



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

JOSÉ ROBERTO BASTIDA LUNA

ASESOR: DR. PEDRO EDUARDO MIRAMONTES VIDAL

CIUDAD DE MÉXICO, 2021



Capítulo 1

Introducción

La biología matemática, también llamada biología teórica, es una rama de la biología que busca modelar la estructura y comportamiento de organismos vivos a partir del uso de herramientas matemáticas. La bioinformática es otra rama de la biología que, haciendo uso del creciente poder de cómputo que nos es accesible así como de los modelos dimanados de la biología matemática, busca manipular, visualizar, comparar y modificar información genética con el fin de entender sus características, estructura y posiblemente su comportamiento ante cambios estructurales.

Además, las ciencias de la complejidad han captado cada vez más atención recientemente pues sugieren formas de analizar sistemas conformados por un ingente número de elementos donde, si bien el comportamiento de cada uno de dichos elementos que lo constituyen puede ser determinista, el comportamiento global es más bien estadístico.

Los avances en el desarrollo de las herramientas computacionales, la tecnología en materia de secuenciación del genoma entero y la teoría en dinámica no lineal han permitido cada vez una mejor integración de la biología, las matemáticas y la computación.

El presente reporte tiene por objeto presentar una propuesta de aplicación de la biología matemática que permita a estudiantes tanto de biología, matemáticas así como cualquier otra ciencia afín aproximarse a paqueterías computacionales en bioinformática, visualización y cómputo numérico aplicado en genomas.

Se desarrolló un *notebook* en el lenguaje de programación python con el fin de que cualquier persona interesada en la implementación computacional de los experimentos de este reporte pueda descargarlo, replicarlo y modificarlo. Deseamos con esto que el material didáctico sea de apoyo a la docencia y el aprendizaje.

Específicamente se usan regiones de genoma humano, se aplica el algoritmo del juego de caos y se calcula la dimensión fractal con el algoritmo de Minkoski-Bouligand (también conocida como *dimensión por conteo de cajas*). Se concluye que, desde la perspectiva del cálculo de la dimensión fractal, hay básicamente dos familias de posibles atractores para las seis diferentes configuraciones del juego del caos que se eligieron.

Se introducen también los siguientes conceptos:

- Las ecuaciones diferenciales discretas y los Sistemas de Funciones Iteradas (SFI).
- El Juego del Caos: Se introduce el triángulo de Sierpinski y el juego del caos aplicado a subcadenas del cromosoma 19.
- La dimensión de Minkowski-Bouligand (también conocido como la dimensión del *conteo de cajas*).

La dimensión de Minkowski-Bouligand converge a la dimensión topológica. Por tal motivo, la primera aproximación consiste en guiar al alumno por los conjuntos que ya conoce, a saber, una recta y un cuadrado con dimensiones topológicas iguales a 1 y 2 respectivamente. Se calcula también la dimensión de Minkowski-Bouligand para:

- El triángulo de Sierpinski: con el fin de validar que coincide con la dimensión fractal teórica calculada a partir de la auto semejanza.
- El atractor de Henón: un Sistema de Funciones Iteradas (SFI) cuadrático cuya dimensión teórica es de 1.26.

Finalmente se introduce el concepto del juego del caos aplicado a subcadenas de DNA de diferentes longitudes y se calcula la dimensión de Minkowski-Bouligand a los conjuntos que dimanan de ésta con el fin de ilustrar que aún sin una auto semejanza clara, es posible calcular la dimensión de cualquier subconjunto de \mathbb{R}^2 siempre que su cardinalidad sea lo suficientemente grande para que el algoritmo converga.

Todos los métodos computacionales asociados al presente proyecto así como la guía para la ejecución e implementación están disponibles en *esta liga*. Consiste en un proyecto desarrollado en el lenguaje de programación python 3.x. Se exhorta al alumno para que visite la página del proyecto, instale la versión más reciente de python 3.x y experimente con las funciones que aquí se presentan.

Capítulo 2

Sistemas dinámicos discretos y fractales

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

$$x_{n+1} = f(\vec{\alpha}, x_n) \quad (2.1)$$

Donde $\vec{\alpha}$ es algún vector de parámetros y x_n es el estado previo del sistema. 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 2.1 y un punto inicial x_0 dentro del dominio de la ecuación, uno puede conocer la dinámica del sistema. Para esto basta componer la ecuación consigo misma alrededor del punto inicial x_0 un número lo suficientemente grande de veces para que aporte información sobre la evolución del sistema:

$$\begin{aligned} & x_0 \\ & x_1 = f(\vec{\alpha}, x_0) \\ & x_2 = f \circ f(\vec{\alpha}, x_0) \\ & \quad \dots \\ & x_n = \underbrace{f \circ \dots \circ f}_{n \text{ veces}}(\vec{\alpha}, x_0) \end{aligned} \quad (2.2)$$

A partir de estas iteraciones, se obtiene una sucesión de puntos $\mathcal{S} = \{x_0, x_1, \dots, x_n\}$ 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 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:

$$x_{n+1} = \lambda x_n (1 - x_n) \quad (2.3)$$

Con $\lambda > 0$. De modo que depende únicamente de un parámetro de 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 en el informe de *Distribuciones de Rango Orden en Genomas Completos*, mismo que es parte del presente proyecto y que puede encontrarse en esta liga.

