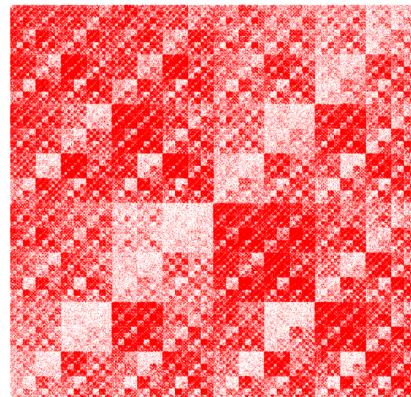


Universidad Iberoamericana  
Ingeniería Física  
PROYECTO ASE III

Alejandro A. Muñoz G.

ANÁLISIS MULTIFRACTAL DE SECUENCIAS  
CDS Y NCDS EN ADN



Trabajo dirigido por  
M. en C. Gabriela Durán Meza

Sinodales  
Mtra. Jeanett López García  
Dra. Ana María Aguilar Molina

Otoño 2020

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

---

## Índice

<b>Índice</b>	<b>1</b>
<b>1. Objetivos</b>	<b>2</b>
<b>2. Presentación</b>	<b>2</b>
<b>3. Marco teórico</b>	<b>3</b>
3.1. Genética . . . . .	3
3.1.1. ADN . . . . .	3
3.1.2. CDS y NCDS . . . . .	3
3.2. Fractales . . . . .	4
3.3. Autosimilaridad y Dimensión . . . . .	5
3.4. Sistema de Funciones Iteradas (IFS) . . . . .	7
3.5. Juego del Caos . . . . .	9
3.6. Intervalo Unitario . . . . .	10
3.6.1. Números como Series Infinitas . . . . .	10
3.6.2. Descomposicion Intervalo Unitario . . . . .	10
3.6.3. Clasificacion de Frecuencias . . . . .	12
3.6.4. Autosimilaridad del Intervalo Unitario . . . . .	14
3.6.5. Teorema de Eggleston . . . . .	15
3.6.6. Multifractalidad de Intervalo Unitario . . . . .	16
<b>4. Desarollo</b>	<b>17</b>
4.1. ADN como Secuencia de Caracteres . . . . .	17
4.2. Juego del Caos extendio . . . . .	18
4.3. Juego del Caos en ADN . . . . .	19
4.4. Separación de Secuencias CDS y NCDS . . . . .	22
4.5. Análisis Multifractal del ADN . . . . .	23
<b>5. Concluciones</b>	<b>27</b>
5.1. Trabajo a Futuro . . . . .	28
5.2. Conclusión . . . . .	28
<b>Bibliografía</b>	<b>29</b>

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

---

## 1. Objetivos

En este proyecto de titulación ASE III, se plantea hacer un estudio multifractal de las secuencias codificantes y no codificantes de ADN.

El proyecto consta de dos objetivos:

1. Demostrar que el ADN tiene una estructura fractal
2. Medir los espectros multifractales de secuencias codificantes y no codificantes

## 2. Presentación

Desde su descubrimiento en 1953, el ADN nos ha revelado muchos de los mecanismos de la vida y la evolución de esta en la tierra. La genética ha cambiado nuestro acercamiento a la medicina, paleontología, investigación forense, agricultura, ganadería, entre muchos otros campos. Nos ha permitido desarrollar vacunas, entender enfermedades hereditarias y desarrollar tomates resistentes a pesticidas. Es casi seguro que la genética dejará huella en nuestra especie a largo plazo.

Sin embargo, aún no entendemos del todo a esta molécula que contiene la información genética en todos los seres vivos. Sabemos que existen genes dentro del ADN, secuencias que le dicen a la célula cómo llevar a cabo sus procesos, que moléculas hay que construir, “Coding Sequence” (CDS). Pero existe una gran parte del código genético que no da instrucciones, no se traduce en la síntesis de proteínas, “Non Coding Sequence” (NCDS). El papel de los genes para los genetistas es bastante claro, en cuanto a las regiones no codificantes, se les ha dado el nombre de ADN basura. Pero investigaciones recientes sugieren lo contrario, ya que representa un 98 por ciento de nuestra secuencia del genoma. Se cree que juega un gran papel cómo regular los genes para saber dónde y cuándo se tienen que activar [8].

En este proyecto se plantea utilizar la teoría multifractal para estudiar desde un acercamiento matemático a la estructura de las CDS y NCDS.

## 3. Marco teórico

En la revisión bibliográfica se estudiaron los conceptos que fungen como la base teórica fundamental para el estudio del ADN como un objeto multifractal. Las ideas de genética, fractalidad, sistema de funciones iteradas, juego del caos y multifractalidad se exploran y desarrollan a continuación.

### 3.1. Genética

#### 3.1.1. ADN

El ácido desoxirribonucleico, es la molécula que contiene la información genética en todos los seres vivos. Está compuesta de cuatro moléculas o cuatro bases: adenina (A), citosina (C), guanina (G), y timina (T). Estas cuatro bases se apilan en dos cadenas que se enrollan entre sí para formar la famosa estructura de doble hélice. La información del organismo reside en el orden de estas bases dentro del ADN. Esto será relevante para el desarrollo del proyecto.

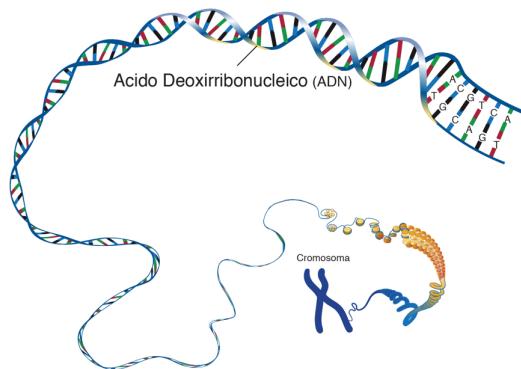


Figura 1: Visualización del ADN y su estructura.

#### 3.1.2. CDS y NCDS

El gen es la unidad física básica de la herencia. Los genes son secuencias de nucleótidos en ADN o ARN que codifican la síntesis algún producto, ya sea ARN o alguna proteína. Los genes están dispuestos, uno tras otro, en estructuras llamadas cromosomas.

Las secuencias no codificantes de ADN no codifican para aminoácidos. La mayor parte del ADN no codificante se encuentra entre los genes en el cromosoma. Investigaciones recientes sugieren que las NCDS si juegan diferentes roles. Entre ellos el de controlar la transcripción de un gen cercano, a estas secciones se les conoce como *cis* y juegan un papel en la evolución y el control de desarrollo del organismo. Dentro de

las NCDS también podemos encontrar vestigios evolutivos conocidos como pseudogenes. Son secuencias de ADN vinculadas con genes conocidos, que han perdido su capacidad de codificación de proteínas o simplemente ya no se expresan en la célula [10].

## 3.2. Fractales

”Fractal geometry will make you see everything differently. There is a danger in reading further. You risk the loss of your childhood vision of clouds, forest, flowers, galaxies, leaves, feathers, rocks, mountains, torrents of water, carpets, bricks, and much else besides. Never again will your interpretation of these things be quite the same.”

-Michael F. Barnsley [1]

La definición formal de fractal tal y como la formuló Mandelbrot es la siguiente: Un fractal es una figura cuya dimensión de Hausdorff es mayor que su dimensión topológica [6].

Existen varias definiciones de fractal en distintos contextos, la mayoría son enunciados matemáticos muy elaborados. Para fines prácticos pensaremos que un fractal es una estructura que se repite a diferentes escalas. Es decir que si se divide al fractal en pequeños pedazos cada “pedacito” reproduce la estructura o forma original. Estos objetos fractales pueden presentar dimensiones no enteras y una estructura autosimilar, como se muestra en la figura 2.

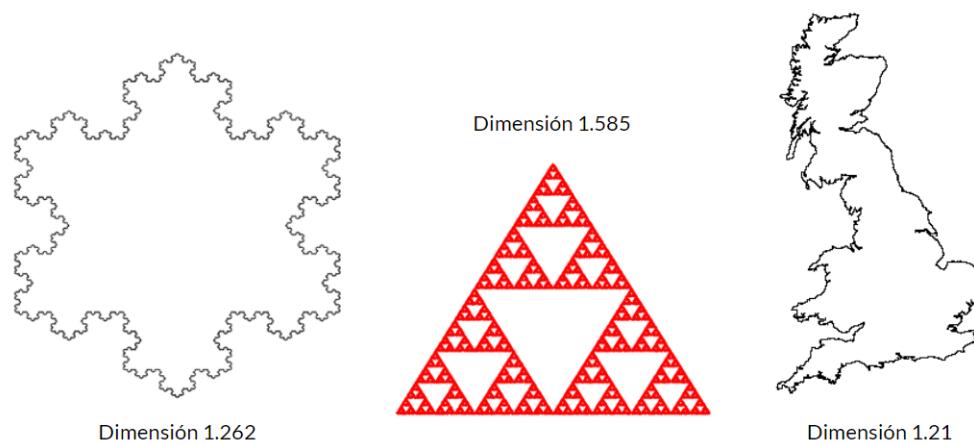


Figura 2: Ejemplos de objetos fractales con sus dimensiones.

