



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

---

FACULTAD DE CIENCIAS

ANÁLISIS NUMÉRICO E IMPLEMENTACIÓN  
COMPUTACIONAL DE LA DIMENSIÓN FRACTAL EN  
SISTEMAS DINÁMICOS EN BIOLOGÍA

REPORTE DE SERVICIO SOCIAL

QUE PARA OBTENER EL TÍTULO DE:

FÍSICO

P R E S E N T A:

JOSÉ ROBERTO BASTIDA LUNA

TUTOR:

DR. PEDRO EDUARDO MIRAMONTES VIDAL  
DR.



CIUDAD DE MÉXICO, 2021

# Capítulo 1

## Sistemas dinámicos discretos y fractales

En general, las ecuaciones diferenciales discretas se pueden representar como:

$$x_{n+1} = f(\vec{\alpha}, x_n) \quad (1.1)$$

Donde  $\vec{\alpha}$  es algún vector de parámetros y  $x_n$  es el estado previo del sistema. Con esto se indica que cada estado del sistema depende del estado inmediato anterior en el que el sistema se encontró.

Dada una ecuación diferencial discreta como la que se muestra en 1.1 y un punto inicial  $x_0$  dentro del dominio de la ecuación, se compone la ecuación consigo misma al rededor del punto inicial  $x_0$  un número lo suficientemente grande de veces para que arroje información sobre la evolución del sistema:

$$\begin{aligned} & x_0 \\ & x_1 = f(\vec{\alpha}, x_0) \\ & x_2 = f \circ f(\vec{\alpha}, x_0) \\ & \dots \\ & x_n = \underbrace{f \circ \dots \circ f}_{n \text{ veces}}(\vec{\alpha}, x_0) \end{aligned} \quad (1.2)$$

A partir de estas iteraciones, se obtiene una sucesión de puntos  $x_0, x_0, \dots, x_n$  cuyo análisis permite conocer la dinámica del sistema. La relevancia que esto tiene es que permite modelar los escenarios en que el sistema evoluciona y comprender la dependencia que ésta evolución tiene con los parámetros  $\vec{\alpha}$  y con el punto inicial  $x_0$ .

Un ejemplo de una ecuación diferencial discreta es la ecuación logística. Ésta modela la evolución en el número de individuos dentro de una población:

$$x_{n+1} = k x_n \left(1 - \frac{x_n}{M}\right)$$

Con  $k$  y  $M$  ambas positivas. Si consideramos  $x_n$  como la fracción de población máxima entonces ésta toma valores entre  $1 \leq x_n \leq 1$  y la ecuación se convierte en:

$$x_{n+1} = \lambda x_n (1 - x_n) \quad (1.3)$$

Con  $\lambda > 0$ . De modo que depende únicamente de un parámetro  $\lambda$  y de la población inicial.

### Análisis gráfico y cualitativo de sistemas

Para el análisis de la evolución de los sistemas, comúnmente se usa alguna de las herramientas: series de tiempo, diagrama fase o diagrama de bifurcación. Para los fines del presente informe basta con mencionar el diagrama fase aunque más detalle sobre la serie de tiempo y los diagramas de bifurcación puede encontrarse.

#### Diagramas fase

Ilustra los *saltos* que da el sistema entre estados consecutivos. Para esto se toma la sucesión de puntos que resulta de aplicar la ecuación 1.2,  $\mathcal{S} = \{x_0, x_1, \dots, x_n\}$  y se grafican las coordenadas  $(x_0, x_1), (x_1, x_2), \dots, (x_{n-1}, x_n)$ , es decir, en el eje de las  $x$  se grafica el punto del cual partió el sistema y en el eje de las  $y$  se grafica el punto al que *llegó* después de que se le aplicara la función  $f$ .

