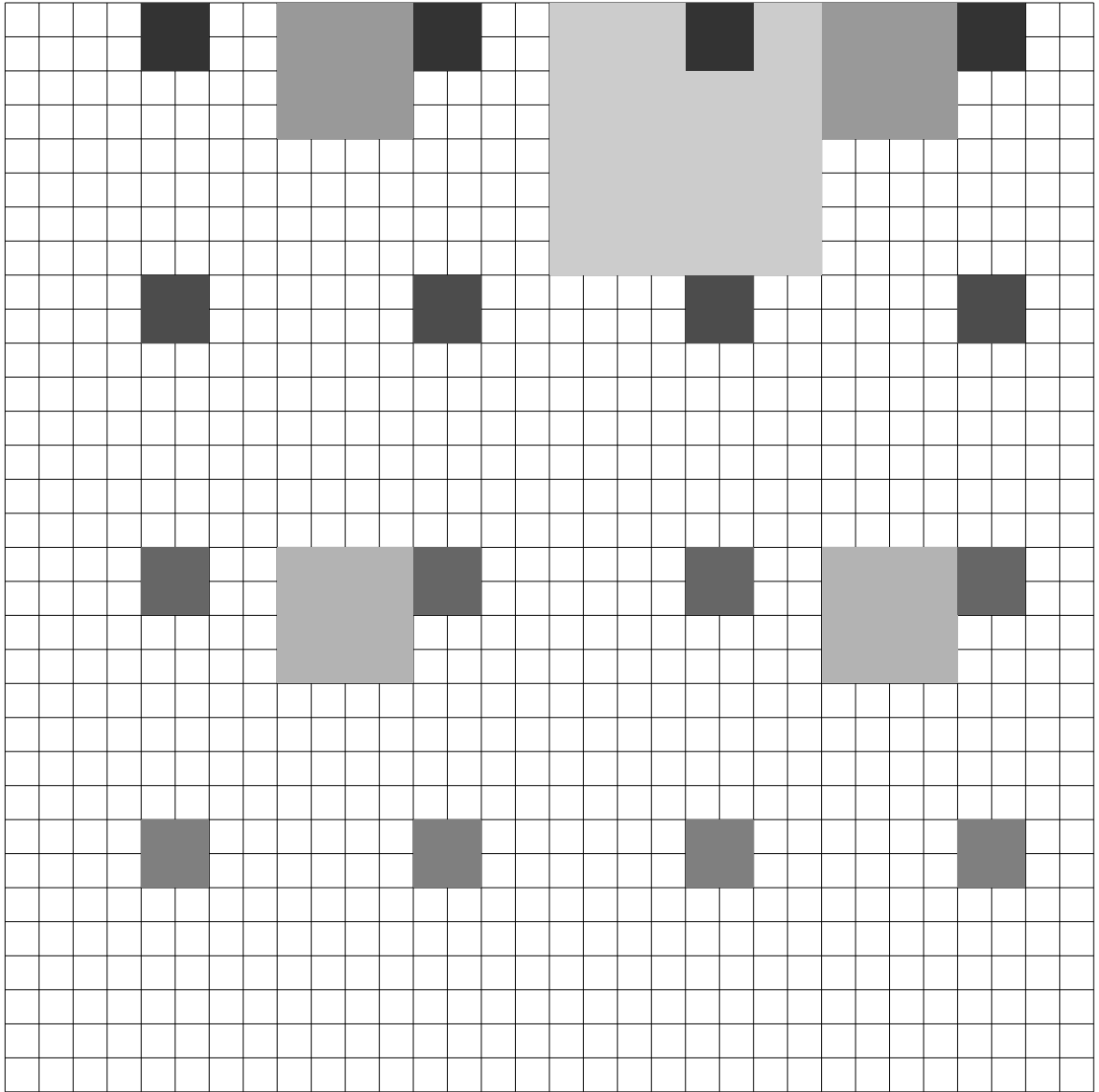
tag

**In the non-biological limit $K \rightarrow \infty$
We get a neat biology-inspired
Mathematical problem, namely,**