Diagramas fase

Ilustra los *saltos* que da el sistema entre estados consecutivos. Para esto se toma la sucesión de puntos $S = \{x_0, x_1, \dots, x_n\}$ 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 .

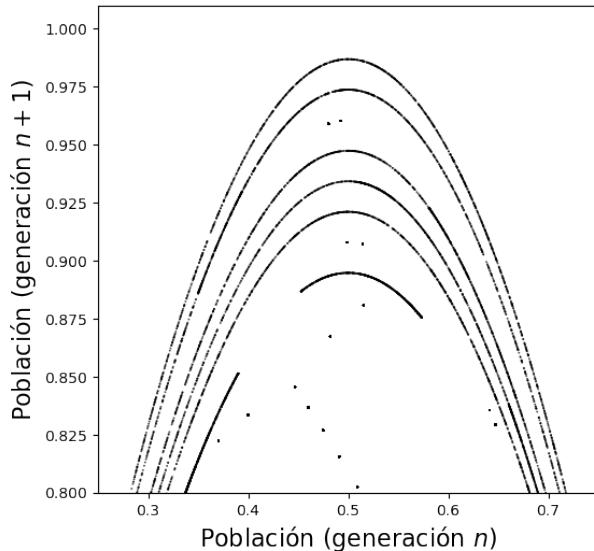


Figura 2.1: Diagrama fase que nos permite visualizar si el sistema se estabiliza. Básicamente nos dice de dónde proviene y a dónde llega el estado del sistema.

En la figura 2.1, cada arco representa la evolución de la ecuación logística 2.3 a un valor de λ . 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.

Sistemas de Funciones Iteradas

Los Sistemas de Funciones Iteradas (o SFI como de ahora en adelante nos referiremos) son un tipo particular de ecuaciones diferenciales discretas en cuanto a que su estructura básica es la misma que en 2.2. Usualmente se les asocia con conjuntos auto semejantes aunque hay ciertas condiciones que deben cumplirse para que esto ocurra.

Unos de los sistemas de funciones iteradas más estudiados es el mapeo de Henón cuyas coordenadas a cada iteraciones se obtienen mediante las ecuaciones:

$$\begin{cases} x_{n+1} = 1 - a x_n^2 + y_n \\ y_{n+1} = b x_n \end{cases} \quad (2.4)$$

El atractor de este sistema para un millón de iteraciones se muestra en la figura 2.2.

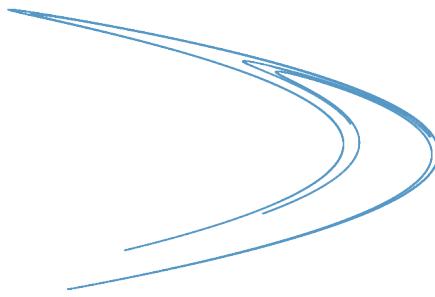


Figura 2.2: Atractor del conjunto de henón cuyo sistema está representado por la ecuación 2.4

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 repartirse.

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_{κ_1} . Este nuevo punto es p_1 , el correspondiente a la iteración 1. Más concisamente, $p_1 = PM(p_0, v_{\kappa_1})$.
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 2.3a 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 2.3b 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 2.3c se cambiaron los colores de tres triángulos más pequeños para evidenciar que éstos, a su vez, tienen triángulos blancos invertidos todos ellos semejantes.

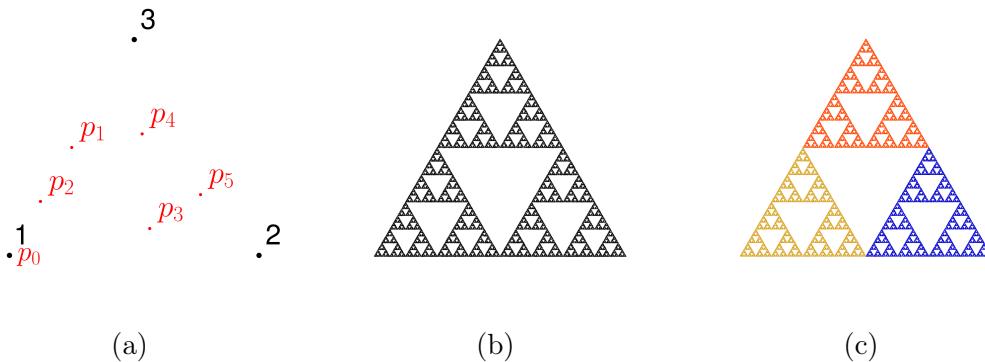


Figura 2.3: (a): Cinco puntos generados con el algoritmo del triángulo de Sierpinski. (b): Gráfica de los puntos después de un millón iteraciones. (c): Los tres triángulos pequeños reproducen la forma del original en (b), la gráfica es auto semejante.