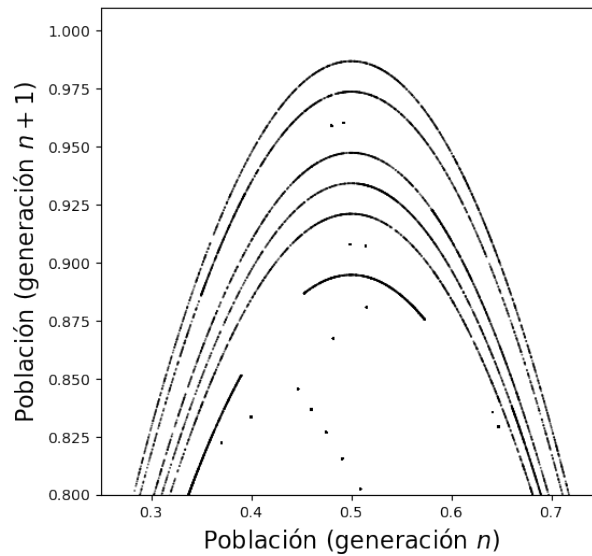


Figura 1.1: Diagrama fase que nos permite visualizar si el sistema se estabiliza. Básicamente nos dice de dónde proviene y a dónde llega el estado del sistema.

En la figura 1.1, cada arco representa la evolución de la ecuación logística 1.3 a un valor de  $\lambda$ . Es interesante notar que los arcos nunca se cruzan. Cada uno tiene una evolución diferente a las demás. Los pequeños arcos cerca del centro de la figura con sólo unos cuantos puntos (o incluso únicamente uno) representan los casos donde el sistema converge rápidamente a un punto o un ciclo límite.

## Juego del caos

El juego del caos, similar a los sistemas dinámicos discretos, consiste en un conjunto de condiciones iniciales y una función o acción a repetirse.

Uno de los más emblemáticos es el *Triángulo de Sierpinski* donde, sobre un triángulo equilátero con vértices  $V = \{v_1 = (x_1, y_1), v_2 = (x_2, y_2), v_3 = (x_3, y_3)\}$  y etiquetados por  $V_e = \{1, 2, 3\}$ , se aplica el siguiente algoritmo:

1. Se elige aleatoriamente un número  $\kappa \in V_e$  y se genera con ello el punto inicial ( $p_0 = v_\kappa$ ) que esté entre los vértices de  $V$ .
2. Se elige aleatoriamente una nueva etiqueta  $\kappa_1 \in V_e$  y se calcula el punto medio entre  $p_0$  y  $v_{\kappa_1}$ . Este nuevo punto es  $p_1$ , el correspondiente a la iteración 1. Más sucintamente,  $p_1 = PM(p_0, v_{\kappa_1})$ .
3. Se itera el paso anterior con una sutil pero importante diferencia: no se toma nuevamente  $p_0$  para calcular el punto medio sino el punto generado durante la previa iteración. Es decir,  $p_{n+1} = PM(p_n, v_{\kappa_{n+1}})$  para  $n \geq 1$ .

Se muestra en la figura 1.2a la gráfica de los puntos resultantes de cinco iteraciones; se puede inferir cuál fue la secuencia de etiquetas que los generó:  $\{1, 3, 1, 2, 3, 2\}$ . La gráfica en 1.2b se obtuvo con un millón de iteraciones y se puede decir que ningún punto cayó dentro del triángulo blanco invertido que está en el centro de la figura y que es semejante al original. Más aún, en 1.2c se cambiaron los colores de tres triángulos más pequeños para evidenciar que éstos, a su vez, tienen triángulos blancos invertidos todos ellos semejantes.

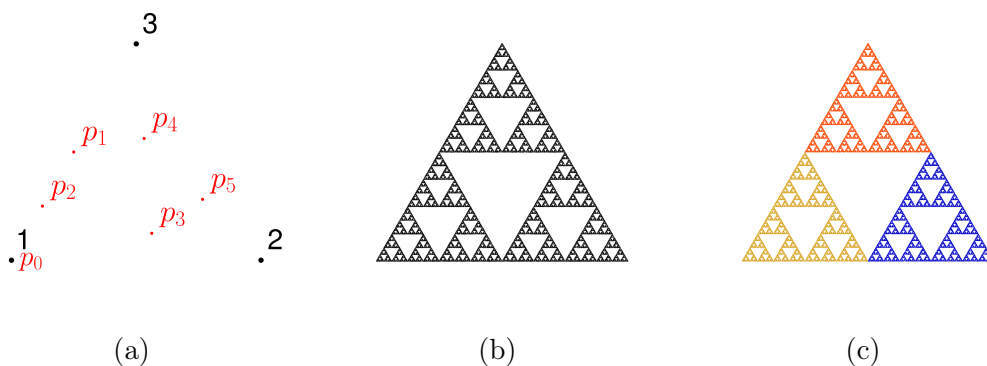


Figura 1.2: (a): Cinco puntos generados con el algoritmo del triángulo de Sierpinski. (b): Gráfica de los puntos después de un millón iteraciones. (c): Los tres triángulos pequeños reproducen la forma del original en (b), la gráfica es auto-semejante.