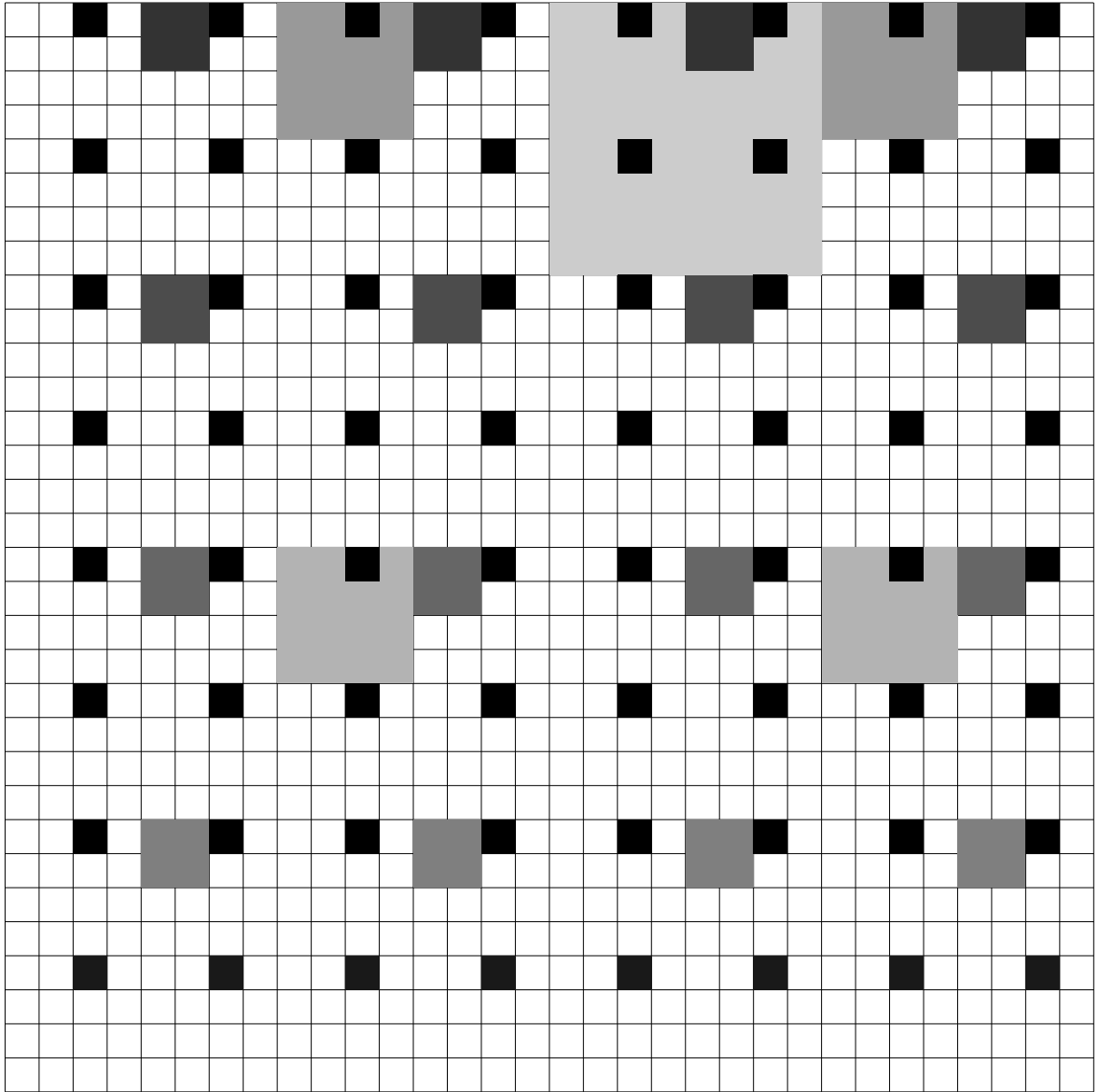
**Our First Case-Study:
The Fractal Dimension of
The Complementary Set of
Tagged-String Templates**

There may be one or more tags.