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 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:

- Dibujar un cuadrado donde a cada vértice 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 subcadena del DNA:

ATGCCGATA

En la figura 2.4 se ilustra cómo luce la gráfica del algoritmo para las primera 9 posiciones de la cadena.

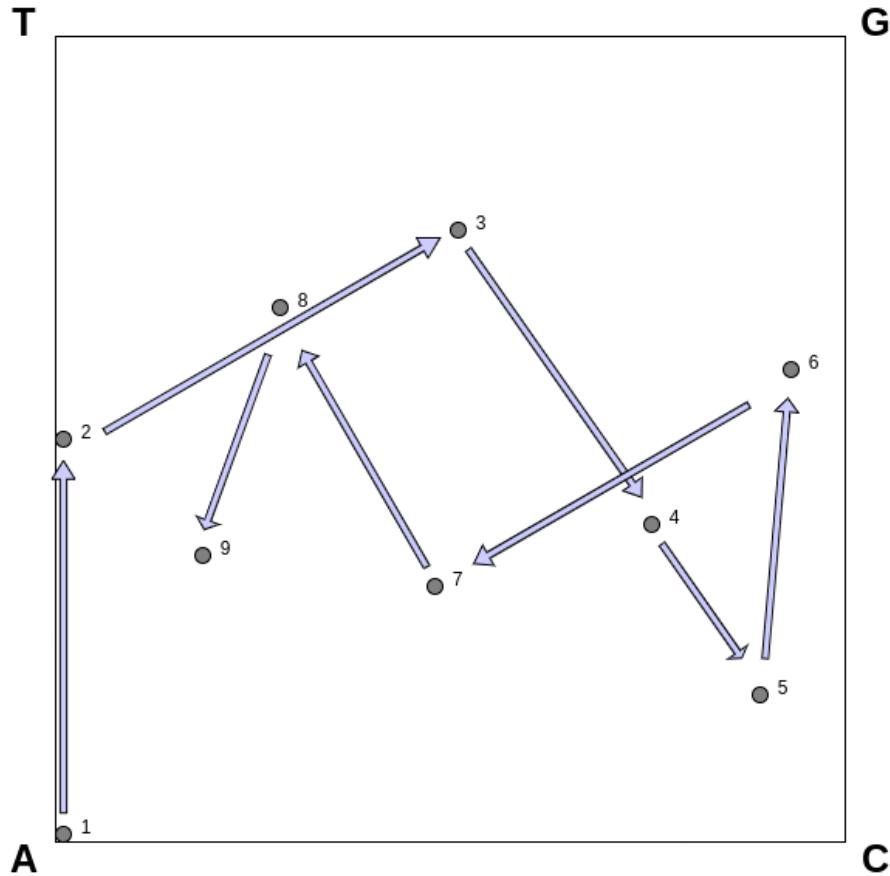


Figura 2.4: Algoritmo del juego del caos en una pequeña subcadena del DNA.

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 2.5 ilustra cómo ésto se relaciona con la dimensión topológica. En 2.5a, 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 2.5b, 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 2.5c se muestra únicamente la silueta de los tres triángulos de 2.3c. Duplicar las dimensiones del triángulo resulta en tres copias de éste, su dimensión es entonces $k = \ln(3)/\ln(2) \approx 1.584$.

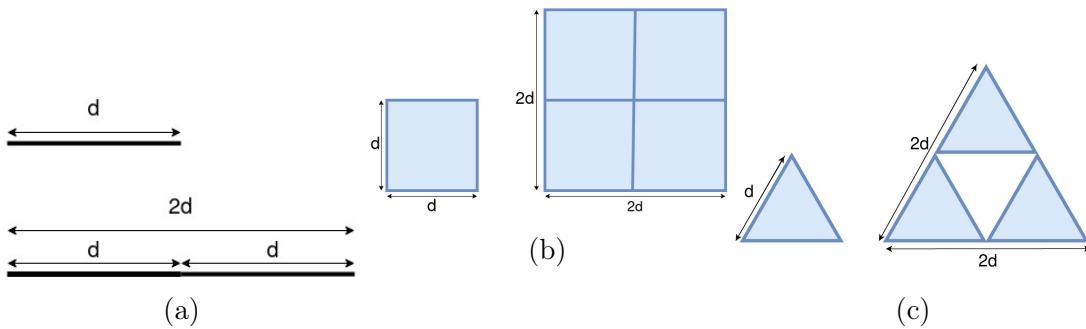


Figura 2.5: El cálculo de la dimensión fractal coincide con la topológica para una y dos dimensiones. El número de copias crece como potencia de 2 al duplicar la longitud de un objeto. Cuando el conjunto es auto semejante y no llena el espacio; la dimensión fractal no es un número entero.