## Juego del Caos en el DNA

El DNA yace dentro de cada célula de cada ser vivo. Tiene la información suficiente para que la célula cumpla con las todas funciones necesarias para sustentar la vida. Es una

macromolécula que forma una espiral y en cuyo interior hay únicamente cuatro posibles moléculas, llamadas nucleótidos: Adenina, Guanina, Citosina, Timina. De ahora en adelante abreviaremos estos elementos con el diccionario: {Adenina: A, Guanina: G, Citosina: C, Timina: T}.

Para fines prácticos, consideraremos que el DNA es únicamente una larga cadena de caracteres donde a cada posición la ocupa alguno de los elementos del diccionario de arriba.

El juego del caos en el DNA consiste en:

- A cada vértice de un cuadrado se le etiqueta con uno de los elementos del diccionario  $A, T, C, G$  de modo que a cada vértice le corresponda únicamente un elemento del diccionario.
- Se identifica la letra que ocupa la primera posición dentro de la secuencia y se coloca un punto sobre el vértice que la representa en el cuadrado del paso anterior. A este punto inicial lo llamamos  $x_0$ .
- Se identifica la letra que le sucede a la del punto anterior y se coloca un nuevo punto ( $x_1$ ) en el punto medio entre  $x_0$  y el vértice que representa dicha letra.
- Se itera la acción anterior. Se fija qué letra ocupa la posición  $n$  de la cadena y se traza el punto medio entre el punto  $x_{n-1}$  y el vértice que representa a la letra en la posición  $n$ .

Tomemos, a modo de ejemplo, la siguiente sub-cadena del DNA:

**ATGCCGATA**

En la figura 1.3 se ilustra cómo luce la gráfica del algoritmo para las primera 9 posiciones de la cadena.

## Fractales

Los fractales son conjuntos cuya principal característica es que tienen una *dimensión fraccionaria*. Otra manera de decir esto requiere que pensemos en las dimensiones como el número de veces que un objeto se replica a sí mismo cuando se aumenta o se reduce uno de sus lados como una potencia de 2.

La dimensión se calcula con la expresión:  $2^k = M$  donde  $k$  es la dimensión y  $M$  es el número de copias resultantes de duplicar el tamaño de un objeto. Acomodando un poco la expresión anterior llegamos a que  $k = \ln(M)/\ln(2)$ .

La figura 1.4 ilustra cómo esto se relaciona con la dimensión topológica. En 1.4a, duplicar el tamaño de un segmento de recta resulta en dos copias de la original, entonces  $k = \ln(2)/\ln(2) = 1$ . Cuando se lleva esta idea a un cuadrado como en 1.4b, aumentar el tamaño del cuadrado en un factor de 2 produce ahora cuatro copias de la figura inicial:  $k = \ln(4)/\ln(2) = 2$ . Finalmente en 1.4c se muestra únicamente la silueta de los tres