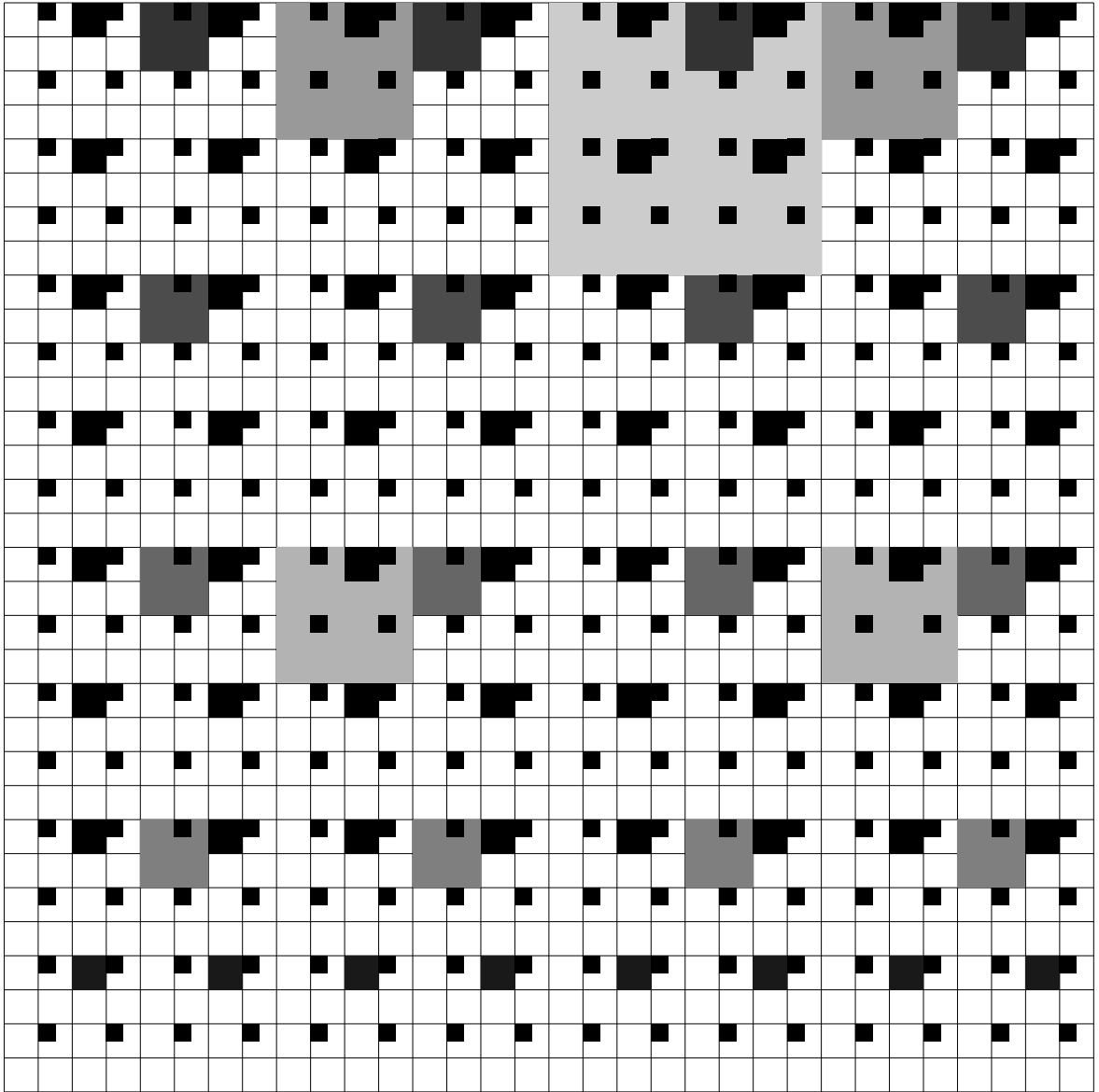
A cg-tagged Template



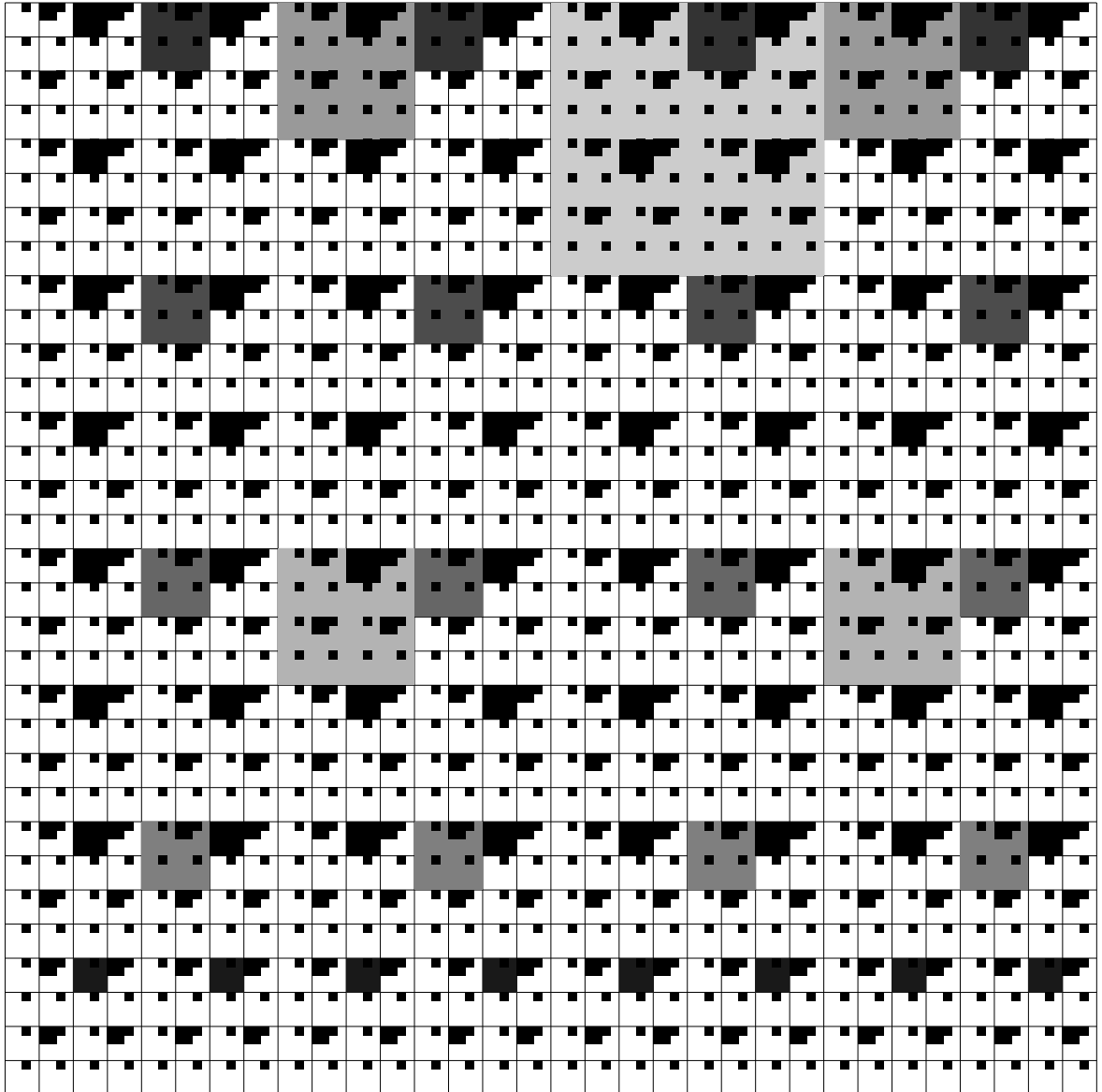
A cg-tagged Template



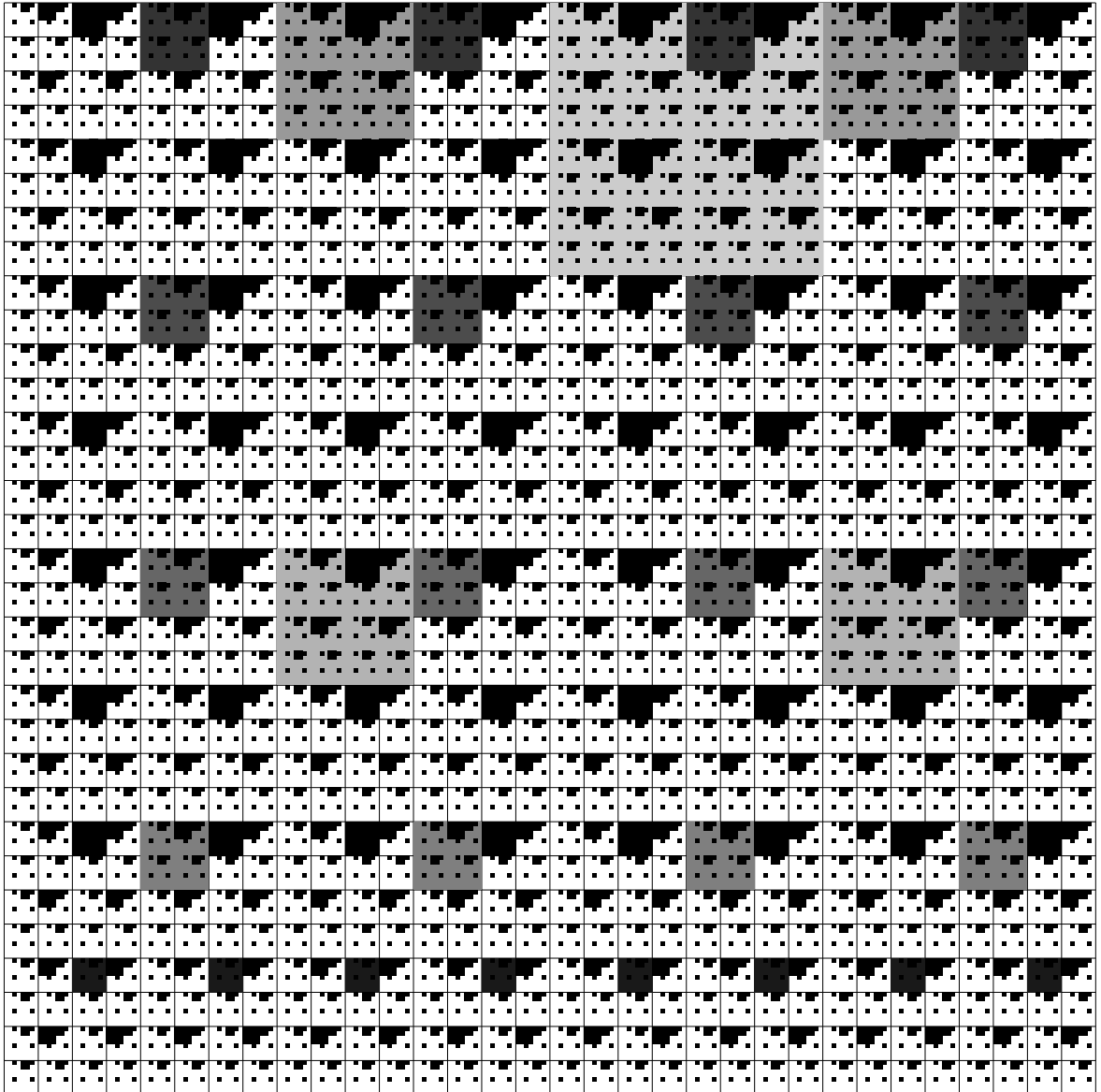
A cg-tagged Template



A cg-tagged Template



A cg-tagged Template



A Trivial Example: g-tagged String Templates

A g -tagged string means any string containing the letter g .

The set of strings that avoid any g -tagged strings are actually made of the three letters $\{a, c, t\}$.

Number of such strings of length K :

$$a_K = 3^K$$

This can be written as a **recursion relation**

$$a_0 = 1, \quad a_K = 3a_{K-1}$$

Trivial Example (continued)

Let us define a **Generating Function** of an auxiliary variable s such that

$$f(s) = \sum_{K=0}^{\infty} a_K s^K.$$

If we could calculate $f(s)$ all a_K would be obtained at one clap.