Una de las implicaciones que esto tiene es que un fractal no necesariamente *lleva* 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 auto semejanza, 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 auto semejanza 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

El método anteriormente descrito para calcular la dimensión fractal se basa en que haya una evidente autosemejanza y que sea posible contar el número de réplicas. Cuando en el conjunto no haya evidente autosemejanza, se usa el algoritmo de *Minkowski-Bouligand*, también conocido como el algoritmo de *conteo de cajas*.

Se basa en la misma expresión $2^k = M$ y busca partitionar el espacio en segmentos cuyo tamaño decrezca como potencia de $\sim \frac{1}{2^n}$ 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:

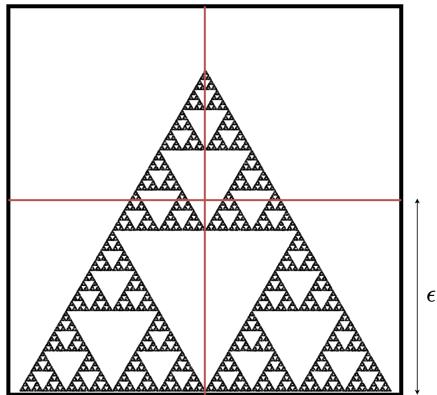
$$\dim_{box}(\mathcal{S}) = \lim_{\epsilon \rightarrow 0} \frac{\log(N(\epsilon))}{\log(1/\epsilon)} \quad (2.5)$$

Donde ϵ 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 .

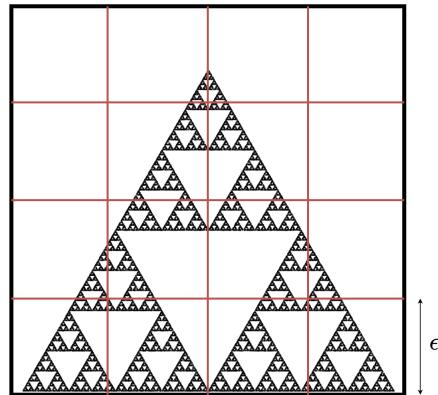
En la figura 2.6 se muestran las primeras cuatro iteraciones del algoritmo del conteo de cajas aplicado sobre el triángulo de Sierpinski. Se aprecia cómo después de unas cuantas iteraciones, hay cuadrados dentro del triángulo invertido del centro donde no hay elementos del conjunto.

La manera de implementar la ecuación 2.5 para el cálculo de la dimensión consiste en iterar para diferentes valores escalas de ϵ 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 .

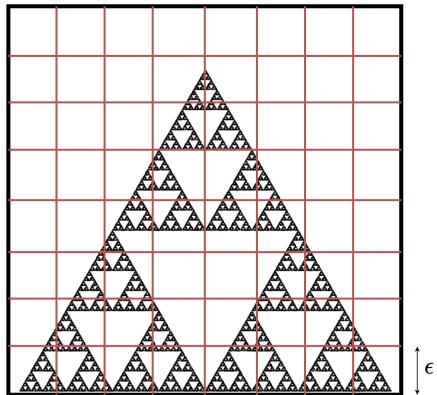
Esto resulta en una recta cuya pendiente aproximarán la dimensión fractal que deseamos calcular. De modo que se hace una regresión lineal con los puntos que resulten de la iteración para diferentes escalas de ϵ .



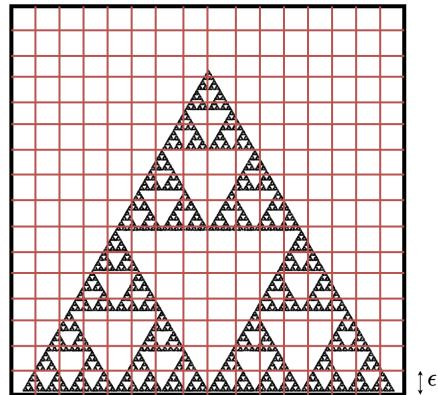
(a)



(b)



(c)



(d)

Figura 2.6: 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 3

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.
- Conjunto de Henón.
- Juego del caos en una subcadena del cromosoma 19.
- Juego del caos en una cadena aleatoria con el mismo diccionario del DNA.

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:

- Validar el correcto funcionamiento de los métodos desarrollados para el cálculo de la dimensión. Para esto se usaron los conjuntos con dimensiones conocidas, a saber: La recta, el cuadrado, el triángulo de Sierpinski y el conjunto de Henón.
- Indagar sobre la convergencia de la dimensión de Minkowski-Bouligand en el juego del caos aplicado al cromosoma 19 y en subcadenas aleatorias generadas a partir del mismo diccionario que el DNA.

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 .

Capítulo 4

Resultados y Discusión

El algoritmo de Minkowski-Bouligand para aproximar la dimensión de los conjuntos anter mencionados arrojará los resultados de la tabla 4.1. Éstos coinciden con la teórica en los casos de la recta, el cuadrado y el conjunto de Sierpinski. Para el conjunto de Henón donde el valor teórico estimado es 1.25, el error asociado es 0.016.