### 3.3. Autosimilaridad y Dimensión

Como ya se mencionó un fractal es una estructura que al dividirse reproduce la estructura o forma original y que estos objetos pueden presentar dimensiones no enteras. Esta noción de dimensión no es muy intuitiva ya que estamos acostumbrados a manejar solo dimensiones enteras. Para entender esta definición tenemos que indagar en el concepto de dimensión.

Desde el punto de vista de álgebra lineal, la dimensión es el número de vectores linealmente independientes que generan un espacio. La intuición es que solo puede haber dimensiones enteras ya que no podemos tener dos vectores y fracción que generen un espacio. Así que esta definición no nos sirve para el presente trabajo.

Para llegar a obtener dimensiones no enteras tenemos que discutir primero el concepto de autosimilaridad. Una estructura autosimilar es una que, si se rompe arbitrariamente en pedazos, un pedazo a una escala menor que el conjunto completo preservará la estructura del objeto original. En las figura 3 se ilustra el concepto de autosimilaridad.

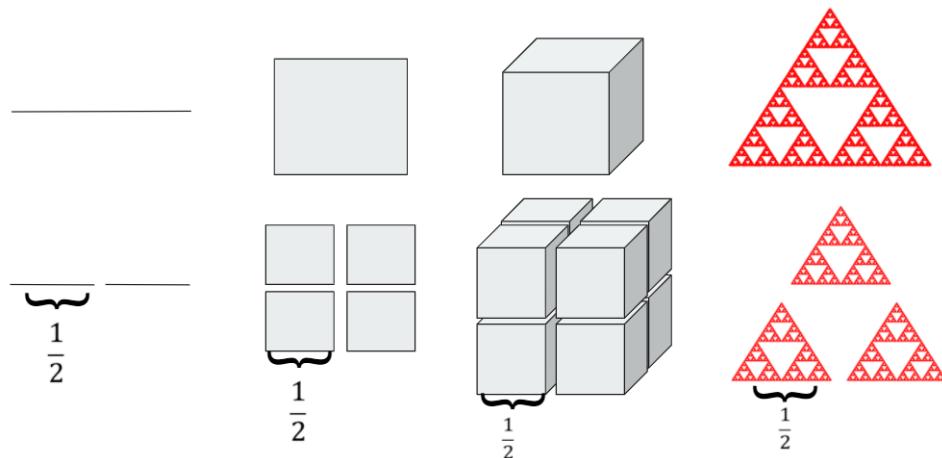


Figura 3: Figuras autosimilares escaladas por un factor de reducción de  $1/2$ .

En la figura 3 podemos apreciar como al aplicar el factor de reducción de  $1/2$  la medida (longitud) de la línea se redujo un factor de  $1/2$ . Al aplicarlo al cuadrado su medida (área) se redujo  $1/4$ . Al aplicarlo al cubo se redujo su medida (volumen) un factor de  $1/8$ . Y al aplicarlo al triángulo de Sierpinsky se redujo un factor de  $1/3$  su medida.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

Lo cual nos permite apreciar que la medida del objeto se redujo  $1/2$  elevado a la dimensión de la figura. Por ejemplo:

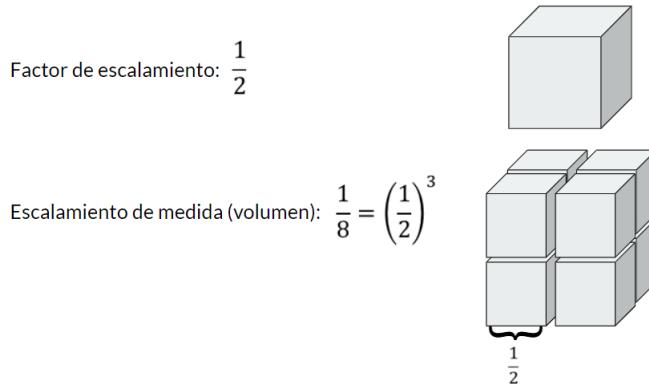


Figura 4: Escalamiento de un cubo por un factor de  $1/2$ .

Aquí se aprecia como  $1/2$  elevado a la potencia 3 (dimensión del cubo), nos da como resultado el factor por el cual se redujo el volumen del cubo, es decir,  $1/8$ . También se puede ver como si se generaron 8 piezas del cubo original.

Así que esta puede ser nuestra definición de dimensión: La dimensión de un objeto es el exponente al cual se tiene que elevar el factor de escalamiento tal que sea igual al factor de escalamiento de su medida. A esta definición también se le conoce como dimensión de fractal autosimilar, la cual está relacionada con la dimensión de Hausdorff.

Despejando a la dimensión de esta relación obtenemos la ecuación 3.1. Siendo  $q$  el número de piezas que se generaron del objeto original al aplicar la reducción, y  $r$  siendo el factor de escalamiento.

$$D = -\frac{\ln(q)}{\ln(r)} \quad (3.1)$$

Por lo tanto la dimensión del triángulo de Sierpinsky está dada por:

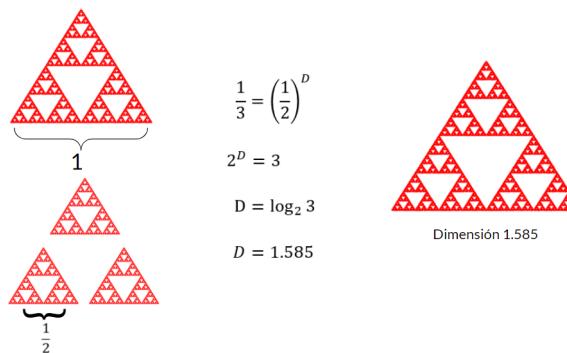


Figura 5: Cálculo dimensión del triángulo de Sierpinsky

### 3.4. Sistema de Funciones Iteradas (IFS)

El Sistema de Funciones Iteradas o IFS por sus siglas en inglés también se le conoce en las matemáticas como Multiple Reduction Copy Machine Metaphor o MRCM. La máquina de IFS toma una imagen como input, aplica diferentes transformaciones (reducciones y traslaciones) y las ensambla en la imagen de output [1].

La idea crucial es que esta máquina corre en retroalimentación (feedback loop); su propio output vuelve a entrar como input una y otra vez. Este experimento aparentemente banal resulta en imágenes asombrosas con el sistema de funciones iteradas adecuado.

Para entender mejor a las IFS exploremos el conjunto de funciones que al iterarse generan el triángulo de Sierpinsky. Para generar un conjunto de Sierpinsky necesitamos tres transformaciones afines que escalen (reducción) y reacomoden (traslación) el objeto input [1]. Dichas funciones son las siguientes:

$$w_1 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad (3.2)$$

$$w_2 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{2} \\ 0 \end{pmatrix} \quad (3.3)$$

$$w_3 = \begin{pmatrix} \frac{1}{2} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} \frac{1}{4} \\ \frac{1}{2} \end{pmatrix} \quad (3.4)$$

Podemos apreciar que 3.2, 3.3, 3.4, contienen la misma reducción que escala a un medio el objeto original, dicha reducción esta escrita en forma de matriz como transformación lineal. Además, cada trasformación incluye una suma de vector la cual nos traslada el objeto ya reducido. El aplicar las tres transformaciones en conjunto se conoce como operador de Hutchinson ecuación 3.5.

$$W(A) = w_1(A) \cup w_2(A) \cup w_3(A) \quad (3.5)$$

Si aplicamos el operador de Hutchinson a un objeto  $A$  obtenemos:

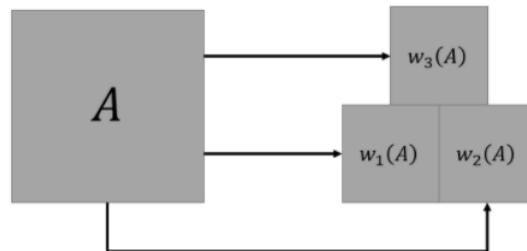


Figura 6: Operador de Hutchinson aplicado a  $A$ .

Como ya se mencionó la gracia de las IFS es iterar varias veces el sistema. Tal y como se muestra en la figura 7.

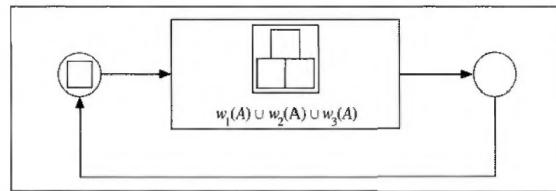


Figura 7: Escencia del IFS.

Iterando varias veces el operador de Hutchinson sobre  $A$  obtenemos:

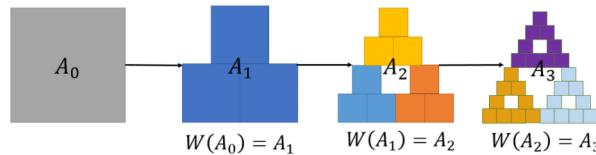


Figura 8: Tres iteraciones del IFS sobre A.