Having a **recursion function** at hand it is easy to derive the generating function. Let

$$sf(s) = \sum_{K=0}^{\infty} a_K s^{K+1} = \sum_{K=1}^{\infty} a_{K-1} s^K$$

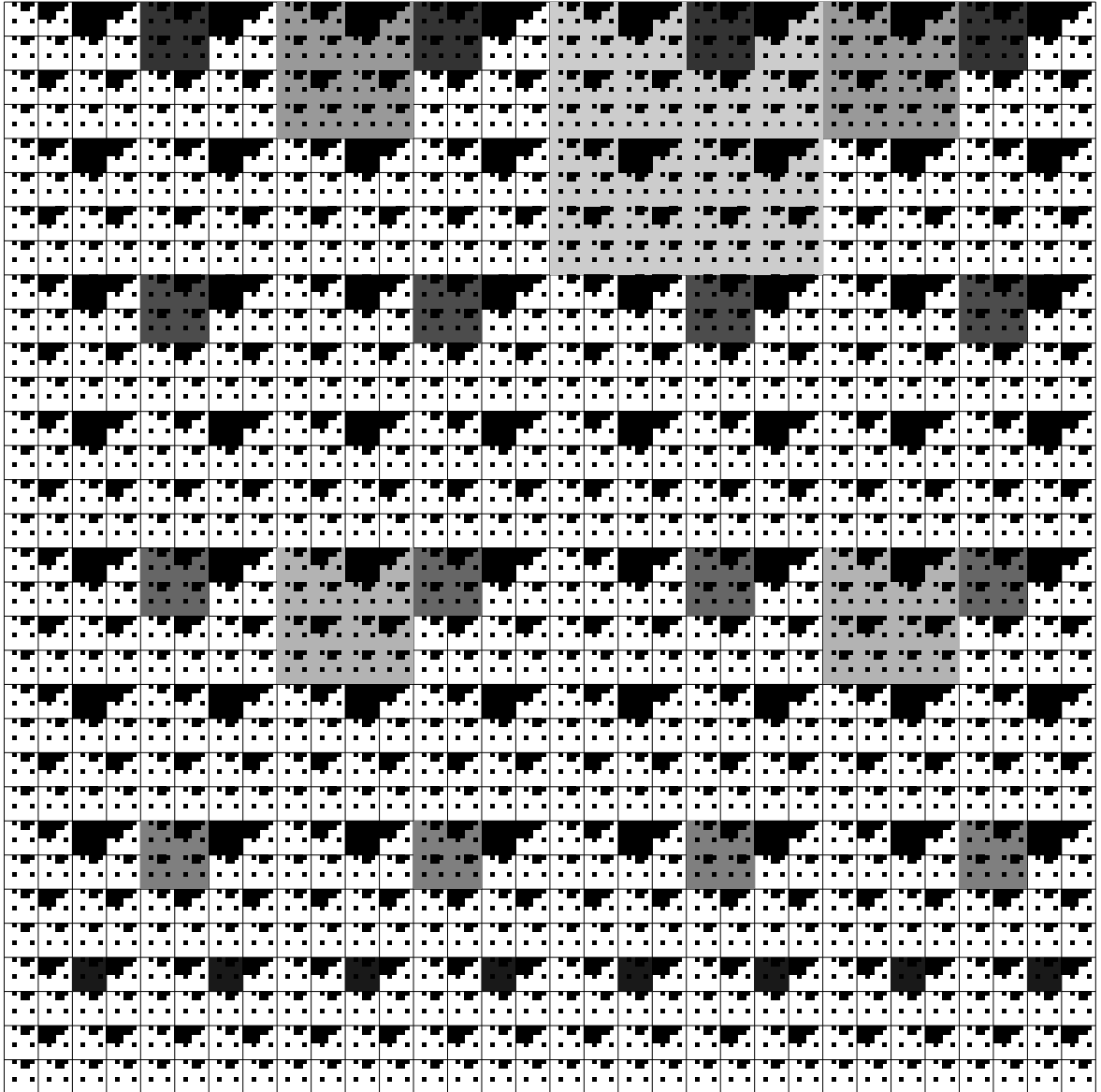
Now combine

$$\begin{aligned} f(s) - 3sf(s) &= \sum_{K=0}^{\infty} a_K s^K - 3 \sum_{K=1}^{\infty} a_{K-1} s^K \\ &= a_0 + \sum_{K=1}^{\infty} [a_K - 3a_{K-1}] s^K \\ &= a_0 \end{aligned}$$

Therefore,

$$f(s) = \frac{1}{1 - 3s}$$

A cg-tagged Template



Note the overlaps of shadowed regions.

Number of cg-free K -Strings

The number a_K is given by expanding the generating function:

$$f(s) = \frac{1}{1 - 4s + s^2}$$

For the time being we need a_K to define the fractal dimension of the complement of the K -string-tagged template.

Dimension of Complementary Set of Tagged-Strings

Σ — an alphabet ($\{a, c, g, t\}$)

Σ^* — collection of all strings on Σ

B — a subset of **avoided** strings

a_K — number of strings of length K in Σ^*
that do not contain any factor from B .

$k = 0, \quad a_0 = 1$ — linear size $\delta_0 = 1$

$K = 1, \quad a_1$ — linear size $\delta_1 = 1/2$

$K = 2, \quad a_2$ — linear size $\delta_2 = 1/4$

$K = 3, \quad a_3$ — linear size $\delta_3 = 1/8$

$K, \quad a_K$ — linear size $\delta_K = \frac{1}{2^K}$

$$D = \lim_{K \rightarrow \infty} \frac{\log a_K}{-\log \delta_K} = \lim_{K \rightarrow \infty} \frac{\log a_K^{1/K}}{\log 2}$$