Tabla 4.1: Dimensión de conteo de cajas para conjuntos cuyo valor teórico se conoce a priori.

| Conjunto | Dimensión |
|------------|-----------|
| Recta | 1.00 |
| Cuadrado | 2.00 |
| Sierpinski | 1.58 |
| Henón | 1.23 |

Juego del caos en DNA

Las gráficas del juego del caos para las subcadenas del cromosoma 19 y para las cadenas de caracteres aleatorios se muestran en las figuras 4.1 y 4.2 respectivamente.

Las dimensiones de Minsowski-Bouligand asociados a estos conjuntos se encuentran la tabla 4. Aquí, N se relaciona con la longitud de cada cadena según la ecuación: $len = 10^N$.

Se podría concluir que las dimensiones por conteo de cajas de ambos sistemas convergen a 2.0, el caso del juego del caos aplicado al DNA nunca llegó al valor de 2.0 sino que lo hizo después de que se redondeara a dos cifras después del punto decimal.

También es menester considerar que, a juzgar por las figuras de los atractores del sistema mostrados en la figura 4.1, es posible que para subcadenas del cromosoma de gran longitud el sistema se esté saturando y la dimensión no sea la correcta.

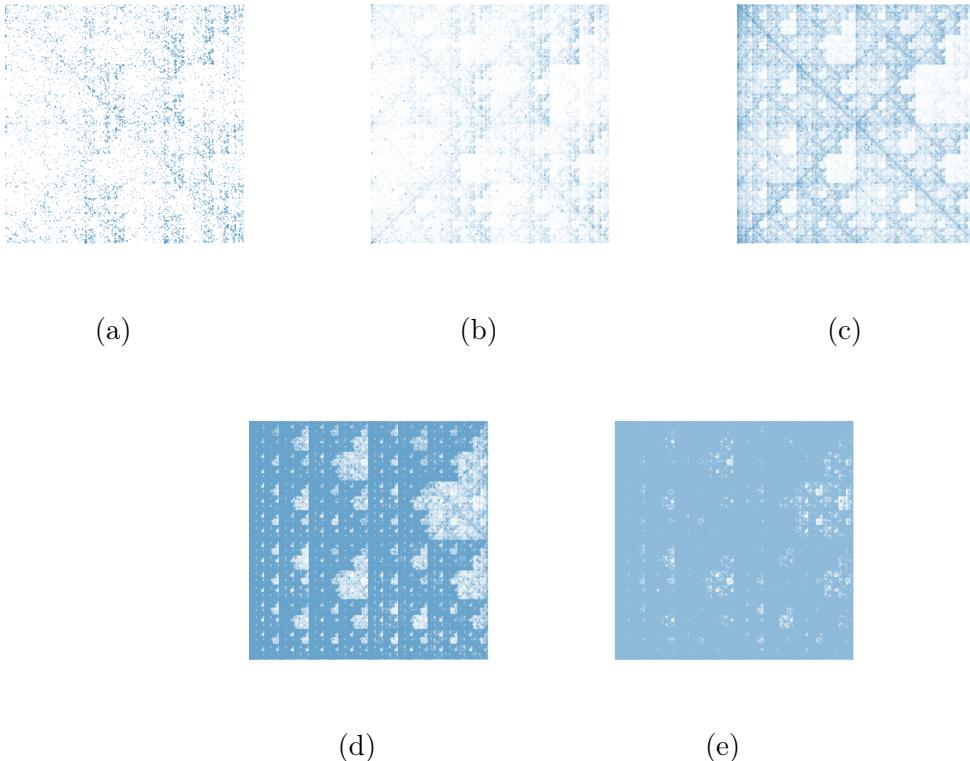


Figura 4.1: Gráficas del juego del caos aplicado a subcadenas del cromosoma 19.

Tabla 4.2: 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: $\text{len} = 10^N$

| 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 | - |

Los resultados anteriormente reportados, si bien aproximan la dimensión fractal del juego del caos en el DNA para diferentes longitudes de la subcadena; no brindan un número definitivo al cual le podamos asociar la dimensión. Por tal motivo se iteró el algoritmo para subcadenas de diferentes longitudes con el fin de aproximar la dimensión de acuerdo con el algoritmo de *kneedle* [9].