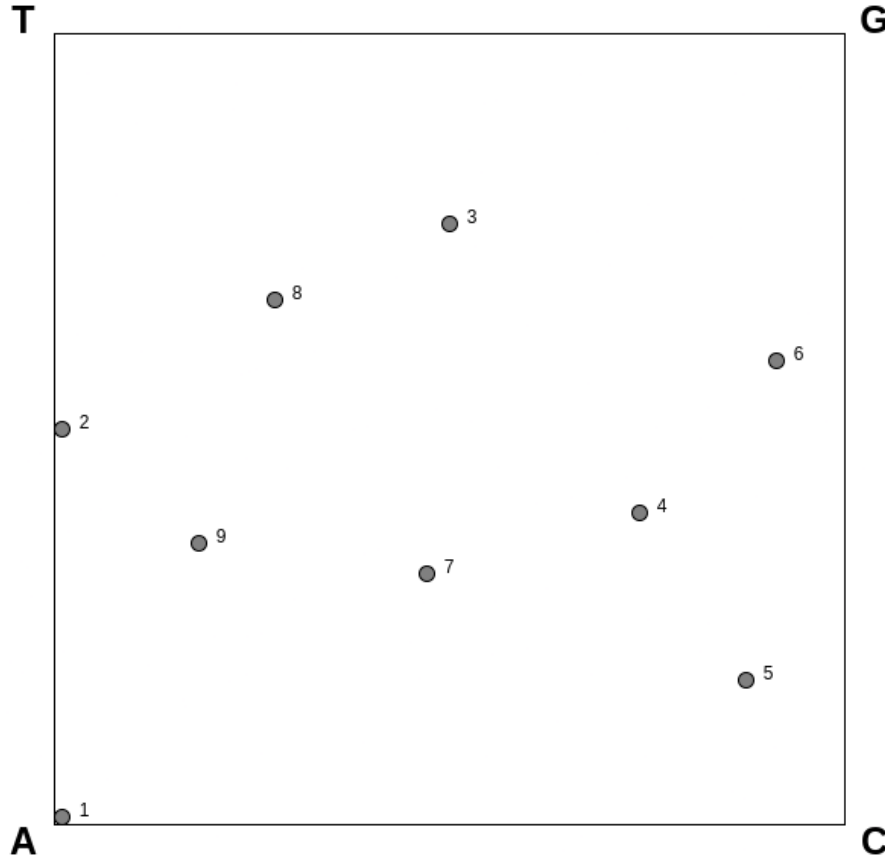


Figura 1.3: Algoritmo del juego del caos en secuencias de genomas. Se muestran los primeros nueve puntos del

triángulos de 1.2c. Duplicar las dimensiones del triángulo resulta en tres copias de éste, su dimensión es entonces  $k = \ln(3)/\ln(2) \approx 1.584$ .

Una de las implicaciones que esto tiene es que un fractal no necesariamente *llena* el espacio donde se encuentra cuando se hacen copias de él, dejando espacios vacíos. La frontera del fractal suele describir vericuetos donde se puede verificar la autosemejanza, es decir, que el objeto se replica a sí mismo independientemente de la escala a la que se le observe.

Resulta sencillo calcular la dimensión fractal de conjuntos con evidente autosemejanza aunque esta cualidad no siempre aparezca durante el estudio de sistemas dinámicos. En la siguiente sección se introduce la dimensión de Minkowski-Bouligand, un método numérico para el cálculo de la dimensión que aproxima la dimensión de fractales.

### Dimensión de Minkowski-Bouligand

Si bien la manera de calcular la dimensión fractal con la definición y el conteo del número de réplicas que se genera de la figura al duplicar el tamaño de uno de los lados; tiene limitaciones cuando la autosemejanza es no trivial o incluso inexistente.

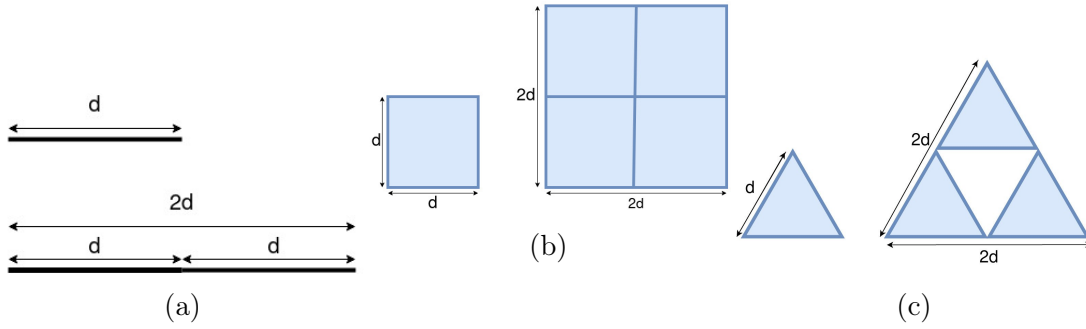


Figura 1.4: El cálculo de la dimensión fractal coincide con la topológica para una y dos dimensiones. El número de copias crece como potencia de 2 al duplicar la longitud de un objeto. Cuando el conjunto es auto-semejante y no llena el espacio; la dimensión fractal no es un número entero.