Donde el resultado de la siguiente iteración esta dado por:

$$A_{k+1} = W(A), k = 0, 1, 2, \dots, \quad (3.6)$$

Tras aplicar repetidamente el operador  $W$ , el IFS genera un grupo de imágenes que tiende a una Imagen final  $A_\infty$ , a la que se le da el nombre de atractor del sistema de IFS. Este atractor es invariante ante  $W$  lo que significa que:

$$W(A_\infty) = A_\infty \quad (3.7)$$

Tras aplicar varias veces  $W$  a  $A$  obtenemos el triángulo de Sierpinsky. Lo cual significa que el Sierpinsky es el atractor del sistema conformado por 3.2, 3.3, 3.4. Tal y como se muestra en la figura 9.

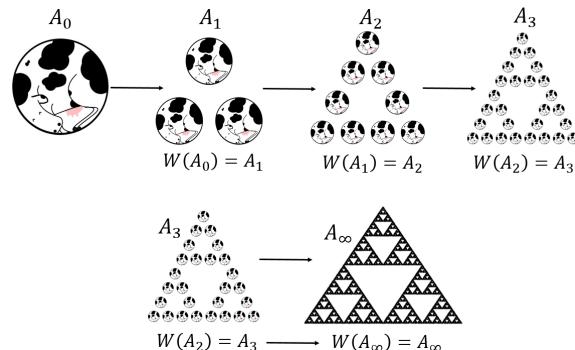


Figura 9: IFS aplicado sobre vaca esférica.

### 3.5. Juego del Caos

El Juego del Caos es un algoritmo, una receta que nos permite generar imágenes de fractales. Para jugar este juego es necesario un dado (con caras A, B, C), papel y lápiz, o una computadora también nos sirve bien [5]. Las reglas del juego son las siguientes:

1. Dibuja tres vértices de un triángulo en tu hoja de papel. A cada vértice se le asocia una letra de la A a la B. Para el primer vértice A, para el segundo B y para el tercero C.
2. Ahora dibuja un punto en cualquier lugar de la hoja, este será el punto inicial.
3. Lanza el dado. Los vértices tienen etiquetas, la letra que salga del dado se refiere a uno de los vértices. Marca un punto a medio camino entre el punto previo y el vértice del resultado.
4. Continúa lanzando el dado y en cada tiro marca un nuevo punto entre el punto previo y el vértice del resultado.

A este procedimiento se le conoce como Juego del Caos. La figura resultante de este juego se le conoce como atractor. En la figura 10 se muestra el atractor de este juego, en particular de tres vértices.

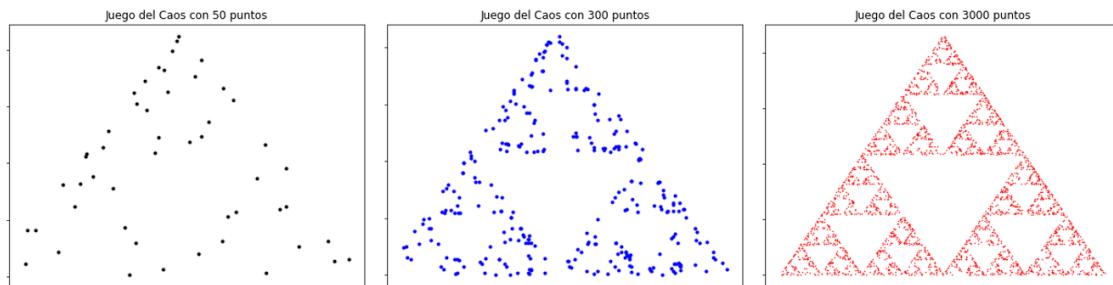


Figura 10: Juego del Caos con 50, 300 y 3000 puntos.

La razón por la que este juego caótico genera un patrón determinado (específicamente el triángulo de Sierpinsky) es porque dentro del método de punto medio en realidad estamos aplicando el sistema de funciones iteradas IFS, ya discutido en la sección anterior del marco teórico. A cada vértice corresponde una función del IFS. Para el primer vértice corresponde 3.2, para el segundo 3.3 y para el tercero 3.4. Para visualizar esta correlación entre el juego del caos y el IFS se muestra en la figura 11 un ejemplo de cálculo de un punto por ambos métodos.

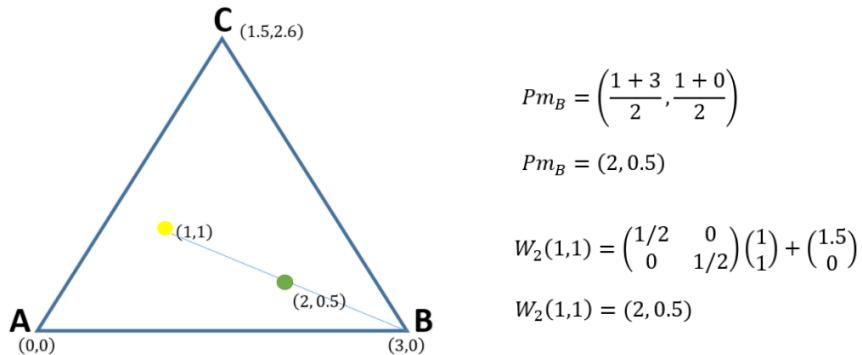


Figura 11: Cálculo por método de IFS y punto medio.

Para este ejemplo se asignaron valores numéricos a las coordenadas de los vértices. Se supuso un punto inicial en  $(1, 1)$  y un resultado  $B$  al lanzar el dado. En la parte derecha de la figura se muestra el cálculo por el método de punto medio y por la correspondiente función del IFS. Como se puede apreciar el resultado de ambos métodos es el mismo, porque en realidad siempre fue el mismo método.

## 3.6. Intervalo Unitario

A continuación, se revisan conceptos previamente desarrollados en mi proyecto ASE II “Multifractales geométricos en base cuaternaria”. Consiste en el estudio del intervalo unitario en base binaria y cuaternaria, mostrando su estructura multifractal fractal. Dejando así las bases para introducir el teorema de Eggleston y mediciones de espectros multifractales.

### 3.6.1. Números como Series Infinitas

Un número  $x$  dentro del intervalo unitario  $\mathbb{I}$  es una serie infinita, por lo que no se puede localizar puntualmente 3.8.

$$x = \sum_{n=1}^{\infty} \frac{z_n(x)}{s^n} \quad (3.8)$$

Dado esto, estudiar  $\mathbb{I}$  con sus números es una tarea complicada, por lo que tenemos qué resignarnos a estudiarlo con intervalos. Siendo estos generados al escoger la cantidad de cifras significativas del “número en cuestión”. Cortando así la serie.

### 3.6.2. Descomposición Intervalo Unitario

El intervalo unitario es el intervalo cerrado  $[0, 1] = \{x \in \mathbb{R} : 0 \leq x \leq 1\}$ , es decir, el conjunto de todos los números reales que son mayores o iguales que 0 y menores o iguales que 1.

En 1874 el matemático George Cantor demostró que el intervalo unitario no es numerable, es decir no existe una lista finita de todos los reales dentro del intervalo

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

(siempre habrá más). Esta característica se extiende a todo el conjunto de los reales ya que los elementos en los reales se pueden poner en correspondencia uno a uno con los elementos en el unitario (exceptuando el intervalo  $[0, 0]$  que tiene un solo valor, el cero). Esto emerge como la imposibilidad de numerar o contar los números en  $\mathbb{I}$ , de hecho cada elemento en  $\mathbb{I}$  se puede escribir como un número decimal con infinitos dígitos diferentes de cero.

Escribimos el intervalo unitario en base binaria con el propósito de describir la metodología necesaria para realizar una clasificación de los elementos de  $\mathbb{I}$  en frecuencias de aparición de sus dígitos. Se elige binario por su practicidad y por la facilidad de observar las particiones y la estructura interna, ya que solo cuenta con los dígitos  $b_i = 0, 1$ . Pero se pretende generalizar a cualquier base, en específico a la base cuaternaria debido a que se puede analizar el ADN como una cuerda abstracta de dígitos y a cada nucleótido se le asigna un dígito diferente de la base cuaternaria.

La primera partición del intervalo unitario consiste en dos subintervalos más pequeños 0.0 y 0.1 los cuales contienen a todos los números que comienzan con 0.0 y 0.1, como 0.0001 o 0.1000101 etc. Podemos seguir dividiendo al intervalo unitario infinitamente. A cada partición la llamaremos “iteración”. Podemos axiomatizar el proceso de partición, ya que podemos verlo como un operador que actúa sobre el sobre los dígitos agregando al primer elemento un “0” por la derecha y luego un “1”, para repetir el proceso con el segundo elemento.

Primerá iteración: 0 1

Segunda iteración: 00 01 10 11

Tercera iteración: 000 001 010 011 100 101 110 111

Y así sucesivamente, tal y como se muestra en la figura 12.