Dimension of Complementary Set

$$D = \lim_{K \rightarrow \infty} \frac{\log a_K}{-\log \delta_K} = \lim_{K \rightarrow \infty} \frac{\log a_K^{1/K}}{\log 2}$$

$$f(s) = \sum_{K=0}^{\infty} a_K s^K$$

$$\lim_{K \rightarrow \infty} a_K^{1/K} = |\lambda| = \frac{1}{|s_0|}$$

λ — radius of convergence (Cauchy)

s_0 — minimal module zero of $f^{-1}(s)$

$$D = -\frac{\log |s_0|}{\log 2}$$

Under-Represented $K = 4$ Palindroms Seen in the Bacterial Genomes

| Bacteria | Avoided Strings | | | |
|----------|-----------------|------|-----------|----------------|
| Ecoli | ctag | | | |
| Tmar | ctag | | | |
| Bsub | ctag | | | |
| Nmen | ctag | | | |
| NmenA | ctag | | | |
| pNGR | ctag | | | |
| Dra1 | ctag | | tata | atat |
| Dra2 | ctag | | tata | atat |
| Aful | ctag | | gcgc | cgcg |
| Mthe | ctag | | gcgc | cgcg |
| Tpal | ctag | | | ggcc |
| Aqua | ctag | | tcga | gcgc cgcg ggcc |
| Mjan | ctag | gatc | gtac | gcgc cgcg |
| Hpyl | | acgt | gtac | tcga |
| Hpyl99 | | acgt | gtac | tcga |
| Hinf | | | | ggcc ccgg |
| Uure | | | | ggcc ccgg |
| Bbur | | | | cgcg |
| Synecho | | | gcgc | cgcg |
| Pyro | | | gcgc | cgcg |
| Aero | | | gcgc | aatt |
| Mgen | | | None seen | clearly |
| Mpneu | | | None seen | clearly |
| Cjej | | | None seen | clearly |
| Ctra | | | None seen | clearly |
| CtraM | | | None seen | clearly |
| Cpneu | | | None seen | clearly |
| CpneuA | | | None seen | clearly |
| CpneuJ | | | None seen | clearly |
| Mtub | | | None seen | clearly |
| Rpxx | | | None seen | clearly |
| Xfas | | | None seen | clearly |

First Avoided Strings in Bacteria

| Species | K_0 | N_{K_0} | N_{K_0+1} | First Avoided Strings |
|----------------|-------|-----------|-------------|------------------------------------|
| <i>Ecoli</i> | 7 | 1 | 173 | gCCTAGG |
| <i>Synecho</i> | 7 | 1 | 149 | aCGCGCG |
| <i>NmenA</i> | 7 | 1 | 729 | aGATCcc |
| <i>CtraM</i> | 7 | 1 | 666 | cgcCCGG |
| <i>Tmar</i> | 7 | 2 | 594 | CCTAGGg tacCTAG |
| <i>CpneuA</i> | 7 | 2 | 452 | cGGCCcg CCGGgcg |
| <i>CpneuJ</i> | 7 | 2 | 447 | cgGGCCg cgcCCGG |
| <i>Hpyl99</i> | 6 | 1 | 130 | GTCGAC |
| <i>Hpyl</i> | 6 | 2 | 192 | GTCGAC TCGAca |
| <i>Mjan</i> | 6 | 3 | 318 | GCGCGC GTCGAC CGATCG |
| <i>Mtub</i> | 7 | 3 | 595 | TATAatg tatgtta taaaata |
| <i>Pabyssi</i> | 7 | 3 | 291 | GCGCGCg CGCGCGa tGCGCGC |
| <i>Aquae</i> | 7 | 4 | 840 | GCGCGCg GCGCGCc cGCGCGC tGCGCGC |
| <i>Aful</i> | 7 | 4 | 365 | GCGCGCg cGCGCGC gcaCTAG cACTAGT |
| <i>Pyro</i> | 7 | 4 | 708 | GCGCgta tGCGCcg ccgtgcg cgtgcga |

The Second Problem: The Number of True and Redundant Avoided Strings at Different String Length K

Suppose that a certain string S of length K_0 is missing in a genome with all its proper substrings present (a **true** avoided string).

Then at the next length $K_0 + 1$ it may take away 8 strings, as any string of form αS or $S\beta$ cannot occur, where α or β is one of $\{a, c, g, t\}$ (**redundant** avoided strings).

Simple induction leads to the conclusion that at length $K_0 + i$ the number of redundant avoided strings is $4^i(i + 1)$.

However, this result is not always true. It is a “mean-field”-type approximation ignoring possible self-overlapping in the string S .

How to get the exact number of redundant avoided strings when there are overlaps among the true ones?

One Problem with Two Solutions

The two problems (fractal dimension and number of avoided strings) turn out to be one and the same, as the first is a graphic representation of the second.

There are two methods to solve the problem:

1. Combinatorial solution using the Goulden-Jackson cluster method
2. Language theory solution making use of so-called factorizable language

Workshop on Combinatorics and Physics Los Alamos National Laboratory 1998

Suggestion by Zeilberger:

Use the Goulden-Jackson cluster method:
generating functions

Realized by XIE Huimin

Construct finite automaton of a factorizable
language defined by the genome

Set of minimal forbidden words → transfer
functions

First Solution: Combinatorics

Using the Goulden-Jackson cluster method to calculate the generating functions

I. Goulden, and D. M. Jackson, “An inversion theorem for cluster decompositions of sequences with distinguished subsequences”, *J. London Math. Soc.* **20** (1979) 567 – 576.