Para calcular la dimensión fractal de conjuntos donde no sea posible contar el número de réplicas, se usa el algoritmo de *Minkowski-Bouligand* también conocido como el algoritmo de *conteo de cajas*. También se basa en la expresión  $2^k = M$  y busca particionar el espacio métrico dentro del cual se encuentra el conjunto en segmentos cuyo tamaño decrezca como potencia de  $\sim \frac{1}{2^n}$  y donde  $n$  es un número natural sobre el cual se itera.

A medida que se particione el espacio métrico, se cuenta el número de cajas dentro de las cuales haya al menos un elemento del conjunto. La dimensión se calcula con la expresión:

$$\dim_{box}(\mathcal{S}) = \lim_{\epsilon \rightarrow 0} \frac{\log(N(\epsilon))}{\log(1/\epsilon)} \quad (1.4)$$

Donde  $\epsilon$  es la longitud de cada elemento de la partición y  $N(\epsilon)$  es el número de cajas que contienen al menos un elemento del conjunto  $R$ .

En la figura 1.5 se muestran las primeras cuatro iteraciones del algoritmo del conteo de cajas aplicado sobre el triángulo de Sierpinski. Se aprecia cómo después de unas cuantas iteraciones, hay cuadrados dentro del triángulo invertido del centro donde no hay elementos del conjunto.

La manera de implementar la ecuación 1.4 para el cálculo de la dimensión consiste en iterar para diferentes valores escalas de  $\epsilon$  y registrar  $N(\epsilon)$ . Luego se traza una recta con los valores de  $\log(1/\epsilon)$  sobre el eje de las  $x$  y  $\log(N(\epsilon))$  sobre el eje de las  $y$ .

Esto resulta en una recta cuya pendiente aproximará la dimensión fractal que deseamos calcular. De modo que se hace una regresión lineal con los puntos que resulten de la iteración para diferentes escalas de  $\epsilon$ .