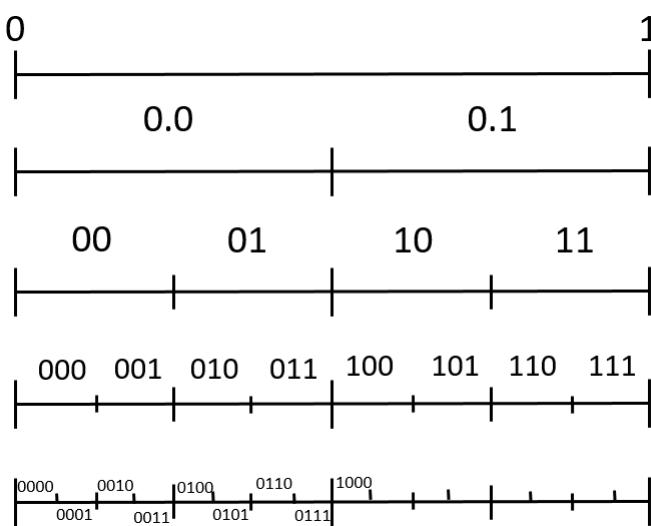


Figura 12: Partición del intervalo unitario en base binaria.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

Del mismo modo podemos descomponer  $\mathbb{I}$  en base cuaternaria, generando así subconjuntos. Como se muestra en la figura 13.

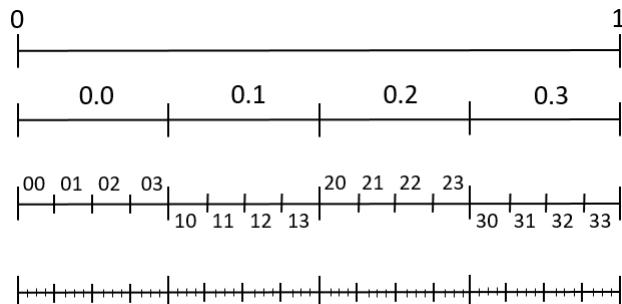


Figura 13: Partición del intervalo unitario en base cuaternaria.

### 3.6.3. Clasificación de Frecuencias

Para el estudio de  $\mathbb{I}$  en este proyecto nos enfocamos en estos subintervalos generados. Siendo estos clasificados por la frecuencia de aparición de dígitos. Es decir que para nuestros fines el intervalo “001”, “100” y “010” son equivalentes. Solo importa la frecuencia de aparición de los dígitos “0” y “1”, las cuales en este caso son  $f_0 = 2$  y  $f_1 = 1$ . Para apreciar la estructura multifractal a estos subintervalos equivalentes les hemos asignado un color, como se muestra en la figura 14.

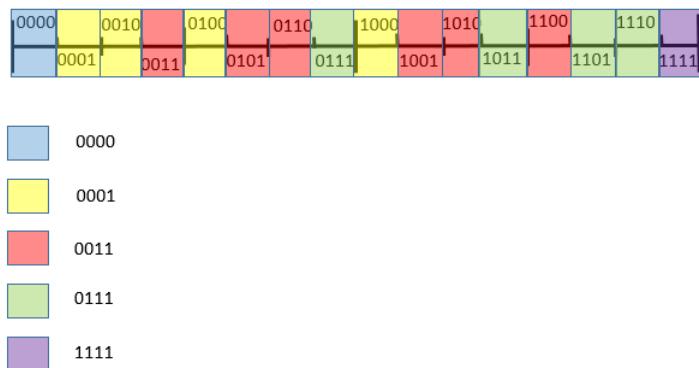


Figura 14: Visualización de subintervalos en la cuarta partición del intervalo  $I$  en base binaria.

A través de esta visualización somos capaces de apreciar la multifractalidad del intervalo  $\mathbb{I}$ . Este mismo proceso se lo aplicamos al intervalo  $\mathbb{I}$  en base cuaternaria, como se muestra en la figura 15.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

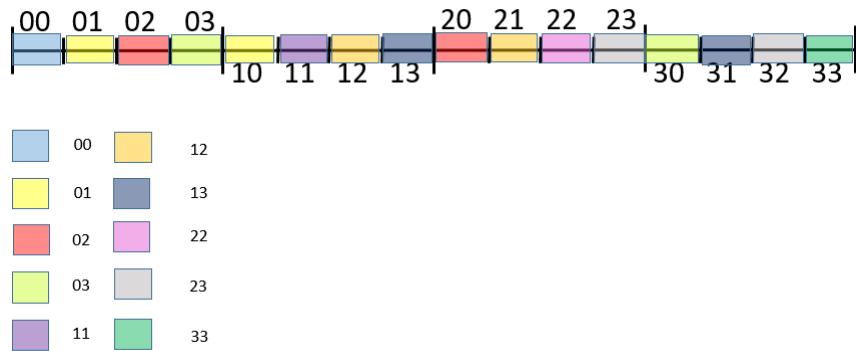


Figura 15: Visualización de subintervalos en la segunda partición del intervalo  $I$  en base cuaternaria.

Esta tarea de agregar números manualmente es poco práctica y para los propósitos de este proyecto se necesitan muchas iteraciones para apreciar el comportamiento multifractal del intervalo unitario. Por lo que se desarrollaron algoritmos en Python para automatizar esta tarea. Este algoritmo hace las particiones del intervalo, genera la clasificación de frecuencias, asigna un color a cada frecuencia y regresa una gráfica para su visualización, obteniendo así la figura 16.

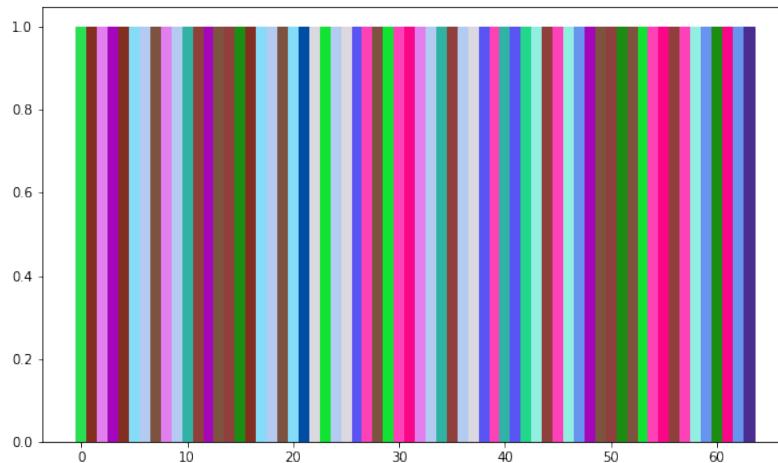


Figura 16: Graficación de subintervalos con colores dependiendo de su frecuencia de dígitos.

Podemos apreciar mejor la multifractalidad del intervalo refinando la definición, aumentando el número de iteraciones, el número de particiones y generando más colores para graficarlos. La figura 17 muestra la quinta iteración dónde se generan 262144 subintervalos, con 56 diferentes combinaciones y colores asociados.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

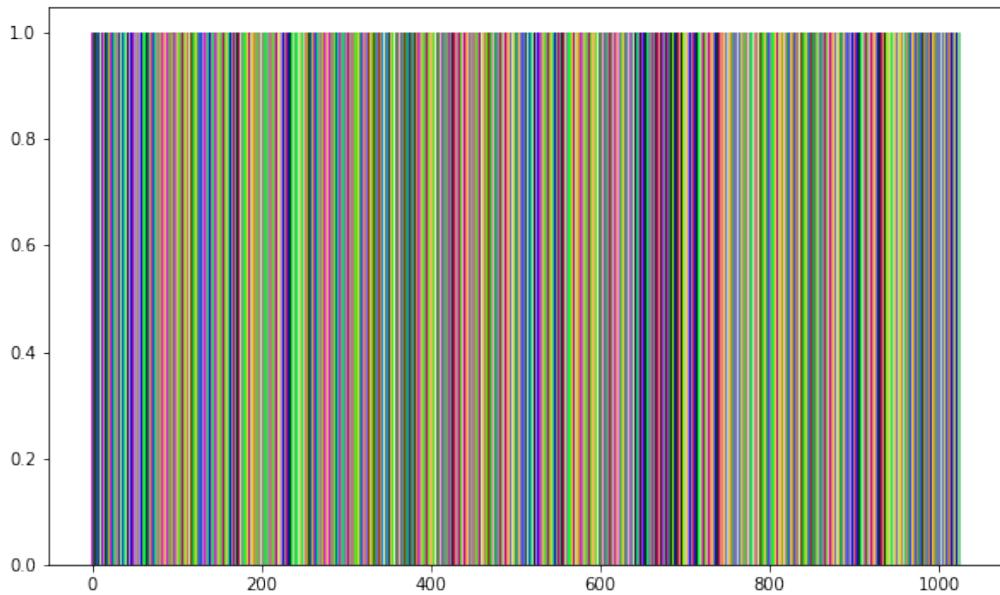


Figura 17: Quinta iteración del algoritmo en base cuaternaria.

#### 3.6.4. Autosimilaridad del Intervalo Unitario

Para evidenciar la estructura fractal del intervalo unitario debemos esclarecer la autosimilaridad en la generación de sus subconjuntos.

Analicemos  $\mathbb{I}$  en  $s=2$ .