I. Goulden, and D. M. Jackson, *Combinatorial Enumeration*, Wiley, New York, 1983.

J. Noonan, D. Zeilberger, “The Goulden-Jackson cluster method: extension, applications and implementations” (1998), available at:
<http://www.math.temple.edu/~zeilberg/>

Generating Functions for Single Tags

| <i>Tag</i> | $f(s)$ | D |
|-------------|---|-------------------------|
| <i>g</i> | $\frac{1}{1-3s}$ | $\frac{\log 3}{\log 2}$ |
| <i>gc</i> | $\frac{1}{1-4s+s^2}$ | 1.89997 |
| <i>gg</i> | $\frac{1+s}{1-3s-3s^2}$ | 1.92266 |
| <i>gct</i> | $\frac{1}{1-4s+s^3}$ | 1.97652 |
| <i>gcg</i> | $\frac{1+s^2}{1-4s+s^2-3s^3}$ | 1.978 |
| <i>ggg</i> | $\frac{1+s+s^2}{1-3s-3s^2-3s^3}$ | 1.98235 |
| <i>ctag</i> | $\frac{1}{1-4s+s^4}$ | 1.99429 |
| <i>ggcg</i> | $\frac{1+s^3}{1-4s+s^3-3s^4}$ | 1.99438 |
| <i>gcgc</i> | $\frac{1+s^2}{1-4s+s^2-4s^3+s^4}$ | 1.99463 |
| <i>gggg</i> | $\frac{1+s+s^2+s^3}{1-3s-3s^2-3s^3-3s^4}$ | 1.99572 |

Number of Different $f(s)$

n — length of tag

$G(n)$ — number of Generating Function types

| | | | | | | | | | |
|--------|---|---|---|---|---|---|----|----|----|
| n | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| $G(n)$ | 1 | 2 | 3 | 4 | 6 | 8 | 10 | 13 | 17 |

$G(n)$: 1, 2, 3, 4, 6, 8, 10, 13, 17, 21, 27, 30, 37, 47, 57, 62, 75, 87, 102, 116, 135, 155, 180, 194, ...

$G(n)$ are given by **correlations of n** , i.e., integer sequence M0555 in N. j. A. Sloane and S. Plouffe, *The Encyclopedia of Integer Sequences*, Academic Press, 1995.

[http://akpublic.research.att.com/
~njas/sequences](http://akpublic.research.att.com/~njas/sequences)

Aquifex aeolicus

— a hyperthermophilic bacterium

Nature **392** (1998) 353-358

155 1335 bp

$$K_0 = 7$$

$$N_{K_0} = 4$$

$$B = \left\{ \begin{array}{cc} GCGCGC_g & GCGCGC_a \\ cGCGCGC & tGCGCGC \end{array} \right\}$$

Generating function:

Goulden-Jackson cluster method (1979)

$$f(s) = \frac{1 + s^2 + s^4 + s^6 + s^8 + s^{10} + s^{12}}{1 - 4s + s^2 - 4s^3 + s^4 - 4s^5 + s^6 - 4s^8 - 4s^{10} - 4s^{12}}$$

$$\frac{1}{1 - 4s} - f(s) = 4s^7 + 27s^8 + 152s^9 + 784s^{10} + 3840s^{11} + \dots$$

hao: a01

Language Theory Solution

Formal language theory is not just formal.

Given a right problem and context it may provide a computational framework and useful constructions.

How it works?

hao: a24

Factorizable Language

Definitions:

Σ — alphabet $\{a, c, g, t\}$

Σ^* — collection of all strings on Σ including the empty string ϵ

Any subset $L \in \Sigma^*$ is a **Language**.

A language is called **factorizable** if any substring of an admissible word $x \in L$ is also admissible, i.e., if $x = yz$ then both $y \in L$ and $z \in L$.

Languages defined by symbolic dynamics are factorizable.

Given a genome G , cut it in all possible ways, from single letters to the whole genome. Collect all these strings and include ϵ we get a factorizable language by construction.

A factorizable language is determined entirely by a minimal set of forbidden words or Distinct Excluded Blocks (DEB) by S. Wolfram.

Equivalence Relation R_L :

Any language $L \subset \Sigma^*$ introduces an Equivalence Relation R_L in Σ^* with respect to L :

For any pair $x, y \in \Sigma^*$

$$xR_L y \text{ or } x \sim y$$

iff for each $z \in \Sigma^*$ either both

$$xz, yz \in L$$

or both

$$xz, yz \notin L.$$

Index of R_L :

$\text{index}(R_L)$ = number of Equivalence Classes in Σ^* with respect to L .

Myhill-Nerode Theorem (1957-58):

1. L is regular iff $\text{index}(R_L)$ is finite.
2. L regular \Rightarrow $\text{minDFA}(L)$ is unique up to an isomorphism.
3. Number of states of $\text{minDFA}(L) = \text{index}(R_L)$.

Let L be a factorizable language and L'' be its set of all DEB's.

Define

$$V = \{v \mid v \text{ is a proper prefix of some } y \in L''\}$$

For each word $x \in L$ there exists a string $v \in V$ such that

$$xR_Lv.$$

In other words, all Equivalence Classes are represented in the set V . In order to find all equivalence Classes of Σ^* with respect to L it is enough to work with L'' .

$[\epsilon]$ is an equivalence class.

L' is an equivalence class.

For two given strings $u, v \in V$,
 $uR_L v$ iff for each $z \in \Sigma^*$
 uz contains a DEB as its suffix $\Leftrightarrow vz \in L'$
and *vice versa*.