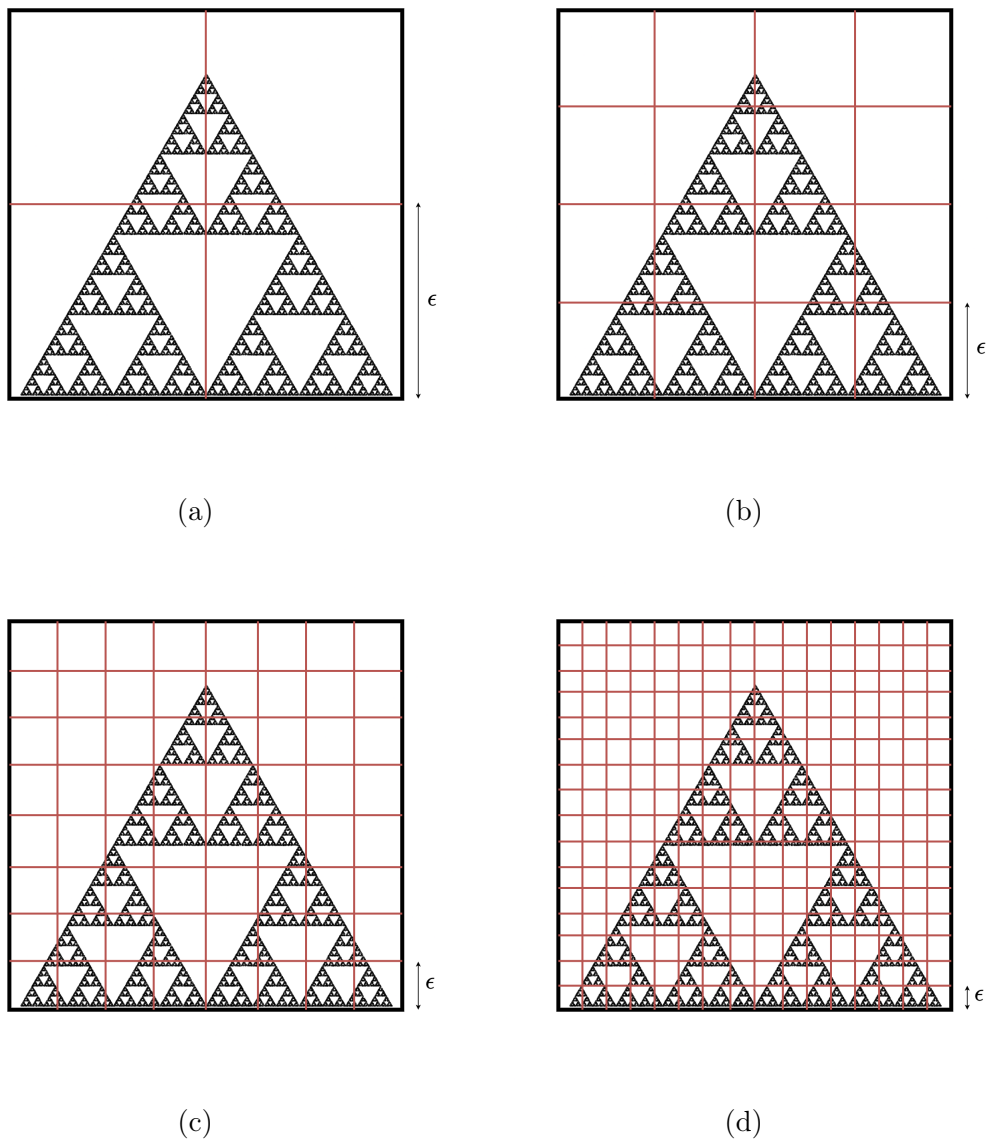


Figura 1.5: Particiones del espacio que contiene al conjunto. El número de particiones dentro de  $\mathbf{R}^2$  incrementa como función de  $2^n$ . A cada iteración se cuenta el número de *cuadrados* que tengan al menos un elemento del conjunto  $\mathcal{S}$  en su interior.



# Capítulo 2

## Implementación

Se desarrolló un proyecto en el lenguaje de programación python 3.x que calcula la dimensión de Minkowski-Bouligand de cualquier subconjunto de  $\mathbb{R}^2$  dadas sus coordenadas. Se probó el algoritmo con los siguientes conjuntos:

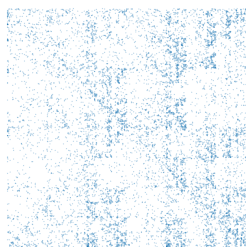
- Una recta.
- Un cuadrado.
- Triángulo de Sierpinski.
- Código de barras.
- Juego del caos en una subcadena del cromosoma 19.
- Juego del caos en una cadena aleatoria con el mismo diccionario del DNA.

Puede modificarse el número de elementos y extenderse para cualquier la dimensión de cualquier otro conjunto mediante la ejecución del método `Frac_dimension_computation()` incluido en la documentación del proyecto, mismo que puede encontrarse *dando click aquí*.

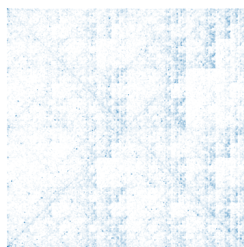
Se usó este desarrollo para reproducir el juego del caos en subcadenas del cromosoma 19 de longitud variable. También se generaron cadenas aleatorias a partir del diccionario de nucleótidos y se reprodujo el juego del caos en éstas también.

A cada uno de los conjuntos antes mencionados se le calculó la dimensión de Minkowski-Bouligand con el fin de indagar en la convergencia de la dimensión entre conjuntos aleatorios y conjuntos con cierto arreglo a partir del DNA real.

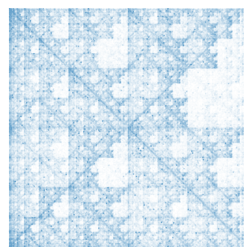
Las longitudes de la subcadena del cromosoma 19 que se usaron para el experimento fueron:  $[10^4, 10^5, 10^6, 10^7, 10^8]$  y las longitudes de la cadena generada aleatoriamente fueron:  $[10^4, 10^5, 10^6, 10^7]$ . Esto se hizo así porque se observó que la convergencia en la dimensión para el caso de la cadena aleatoria siempre ocurría entre las potencias  $10^5$  y  $10^6$ . Además que el equipo de cómputo donde se desarrolló el proyecto tenía dificultades al generar la cadena de caracteres de longitud  $10^8$ .