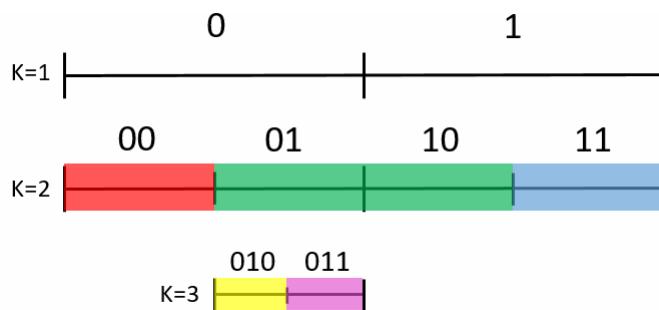


Figura 18: Iteraciones 1, 2 y 3 en base binaria.

En la figura 18 se aprecia como un subintervalo con un color dado se vuelve a partir y se obtienen nuevos colores, concretando esta idea con la figura 19.

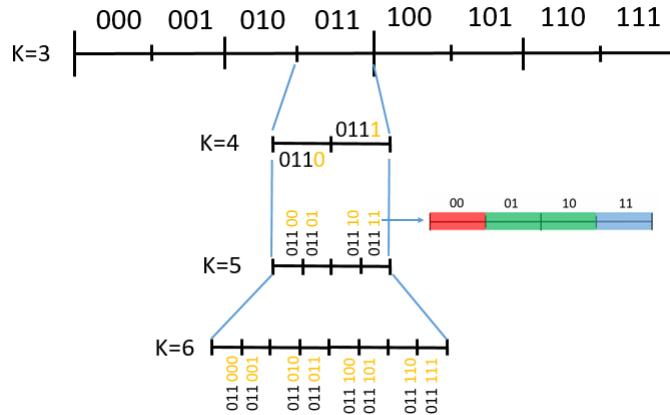


Figura 19: Iteraciones 3, 4, 5 y 6 en base binaria.

La característica que refleja la estructura fractal del intervalo unitario yace en las escalas; esto es, dada un partición como  $k=3$ , si se escoge un subconjunto cuyo color ya ha sido asignado y a ese subintervalo le “hacemos zoom” aumentando el número de iteraciones y cifras significativas dentro de él. Si agregamos dos cifras más obtenemos nuevamente el patrón de partición de  $\mathbb{I}$  en  $s=2$  y  $k=2$ . Y si agregamos dos cifras más, obtenemos análogamente la partición para  $s=2$   $k=3$  se reproduce en el zoom del subintervalo ‘011’ con dos cifras significativas más que estan pintadas en amarillo en la figura 19. Esto es una muestra de la estructura autosimilar de  $\mathbb{I}$ .

### 3.6.5. Teorema de Eggleston

Teorema de Eggleston establece una conexión entre la dimensión de Hausdorff y la entropía de frecuencias [4], la cual proviene del proceso límite cuando  $k \rightarrow \infty$ , es decir, cuando realizamos una mejor aproximación, el número de cifras significativas aumenta y las frecuencias toman valores en el continuo. En este proceso se calcula la entropía de Shanon de la siguiente forma:

$$S(\varphi_0, \varphi_1, \dots, \varphi_{s-1}) = - \sum_{j=0}^{s-1} \varphi_j \ln \varphi_j \quad (3.9)$$

Se puede obtener una expresión para la dimensión de Hausdorff (dimensión fractal) del conjunto  $M(\varphi_0, \varphi_1, \dots, \varphi_{s-1})$ , esto es llamado Teorema de Eggleston, el cual establece la dimensión de cada conjunto de frecuencias que conforman a todo el intervalo unitario.

$$\text{Dim}_H M(\varphi_0, \varphi_1, \dots, \varphi_{s-1}) = -\frac{1}{\ln s} \sum_{j=0}^{s-1} \varphi_j \ln \varphi_j \quad (3.10)$$

Midiendo la entropía de Shanon de cada subconjunto  $M$ , se puede encontrar una dimensión fractal diferente para cada subconjunto como se muestra en la figura 20 [2].

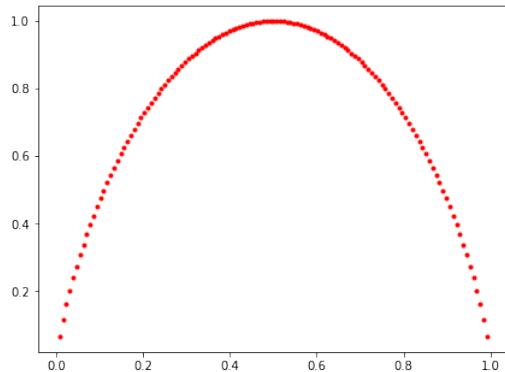


Figura 20: Dimensión de subconjuntos en sistema binario partición  $k = 7$ .

### 3.6.6. Multifractalidad de Intervalo Unitario

Midiendo la entropía de Shanon de cada subconjunto  $M$ , se puede encontrar una dimensión fractal diferente para cada subconjunto, de aquí viene la característica multifractal del intervalo unitario, ya que se observa que es un conjunto compuesto por subconjuntos de distinta dimensión fractal, todas menores o iguales que 1. Es decir para generar la figura 22 debemos combinar las estructuras fractales de cada subconjunto, mostrados en la figura 21.

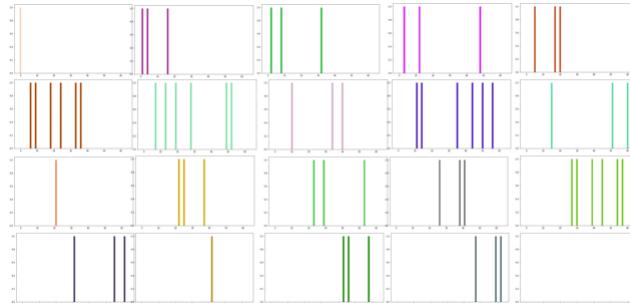


Figura 21: Estructuras fractales de cada subconjunto en  $k=3$  base cuaternaria

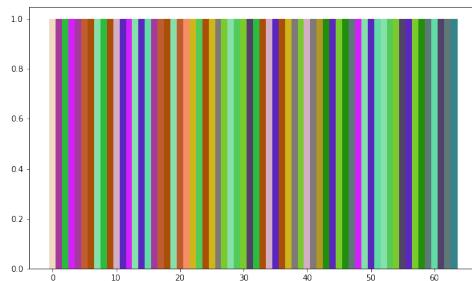


Figura 22: Composición multifractal de subconjuntos en  $k=3$  base cuaternaria.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

## 4. Desarrollo

En el marco teórico se expusieron diversos temas. Desde las secciones no codificantes, juego del caos, dimensiones, hasta el intervalo unitario. A primera vista pareciera ser una sopa de temas desconectados, así que empecemos a poner uso a estas herramientas ya discutidas con ayuda de Python.

### 4.1. ADN como Secuencia de Caracteres

Para el desarrollo de este proyecto solo se utilizó una computadora y conexión a internet. Nuestro estudio del ADN no requirió el sacrificio de ningún espécimen de planta, animal o bacteria. Así que ¿cómo hacemos un estudio multifractal del ADN? ¡Con experimentos computacionales!

Necesitamos concebir a la doble hélice como una secuencia abstracta de caracteres. Lo único que nos interesa del ADN es el orden en el que están acomodadas las bases. Esta concepción se ilustra en la figura 23.

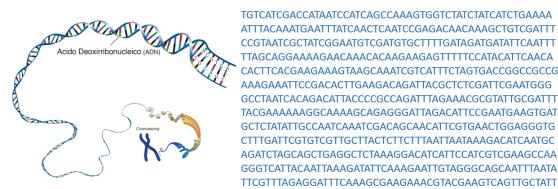


Figura 23: ADN como cadena abstracta de caracteres

Para ello se descargaron archivos de diversos organismos de bancos mundiales de información genética. En específico se ultizó bastante el “gene bank” del National Center for Biotechnology Information (NCBI).

Dichos archivos vienen en formato “.txt” como una secuencia de renglones con los pares de las bases del genoma. Se desarrolló un algoritmo para leer, limpiar y transformar el archivo en una “string” gigante con toda la secuencia, obteniendo así una cadena abstracta de caracteres.

```
with open('gen1.txt', 'r') as f:  
    file_data = f.readlines()  
  
file_data = file_data[1:]  
#print(file_data[:5])  
  
converted_list = []  
  
for element in file_data:  
    converted_list.append(element.strip())  
  
#print(converted_list[:5])  
  
file_string=''  
adn = file_string.join(converted_list)
```

Figura 24: Algoritmo que lee y convierte archivo .txt en string.

## 4.2. Juego del Caos extendio

En la revisión bibliográfica se expuso el Juego del Caos con tres vértices, ahora vamos a extender este juego a cuatro vértices. Dando lugar a un cuadrado en vez de un triángulo. Ahora necesitamos un dado que tenga cuatro posibles resultados, cada uno representando a un vértice nuevamente. El resto de las reglas siguen siendo las mismas. Se comienza dibujando un punto en cualquier sitio, se lanza el dado, se dibuja el siguiente punto entre el punto anterior y el vértice del resultado y así sucesivamente.