Para aproximar la dimensión de Minkowski-Bouligand para el DNA, se calculó la dimensión para 50 subcadenas cuyas longitudes particionaron el intervalo $[10^5, 10^6]$ en segmentos de igual tamaño. La gráfica de *longitud de la cadena vs dimensión* se muestra en la figura

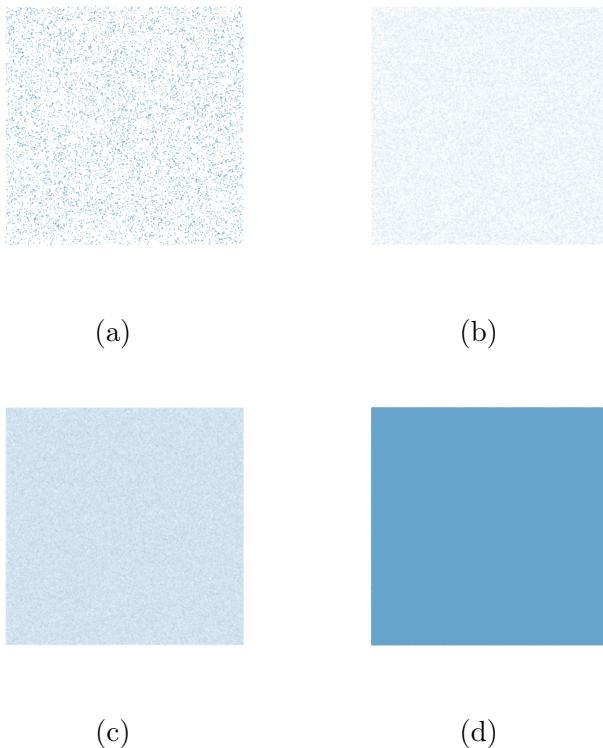


Figura 4.2: Gráficas del juego del caos en cadenas generadas aleatoriamente

4.3.

Aplicación del algoritmo de kneedle

De la figura 4.3 se aprecia que el valor de la dimensión se estabiliza rápidamente y se acerca asintóticamente a 2.0. Para aproximar la dimensión del sistema se requiere encontrar *rodilla* o la *meceta* de la curva cuya gráfica se muestra en 4.3. Aquí se usa el algoritmo de *Kneedle* [9].

Éste consiste en los pasos:

- Estandarizar los datos, aquí se usó la transformación MinMax.
- Se grafica la curva de $(x, y - x)$
- Identificar el máximo de las coordenadas $y' = y - x$ y posteriormente transformar nuevamente a la escala de dimensiones originales.

Los pasos que se enunciaron anteriormente se ilustran gráficamente en las figuras de 4.3. Lo que se busca con esto es aproximar el punto de inflexión de la curva de dimensiones. Consideraremos este valor como la mejor aproximación de la dimensión de Minkowski-Bouligand para el juego del caos aplicado al cromosoma 19.

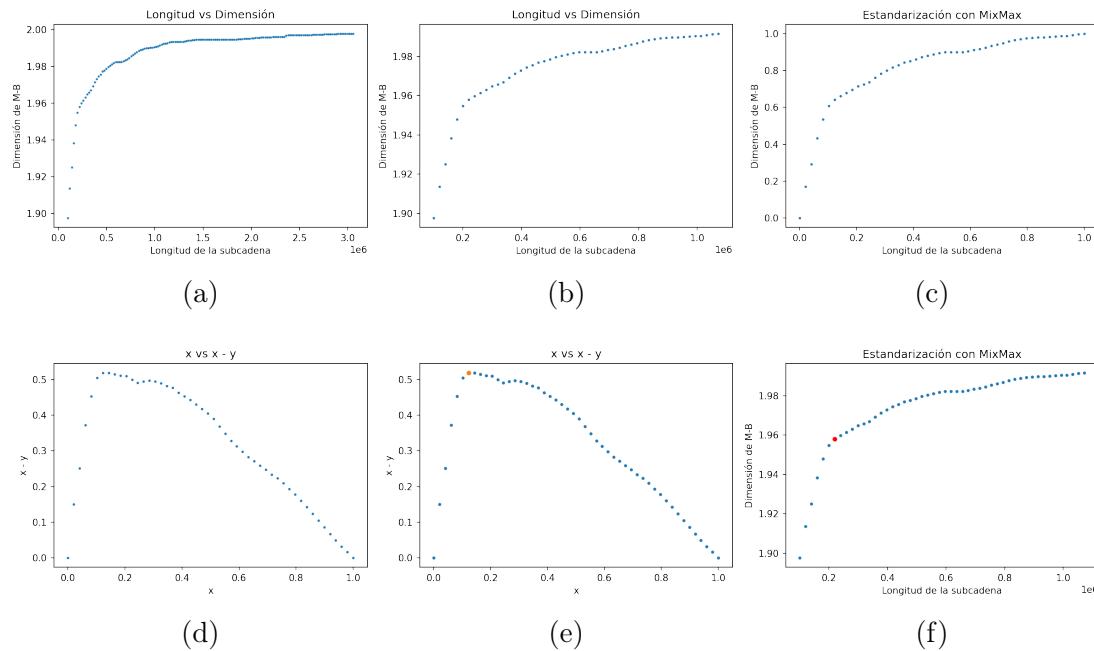


Figura 4.3: Ilustración de pasos para determinar la relación entre la longitud de la subcadena y la dimensión de Minkowski-Bouligand y así determinar el punto de inflexión que consideraremos la dimensión del atractor.