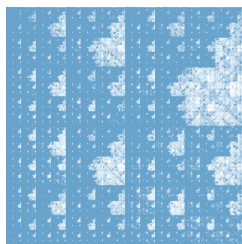
(a)



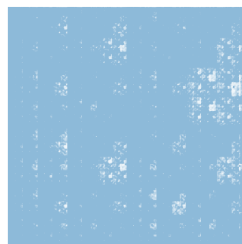
(b)



(c)

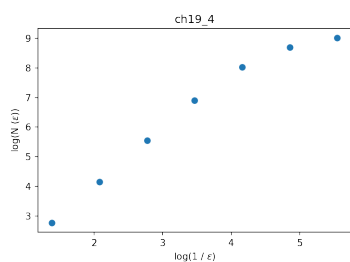


(d)

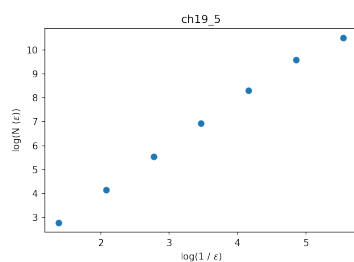


(e)

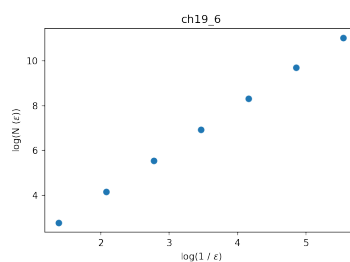
Figura 2.1: Gráficas del juego del caos aplicado a sub-cadenas del cromosoma 19.



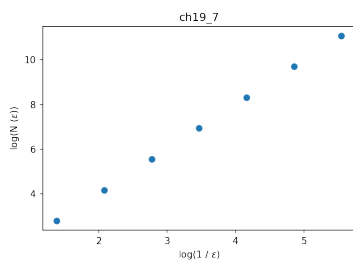
(a)



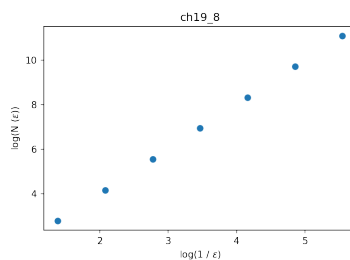
(b)



(c)



(d)



(e)

Figura 2.2: Gráficas del juego del caos aplicado a sub-cadenas del cromosoma 19.