Como ya se demostró empírica y experimentalmente el resultado de este juego con tres vértices es el triángulo de Sierpinsky. Si jugamos este juego extendido con cuatro vértices se esperaría encontrar algún otro fractal o patrón interesante. Sin embargo, se genera un cuadrado lleno densamente, como se muestra en la figura 25.

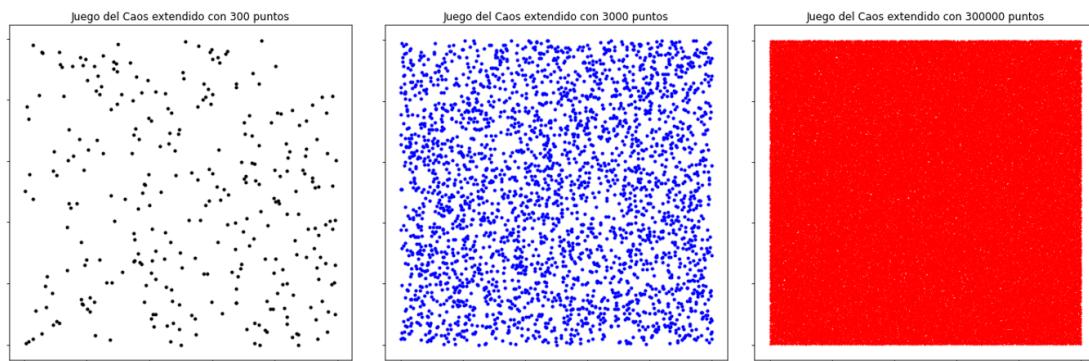


Figura 25: Juego del caos extendido con 300, 3000 y 300000 puntos.

La razón por la que el juego del caos extendido no genera ningún patrón específico es porque su IFS (figura 26) al aplicar las transformaciones recupera la misma forma que entró dentro del sistema, como se ilustra en la figura 27. Así que tras lanzar el dado aleatorio 300000 veces el cuadrado se llena.

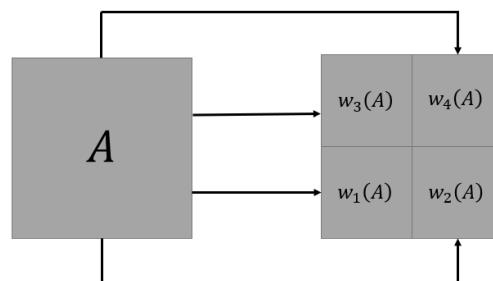


Figura 26: IFS para Juego del caos extendido.

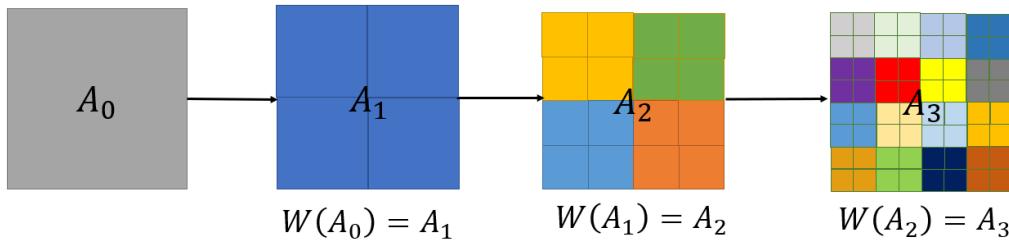


Figura 27: Iteraciones del IFS para Juego del caos extendido .

### 4.3. Juego del Caos en ADN

Ahora que tenemos al ADN como una secuencia abstracta de caracteres vamos a jugar al Juego del Caos extendido usando a esta secuencia como una lista de resultados de tiros. Para ello se asocia a cada vértice del cuadrado una letra de los pares de bases: adenina (A), citosina (C), guanina (G), y timina (T) [7].

Se desarrolló un algoritmo para escanear la “string” de ADN e interpretar a cada letra del genoma como un “resultado del dado”. Dicho algoritmo se muestra en la figura 31.

```

a = (0,0)
c = (100,0)
b = (100, 100)
d = (0, 100)

def adn_fractal3(strg_adn):
    #####
    posx = []
    posy = []
    pos=[0,0]

    for i in strg_adn:
        if i == 'A':
            pos[0] = (pos[0]+a[0])/2
            pos[1] = (pos[1]+a[1])/2
            posx.append(pos[0])
            posy.append(pos[1])

        elif i == 'G':
            pos[0] = (pos[0]+b[0])/2
            pos[1] = (pos[1]+b[1])/2
            posx.append(pos[0])
            posy.append(pos[1])

        elif i == 'T':
            pos[0] = (pos[0]+c[0])/2
            pos[1] = (pos[1]+c[1])/2
            posx.append(pos[0])
            posy.append(pos[1])

        elif i == 'C':
            pos[0] = (pos[0]+d[0])/2
            pos[1] = (pos[1]+d[1])/2
            posx.append(pos[0])
            posy.append(pos[1])
        else:
            continue
    return posx, posy

```

Figura 28: Algoritmo que aplica Juego del Caos sobre ADN.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

Los resultados de este pequeño experimento son visualmente satisfactorios y gratificantes. A continuación, se muestra el resultado de aplicar estos algoritmos a diferentes organismos.

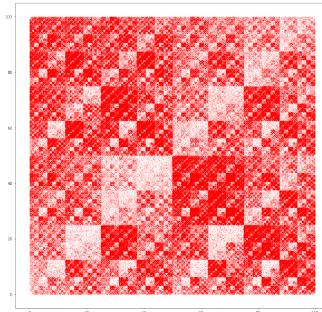


Figura 29: Algoritmo sobre *Halobacterium salinarum*.

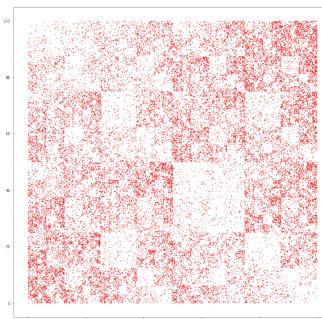


Figura 30: Algoritmo sobre SARS-CoV-2.



Figura 31: Algoritmo sobre Fungi.

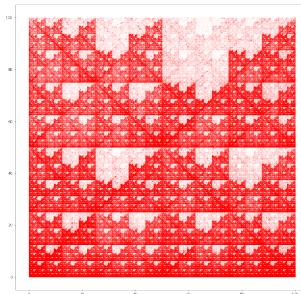


Figura 32: Algoritmo sobre Cromosoma Y Homo Sapiens.

Vemos que existe un atractor aparentemente fractal autosimilar. Esta propiedad se ejemplifica tomando la figura 31 y aplicándole el sistema IFS de la figura 26. Obteniendo así la figura 33.

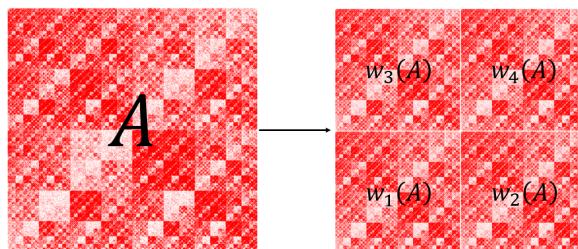


Figura 33: Aplicación del sistema IFS para Juego del caos extendido.

En la figura 33 se puede apreciar que al aplicar el IFS recuperamos el mismo atractor, quedando así más clara la autosimilaridad del conjunto. Sin embargo, esta autosimilaridad no es del todo perfecta. Esto queda mas claro aplicando el IFS sobre el atractor del Cromosoma Y (figura 32).

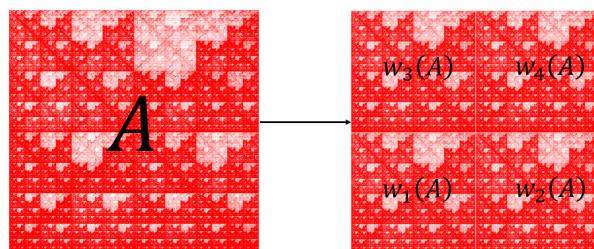


Figura 34: Aplicación del sistema IFS para Juego del caos extendido Cromosoma Y.

En la figura 34 se puede observar una gran autosimilaridad, pero imperfecta, ya no recuperamos exactamente el mismo fractal tras aplicar el IFS. En la esquina superior derecha del objeto  $A$  se puede apreciar un hueco blanco que no existe en el cuadrante correspondiente a  $w_4(A)$ .

Esta irregularidad en la autosimilaridad de estos fractales nos incita a aplicar la teoría multifractal para su estudio.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

## 4.4. Separación de Secuencias CDS y NCDS

El objetivo de este proyecto es estudiar la estructura multifractal del ADN, pero antes de aplicar la teoría multifractal debemos obtener las secuencias codificantes y no codificantes de los organismos.