This statement sets the computation rule:

1. Start from a given u .
2. Pick up those $z \in \Sigma^*$ that
 $z \notin L'$ and form all uz that contains
a DEB as its suffix;
3. For each v check whether $vz \in L'$.
4. If not then $[u] \neq [v]$.
5. If yes, check v for each $z \in \Sigma^*$
in the other way around.
6. Go through the upper triangle in $\{u, v\}$.

Minimal Deterministic Automaton

$$\text{minDFA}(Q, \Sigma, \delta, q_0)$$

$\Sigma = \{a, c, g, t\}$ — alphabet

$Q = \{[x_i]\}, x_i \in \Sigma^*$ — set of states

q_0 — unique initial state

δ — transfer function defined by

$$\delta([x_i], s) = [x_i s],$$

where $[x_i] \in \Sigma^*, s \in \Sigma$.

Aquifex aeolicus

4 avoided strings at $K = 7$:

gcgcgcg gcgcgca cgcgcgc tgcgcgc

14 equivalence classes:

| | | | |
|--------------|-----------|------------|-----------|
| $[\epsilon]$ | $[g]$ | $[gc]$ | $[gcg]$ |
| $[gcgc]$ | $[gcgcg]$ | $[gcgcgc]$ | $[c]$ |
| $[cg]$ | $[cgc]$ | $[cgcg]$ | $[cgcgc]$ |
| $[cgcgcg]$ | L' | | |

Transfer Function $\delta(i, j)$:

| | <i>a</i> | <i>c</i> | <i>g</i> | <i>t</i> |
|--------------|--------------|------------|------------|----------|
| $[\epsilon]$ | $[\epsilon]$ | $[c]$ | $[g]$ | $[c]$ |
| $[g]$ | $[\epsilon]$ | $[gc]$ | $[g]$ | $[c]$ |
| $[gc]$ | $[\epsilon]$ | $[c]$ | $[gcg]$ | $[c]$ |
| $[gcg]$ | $[\epsilon]$ | $[gcgc]$ | $[g]$ | $[c]$ |
| $[gcgc]$ | $[\epsilon]$ | $[c]$ | $[gcgcg]$ | $[c]$ |
| $[gcgcg]$ | $[\epsilon]$ | $[gcgcgc]$ | $[g]$ | $[c]$ |
| $[gcgcgc]$ | L' | $[c]$ | L' | $[c]$ |
| $[c]$ | $[\epsilon]$ | $[c]$ | $[cg]$ | $[c]$ |
| $[cg]$ | $[\epsilon]$ | $[cgc]$ | $[g]$ | $[c]$ |
| $[cgc]$ | $[\epsilon]$ | $[c]$ | $[cgcg]$ | $[c]$ |
| $[cgcg]$ | $[\epsilon]$ | $[cgcgc]$ | $[g]$ | $[c]$ |
| $[cgcgc]$ | $[\epsilon]$ | $[c]$ | $[cgcgcg]$ | $[c]$ |
| $[cgcgcg]$ | $[\epsilon]$ | L' | $[g]$ | $[c]$ |

$$(M^n)_{1j}$$

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|----|----|-----|------|------|-------|-------|--------|---------|
| 1 | 4 | 16 | 64 | 256 | 1024 | 4095 | 16378 | 65501 | 261960 |
| 1 | 2 | 8 | 32 | 128 | 512 | 2048 | 8190 | 32756 | 131002 |
| 0 | 1 | 2 | 8 | 32 | 128 | 512 | 2048 | 8190 | 32756 |
| 0 | 0 | 1 | 2 | 8 | 32 | 128 | 512 | 2048 | 8190 |
| 0 | 0 | 0 | 1 | 2 | 8 | 32 | 128 | 512 | 2048 |
| 0 | 0 | 0 | 0 | 1 | 2 | 8 | 32 | 128 | 512 |
| 0 | 0 | 0 | 0 | 0 | 1 | 2 | 8 | 32 | 128 |
| 2 | 7 | 28 | 112 | 448 | 1792 | 7168 | 28665 | 114640 | 458483 |
| 0 | 2 | 7 | 28 | 112 | 448 | 1792 | 7168 | 28665 | 114640 |
| 0 | 0 | 2 | 7 | 28 | 112 | 448 | 1792 | 7168 | 28665 |
| 0 | 0 | 0 | 2 | 7 | 28 | 112 | 448 | 1792 | 7168 |
| 0 | 0 | 0 | 0 | 2 | 7 | 28 | 112 | 448 | 1792 |
| 0 | 0 | 0 | 0 | 0 | 2 | 7 | 28 | 112 | 448 |
| 4 | 16 | 64 | 256 | 1024 | 4096 | 16380 | 65509 | 261992 | 1047792 |

hao: a12

Generating Function:

$$f(s) = \sum_0^{\infty} a_K s^K$$

The 1st Rows of M^K :

$$a_K = \sum_{j=1}^{13} (M^K)_{1j}$$

Sum over all equivalence classes except for L' , using a result of S. Wolfram (1984)

M contains more detailed information than $f(s)$.

hao: a04

The 3rd Case-Study

Decomposition

and

Reconstruction

of Protein

(Amino Acid)

Sequences

How unique They Are?

(Change to PowerPoint)