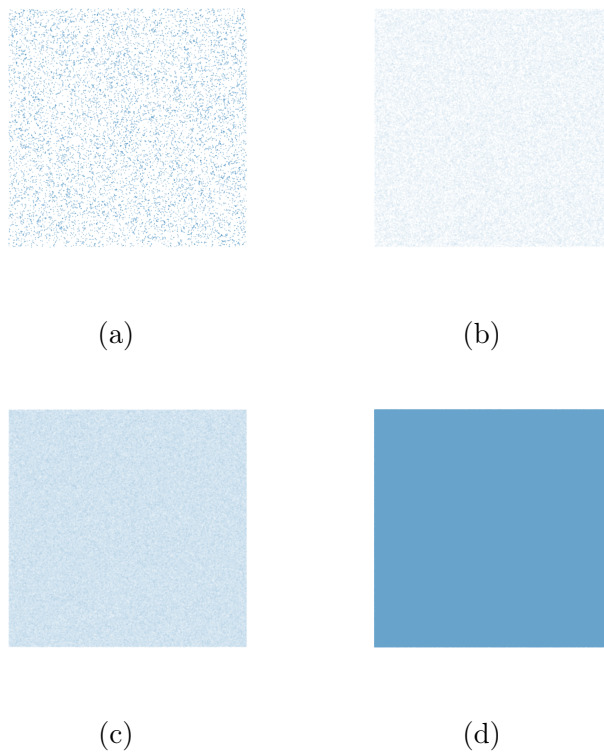


Figura 2.3: Gráficas del juego del caos en cadenas generadas aleatoriamente

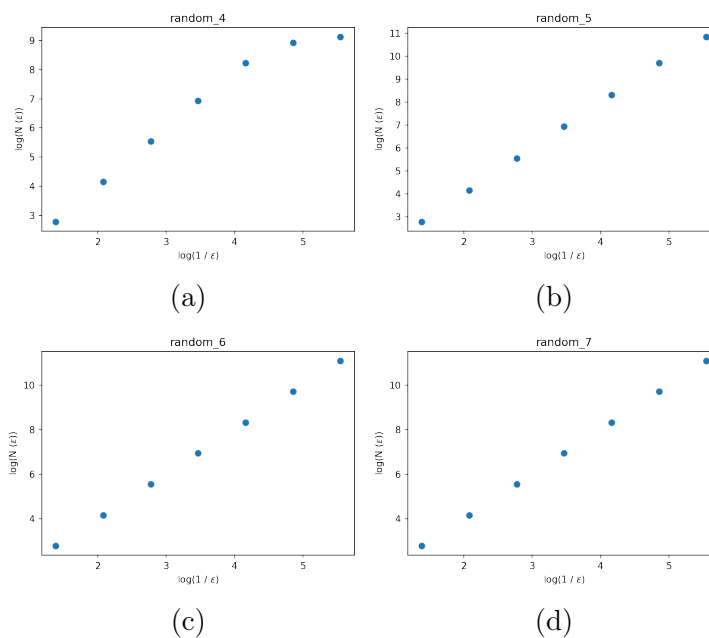


Figura 2.4: Gráficas de la recta que se forma por la relación de la ecuación 1.4 en el conjunto de cadenas de caracteres aleatorios.

Tabla 2.1: Dimensión de los conjuntos anteriormente descritos. *str* representa el origen de los datos y  $N$  se relaciona con la longitud de la cadena mediante la expresión:  $len = 10^N$

N	str	chr_19	aleatoria
4		1.56	1.61
5		1.90	1.96
6		1.99	2.00
7		2.00	2.00
8		2.00	-

En la tabla 2 se incluyen las dimensiones aproximadas de los conjuntos anteriormente descritos para diferentes longitudes.  $N$  se relaciona con la longitud de cada cadena según la ecuación:  $len = 10^N$ .

## Discusión

De acuerdo con los resultados de la tabla 2, las dimensiones de Minkowski-Bouligand para juego del caos aplicado al cromosoma 19 y a cadenas aleatorias de caracteres convergen ambos a 2.0.

Si bien el punto al que convergen es el mismo, sí puede decirse que lo hacen a través de trayectorias diferentes. En las gráficas de las figuras 2.1 y 2.3 se muestran las gráficas con el mismo número de puntos y con las mismas dimensiones. Es evidente que juego del caos produce geometrías muy diferentes dependiendo del origen de las cadenas en que se aplicó el algoritmo; en 2.1, el llenado del cuadrado ocurre eventualmente aunque haya estados que el sistema no suela visitar (regiones con manchas blancas que se repiten con aparente regularidad en el cuadrado).

Esto se corrobora con las dimensiones de la tabla 2 donde la convergencia al 2.0 ocurra un orden de magnitud antes para el caso de la cadena aleatoria.

El proyecto computacional completo con los métodos desarrollados para el cálculo de la dimensión así como varios ejemplos puede encontrarse *aquí*.

# Bibliografía

- [1] Boeing, G. *Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction*. Systems 2016, 4, 37.
- [2] M. Barnsley, A. Vince, *The chaos game on a general iterated function system*. *Ergodic Theory and Dynamical Systems*, 31(4), 1073-1079. doi:10.1017/S0143385710000428, (2011)
- [3] M. Barnsley, *Fractals Everywhere* 2nd ed. Academic Press, Atlanta, Georgia, (1993).
- [4] M. Schroeder *Fractals, chaos and power laws - Minutes from an Infinite Paradise* Freeman (1996).
- [5] M. Hirsch, S. Smale, R. Devaney *Differential Equations, Dynamical Systems and An Introduction to Chaos* 2nd ed. Elsevier Academic Press, 2004.
- [6] M. Newville, et al. *LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python*. Zenodo. 10.5281/zenodo.11813, 2014.