En el “gene bank” del National Center for Biotechnology Information podemos encontrar archivos “.txt” con el genoma completo de organismos y archivos con los genes de dicho organismo (CDS), pero no hay archivos que nos den la NCDS. Por lo que se desarrolló un algoritmo que limpiara el archivo de genes, para obtener una lista de Python con un gen por cada elemento. Dicho algoritmo se muestra en la figura 35.

```
def genesis(prueba):

    indicadores = [] #Lista que me dice donde empieza cada gen
    for i in range(len(prueba)):
        if prueba[i][0] == '>':
            indicadores.append(i)
    #print('indicadores', indicadores)

    igenes= list(range(len(indicadores)))
    #creo una lista que me dirá donde recortar cada gen con ayuda de los indicadores
    for e in range(len(indicadores)):
        if e + 1 < len(indicadores):
            igenes[e] = [indicadores[e] + 1 , indicadores[e+1]]

    igenes = igenes[:-1] #quito el último elemento, porque le hace falta parametro
    #print('igenes',igenes)

    #creo una lista que recorta cada gen y los guarda en listas dentro de la lista
    genes = list(range(len(igenes)))
    for i in range(len(igenes)):
        genes[i] = prueba[igenes[i][0]: igenes[i][1]]
    #print('genes', genes)

    #remuevo los saltos de linea
    gen_l = genes[:]
    for i in range(len(genes)):
        for j in range(len(genes[i])):
            gen_l[i][j] = genes[i][j].replace("\n","");
    #print('gen_l',gen_l)

    #uno los caracteres de cada gen en uno solo
    gen_list = list(range(len(gen_l)))

    for i in range(len(gen_l)):
        gen_list[i] = "".join(gen_l[i])

    #print('gen_list', gen_list)

    #finalmente agrego el último gen manualmente
    s = prueba[indicadores[-1]+1:]
    strg = s[0]
    strg = strg.replace("\n","");
    #print('strg', s, strg)
    gen_list.append(strg)

    return gen_list
```

Figura 35: Algoritmo que genera lista de genes.

Una vez obtenida esta lista de genes, se desarrolló otro algoritmo que buscara estos genes dentro del genoma completo y los removiera, dejando así solo la sección no codificante. El algoritmo se muestra en la figura 36.

```
def remove_genes(genes, genoma):
    ''' remueve genes de un genoma completo,
    regresa el ADN no codificante'''

    for i in range(len(genes)):
        if genes[i] in genoma:
            genoma = genoma.replace(genes[i], "")
    return genoma
```

Figura 36: Algoritmo que regresa NCDS.

## 4.5. Análisis Multifractal del ADN

Ahora que tenemos nuestras CDS y NCDS separadas estamos listos para aplicar un análisis multifractal. Para ello se utilizará un símil al procedimiento del intervalo unitario.

En la sección 3.6.3 del marco teórico se introdujo la noción de clasificación de frecuencias, para clasificar a los subintervalos del Intervalo Unitario en subconjuntos. Dicha clasificación consistía en discriminarlos por la frecuencia de aparición de dígitos. Para el ADN se generarán “intervalos” del orden  $k = 6$ , siendo estos pequeñas subsecuencias de 6 caracteres. El primer “intervalo” o subsecuencia se construye tomando los primeros 6 caracteres de la cadena de ADN, el segundo toma 6 caracteres nuevamente, pero saltando un carácter al inicio de la cadena, el tercero salta dos lugares y toma 6 caracteres y así sucesivamente. Tal y como se ilustra en la figura 37.

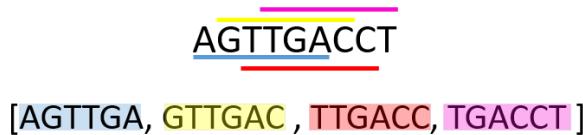


Figura 37: Ilustración de generación de subsecuencias.

Estas subsecuencias se guardan en una lista. El algoritmo que hace este trabajo se muestra en la figura 38.

```
def subsecuencias(string, k):
    ls = []
    for i in range(len(string)):
        ls.append(string[i:i+k])
    ls = ls[:-1]
    return ls
```

Figura 38: Algoritmo que genera subsecuencias.

Ahora que tenemos todas las subsecuencias el siguiente paso es contar el número de veces que aparece cada subsecuencia dentro de la cadena, en otras palabras,

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

obtener la frecuencia de aparición. Nuevamente se desarrolló el algoritmo requerido, se muestra en la figura 39.

```
def contador(adn, k):

    lista_secuencias = subsecuencias(adn, k) #genera lsita con todas las secuencias

    lista_sin_rep = [] #lista sin repeticiones de secuencia, se va llenando con el for
    for i in range(len(lista_secuencias)):
        if lista_secuencias[i] in lista_sin_rep:
            continue
        else:
            lista_sin_rep.append(lista_secuencias[i])

    lista_ordenada = sorted(lista_sin_rep)#lista sin rep ordenada en orden alfabetico

    cuenta = [] #lleva la cuenta de las apariciones de las subsecuencias
    for i in lista_ordenada:
        cuenta.append(lista_secuencias.count(i))
    return cuenta, lista_ordenada
```

Figura 39: Algoritmo que cuenta aparición de subsecuencias.

Si ordenamos a las subsecuencias en orden alfabético y graficamos la frecuencia con la que aparecen, obtenemos la figura 40. Se utilizó el genoma de la Arquea Salinarium. A este resultado también se le conoce como vector de frecuencias [3].

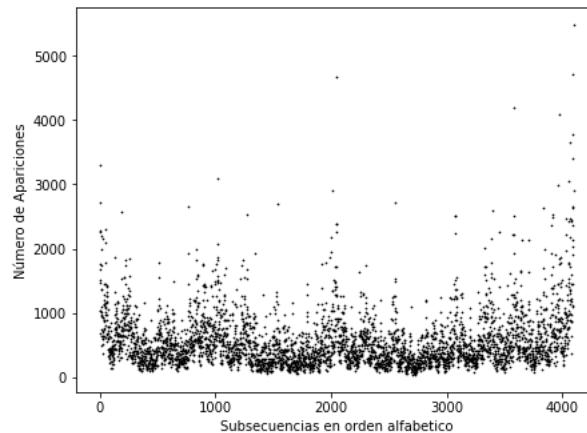


Figura 40: Vector de frecuencias orden  $k = 6$ .

Si ordenamos estas en un histograma se genera la figura 41. Se puede apreciar que la mayoría se encuentran en un rango entre 0 y 1000 apariciones dentro del genoma.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

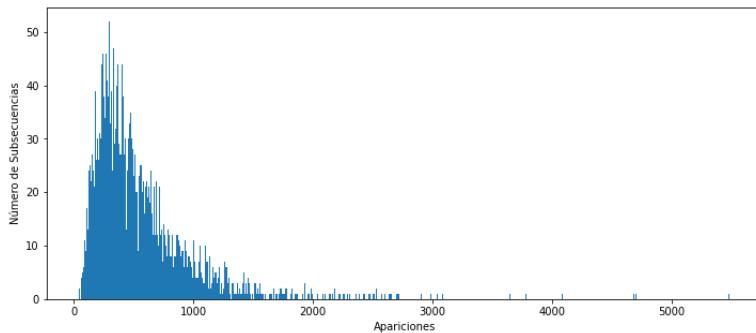


Figura 41: Histograma de frecuencias de aparición de subsecuencias.

Finalmente, con el vector de frecuencias se utilizó un algoritmo para medir los espectros multifractales de las secuencias CDS y NCDS. Este algoritmo no se desarrolló en el proyecto, sino que se recurrió a él como paquetería. Obteniendo así los espectros multifractales de dos bacterias y dos arqueas.

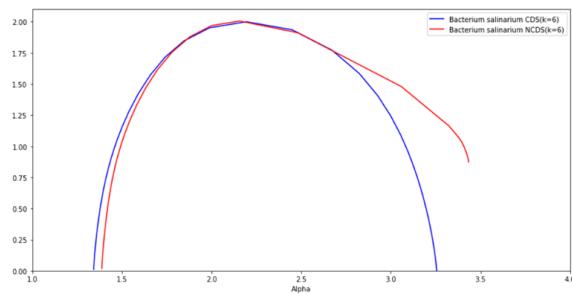


Figura 42: Espectros multifractales de CDS y NCDS Arquea *Halobacterium Salinarium*.

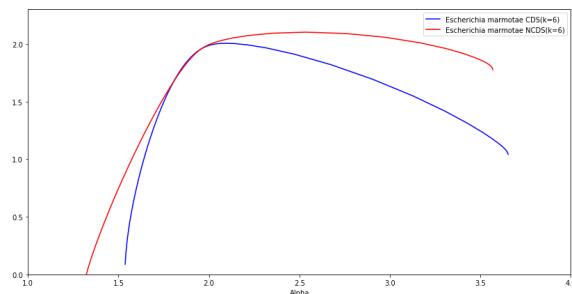


Figura 43: Espectros multifractales de CDS y NCDS Arquea *Escherichia marmotae*.

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

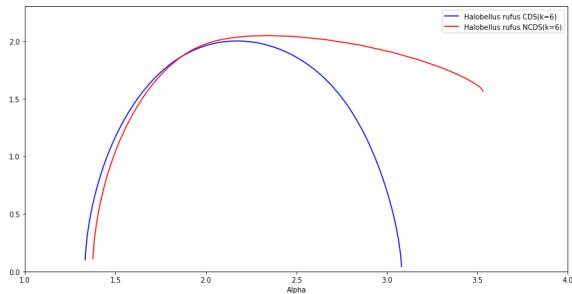


Figura 44: Espectros multifractales de CDS y NCDS Arquea Halobellus rufus.