Esto condujo a identificar el punto indicado en la figura 4.3 y el valor aproximado de la dimensión de 1.97.

Una vez que se contró con un método para reproducir el juego del caos en regiones del cromosoma 19 del genoma humano y calcular la dimensión fractal del atractor resultante. Se antoja la pregunta sobre cómo cambiaría los resultados el cambiar de orden las etiquetas del cuadrado en el juego del caos como se muestra en la figura 2.4.

Es decir ¿cómo lucirán los atractores y cuáles serán las dimensiones fractales cuando se intercambian las etiquetas de los nucleótidos **C - G**, **C - G** o se aplica cualquier permutación? Se tienen doce maneras de permutar los vértices de un cuadrilátero con etiquetas sin repetición. Aquí se consideraron únicamente las seis permutaciones que resultan de dejar uno de los vértices fijos puesto que las seis restantes se obtienen de aplicar rotaciones y reflexiones. Éstas se muestran en la figura 4.4

En la figura 4.5 se muestran los atractores que resultan de reproducir el juego del caos cuando los vértices son los de la figura 4.4. Para todos ellos se approximó una dimensión fractal de 1.97 usando el algoritmo de Minkowski-Bouligand.

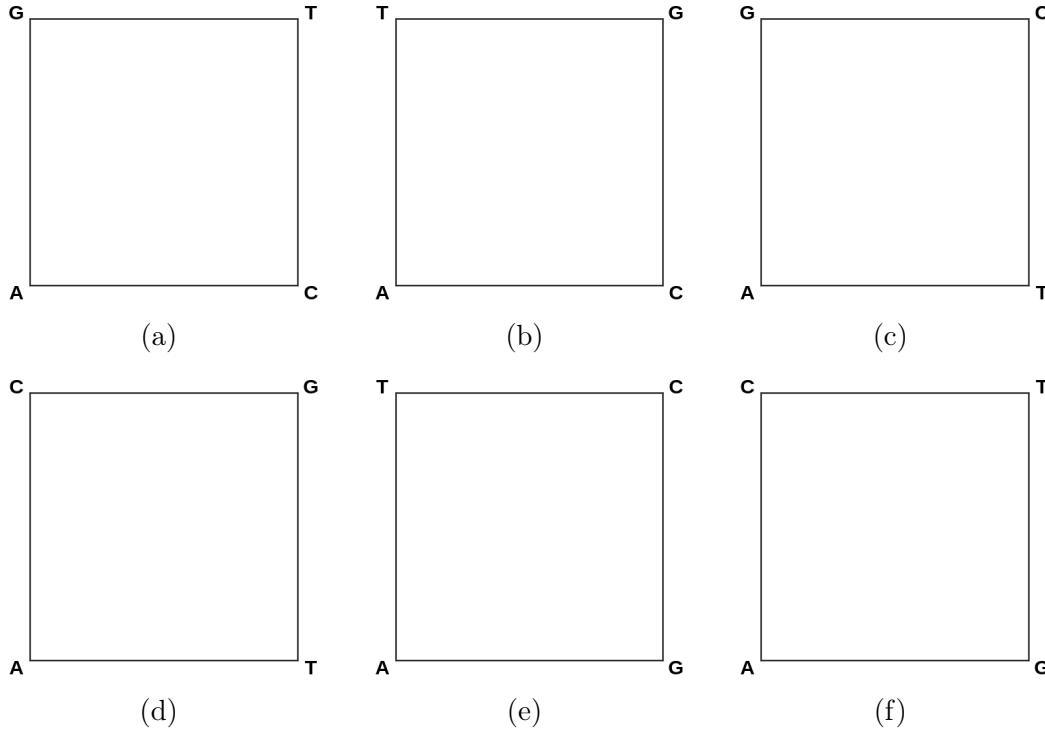


Figura 4.4: Diferentes configuraciones sobre las cuales se aplicó el juego del caos para el DNA.

Discusión

Si bien esto es cierto que dimensiones de Minkowski-Bouligand para juego del caos aplicado al cromosoma 19 y a cadenas aleatorias de caracteres convergen ambos a 2.0. Se observó que para el caso de la subcadena, esto ocurre cuando la longitud de la es lo suficientemente grande.

Por tal razón se realizó un análisis de concavidad con el algoritmo *Kneedle* [9] para determinar el punto de mayor tasa de cambio en la gráfica de las longitudes contra la dimensión calculada. Este procedimiento se ilustra en la figura 4.3 y tuvo como finalidad el determinar el punto que se usaría como la mejor aproximación a la dimensión fractal de todas las que se calcularon.

Esto condujo a una aproximación de la dimensión fractal igual a 1.97 para todos los atractores mostrados en la figura 4.5.

Podría argumentarse que la dimensión del atractor habría de ser aquella a la que se converga aún con un gran número de elementos, es decir, dos. Consideramos para el presente trabajo que también es una pregunta de interés conocer el comportamiento del sistema no sólo el resultado al que se llega sino *cómo* es que se llega dicho resultado.

Nos referimos con esto a que podría ser de interés conocer no sólo el atractor final del juego del caos aplicado a algún cromosoma usando la cadena en su totalidad sino también

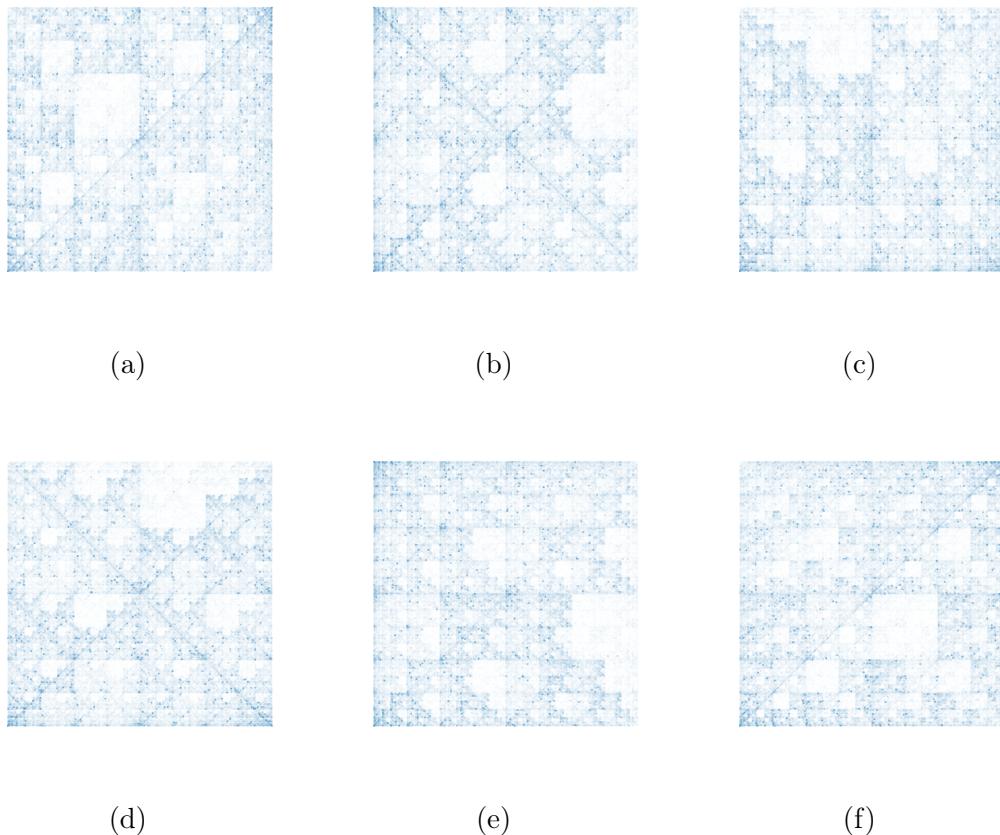


Figura 4.5: Atractores para las seis configuraciones de vértices en la aplicación del juego del caos.

la manera en la que se va *llenando* el cuadrado a medida que se incrementa la longitud de la cadena. Esto pudiera ser de igual interés al momento de conocer el comportamiento del sistema.

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

Bibliografía

- [1] Martínez-Meckler G, Martínez RA, Mansilla R, Miramontes P, et al. *Universality of Rank-Order Distributions in the Arts and Sciences*, PLoS ONE 4(3): e4791, doi:10.1371/journal.pone.0004791.
- [2] O. Fontanelli, P. Miramontes, R. Mansilla. *Distribuciones de probabilidad en las ciencias de la complejidad: una perspectiva contemporánea*, <https://arxiv.org/abs/2002.09263>, 2019.
- [3] Boeing, G. *Visual Analysis of Nonlinear Dynamical Systems: Chaos, Fractals, Self-Similarity and the Limits of Prediction*. Systems 2016, 4, 37.
- [4] 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)
- [5] M. Barnsley, *Fractals Everywhere* 2nd ed. Academic Press, Atlanta, Georgia, (1993).
- [6] M. Schroeder *Fractals, chaos and power laws - Minutes from an Infinite Paradise* Free-man (1996).
- [7] M. Hirsch, S. Smale, R. Devaney *Differential Equations, Dynamical Systems and An Introduction to Chaos* 2nd ed. Elsevier Academic Press, 2004.
- [8] M. Newville, et al. *LMFIT: Non-Linear Least-Square Minimization and Curve-Fitting for Python*. Zenodo. 10.5281/zenodo.11813, 2014.
- [9] V. Satopaa, et al. *Finding a “Kneedle” in a Haystack: Detecting Knee Points in System Behavior*. 2011 31st International Conference on Distributed Computing Systems Workshops, 2011, pp. 166-171, doi: 10.1109/ICDCSW.2011.20.