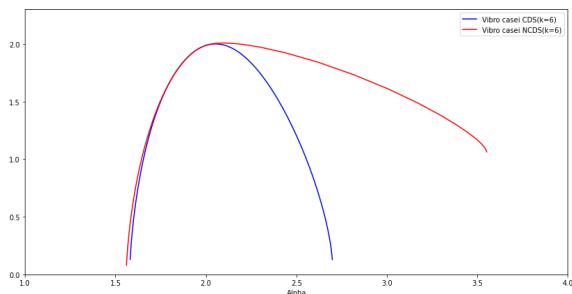


Figura 45: Espectros multifractales de CDS y NCDS Bacteria Vibrio casei.

El espectro multifractal vive en el espacio  $D(\alpha)$  (eje Y) y  $\alpha$  (eje X), donde fácilmente se puede calcular  $\alpha_{min}$  y  $\alpha_{max}$ . Podemos definir la anchura del espectro como  $W$ :

$$W = \alpha_{max} - \alpha_{min} \quad (4.11)$$

Tal como se ilustra en la figura 46.

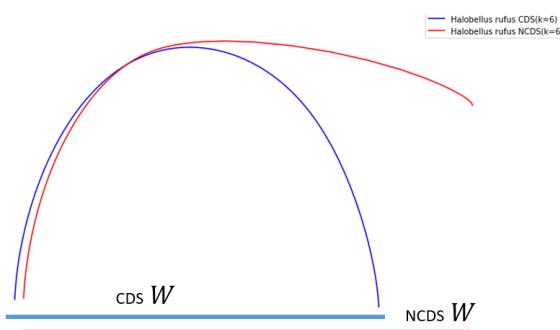


Figura 46: Ilustración Anchura  $W$ .

## 5. Conclusiones

Jugando al Juego del Caos en ADN pudimos observar como se generaban atractores para diferentes microorganismos, lo cual nos permite llegar a la conclusión (un tanto trivial) de que el ADN no es una secuencia aleatoria, de lo contrario abríamos obtenido cuadrados densamente llenos (figura 47). Además, no solo vimos que no eran aleatorios, sino que presentaban una estructura fractal estadísticamente autosimilar. Por lo que se procedió a aplicar la teoría multifractal.

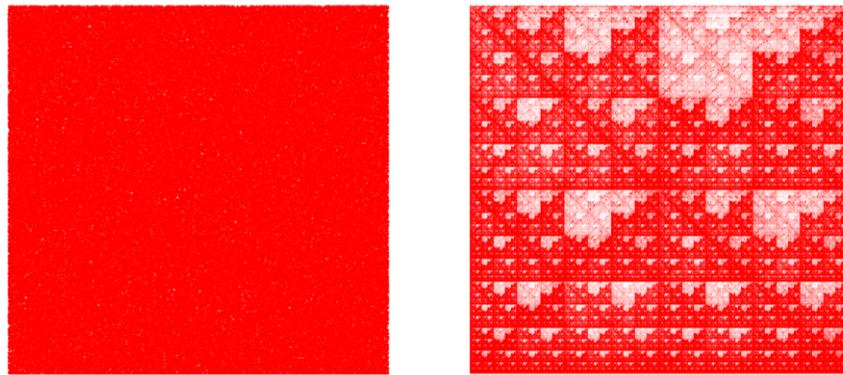


Figura 47: Juego del Caos con secuencia aleatoria y secuencia de ADN.

Aplicando la teoría multifractal fuimos capaces de producir espectros multifractales de las secuencias CDS y NCDS de distintos organismos. Los cuales se muestran apilados en la figura 48.

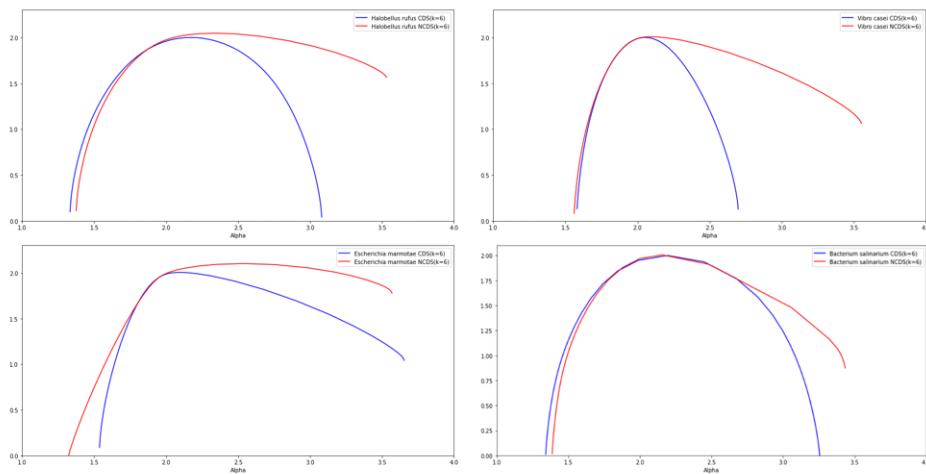


Figura 48: Espectros multifractales CDS y NCDS.

Podemos concluir que ambas secuencias CDS y NCDS son estructuras multifractales, pero presentan espectros diferentes. Si comparamos la  $W$  de los espectros, encontramos que las NCDS presenta una  $W$  más grande que la anchura de la CDS,

# Análisis Multifractal de Secuencias CDS y NCDS en ADN

ASE III - Otoño 2020

Alejandro A. Muñoz G.

como se puede apreciar a simple vista en la figura 48 (recordando que la curva roja corresponde a la NCDS y la azul a la CDS). Lo cual nos indica que la sección no codificante presenta una estructura multifractal más compleja [9], tal y como se muestra en la siguiente tabla.

Medición de $W$ de espectros		
Organismo	$W_{CDS}$	$W_{NCDS}$
Vibrio casei	1.116698	1.990883
Halobellus rufus	1.747979	2.154722
Escherichia marmotae	2.116088	2.246019
Halobacterium	1.91272	2.045809
Salinarium		

Lo relevante de estos resultados es que de una misma cadena de ADN se obtuvieron dos subsecuencias (CDS, NCDS) y estas presentaron grados de complejidad distintos. Lo cual no es trivial o fácil de ver sin el análisis realizado en este trabajo.

## 5.1. Trabajo a Futuro

Aún quedan áreas abiertas de investigación. Como realizar más mediciones e incluir otros parámetros de la teoría multifractal que sirvan para comparar la complejidad de secuencias CDS y NCDS. Quizá con un análisis más riguroso de los resultados se podría dar una interpretación física biológica de las distintas regiones no codificantes.

## 5.2. Conclusión

En conclusión, en este trabajo estudiamos al ADN desde un acercamiento multifractal, primero demostrando que la estructura del ADN presentaba este comportamiento y después midiendo espectros de las secciones codificantes y no codificantes. Lo cual nos permitió ver una cara más de la complejidad de la vida.

## Bibliografía

- [1] M.F. Barnsley. *Fractals Everywhere: New Edition*. Dover Books on Mathematics. Dover Publications, 2013.
- [2] JL del Río-Correa and J López-García. Shannon entropy and hausdorff dimension in multifractals. *Revista Mexicana de Física*, 58(1):13–20, 2012.
- [3] G Durán-Meza, J López-García, and JL del Río-Correa. The self-similarity properties and multifractal analysis of dna sequences. *Applied Mathematics and Nonlinear Sciences*, 4(1):261–272, 2019.
- [4] HG Eggleston. The fractional dimension of a set defined by decimal properties. *The Quarterly Journal of Mathematics*, (1):31–36, 1949.
- [5] H Joel Jeffrey. Chaos game visualization of sequences. *Computers & Graphics*, 16(1):25–33, 1992.
- [6] Benoit B Mandelbrot. The fractal geometry of nature/revised and enlarged edition. *whf*, 1983.
- [7] Heinz-Otto Peitgen, Hartmut Jürgens, and Dietmar Saupe. *Chaos and fractals: new frontiers of science*. Springer Science & Business Media, 2006.
- [8] Svetlana A Shabalina and Nikolay A Spiridonov. The mammalian transcriptome and the function of non-coding dna sequences. *Genome biology*, 5(4):105, 2004.
- [9] Longfeng Zhao, Wei Li, Chunbin Yang, Jihui Han, Zhu Su, and Yijiang Zou. Multifractality and network analysis of phase transition. *PloS one*, 12(1):e0170467, 2017.
- [10] Deyou Zheng, Adam Frankish, Robert Baertsch, Philipp Kapranov, Alexandre Reymond, Siew Woh Choo, Yontao Lu, France Denoeud, Stylianos E Antonarakis, Michael Snyder, et al. Pseudogenes in the encode regions: consensus annotation, analysis of transcription, and evolution. *Genome research*, 17(6):839–851, 2007.